Virtuoso Layout Suite XL: Basic Editing User Guide

Product Version IC23.1
November 2023

© 2023 Cadence Design Systems, Inc.
Printed in the United States of America.

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

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

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
Layout Editor Basics 47
Reserved Parameter and Property Names for Setting Up Layout XL 48
Starting the Design Environment for Layout XL 49
Layout XL Launched from a Schematic View 50
Generating a Schematic View from a CDL Netlist 51
Starting Layout XL with an Automatic Physical Configuration View 52
Starting Layout XL with a New Physical Configuration View 55
Starting Layout XL with an Existing Physical Configuration View 60
Starting Layout XL from a Layout View 63
Starting Layout XL from a Physical Configuration View 64
Layout XL Constraint Group 65
Layout XL Desktop 66
Working with Designs 68
Opening a New Design 69
Opening an Existing Cellview 71
Making a Design Read Only or Editable 72
Saving a Design 73
Closing a Design 76
Basic Command Functions in Layout XL 77
Bindkeys in Virtuoso Layout Suite 79
Displaying Bindkeys for Layout XL 80
Layout Editor Default Setup 83
Startup Files 83
Forms for Setting Layout XL Default Values 83
Default Application Used for a Cellview Type 84
Technology File 84
Workspaces in Layout XL 85
Display and Selection Controls 95
Printing in the CIW 95

© 2023 Cadence Design Systems, Inc.

Multiple Cellviews Opened in Layout XL . . . 96
Shortcut Menus . . . 96
Object Selection Modes in Layout XL . . . 98
Displaying Options Forms Automatically . . . 99
Layer Display Colors . . . 100
Layout Editor Strokes . . . 102
Loading and Unloading Strokes . . . 104

2
Design Display and Search Controls . . . 105

Design Display . . . 106
Setting an Area Display . . . 107
Viewing Coordinates . . . 108
Find Functionality . . . 115
Search Criteria . . . 115
Finding Objects . . . 117
Replace Functionality . . . 119
Replacing Instances . . . 121
Finding and Replacing Text of Labels . . . 123
Object Search Criteria . . . 126

3
Design Tools . . . 145

Calculating Area and Density . . . 145
Rulers . . . 147
Saving and Deleting Rulers . . . 148
Ruler Segment Modes . . . 149
Mouse and Key Actions in Ruler Segment Modes . . . 151
Ruler Snapping Support . . . 153
Creating Rulers with Snapping On . . . 154
Creating Rulers with Smart Snapping Off . . . 157
Instance Remastering . . . 159
Terminal Checking During Instance Remastering . . . 162
Replacing Instance Abstract Views with Layout Views . . . 163
Object Tapping . . . 164

© 2023 Cadence Design Systems, Inc.

Tapping Objects Using the Tap Command . . . 166
Customizing Tap Settings Using the Layout Editor Options Form . . . 167
Layer Generation . . . 169
Making Changes to Objects on Layers . . . 170
Making Boolean Edits by Layer . . . 171
Markers . . . 172
Displaying Marker Information . . . 173
Finding Markers in a Design . . . 174
Deleting Markers . . . 174
Checking Layout Integrity . . . 178
Pad Shapes and Labels . . . 180
Generating a Standard Report for Pad Shapes . . . 180
Generating a Customized Report for Pad Shapes . . . 183
Setting Filters for a Customized Report . . . 186
Generating Labels for Bond Pads . . . 187
Promoting Labels to the Top Level . . . 188
Status Toolbar Customization . . . 189
Status Toolbar Information Formatting . . . 191
Capture and Replay Your Design Actions . . . 193
Capture and Replay Assistant . . . 194
Recording Your Actions in a Design . . . 197
Replaying Your Recorded Actions in a Design . . . 199
Deleting a Recording in the Capture and Replay Assistant . . . 202
Renaming a Recording in the Capture and Replay Assistant . . . 203
Cross Section Viewer . . . 205
Viewing the Cross Section of a Layout . . . 207
Specifying Cross Section Viewer Settings . . . 209
Cross Section Viewer Configuration File . . . 213

4
Hierarchical Designs . . . 217
Checking for Instance Recursion . . . 218
Changing the Origin of a Cellview . . . 219
Descending into a Cellview . . . 220
Returning to a Different Hierarchy Level . . . 221

© 2023 Cadence Design Systems, Inc.

Editing a Cell in Place . . . 222
Edit In Place in Post-Selection Mode . . . 223
Freezing and Unfreezing Pcell Instances . . . 225
Making Hierarchy Editable . . . 226
Making Cells . . . 227
Instance Hierarchy Flattening . . . 228
Flattening Instances . . . 230
Copying and Cutting Through the Hierarchy . . . 231
Uniquifying a Cellview . . . 232
Viewing Cellview Summary . . . 233
Refreshing Cellviews to Reflect Updated Masters . . . 234

5
Object Creation in Layout XL . . . 237

Object Creation by Entering Points . . . 237
Entering Points in the CIW to Create Objects . . . 239
Entering Points Using the Enter Points Form . . . 239
Points Formats Supported in the Enter Points Forms . . . 241
ROD Support for Object Creation . . . 242
Creating a Rectangle . . . 245
Creating a Polygon . . . 247
Creating an Arc in a Polygon . . . 248
Creating a Path . . . 249
Creating a Path with Slots . . . 252
Controlling the Display of Paths . . . 255
Creating a Circle . . . 255
Creating an Ellipse . . . 256
Creating a Donut . . . 257
Creating Instances . . . 258
Pin Creation . . . 260
Creating Pins . . . 260
Placing a Pin Array . . . 263
Setting Pin Label Text Style Options . . . 264
Support for Orthogonal WSP Grid . . . 266
Creating Pins from Labels . . . 267

© 2023 Cadence Design Systems, Inc.

Label Creation . . . 269
Label Creation Patterns in Manual Mode . . . 270
Label Creation on LPPs Derived from stampLabelLayers . . . 273
Configuring Label Creation Options . . . 276
Generating Labels Using the Scan Line . . . 281
Generating Labels Using Auto Step in Manual Mode . . . 284
Label Creation in Manual and Auto Modes . . . 286
Label Generation for Overlapping Shapes . . . 291
Via Creation . . . 296
Installing a cdsVia Device in the Technology File . . . 297
Adding the cdsVia Device to the validVias Constraint . . . 300
Setting Up Via Creation Options . . . 304
Creating a Single Via . . . 307
Creating a cdsVia . . . 310
Creating a Stack Via . . . 312
Creating or Updating a Via Variant . . . 316
Creating a Via Automatically in Single and Stack Modes . . . 318
Auto Mode Via Creation on Intersecting Shapes . . . 321
Placing a Via Automatically with a Single Click . . . 323
Placing a Via Automatically Based on Connectivity Information of Intersecting Shapes 328
Auto and Fast Edit Via Mode Options . . . 329
Analyzing a via using Auto Via Assistant . . . 337
Creating a via using Auto Via Assistant . . . 341
Setting the Technology to Create Vias . . . 345
Calculation Modes for Via Parameters . . . 345
viaEngineContext User Type . . . 346
Supported Constraints . . . 347
Support for Voltage-Dependent Rules . . . 352
Creating Slots . . . 353
Wide Wires Slotting Flow . . . 355
Creating a Group . . . 356
Placement and Routing Objects . . . 357
Blockages . . . 358
Creating a Blockage . . . 360
Blockage Halos . . . 361

© 2023 Cadence Design Systems, Inc.

Rows . . . 362
Creating Rows . . . 364
Creating a Custom Placement Area . . . 368
Creating Clusters . . . 369
Track Patterns . . . 371
Technology File Requirements for Track Patterns . . . 372
Displaying Track Patterns . . . 377
Creating a Track Pattern . . . 378
Creating a PR Boundary . . . 379
Creating a Snap Boundary . . . 380
Creating an Area Boundary . . . 381
Creating a Cluster Boundary . . . 382
Create Commands Supporting Snap Pattern Snapping . . . 384

6
Object Editing in Layout XL . . . 385

Object Moving . . . 386
Moving an Object by Direct Manipulation . . . 387
Moving Objects Manually . . . 388
Moving an Object by Using the Move Command . . . 389
Moving or Copying an Object to Another Layer . . . 389
Moving an Object to Another Cellview . . . 390
Moving a Multipart Rectangle . . . 390
Moving Objects Using the Move Form . . . 391
Object Copying . . . 393
Copying an Object . . . 395
Array of Copied Objects . . . 399
Creating an Array by Using the Copy Command Interactively . . . 400
Creating an Array by Specifying Spacing Mode and Values . . . 401
Copying an Object to Another Cellview . . . 404
Cellviews Copied Across Libraries . . . 406
Copying an Object in Repeat Mode . . . 406
Object Stretching . . . 410
Stretching an Object by Direct Manipulation . . . 412
Stretching a Circle . . . 413

© 2023 Cadence Design Systems, Inc.

Stretching a Via . . . 414
Changing the Angle of a Corner by Stretching . . . 415
Stretching and Moving Objects . . . 416
Stretching Paths . . . 417
Stretching Multipart Rectangles . . . 418
Object Deletion . . . 420
Deleting Objects . . . 420
Object Flipping . . . 421
Flipping Objects . . . 422
Flipping an Object by Specifying a Reference Point on Canvas . . . 424
Object Rotation . . . 426
Rotating Objects by Using the Rotate Form . . . 428
Rotating a Geometric Object by Any Angle . . . 429
Rotating Multiple Objects . . . 431
Rotating or Flipping an Object During a Move Operation . . . 434
Object Chopping . . . 436
Array Chopping . . . 439
Chopping an Object . . . 442
Via Chopping . . . 444
Object Merging . . . 446
Merging Objects . . . 447
How Merging Converts Objects . . . 448
How Merging Converts Paths . . . 449
Object Yanking and Pasting . . . 453
Yanking and Pasting Objects in Pre-Selection Mode . . . 453
Yanking and Pasting Objects in Post-Selection Mode . . . 455
Object Reshaping . . . 456
Reshaping a Polygon . . . 457
Adding a Rectangle to a Polygon . . . 458
Reshaping a Path . . . 459
Object Splitting . . . 459
Splitting an Object . . . 460
Changing the Angle While Splitting an Object . . . 464
Splitting Multiple Objects . . . 466
Modifying Corners of a Polygon . . . 468
Modifying Object Size . . . 469

© 2023 Cadence Design Systems, Inc.

Parent-Child Relationship Between Objects . . . 470
Attaching and Detaching Objects . . . 472
Object Conversion . . . 473
Instance-to-Mosaic Conversion . . . 474
Converting Instances to Mosaics . . . 478
Converting Circles, Donuts, and Ellipses to Polygons . . . 479
Converting Polygons to Paths . . . 480
Object Alignment . . . 482
Object Alignment by Using the Quick Align Command . . . 483
Object Highlighting When Quick Align is Active . . . 484
Customizing Quick Align Settings . . . 486
Quick Align Functionality in Full and Partial Selection Modes . . . 493
Aligning Edges and Centerlines with a Target Edge in Full Selection Mode . . . 494
Aligning Edges and Centerlines with a Target Edge in Partial Selection Mode . . . 499
Alignment When Reference and Target Are Set on the Same Edge . . . 505
Aligning Points with a Target Point . . . 506
Aligning Objects with a Target Axis . . . 512
Object Alignment by Using the Align Command . . . 516
Aligning Objects with a Point . . . 520
Aligning Objects with a Reference Object . . . 523
Aligning Objects with a Layer . . . 527
Aligning a Layer with Another Layer . . . 529
Aligning a Layer with Objects . . . 532
Adding Objects with a Pre-Selected Aligned Group . . . 533
Aligning Objects by Using the Align Toolbar . . . 534
Editing Slots . . . 540
Editing Groups . . . 541
Adding Objects to a Group . . . 541
Adding Objects to a Group By Using the Select Target Group Form . . . 543
Ungrouping All Objects in a Group . . . 544
Removing Objects from a Group . . . 544
Editing Group Contents by Using Edit In Place . . . 546
Editing Group Contents by Using Transparent Group Mode . . . 546
Edit Commands Supporting Snap Pattern Snapping . . . 548
Pin-Path and Pin-PathSeg Connections Preserved . . . 548

© 2023 Cadence Design Systems, Inc.

7
Multipart Paths

Multipart Paths . . . 553
Multipart Path Creation Overview . . . 554
Flow for Creating Multipart Paths . . . 556
Defining the Master Path of an MPP . . . 557
Defining Connectivity for the Master Path . . . 562
Subparts of a Multipart Path . . . 564
Adding an Offset Subpath . . . 564
Adding an Enclosure Subpath . . . 567
Sets of Subrectangles . . . 569
Adding a Set of Subrectangles Subpart . . . 571
Defining Connectivity for a Subpart . . . 575
Editing a Subpart of an MPP . . . 576
Deleting a Subpart of an MPP . . . 577
Drawing a Multipart Path in the Layout Window . . . 579
Multipart Path Values Saved as a Template . . . 580
Saving MPP Templates to an ASCII File . . . 581
Saving MPP Templates to Your Technology Library . . . 582
Saving MPP Templates to Your Technology Library on the Disk . . . 583
Loading an ASCII Multipart Path Template File . . . 584
Creating an MPP Guard Ring . . . 586
Multipart Path Environment Variables . . . 587
mppTemplate Environment Variable . . . 587
mppASCIIFileName Environment Variable . . . 588
Multipart Path Selection . . . 589
Selecting a Whole Multipart Path . . . 590
Selecting Multipart Path Ends . . . 591
Selecting Chopped Subpath Ends in Multipart Paths . . . 593
Selecting Multipart Path Segments . . . 594
Selecting a Multipart Path Vertex . . . 597
Moving Multipart Paths . . . 598
Stretching Multipart Paths . . . 600
Stretching Chopped Subpath Ends in Multipart Paths . . . 602
Splitting Multipart Paths . . . 605
Reshaping a Multipart Path . . . 608

© 2023 Cadence Design Systems, Inc.

How Chopping Affects Multipart Paths . . . 610
Chopping a Multipart Path . . . 613

8
Object Properties . . . 617

Displaying Object Properties . . . 618
Editing Properties of an Object . . . 620
Editing Properties of a Group of Objects . . . 622
Searching and Replacing Object Properties . . . 623
Adding a New Cellview Property . . . 624
Defining a New Object Property . . . 625
Modifying a Property . . . 625
Deleting a Property . . . 626
Editing Object Attributes . . . 627
Editing Object Connectivity . . . 627
Via Editing . . . 628
Editing Cut Patterns . . . 630
Inherited Connections and Net Expressions . . . 632
Inherited Connections . . . 632
Net Expressions . . . 632
Creating and Editing a Net Expression . . . 633
Property Editor Assistant . . . 634
Net Setting on Instance Terminals . . . 635
Route Status of Objects . . . 636
Voltage Range Information . . . 636
Interactively Setting Color on a Selected Object . . . 638
Setting Color on Shape Objects . . . 638
Setting Color on Vias . . . 639

9
Group Arrays . . . 641

Group Array Creation . . . 643
Group Array Considerations . . . 643
Holes in a Group Array . . . 644
Locked Group Arrays . . . 647

© 2023 Cadence Design Systems, Inc.

Predefined Orientation Patterns for a Group Array . . . 649
Group Array SKILL Functions . . . 652
Creating a Group Array by Using the Copy Form . . . 652
Specifying Group Array Orientation Using the Array Pattern Form . . . 655
Saving a Custom Orientation Pattern for a Group Array . . . 658
Group Array Editing . . . 660
Modifying the Number of Cells in a Group Array . . . 662
Creating Holes in a Group Array . . . 664
Locking Group Array Spacing . . . 666
Converting a Mosaic to a Group Array . . . 667

10
Connectivity in Layout XL . . . 669

Pin Connectivity . . . 670
Benefits of Defining Pin Connectivity . . . 670
Defining Must-Connect Pins . . . 677
Defining Strongly Connected Pins . . . 679
Defining Weakly Connected Pins . . . 681
Defining Pseudo Parallel Connected Nets . . . 683
Checking Connectivity of Pins . . . 685
Propagating Nets . . . 685
Adding and Deleting Shapes on Nets . . . 687
Adding Shapes to a Net . . . 687
Removing Shapes from a Net . . . 688
Assigning and Unassigning Nets on Shapes . . . 689
Assigning Nets to Shapes in Pre-Selection Mode . . . 689
Assigning Nets to Shapes in Post-Selection Mode . . . 689
Unassigning Nets from Shapes in Pre-Selection Mode . . . 690
Unassigning Nets from Shapes in Post-Selection Mode . . . 690
Ways to Select Shapes Based on Nets . . . 690
Net Tracing Using the Mark Net Command . . . 693
Layers Specification for Tracing a Net . . . 695
Tracing Nets . . . 699
Saving Marked Nets . . . 707
Unmarking Nets . . . 707

© 2023 Cadence Design Systems, Inc.

Probing Nets, Instances, Pins, and Shapes . . . 708
Probing Objects Using Pre-Selection Mode . . . 710
Probing Objects Using Post-Selection Mode . . . 714

11
Microwave Commands . . . 717

Creating Transmission Lines . . . 718
Creating Bends . . . 719
Transmission Line Bend Styles . . . 721
Standard Bends . . . 721
Chamfer Bends . . . 723
Radial Bends . . . 724
Creating Advanced Transmission Lines . . . 725
Creating Tapers . . . 726

12
Troubleshoot Basic Editing Issues in Layout XL . . . 729

Troubleshoot Design Access Issues . . . 729
Cannot Find a Library . . . 730
Cannot Open a Cellview . . . 730
Cannot Write to a Cellview . . . 731
Cell Instances Are Missing . . . 731
ASCII MPP Template File Loading Returns a CIW Warning . . . 732
Troubleshoot Entered-Points Issues . . . 733
Points Entered Incorrectly . . . 734
Cannot Move Objects or Enter Points As Required . . . 735
Object Moves Instead of Stretching . . . 736
Object Changes Shape When Resizing . . . 737
Cannot Paste an Object . . . 737
Pcell Parameters Not Appearing . . . 738
Cannot Edit the Properties of Selected Instances . . . 739
Troubleshoot Mouse and Cursor Issues . . . 739
Pointer Changes Shape . . . 739
Cursor Sticks to Objects . . . 739
Cursor Snaps to the Wrong Object . . . 740

© 2023 Cadence Design Systems, Inc.

Cursor Does not Snap to a Point . . . 740
Right Mouse Button Does not Work . . . 740
Troubleshoot Object Selection Issues . . . 741
Unable to Select Instances . . . 742
Unable to Select Objects on a Layer . . . 742
Unable to Select All Objects During Edit-In-Place . . . 743
Unable to Select Objects in a Dense Design . . . 743
Unable to Select Copied Objects . . . 744
Selection Box Zooms the Cellview Instead of Selecting Objects . . . 745
How to Select Invisible Objects . . . 745
Troubleshoot Pcell Stretching Issues . . . 746
Pcell Stretch Handles Are Not Visible . . . 746
Difficulty Selecting Pcell Stretch Handles . . . 747
Troubleshoot Basic Command Issues . . . 748
Basic Editing Frequently Asked Questions . . . 749
Window Display FAQs . . . 749
Object Creation FAQs . . . 751
Object Editing FAQs . . . 755
Ruler-Related FAQs . . . 757
Layer-Related FAQs . . . 760

13
Snap Pattern Snapping for FinFET Devices in Layout Editor . . . 763

Turning On Snap Pattern Snapping . . . 764
Controlling the Display of Snap Patterns . . . 764
Object Snapping to Local Snap Pattern Shapes . . . 766
Object Placement Inside the Fin Boundary . . . 767
Object Snapping . . . 769
Snap-Pattern-to-Snap-Pattern Snapping . . . 770
Instance Snapping . . . 771
Shape Snapping . . . 772
Via Snapping . . . 774
Group Snapping . . . 775
Track Group Snapping . . . 776

© 2023 Cadence Design Systems, Inc.

Shape Snapping for Track Groups . . . 776
Instance Snapping for Track Groups in Detail Mode . . . 778
Object Snapping to Global Snap Pattern Grid . . . 779
Global Grid Anchoring . . . 780
Detail Mode Snapping to Specific Tracks of a Global Grid . . . 784
Snap Pattern Display Controls in the Palette Assistant . . . 786
Instance Placement Modes . . . 788
Abstract Mode Placement . . . 788
Manual Mode Placement Override . . . 790
Object Snapping in singleTrackCenter Snapping Mode . . . 791
Rectangles Created on a Layer with singleTrackCenter Snapping Mode . . . 791
Snapping Mode Behavior in Edit Commands . . . 793
Layer Shapes Exempted from Snapping by Using exceptOverlap . . . 794
Snap Pattern Display Packets . . . 794
Snap Pattern Display Based on Track Index . . . 797

14
Width Spacing Patterns in Layout Editor . . . 801

Controlling the Display of Width Spacing Patterns . . . 802
Object Snapping Behavior in Width Spacing Patterns . . . 808
Pin Snapping . . . 808
Region Shape Snapping . . . 812
Region Group Snapping . . . 813
Instance Snapping in Abstract Mode Placement . . . 813
Object Snapping to WSPs During Editing . . . 814
Pin Snapping to Width Spacing Patterns . . . 816
Dynamic Display During Pin Creation and Editing . . . 817
Controlling Transparency of Tracks With Alpha Blending . . . 818
Interactive Snapping . . . 820

A
Layout XL Basic Editing Forms . . . 825

Add Property Form . . . 828
Add Shape to Net Form . . . 829
Align Form . . . 830

© 2023 Cadence Design Systems, Inc.

Reference . . . 830
Align Using . . . 833
Spacing and Sorting . . . 834
Area and Density Calculator Form . . . 838
Array Pattern Form . . . 840
Assign Net Form . . . 841
Auto Via Assistant Options Form . . . 842
Background Cellview Form . . . 845
Choose Object for Label Creation Form . . . 846
Chop Form . . . 847
Copy Form . . . 850
Create Area Boundary Form . . . 856
Create Blockage Form . . . 857
Create Circle Form . . . 860
Create Cluster Boundary Form . . . 861
Create Custom Placement Area Form . . . 864
Create and Edit Clusters Form . . . 866
Create Encore Report Form . . . 869
Create Group Form . . . 871
Create Instance Form . . . 872
Create Label Form . . . 876
Manual . . . 877
Auto . . . 880
Create Measurement Form . . . 884
Ruler Shortcut Menu . . . 892
Create Multipart Path Form . . . 894
Create Path Form . . . 900
Path Options . . . 900
Create Pin Form . . . 904
Manual Mode . . . 904
Auto Mode . . . 908
Create Pins From Labels Form . . . 910
Create Polygon Form . . . 912
Create P&R Boundary Form . . . 913
Create Rectangle Form . . . 914
Create Row Form . . . 917

© 2023 Cadence Design Systems, Inc.

Definition . . . 917
Templates . . . 919
Create Method . . . 919
Create Slot Form . . . 922
Calculating Dimensions Associated with Slots . . . 926
Create Snap Boundary Form . . . 928
Create Via Form . . . 929
Single Via Mode . . . 929
Stack Via Mode . . . 944
Auto Via Mode . . . 946
Fast Edit Via Mode . . . 950
Create Via Options Form . . . 952
Cross Section Viewer Form . . . 956
Custom Show Layer/Custom Show Purpose Form . . . 962
Define Net/Pin Connections Form . . . 963
Delete Form . . . 964
Delete All Markers Form . . . 965
Delete Configuration Form . . . 967
Delete Marker Form . . . 968
Descend Form . . . 969
Descend Options Form . . . 970
Display Options Form . . . 971
Display Controls . . . 971
Zoom/Pan Controls . . . 978
Grid Controls . . . 979
Resolution . . . 980
Snap Modes . . . 981
Display Abstraction Controls . . . 981
Dimming . . . 982
Dynamic Display Form . . . 986
Setup . . . 987
Info Balloon . . . 988
Dynamic Measurement . . . 989
Highlight Aligned Edges . . . 989
Parameter Types for Shapes and Objects . . . 990
Dynamic Selection Assistant Options Form . . . 993

© 2023 Cadence Design Systems, Inc.

Dynamic Selection Shortcut Menu . . . 995
Edit Cellview Properties Form . . . 997
Cell Types . . . 999
Edit Properties Form . . . 1003
Connectivity Tab . . . 1005
Property Tab . . . 1007
ROD Tab . . . 1007
Property Editor Shortcut Menu . . . 1010
Attribute Tab of Various Objects . . . 1011
Edit Cut Pattern Form . . . 1057
Edit Synchronous Clone Form . . . 1061
Edit Via Array Pattern Form . . . 1062
Enter Points Form . . . 1063
Export Image Form . . . 1064
Export Label Form . . . 1067
Find Marker Form . . . 1068
Find/Replace Form . . . 1071
Flatten Form . . . 1075
Flip Form . . . 1078
Flip Vertical Form . . . 1079
Flip Horizontal Form . . . 1080
Freeze Pcell Instance Form . . . 1081
Group Return Form . . . 1082
Guard Ring Form . . . 1083
Halo Editor Form . . . 1084
Highlight Options Form . . . 1086
Instance to Mosaic Form . . . 1089
Layer Generation Form . . . 1090
Layout Editor Options Form . . . 1091
Editor Controls . . . 1092
Foundry Constraints . . . 1096
Gravity Controls . . . 1096
Tap . . . 1097
Snapping . . . 1099
Wire Editing . . . 1102
Layout Integrity Check Form . . . 1112

© 2023 Cadence Design Systems, Inc.

Load Multipart Path Template Form . . . 1113
Magnifier Options Form . . . 1114
Make Cell Form . . . 1117
Make Readonly/Editable Form . . . 1120
MarkNet Options Form . . . 1123
Options Tab . . . 1124
Via Layers Tab . . . 1128
Connected Layers Tab . . . 1129
Stop Layers Tab . . . 1130
Marker Text Window . . . 1132
Modify Corner Form . . . 1133
Modify Property Form . . . 1134
Move Form . . . 1135
Open Form . . . 1138
Pad Opening Info Form . . . 1139
Search Criteria . . . 1139
Sort . . . 1142
Reporting . . . 1142
Label Control . . . 1143
Promote Label . . . 1144
Pad Opening Info: Filter Form . . . 1146
Palette Options Form . . . 1147
Paste Form . . . 1150
Print Hierarchy Tree Form . . . 1151
Quick Align Form . . . 1152
Quick Align Shortcut Menu . . . 1155
Refresh Cellviews Form . . . 1157
Remaster Instances Form . . . 1158
Remove Shape From Net Form . . . 1160
Rename Command Name Form . . . 1161
Repeat Copy Form . . . 1162
Reshape Form . . . 1164
Restore View Form . . . 1166
Return To Level Form . . . 1167
ROD Subpart Form . . . 1168
Offset Subpath Fields . . . 1169

© 2023 Cadence Design Systems, Inc.

Enclosure Subpath Fields 1173
Subrectangle Fields 1174
Rotate Form 1182
Rotate Left Form 1184
Rotate Right Form 1185
Save a Copy Form 1186
Save All Mark Nets Form 1187
Save Configuration Form 1188
Save Multipart Path Templates Form 1189
Save/Restore Selection Set Form 1190
Save Via Variant Form 1191
Save View Form 1192
Save Recording Form 1193
Select Area Shape Type Form 1194
Select Array Member Form 1195
Selection Options Form 1196
Selection Protection Highlight Options Form 1202
Set Area View Level Form 1204
Set Default Application Form 1205
Set Pin Label Text Style Form 1207
Show Selected Set Window 1208
Size Form 1209
Split Form 1210
Stretch Form 1211
Summary Window 1218
Tap Form 1220
Track Pattern Editor Form 1222
Transmission Line Bend Form 1224
Transmission Line Form 1225
Transmission Line Taper Form 1226
Transparency Options Form 1227
Tree Form 1229
Uniquify Form 1230
Unmark Net Form 1231
View By Form 1232
Yank Form 1233

© 2023 Cadence Design Systems, Inc.

B
Layout XL Basic Editing Environment Variables 1235

System Evaluation of Environment Variables 1235
Ways to Set Environment Variables 1236
Changing Graphic Environment Variables That Are Cellview Properties 1238
abeFixShapePartitioning 1239
abutServerOn 1240
accessEdgesOn 1241
acuteAnglePath 1242
addDesignLppsToTech 1243
alignApplySpacing 1244
alignDirection 1245
alignLayer 1246
alignMatchRefLayer 1247
alignMatchRefSize 1248
alignMethod 1249
alignObjType 1251
alignReference 1252
alignRefMethod 1253
alignRefObjType 1255
alignReverseSort 1256
alignSnapToGridType 1257
alignSortMethod 1258
alignSpacingType 1259
alignSpacingValue 1260
alignToolBarReferenceObject 1261
alignUpdateLabelLayer 1262
alignUsing 1263
alignUsingInstanceDrawingMode 1264
allowPartialDelete 1265
alphaBlendingBoundaryFillValue 1266
alphaBlendingBoundaryHaloValue 1267
alphaBlendingBoundaryOutLineValue 1268
alphaBlendingFinFillValue 1269
alphaBlendingFinHaloValue 1270

© 2023 Cadence Design Systems, Inc.

alphaBlendingFinOutLineValue 1271
alphaBlendingTrackFillValue 1272
alphaBlendingTrackHaloValue 1273
alphaBlendingTrackOutLineValue 1274
anchor 1275
applyBounce 1276
arrayDisplay 1277
askSaveOnReturn 1278
autoCopyTemplateCellViewList 1279
autoCircleCreation 1280
autoExtractSaveEnabled 1281
autoLoadBindKeys 1282
autoModePinSize 1283
autoModePinSizeFromPath 1284
autoNetName 1285
autoPermutePins 1286
autoPinLayerFunctionsList 1287
autoPinRectHeight 1288
autoPinRectWidth 1289
autoRefresh 1290
autoSaveMarkNetSetupFile 1291
autoSetRefPoint 1292
autoStoreSelectionSetOn 1293
autoTapShape 1294
autoTapType 1295
autoViaOverrideVariantParameters 1296
autoviaOverrideVariantParameters 1297
autoZoomMode 1298
autoZoomPan 1299
autoZoomScale 1300
balloonFadeTime 1301
balloonOn 1302
balloonPauseTime 1303
balloonTransparency 1304
bannerFields 1305
blockageAllowPGNet 1307

© 2023 Cadence Design Systems, Inc.

blockageAttachOwner 1308
blockageAttributeMode 1309
blockageCreateMethod 1310
blockageDerivePoints 1311
blockageIsPushedDown 1312
blockageKeepOwner 1313
blockageMaterial 1314
blockageMaxDensity 1315
blockageSpacingType 1316
blockageSpacingValue 1317
blockageType 1318
bondPadLPP 1319
boundarySnappingLayer 1320
boundarySnapXGrids 1321
boundarySnapYGrids 1322
carDefaultGroup 1323
chainMode 1324
checkAllSPGridsEnclosures 1325
checkHalfWidthPathSeg 1326
checkLargeCoordinates 1327
chopArray 1328
chopCircleJustify 1329
chopCircleRadius 1330
chopLineJustify 1331
chopLineType 1332
chopLineUserWidth 1333
chopPRBDialog 1334
chopRectHeight 1335
chopRectJustify 1336
chopRectWidth 1337
chopShape 1338
chopSpecifySize 1339
chopSpecifySpacing 1340
circleRadius 1341
circleShapeMode 1342
clearAllMeasurementsUseEditMode 1343

© 2023 Cadence Design Systems, Inc.

clusterBoundaryAspectRatio 1344
clusterBoundaryUtilization 1345
commandVerboseLevel 1346
confirmDelete 1347
convertEachInstanceIntoMosaic 1348
copyColumns 1349
copyConnectivity 1350
copyCreateGroupArray 1351
copyGroupArrayLockXY 1352
copyIncrementInstName 1353
copyIncrementNetName 1354
copyOrder 1355
copyResetRowCol 1356
copyRows 1357
copySpacingMode 1358
copyToLayer 1359
copyXField 1360
copyYField 1361
createAsLabel 1362
createInstResetCDF 1363
createInstResetCDFParamsFile 1364
createPinLabel 1365
createPinPathSpineAware 1366
createPinSchematicAware 1367
createRectilinearBoundary 1368
createViaAlwaysUsesDefinedCalcMode 1369
cursorShape 1370
crossSectionCutLineColor 1371
crossSectionDisplayForwardCut 1372
crossSectionDisplayLayerName 1373
crossSectionDisplayNetName 1374
crossSectionDisplayUnusedLayer 1375
crossSectionOrientationHorizontal 1376
crossSectionSizeCutLine 1377
crossSectionSyncWithPaletteVisibility 1378
crossSectionViewModeAssistant 1379

© 2023 Cadence Design Systems, Inc.

defaultNewCellName 1380
defaultNewLibName 1381
defaultNewViewName 1382
defaultOpenCellName 1383
defaultOpenLibName 1384
defaultOpenViewName 1385
defaultToolName 1386
defCreatePinConnModel 1387
delDetachedBlockages 1388
deleteFluidShapes 1389
deleteTarget 1390
descendEIPPopupSelectionWindow 1391
descendPromptForView 1392
dimGridLines 1393
dimmingAutoEnabled 1394
dimmingInGray 1395
dimmingIntensity 1396
dimmingOn 1397
dimmingScope 1398
disableLayerFuncSnapToMfgGrid 1399
disableMissingLayerPopUp 1400
disableRoundPathStyle 1401
discardCVEditOnWindowClose 1402
displayFigGroupName 1403
displayFigGroupShapes 1404
displayInstanceOrientation 1405
displayMissingMasterMarkers 1406
displayOptionsFile 1407
displayPCellOrientation 1408
displayPinNames 1409
displayPRBoundaryAtCurrentLevelOnly 1410
displayPRHighestResolution 1411
displayRefPoint 1412
displayResolution 1413
displayRowName 1414
displayRowSite 1415

© 2023 Cadence Design Systems, Inc.

displayRulerDistanceAtEnd 1416
displayRulerInHierarchy 1417
displayRulerMiddlePoint 1418
displaySchDashText 1419
displaySelectionHighlightWhenDimmed 1420
displaySigDigits 1421
displayStartLevel 1422
displayStopLevel 1423
displayStretchHandles 1424
displayViaShapes 1425
doNotShowChopPRBDialog 1426
dotsOn 1427
dotStyleMajorGrid 1428
dragDisplayNetName 1429
drawDottedGridOn 1430
drawEngine 1431
drawGridOn 1432
drawInstancePinNames 1433
drawInstancePins 1434
drawLabelsSurround 1435
drawPRBoundary 1436
drawSmallLabelOrigin 1437
drawSurroundingOn 1438
drawTraversedInstBbox 1439
drawTraversedInstName 1440
drdEditSmartSnapAllowedWidthSnap 1441
drfPath 1442
dynamicHilightOn 1443
dynamicMeasurementBackgroundColor 1444
dynamicMeasurementDrawBackground 1445
dynamicMeasurementDisplay 1446
dynamicMeasurementEdge 1447
dynamicMeasurementFontSize 1448
dynamicMeasurementFromCursor 1449
dynamicMeasurementLayerScope 1450
dynamicMeasurementTextColor 1451

© 2023 Cadence Design Systems, Inc.

dynamicProbe 1452
editInstanceBoundaryMode 1453
editorOptionsFile 1454
eipHighlightColor 1455
enableAlignToolBarReferenceObject 1456
enableCopyAndPaste 1457
enableEIPEnhancements 1458
enableLayoutIntegrityCheck 1459
exceptOverlapSnapMode 1460
expandPinBusExpression 1461
extendPathLengthCheck 1462
extractAfterGenerateAll 1463
extractCellviewIllegalConnectionLimit 1464
extractCellviewInvalidConnectionLimit 1465
extractCellviewOpenViolationLimit 1466
extractCellviewShortViolationLimit 1467
extractCellviewUnnecessaryPseudoParallelConnectionLimit 1468
extractConnectUnassignedInstTerms 1469
extractConsiderAllCutLayersInVias 1470
extractCheckShortedTerminalsInPcells 1471
extractDeepOffPinConnectionsToLevel 1472
extractEnabled 1474
extractEnableVirtualConnections 1475
extractHierExtractSearchDepth 1476
extractIgnoreColorLockState 1477
extractKeepShapeAssignment 1478
extractNetInvalidConnectionLimit 1479
extractNetOpenViolationLimit 1480
extractNetShortViolationLimit 1481
extractPromoteAllStopShapes 1482
extractReinitOnStopLevelChange 1483
extractRemoveStickyProperty 1484
extractRunOnInit 1485
extractStopLevel 1486
extractVerifyAnalogSigType 1487
extractVerifyBigMPP 1488

© 2023 Cadence Design Systems, Inc.

extractVerifyClockSigType 1489
extractVerifyGroundSigType 1490
extractVerifyIllegalHierConnections 1491
extractVerifyIllegalHierConnectionsIgnoreUnassignedHierShapes 1492
extractVerifyIllegalOverlapsForStoppedColoredShapes 1493
extractVerifyIncompleteMustConnectAllPins 1494
extractVerifyIncNetsWithShapesOnNonRoutingLayers 1495
extractVerifyInvalidConnections 1496
extractVerifyMustConnectMarkers 1497
extractVerifyOpenViolations 1498
extractVerifyOpticalPortRadiusMatch 1499
extractVerifyPowerSigType 1500
extractVerifyResetSigType 1501
extractVerifyScanSigType 1502
extractVerifySignalSigType 1503
extractVerifyTieHighSigType 1504
extractVerifyTieLowSigType 1505
extractVerifyTieOffSigType 1506
extractVerifyWellSubstrateConnections 1507
extractVerifyUnimplementedInstTerms 1508
extractVerifyUnnecessaryPseudoParallelConnections 1509
extractVerifyWeakMarkers 1510
extractVirtualConnectionSeparator 1511
filteredNetName 1512
filterSizeForResolution 1513
findReplaceReportFileName 1515
fitIntoClusterBoundary 1516
fixedWidthPaths 1517
flattenExcludePcellListFile 1518
flattenMode 1519
flattenPCells 1520
flattenVias 1521
flipAroundCombinedCenter 1522
flipReference 1523
followRoutingDirection 1525
fullSelectionMode 1526

© 2023 Cadence Design Systems, Inc.

ghostOfDraggedObjects 1527
glyphTransparency 1528
gravityApertureInPixels 1529
gravityDistance 1530
gravityEditCellViewOnly 1531
gravityOn 1532
gravitySelectableOnly 1533
gravityType 1534
gravityUse5PixelApertureWhenOff 1535
grEnclosedBy 1536
gridMultiple 1537
gridSpacing 1538
groupArrayDragMode 1539
groupArrayXSpacing 1540
groupArrayYSpacing 1541
highlightSelProtectedItems 1542
haloBottomOffset 1543
haloLeftOffset 1544
haloRightOffset 1545
haloTopOffset 1546
highlightAlignedEdges 1547
highlightAlignedEdgesColor 1548
highlightAlignedEdgesWidth 1549
iconsOn 1550
ignoreBounce 1551
ignoreParallelMinWidthForSquarePinResize 1552
ignorePCellSnapPropertyForDevice 1553
instanceDrawingMode 1554
instanceSelectionMode 1555
instanceVisibilityMode 1556
instCellName 1557
instColumns 1558
instDelX 1559
instDelY 1560
instHalosEnabled 1561
instLabel 1562

© 2023 Cadence Design Systems, Inc.

instLibName . . . 1563
instMosaicCreation . . . 1564
instName . . . 1565
instPhysicalOnly . . . 1566
instRows . . . 1567
instViewName . . . 1568
keepCopiedInsideGroup . . . 1569
keepFirstTermName . . . 1570
keepGuardRingEndsConnected . . . 1571
keepWiresConnectedToNonWires . . . 1572
labelAttach . . . 1573
labelAutoConn . . . 1574
labelAutoInst . . . 1575
labelDrafting . . . 1576
labelExpand . . . 1577
labelFontStyle . . . 1578
labelHeight . . . 1579
labelJustify . . . 1580
labelKeepLabel . . . 1581
labelLPP . . . 1582
labelMode . . . 1583
labelOverbar . . . 1584
labelPurpose . . . 1585
labelText . . . 1586
leCheckMissingLayers . . . 1587
leCheckMissingMasters . . . 1588
leChopPreservePathSeg . . . 1589
leDispNetExpr . . . 1590
leWindowBBox . . . 1591
lockAngles . . . 1592
lppVisibilityMode . . . 1593
magAnchorPosition . . . 1594
magConfined . . . 1595
magCursorOffset . . . 1596
magDisplayMode . . . 1597
magFloatPosition . . . 1598

© 2023 Cadence Design Systems, Inc.

magOffsetCorner . . . 1599
magScale . . . 1600
magSize . . . 1601
maintainConnections . . . 1602
makeCellCellType . . . 1603
makeCellGridAlignment . . . 1604
makeCellUseInstanceDrawingMode . . . 1605
markerAccessMode . . . 1606
markerDeleteAccessModeIsEdit . . . 1607
markerDeleteConfirmModeChange . . . 1608
markerDeleteScope . . . 1609
markerDeleteSeverities . . . 1610
markerDeleteSources . . . 1611
markerDeleteSourcesMode . . . 1612
markerSearchBy . . . 1613
markerSearchScope . . . 1614
markerSeverity . . . 1615
markerZoom . . . 1616
markerZoomScale . . . 1617
markNetAutoColorCycle . . . 1618
markNetConnectedLayer . . . 1619
markNetExcludePurposes . . . 1620
markNetHilightObj . . . 1621
markNetNumThreads . . . 1622
markNetOptionsFile . . . 1623
markNetRange . . . 1624
markNetSaveViaInfo . . . 1625
markNetSelectOverlapLayer . . . 1626
markNetStartLevel . . . 1627
markNetStopLayer . . . 1628
markNetStopLevel . . . 1629
markNetThickline . . . 1630
markNetViaLayer . . . 1631
maskLayoutDefaultApp . . . 1632
maxAutoStoreSelectionObj . . . 1633
maxAutoStoreSelectionSet . . . 1634

© 2023 Cadence Design Systems, Inc.

maxPathSegInSpineAreaSelection 1635
measurementDisplayOn 1636
measurementTextHeight 1637
mergeEnable 1638
mergeScope 1639
mergeVias 1640
mergeXStippleShapes 1641
meshNBThread 1642
minInstSizeInPixelsForStretchHandles 1643
modalCommands 1644
modCornerRadius 1645
modCornerType 1646
moveConnectedInstancesAndWires 1647
mppASCIIFileName 1648
mppTemplate 1649
multiMfgResolutionSnap 1650
mustConnectAllPins 1651
multiThreadInfo 1652
muteUnstretchableOption 1653
netBasedSelectionNetName 1654
netBasedSelectionOn 1655
netBasedSelectionReverse 1656
netExprDisplay 1657
netName 1658
netNameDisplayDrawBackground 1659
netNameDisplayDrawOnTop 1660
netNameDisplayEnabled 1661
netNameDisplayFilter 1662
netNameDisplayFilterDefaultList 1663
netNameDisplayFilterLayers 1664
netNameDisplayFilterMode 1665
netNameDisplayFontStyle 1666
netNameDisplayMaxLabelSize 1667
netNameDisplayMaxLabelSizeFull 1668
netNameDisplayNetShow 1669
netNameDisplayScope 1670

© 2023 Cadence Design Systems, Inc.

netNameDisplaySkipInstances 1671
netNameDisplaySkipPcells 1672
netNameDisplaySkipLayers 1673
netNameDisplaySkipVias 1674
netNameDisplayTermShow 1675
netNameDisplayTermSkipInstances 1676
netNameDisplayTermSkipPcells 1677
netNameDisplayUserColor 1678
netNameDisplayUseShapeColor 1679
netsOn 1680
nPtsToTaper 1681
numLevels 1682
numSides 1683
openSavedCellViewInWindow 1684
openToStopLevel 1685
orientation 1686
originMarkersOn 1687
overrideFontName 1688
packetDialogBoxes 1689
padLabelLPP 1690
partialSelect 1691
partialSelectProximityMaxPixel 1692
partialSelectionMode 1693
pasteRef 1694
pathBeginExt 1695
pathCL 1696
pathEndExt 1697
pathJustify 1698
pathOffset 1699
pathStyle 1700
pathWidth 1701
pinAccessDir 1702
pinDefaultObjectType 1703
pinLabelPlacementMode 1704
pinLayerPurpose 1705
pinLPP 1706

© 2023 Cadence Design Systems, Inc.

pinMode . . . 1707
pinSelectionMode . . . 1708
pinShape . . . 1709
pinTextCreateAsLabel . . . 1710
pinTextLayer . . . 1711
pinTextLayerNames . . . 1712
pinTextPurpose . . . 1713
pinTextPurposeNames . . . 1714
pinTextSamePurpose . . . 1715
pinTextSameLayer . . . 1716
pinTextOrientation . . . 1717
pinWidth . . . 1718
pinXPitch . . . 1719
pinYPitch . . . 1720
pointSelectAperture . . . 1721
pointSelectCycleMode . . . 1722
polygonArc . . . 1723
preservePins . . . 1724
preservePinFigs . . . 1725
preserveRodObjects . . . 1726
preserveTerminalContacts . . . 1727
preserveTermName . . . 1728
preventDeletionOfLogicalInstance . . . 1729
probeCycleBool . . . 1730
probeFilledInstance . . . 1731
probeFilledInstanceParam . . . 1732
probeHighlight . . . 1733
probeHiliteLppList . . . 1734
probeHiliteLppString . . . 1735
probeInstDisplayRadio . . . 1736
probeNetDisplayCheckInstance . . . 1737
probeNetDisplayCheckPins . . . 1738
probeNetDisplayFlightLines . . . 1739
probeNetDisplayHierarchyLevel . . . 1740
probeNetNameFilterByName . . . 1741
probeNetNameFilterByNameDefaultList . . . 1742

© 2023 Cadence Design Systems, Inc.

probeNetNameFilterByNameMode . . . 1743
probeNetStopAtDisplayLevel . . . 1744
probePromptBusBit . . . 1745
probeShapeDisplay . . . 1746
probeTermDisplayRadio . . . 1747
probeVisibleLpp . . . 1748
propEditHideTopSection . . . 1749
propEditDefaultSelectAll . . . 1750
propEditorHeight . . . 1751
propEditorWidth . . . 1752
propertyEditorAutoResize . . . 1753
propEditStretchableTree . . . 1754
protectedObjHighlightHaloLevel . . . 1755
protectedObjHighlightHaloMode . . . 1756
protectedObjHighlightHaloTrans . . . 1757
protectedObjHighlightHaloWidth . . . 1758
pteAllLayersScope . . . 1759
pteApplyParentState . . . 1760
pteAutoRedrawMode . . . 1761
pteCheckableColumnClickMode . . . 1762
pteCollapseLayersInViewBy . . . 1763
pteColoredLppSortOrder . . . 1764
pteColorMixedSort . . . 1765
pteCompactMPT . . . 1766
pteDeferRedrawOnMouseMove . . . 1767
pteDisplayActiveLppInForeground . . . 1768
pteEnableMouseBindings . . . 1769
pteFindModeOn . . . 1770
pteIncludeCutLayers . . . 1771
pteInfraDefValueObjGrids . . . 1772
pteInfraObjSelFollowVis . . . 1773
pteInfraWirebondProfileColors . . . 1774
pteInverseColorScheme . . . 1775
pteLayerSetActivationMode . . . 1776
pteLayerSetManagerLeftClickMode . . . 1777
pteLayersSelectabilityDepth . . . 1778

© 2023 Cadence Design Systems, Inc.

pteLayersVisibilityDepth 1779
pteLinkColorVisibilityToShapeVisibility 1780
pteLppGroup 1781
pteLppsDisplayOrderControl 1782
pteLSManagerAsAssistant 1783
pteLSManagerRespectFilters 1784
pteMPTAwareSelectionTrigger 1785
pteMPTMode 1786
pteMPTScope 1787
pteMovableTools 1788
pteMultiSelectionMode 1789
pteProhibitToggleOnMouseMove 1790
pteRetainFilterSettings 1791
pteRoutingOnlyScope 1792
pteShowColorControlsForMPT 1793
pteShowColoredLSManager 1794
pteShowLSManager 1795
pteShowMPTScope 1796
pteShowUsedSystemLpps 1797
pteUsedLayersEipScope 1798
pteWindowSynchronizationMode 1799
QAHoldTimeMilliSec 1800
QAMaxPixelsPerMilliSec 1801
quickAlignMode 1802
quickAlignSameLayerTarget 1803
quickAlignSnapTarget 1804
quickAlignSpacingType 1805
quickAlignUserSpacing 1806
realColorSelection 1807
realignLabelToPinCenter 1808
rectangularWSPPinAsSquare 1809
rectBaseMode 1810
rectHeight 1811
rectOnSnapPatternGrid 1812
rectSmartSnapEnable 1813
rectSmartSnapOnActiveLayer 1814

© 2023 Cadence Design Systems, Inc.

rectWidth . . . 1815
recursionCheck . . . 1816
remasterInstCheckTerms . . . 1817
remasterInstSearchCellText . . . 1818
remasterInstSearchLibText . . . 1819
remasterInstSearchViewText . . . 1820
remasterInstUpdateCellText . . . 1821
remasterInstUpdateLibText . . . 1822
remasterInstUpdateViewText . . . 1823
removeChopShape . . . 1824
removeEmptyNetTerm . . . 1825
removeTerminals . . . 1826
replaceFigs . . . 1827
resetOrientation . . . 1828
reshapeType . . . 1829
retainSelectionSet . . . 1830
reuseExistingFrozenMaster . . . 1831
rodAutoName . . . 1832
rotateAroundCombinedCenter . . . 1833
rotateReference . . . 1834
rotateInstParam . . . 1836
rowAbutFirstUnit . . . 1837
rowAbutNumberOfUnits . . . 1838
rowAbutUnit . . . 1839
rowBaseName . . . 1840
rowCreateMethod . . . 1841
rowChannelWidth . . . 1842
rowFirstUnitOrient . . . 1843
rowFlipFirstUnit . . . 1844
rowFlipUnit . . . 1845
rowNumberOfRows . . . 1846
rowPartitionBottomOffset . . . 1847
rowPartitionLeftOffset . . . 1848
rowPartitionRightOffset . . . 1849
rowPartitionTopOffset . . . 1850
rowNumberOfSites . . . 1851

© 2023 Cadence Design Systems, Inc.

rowSiteEnable . . . 1852
rowSiteSymmetry . . . 1853
rowUnitOrient . . . 1854
rulerColorPacketName . . . 1855
rulerDisplayType . . . 1856
rulerEdgeMode . . . 1857
rulerFontBold . . . 1858
rulerFontSize . . . 1859
rulerLabelBackgroundColor . . . 1860
rulerLimitDecimals . . . 1861
rulerSegmentModeType . . . 1862
rulerSegSnapMode . . . 1863
rulerSnapMode . . . 1864
saveACopyLibName . . . 1865
saveFindReplaceReportToFile . . . 1866
saveRulers . . . 1867
schematicDefaultApp . . . 1868
schematicDefaultTier . . . 1869
schematicSymbolDefaultApp . . . 1870
scrollPercent . . . 1871
segSnapMode . . . 1872
selectCreatedObj . . . 1873
selectHilightPacket . . . 1874
selectionAperture . . . 1875
selectMarkersAtLast . . . 1876
selectObjectLargerThanWindow . . . 1877
selectObjectInEditCvFirst . . . 1878
selectOnlyInstWithSelectableShapes . . . 1879
selectPartialPathHighlightStyle . . . 1880
selectProtectedItems . . . 1881
selectUnderCursorMenuAlphaSorting . . . 1882
selectUnderCursorMenuLimit . . . 1883
setDxDyToZero . . . 1884
setupConstraintGroup . . . 1885
showFindReplaceReport . . . 1886
showAnglesSigDigits . . . 1887

© 2023 Cadence Design Systems, Inc.

showWhatsNew . . . 1888
slotEdgeToSlotOffset . . . 1889
slotEnable . . . 1890
slotEnableRect . . . 1891
slotEnablePath . . . 1892
slotInterSlotSpacing . . . 1893
slotLength . . . 1894
slotRowSpacing . . . 1895
slotShapeMode . . . 1896
slotSpecifyArea . . . 1897
slotStaggerOffset . . . 1898
slotWarnWhenMoreThanRows . . . 1899
slotWidth . . . 1900
snapDeviceInDetailMode . . . 1901
snapMode . . . 1902
snapPatternDisplayTrackIndex . . . 1903
snapPatternDrawingStyle . . . 1904
snapPatternFastDetailModeSnapping . . . 1905
snapPatternGlobalGridAnchoring . . . 1906
snapPatternIndexSnappingLayers . . . 1907
snapPatternSnapping . . . 1908
snapPatternSnappingDepth . . . 1909
snapPatternSnappingDepthMethod . . . 1910
snapPatternSnappingMode . . . 1911
snapPatternSnapToXYGridFirst . . . 1912
snapRectToWSPGrid . . . 1913
snapResizeWireObjects . . . 1914
snapResizeWireObjectsMaxDragLimit . . . 1915
snapResizeWireObjectsMaxLimit . . . 1916
snapStdCellInDetailMode . . . 1917
snapToGrid . . . 1918
snapToMfgGrid . . . 1920
snapUsingInstanceAsReference . . . 1921
softFence . . . 1922
spineSelection . . . 1923
startLevel . . . 1924

© 2023 Cadence Design Systems, Inc.

statusBarXYdXdYLength 1925
statusToolbarFields 1926
stopLevel 1929
stretchPcellApplyToName 1930
stretchHandlesLayer 1931
stretchViaMode 1932
supressFreezePcellUI 1933
tapConnectedFigsOnly 1934
taperStyle 1935
tapLayerFilter 1936
tapPurposeList 1937
tapType 1938
targetColor 1939
targetLineStyle 1940
targetWidth 1941
termDirection 1942
termName 1943
textJustificationOn 1944
tinyInstDetail 1945
transientRulerColorPacketName 1946
transparentGroup 1947
trimEnable 1948
trueColorProbe 1949
trlBendFac 1950
trlBendStyle 1951
trlChamFac 1952
trlNPtsToPl 1953
trlRadFac 1954
trlWidth1 1955
trlWidth2 1956
turnInfixOffWhenSmartSnapping 1957
undoRedoHistoryLimit 1958
updateLabelLPPWithPin 1959
updatePCellIncrement 1960
useAlphaBlending 1961
useDynamicHilightPacket 1962

© 2023 Cadence Design Systems, Inc.

useFlipReference . . . 1963
useOverrideFontName . . . 1964
useRelativeXY . . . 1965
useRotateReference . . . 1966
useRulerSnapModes . . . 1967
useTapPurposeList . . . 1968
useTrueBBox . . . 1969
validLppFilterOn . . . 1970
viaAlignCuts . . . 1971
viaAlignCuts . . . 1972
viaAllowOverridePurposes . . . 1973
viaAnnotateOverlapsWithNoValidSolution . . . 1974
viaAnnotateViasNotRetained . . . 1975
viaAnnotateViasWithAddedMetal . . . 1976
viaAnnotateViasWithNoAddedMetal . . . 1977
viaAnnotateViasWithSingleCut . . . 1978
viaAttemptToUseBridgeVias . . . 1979
viaAutoFromArea . . . 1980
viaAutoOnlyOnTopLayers . . . 1981
viaAutoOnlyOnTopLayers . . . 1982
viaAutoViaAlignment . . . 1983
viaAutoViaAlignment . . . 1984
viaAutoViaCutArrayDirection . . . 1985
viaAutoViaCutArrayDirection . . . 1986
viaAutoViaEnableMaxStackDepth . . . 1987
viaAutoViaEnclosureDirection . . . 1988
viaAutoViaEnclosureDirection . . . 1990
viaAutoViaExcludeNetsName . . . 1992
viaAutoViaExcludePowerAndGroundNets . . . 1993
viaAutoViaHints . . . 1994
viaAutoViaMaxStackDepth . . . 1995
viaAutoViaMinNumCuts . . . 1996
viaAutoViaMinNumCuts . . . 1997
viaAutoViaOnSelectedNets . . . 1998
viaAutoViaPreviewShow . . . 1999
viaAutoViaPreviewShow . . . 2000

© 2023 Cadence Design Systems, Inc.

viaCheckBlockageTypes . . . 2001
viaCheckBlockageTypes . . . 2002
viaCheckMinNumCut . . . 2003
viaCheckPlateMaxWidth . . . 2004
viaCheckPlateMinWidth . . . 2005
viaCheckShapeConn . . . 2006
viaCheckUserSet . . . 2007
viaColumns . . . 2008
viaCompuEnclosures . . . 2009
viaComputeFromMode . . . 2010
viaConnectivityMode . . . 2011
viaConstraintGroup . . . 2012
viaCreateInGroup . . . 2013
viaCreateInRoute . . . 2014
viaCreateInRoute . . . 2015
viaCutClassInAuto . . . 2016
viaCutClassInAuto . . . 2017
viaCutSpacingMethod . . . 2018
viaCutClassOrientation . . . 2019
viaCutClassOrientation . . . 2020
viaDRCCheckHierarchyRangeMode . . . 2021
viaEnableAddMetal . . . 2022
viaEnableAnnotation . . . 2023
viaEnableDRCCheck . . . 2024
viaEnableUserSkillCallbacks . . . 2025
viaEnableValidPurposes . . . 2026
viaEnableValidPurposes . . . 2027
viaExtendEnclosureBeyondOverlap . . . 2028
viaExtendEnclosureBeyondOverlap . . . 2029
viaFastEditFromSelectedVias . . . 2030
viaFillEnclosureOverlap . . . 2031
viaFillEnclosureOverlap . . . 2032
viaFillpolygonalOverlap . . . 2033
viaFixOverlapShapeViolations . . . 2034
viaHideEnclosures . . . 2035
viaIgnorePurposes . . . 2036

© 2023 Cadence Design Systems, Inc.

viaIgnorePurposes 2037
viaMaximizeCuts 2038
viaMaximizeCuts 2039
viaMaxStackDepth 2040
viaMode 2041
viaMultiProcessMode 2042
viaMultiProcessNumCPU 2043
viaOverrideAddMetalConstraint 2044
viaOverrideAlignmentConstraint 2045
viaOverrideCutArrayDirConstraint 2046
viaOverrideCutClassOrientConstraint 2047
viaOverrideEnclosureDirConstraint 2048
viaOverrideMaximizeCutsConstraint 2049
viaOverrideMinNumCutsConstraint 2050
viaOverrideOffCenterViaConstraint 2051
viaParamCalcMode 2052
viaPreventDRCOnOverlap 2054
viaPreventDRCWithNeighbors 2055
viaPreventOffCenterVia 2056
viaPrintStatisticsReport 2057
viaPrintStatisticsReport 2058
viaPrintStatisticsReportByStackDepth 2059
viaTryWithSmallerCutClass 2060
viaRows 2061
viaShowAllConstraintOverride 2062
viaSelectionMode 2063
viaStackColumns 2064
viaStackRows 2065
viaStackSelection 2066
viaStackSelectionMatchingSizes 2067
viaUseEffectiveCuts 2068
viaUseEntireOverlap 2069
viaUseRowColDefault 2070
viaUseEntireOverlap 2071
viaValidateDRCStandalone 2072
viaValidPurposes 2073

© 2023 Cadence Design Systems, Inc.

viaValidPurposes 2074
viaWallFromSelectedShapesEnabled 2075
viaWallKeepViasEnabled 2076
viaWallPreviewEnabled 2077
viewNameToTypeList 2078
xSnapSpacing 2079
ySnapSpacing 2080
yankShape 2081
zoomToFig 2082
Environment Variables Used Internally 2083

C
Layout XL Assistants 2085
Auto Via Assistant 2086
Analyze Tab 2086
Create Tab 2090

D
Layout Editor Plug-Ins 2095

E
Bindkeys in Virtuoso Layout Suite 2099

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

1

Layout Editor Basics

The Virtuoso® Layout Suite XL layout editor (Layout XL) is an editing tool that provides the basic creation and editing capabilities to work with layout designs. The layout editor lets you create custom layouts or edit any existing layouts. It lets you work with objects, such as rectangles, pathSegs, vias, and multipart paths, and hierarchical designs. You can also use the layout editor to view the connected shapes in a design.

For information about connectivity driven editing, see Virtuoso Layout Suite XL: Connectivity Driven Editing User Guide.

Before you start using the layout editor, it is good to become familiar with the basics of the Virtuoso Studio design environment, such as the Command Interpreter Window (CIW), cellviews, and workspaces. For information, see Virtuoso Studio Design Environment. For information on licensing in the Virtuoso Studio design environment, see *Virtuoso Software Licensing and Configuration User Guide*.

Related Topics

- Starting the Design Environment for Layout XL
- Layout XL Launched from a Schematic View
- Starting Layout XL from a Layout View
- Starting Layout XL from a Physical Configuration View
- Layout XL Constraint Group
- Layout XL Desktop
- Working with Designs
- Basic Command Functions in Layout XL
- Bindkeys in Virtuoso Layout Suite
- Layout Editor Default Setup

© 2023 Cadence Design Systems, Inc.

- Technology File
- Workspaces in Layout XL
- Display and Selection Controls
- Layout Editor Strokes

Reserved Parameter and Property Names for Setting Up Layout XL

Virtuoso Layout Suite XL requires the following parameter and property names to be reserved for correctly setting up the environment.

Parameter/Property Name	Reserved Name Used for Layout XL Setup
Capacitance	c C
Fingers	finger numFingers numFinger
Inductance	l L
Parallel-connected factor	m M
Resistance	r R
Segments	segments
Series-connected factor	s S
Transistor	w

For each parameter, the following is the name that takes precedence:

- Parameter name set at the instance level, if an instance override is set
- Parameter name defined at the cell level, if an instance override is not set
- The first parameter name defined in Common Description Format (CDF), if different names are set at the instance and cell levels

Related Topics

Layout Editor Basics

© 2023 Cadence Design Systems, Inc.

Starting the Design Environment for Layout XL

To use Layout XL, you must have access to one of the following design environments:

- Virtuoso Physical Design Environment (`layoutPlus`)
- Virtuoso Front to Back Design Environment (`virtuoso`)

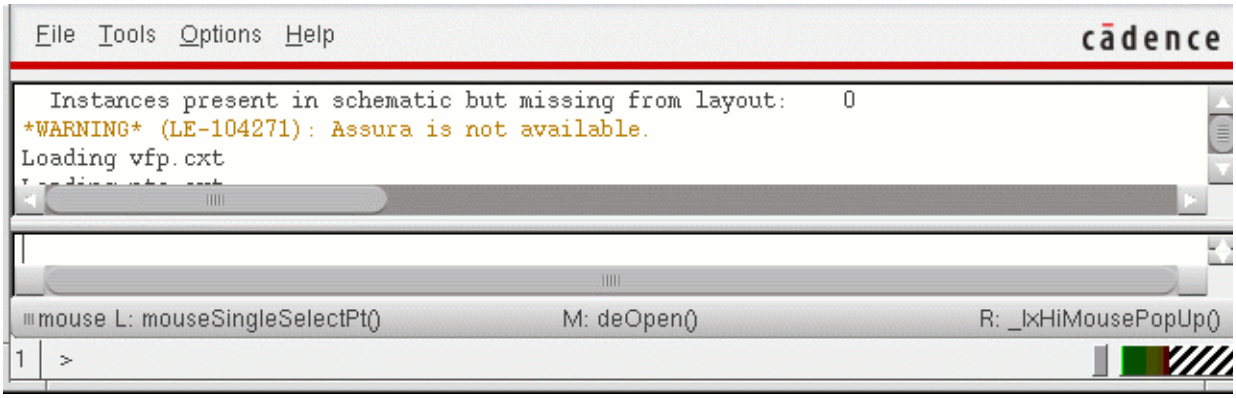
To start a design environment that includes the Layout XL software:

1. In a terminal window, move to the top-level directory of the design that you want to edit.

   The top-level directory is the directory that contains the library definitions (`cds.lib`) file for the design in question. The library definitions file defines the physical locations of all the reference and design libraries. Launching Layout XL loads the library initialization (`libInit.il`) files related to all these reference and design libraries.

2. Type the executable name of the design environment you want to launch; for example, `virtuoso`.

   The Command Interpreter Window (CIW) is displayed. You use the CIW to launch applications on specific cellviews, perform general, non-tool-specific operations, enter SKILL commands, and view messages and warnings.



After you have started the design environment, you can launch Layout XL from any of the following views:

- A schematic view, with or without a physical configuration view. See Layout XL Launched from a Schematic View.
- A layout view, with or without a connectivity reference. See Starting Layout XL from a Layout View.
- A physical configuration view. See Starting Layout XL from a Physical Configuration View.

© 2023 Cadence Design Systems, Inc.

Related Topics

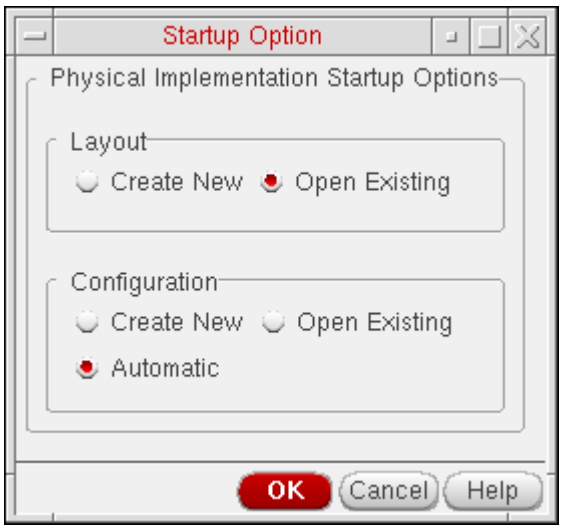
Command Interpreter Window

Layout XL Launched from a Schematic View

A schematic cellview is the only valid connectivity reference for a Layout XL session. If you do not have a schematic view of your design, you first need to generate one.

When you start Layout XL from a schematic, you need to specify:

- Layout: Whether to create a new layout view or use an existing one.
- Configuration: Whether the layout view is to be opened in the context of a new, an existing, or an automatic physical configuration view.



If you do not need to use any of the features provided by the *Configure Physical Hierarchy* command, you can choose *Automatic* mode. This ensures that Layout XL functions the same way as in previous releases, handling all operations related to the physical configuration view automatically.

When Layout XL initializes, the schematic view is re-opened in the context of the physical configuration view being used for the session, which might be different from the configuration the schematic was using previously. The schematic window banner updates to indicate the name of the physical configuration view currently being used. You might need to re-extract the schematic view to take into account the settings in the physical configuration view. The system informs you if this extraction is required.

© 2023 Cadence Design Systems, Inc.

If you close the Layout XL session by closing the layout window only, the schematic view is re-opened with no physical configuration context.

Related Topics

Generating a Schematic View from a CDL Netlist

Starting Layout XL with an Automatic Physical Configuration View

Starting Layout XL with a New Physical Configuration View

Starting Layout XL with an Existing Physical Configuration View

Configuring the Physical Hierarchy

Generating a Schematic View from a CDL Netlist

If your connectivity reference is a CDL netlist, you cannot import it directly into Layout XL. You must first generate a schematic from the netlist.

To generate a schematic view from a CDL netlist:

1. From the CIW, select *File – Import – Spice*.

   The Virtuoso Spice In form opens.
2. Set the *Netlist Language* to *CDL*.
3. Enter the file name in *Netlist File*.
4. Click *OK*.
5. Start Layout XL using any method of your choice.
   - Starting Layout XL with an Automatic Physical Configuration View
   - Starting Layout XL with a New Physical Configuration View
   - Starting Layout XL with an Existing Physical Configuration View

Related Topics

Netlist Import Using Spice In

© 2023 Cadence Design Systems, Inc.

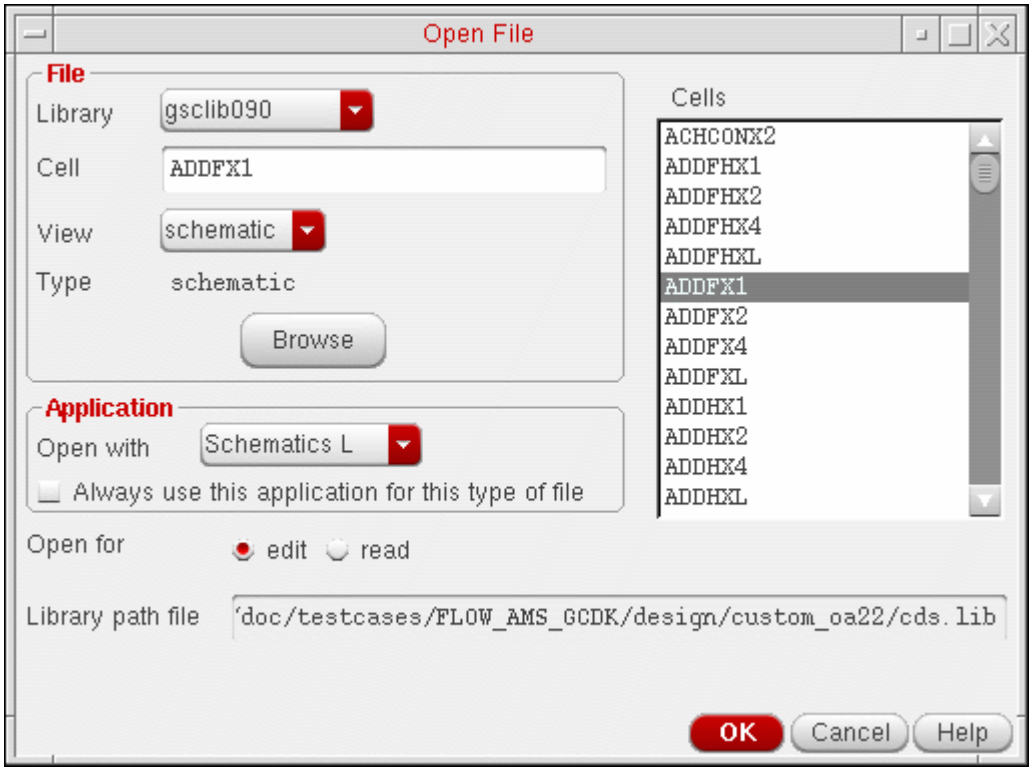
Starting Layout XL with an Automatic Physical Configuration View

If you do not need to use any of the features provided by the Configure Physical Hierarchy command, you should launch Layout XL in *Automatic* mode. This ensures that Layout XL handles all operations related to the physical configuration view automatically.

To start Layout XL from a schematic view with *Automatic* mode of physical configuration view:

1. From the CIW, choose *File – Open*.

   The Open File form is displayed.



2. Specify the *Library* and *Cell* names.
3. Select *schematic* from the *View* list and click *OK*.

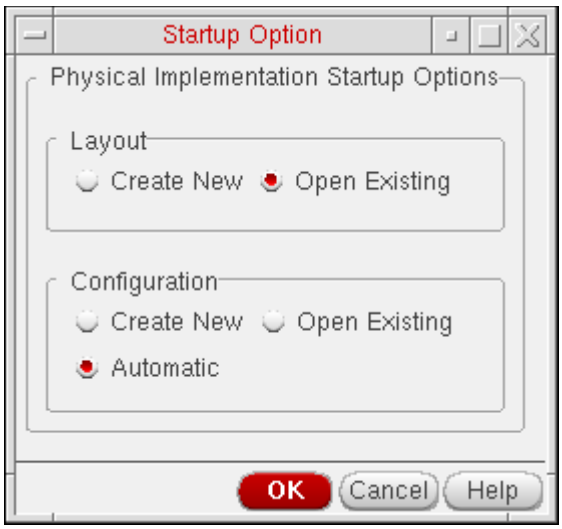
   The Virtuoso Schematic Editor L Editing window appears displaying the cellview you specified.

4. From the schematic window, choose *Launch – Layout XL*.

   Layout XL appears on the *Launch* menu only if the Layout XL software is installed. If you do not see *Layout XL* in the menu, check with your system administrator.

© 2023 Cadence Design Systems, Inc.

The Startup Option form is displayed.



5. In the *Layout* group box, choose whether you want to create a new layout cellview or open an existing one.
6. In the *Configuration* group box, choose *Automatic* and then click *OK*.

   One of the following happens:

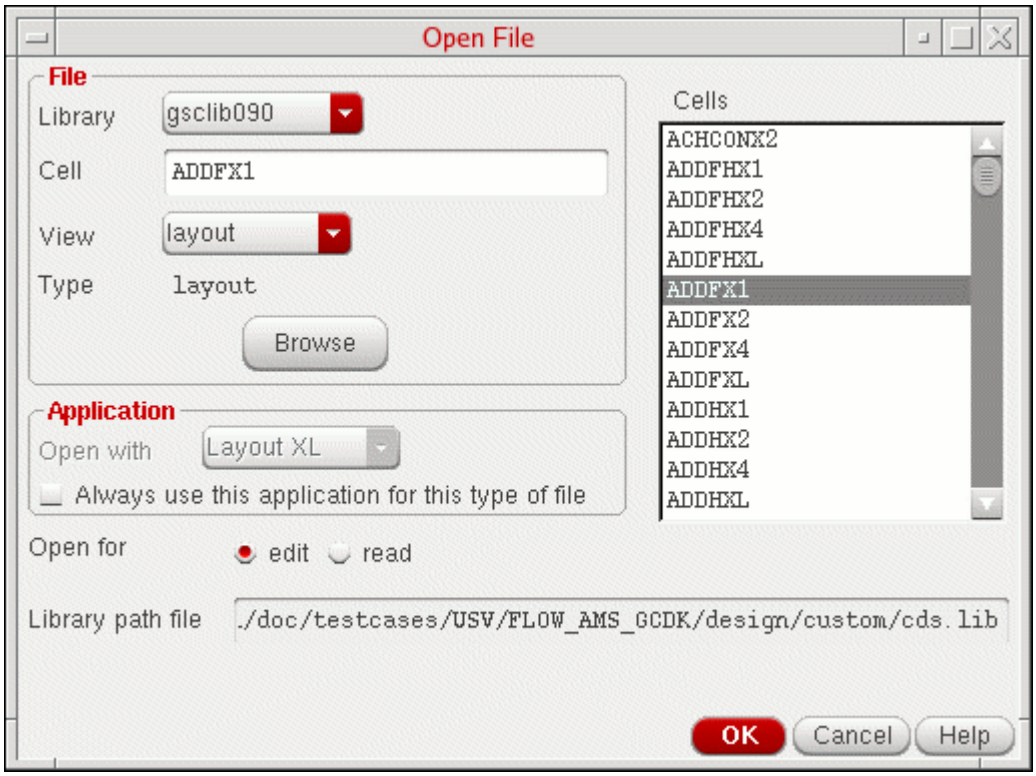
   - If you chose to create a new layout view, the New File form is displayed, with the default view name as `layout`.

     Choose a *Library* name and type the *Cell* and *View* names, and click *OK* to create the new cellview.

     The software automatically creates the new layout cellview and looks for an existing physical configuration view called either `physConfig` or `physConfig_`*`schName`*, which might have been created in a previous Layout XL session or when converting data to use the IC 6.1 Layout XL schema. If neither view exists, it creates a new one called `physConfig` or `physConfig_`*`schName`*.

© 2023 Cadence Design Systems, Inc.

- If you chose to open an existing layout view, the Open File form is displayed again.



Specify the *Library* and *Cell* names, choose the *View* you want to open, and click *OK* to open the cellview.

In the form, you cannot change setting for the *Open with* option in the *Application* group box, regardless of the value set for the maskLayoutDefaultApp environment variable.

If there is an existing physical configuration view associated with the layout cellview, the software opens it in the background. If there is no existing physical configuration, the software creates a temporary, default one called `physConfig` or `physConfig_`*`schName`*.

If the physical configuration view, `physConfig`, associated with a layout cellview is currently open and it is a temporary, default physical configuration view, then the physical configuration view cannot be accessed by any other user at the same time. Therefore, if another user wants to open the same design in read-only mode, another temporary physical configuration view, `physConfig_2`, gets created. Likewise, if a third user now wants to open the same design in read-only mode, another temporary physical configuration view, `physConfig_3` gets created. To conclude, for any incremental views of the same design that require the layout cellview to be opened in read-only mode, a new, temporary physical configuration

© 2023 Cadence Design Systems, Inc.

view gets automatically generated and the name of each new physical configuration view increases incrementally.

The default Layout XL desktop configuration appears, displaying the schematic window, the layout window, the CIW, and the Palette assistant. The schematic view is re-opened in the context of the automatic physical configuration view, which might be different from the configuration it was using previously. The schematic window banner updates to indicate the name of the automatic physical configuration view. You might need to re-extract the schematic view to take into account the settings in the physical configuration view. The system prompts you if this extraction is required. To avoid the need for re-extracting the schematic view and to have the tool automatically copy the physConfig settings from the previous view, you must copy the cellview using the Library Manager.

The Configure Physical Hierarchy window is not displayed in *Automatic* mode. However, you can invoke it at any time by choosing *Launch – Configure Physical Hierarchy* from the layout window menu bar. Even in *Automatic* mode, you still need to use the Configure Physical Hierarchy window if you want to change the switch, stop, and constraint view lists, or make changes to component type for soft block definitions.

If Layout XL does not initialize and instead you see the Layout XL Constraint Group Setting dialog, see Layout XL Constraint Group.

Related Topics

Open File Form

Startup Option Form

Layout XL Desktop

Copying a View in the Library Manager

Configuring the Physical Hierarchy

Data Conversion to Use the Layout XL Schema

Starting Layout XL with a New Physical Configuration View

To start Layout XL from a schematic view in the context of a new physical configuration view:

1. From the CIW, choose *File – Open*.

   The Open File form is displayed.

© 2023 Cadence Design Systems, Inc.

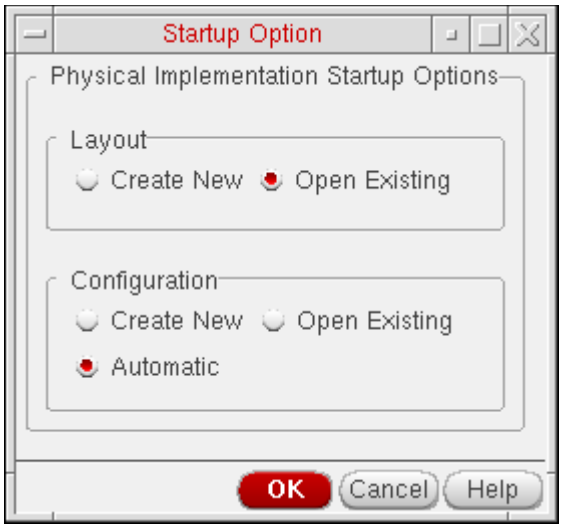
2. Specify the *Library* and *Cell* names, choose *schematic* from the *View* list and click *OK*.

   The Virtuoso Schematic Editor L Editing window appears displaying the cellview you specified.

3. From the schematic window, choose *Launch – Layout XL*.

   Layout XL appears on the *Launch* menu only if the Layout XL software is installed. If you do not see *Layout XL* in the menu, check with your system administrator.

   The Startup Option form is displayed.



4. In the *Layout* group box, choose whether you want to create a new layout cellview or open an existing one.

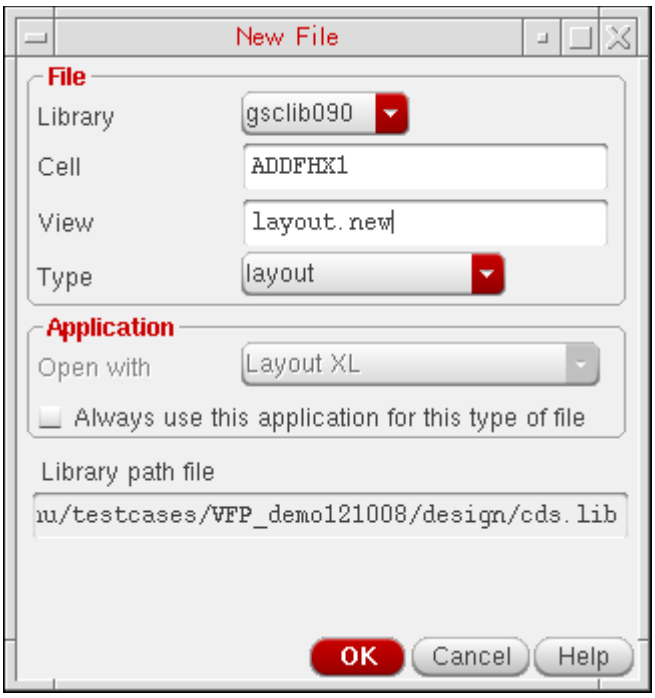
5. In the *Configuration* group box, select *Create New* and then click *OK*.

   One of the following happens:

   - If you chose to create a new layout cellview, the New File form is displayed.

© 2023 Cadence Design Systems, Inc.

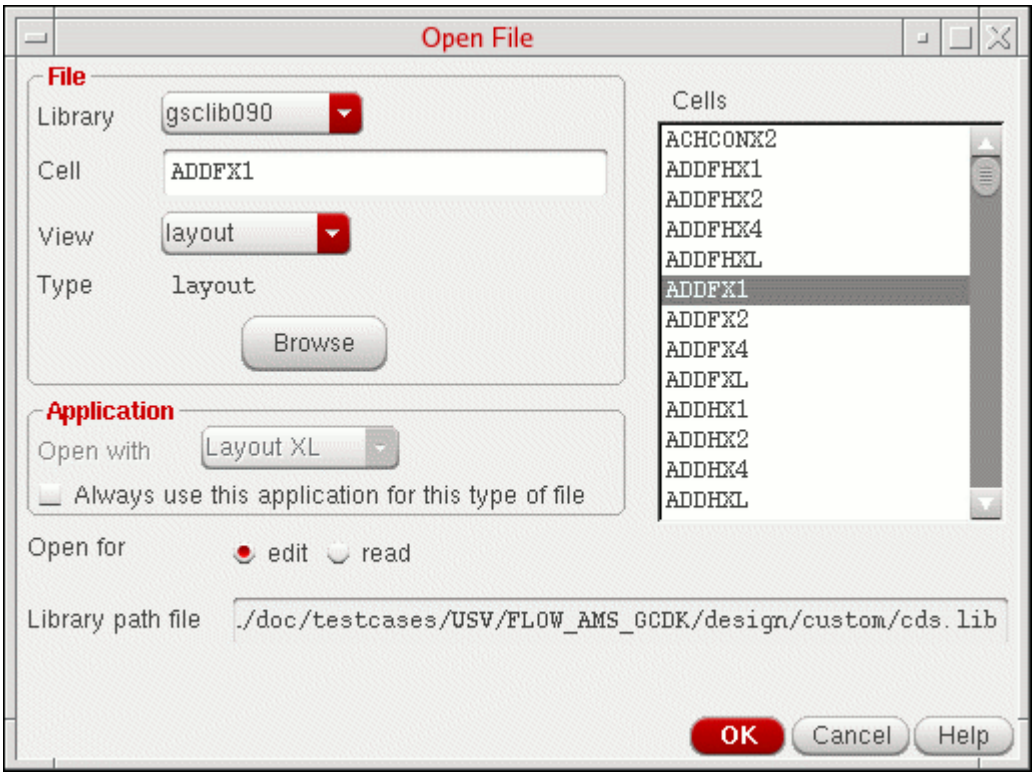
Choose a *Library*, type the *Cell* and *View* names, and click *OK*.



- If you chose to open an existing layout view, the Open File form is displayed again.

© 2023 Cadence Design Systems, Inc.

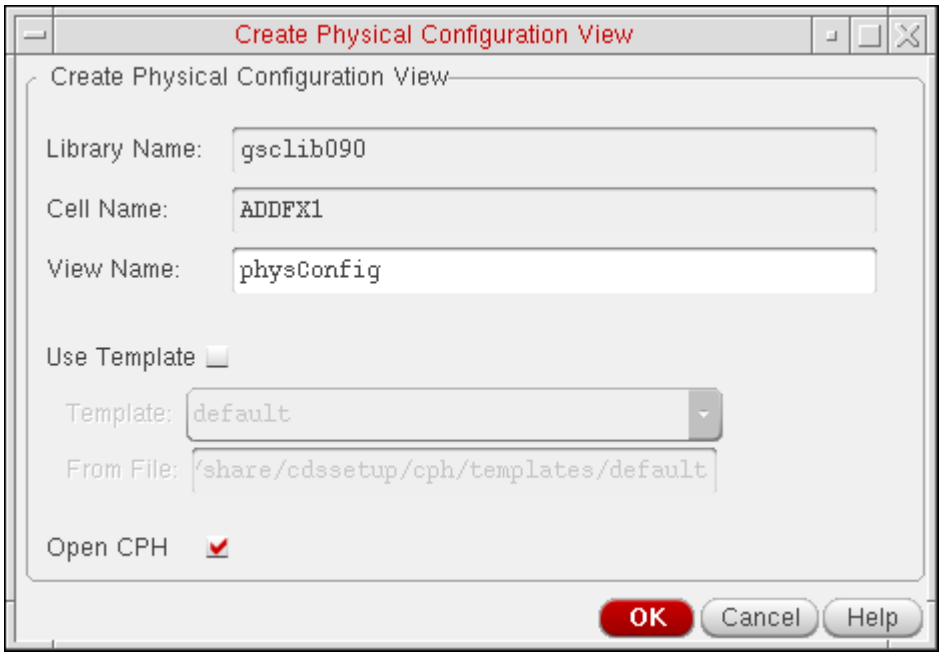
Specify the *Library* and *Cell* names, choose the *View* you want to open, and click *OK*.



You cannot change setting for the *Open with* option in the *Application* group box, regardless of the value set for the maskLayoutDefaultApp environment variable.

© 2023 Cadence Design Systems, Inc.

The Create Physical Configuration View form is displayed. The layout *Library* and *Cell* names are seeded automatically.



6. Specify *View Name* for the physical configuration view.

   To seed the new view with predefined values, check *Use Template* and either choose one of the predefined templates from the drop-down list or specify a file stored in another location. If you do not specify a template, the corresponding environment variables are used to obtain initial values.

7. Select the *Open CPH* check box to open the Configure Physical Hierarchy window automatically.

8. Click *OK*.

   The default Layout XL desktop configuration appears, displaying the schematic window, the layout window, the CIW, and the Palette assistant.

   The schematic view is re-opened in the context of the new physical configuration view, which might be different from the configuration it was using previously. The schematic window banner updates to indicate the name of the new physical configuration view. The Configure Physical Hierarchy window opens to display the physical configuration view you created. You might need to re-extract the schematic view to take into account the settings in the physical configuration view. The system will prompt you if this extraction is required.

© 2023 Cadence Design Systems, Inc.

If Layout XL does not initialize and instead you see the Layout XL Constraint Group Setting dialog, see Layout XL Constraint Group.

Related Topics

Open File Form

Startup Option Form

Layout XL Desktop

Configuring the Physical Hierarchy

Starting Layout XL with an Existing Physical Configuration View

To start Layout XL from a schematic view in the context of an existing configuration view:

1. From the CIW, choose *File – Open*.

   The Open File form is displayed.

2. Specify the *Library* and *Cell* names, choose *schematic* from the *View* list and click *OK*.

   The Virtuoso Schematic Editor L Editing window appears displaying the cellview you specified.

3. From the schematic window, choose *Launch – Layout XL*.

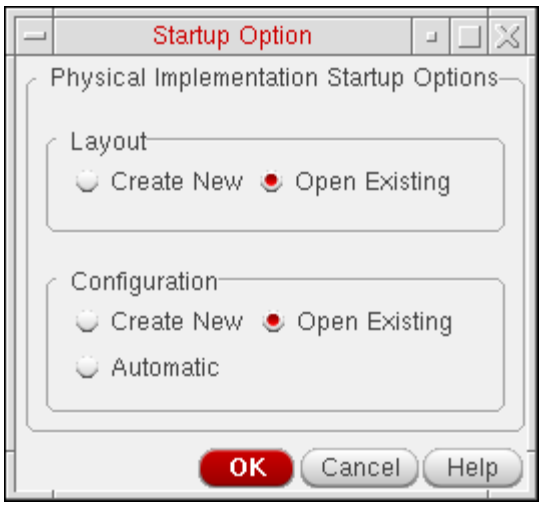
   Layout XL appears on the *Launch* menu only if the Layout XL software is installed. If you do not see *Layout XL* in the menu, check with your system administrator.

   The Startup Option form is displayed.

4. In the *Layout* group box, choose whether you want to create a new layout cellview or open an existing one.

© 2023 Cadence Design Systems, Inc.

5. In the *Configuration* group box, select *Open Existing*.

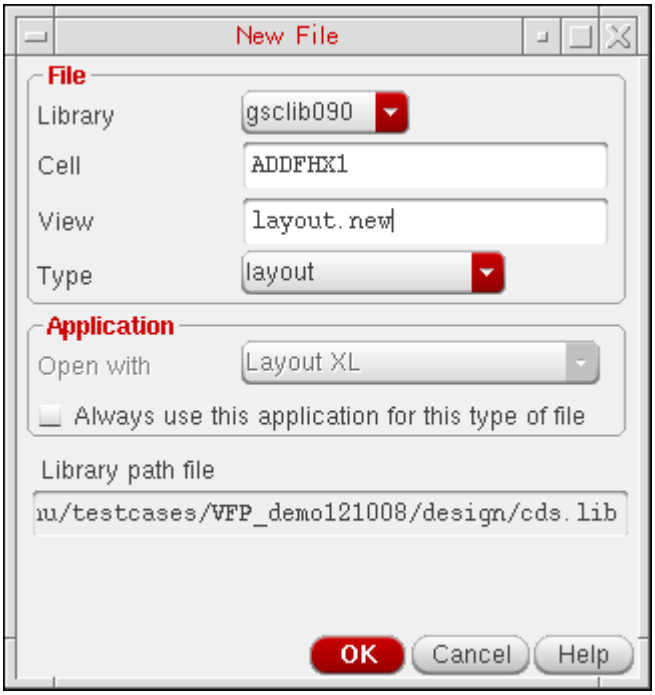


6. Click *OK*.

   One of the following happens:

   - If you chose to create a new layout cellview, the New File form is displayed.

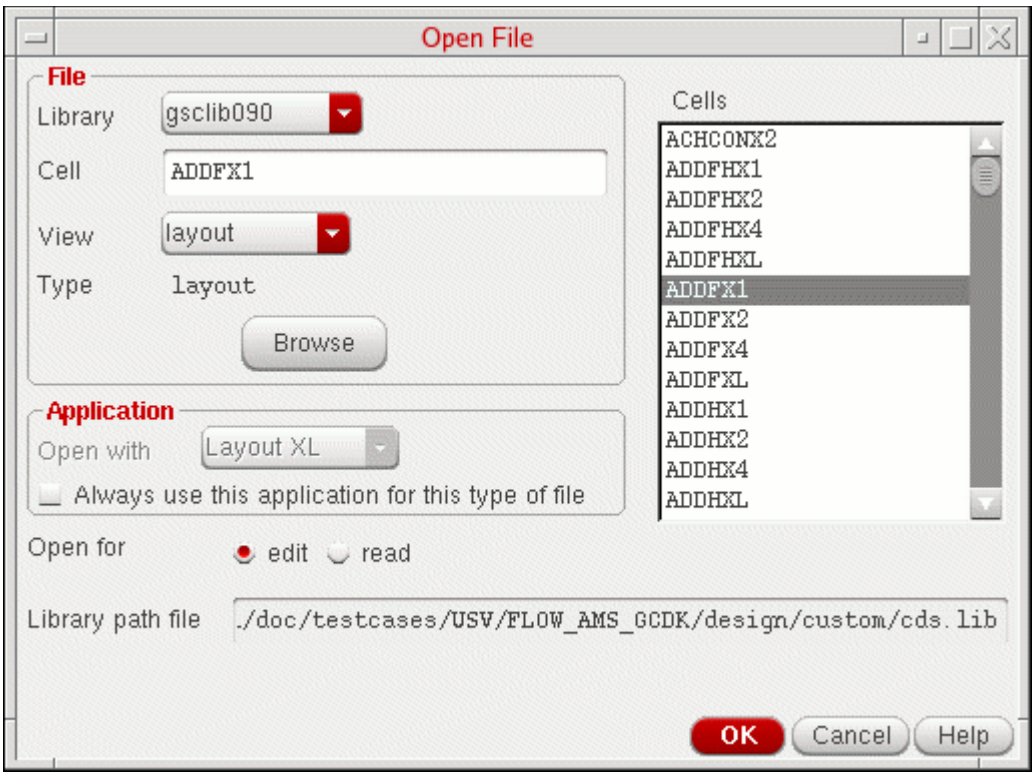
     Choose a *Library* name and type the *Cell* and *View* names.



   - If you chose to open an existing layout view, the Open File form is displayed again.

© 2023 Cadence Design Systems, Inc.

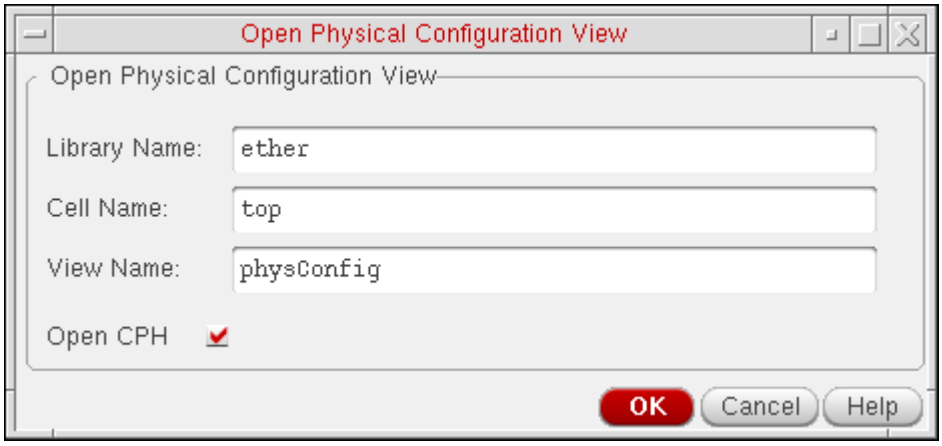
Specify the *Library* and *Cell* names and choose the *View* you want to open.



You cannot change setting for the *Open with* option in the *Application* group box, regardless of the value set for the maskLayoutDefaultAppenvironment variable.

7. Click *OK* to create the new layout cellview or to open an existing one.

   The Open Physical Configuration View form is displayed.



8. Specify the *Library Name*, *Cell Name*, and *View Name*.

© 2023 Cadence Design Systems, Inc.

9. Select the *Open CPH* check box to open the Configure Physical Hierarchy window automatically.
10. Click *OK*.

    The default Layout XL desktop configuration appears: the schematic window, the layout window, the CIW, and the Palette assistant.

    The schematic view is re-opened in the context of the specified physical configuration view, which might be different from the configuration it was using previously. The schematic window banner updates to indicate the name of the specified physical configuration view and the Configure Physical Hierarchy window opens.

    You might need to re-extract the schematic view in order to take into account the settings in the physical configuration view. The system will inform you if this extraction is required.

Related Topics

Open File Form

Startup Option Form

Layout XL Desktop

Configuring the Physical Hierarchy

Starting Layout XL from a Layout View

To start Layout XL from a layout view:

1. From the CIW, choose *File – Open*.

   The Open File form is displayed.
2. Specify the *Library* and *Cell* names.
3. Select *schematic* from the *View* list and click *OK*.

   The Virtuoso Layout Suite editing window appears displaying the cellview you specified.
4. From the layout window, choose *Launch – Layout XL*.

   The default Layout XL desktop configuration appears: the schematic window, the layout window, the CIW, and the Palette assistant. The schematic view is opened in the context of the physical configuration view being used for the session, which might be different from the configuration it uses when it is opened standalone.

© 2023 Cadence Design Systems, Inc.

You might need to re-extract the schematic view in order to take into account the settings in the physical configuration view. The system will inform you if this extraction is required.

By default, the schematic editor launches in the XL mode. To launch the editor in the L mode instead, select the *Schematics L* option for the *Open with* field on the Connectivity form. You can also use the lxSchematicDefaultApp environment variable.

Layout XL appears in the *Launch* menu only if the Layout XL software is installed. If you do not see *Layout XL*, check with your system administrator.

Related Topics

Layout XL Desktop

Connectivity Form

Starting Layout XL from a Physical Configuration View

To start Layout XL directly from a physical configuration view:

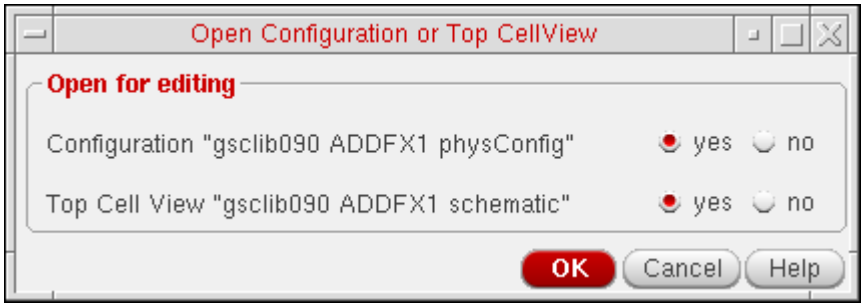
1. From the CIW, choose *File – Open*.

   The Open File form is displayed.

2. Specify the *Library* and *Cell* names.

3. Select *physConfig* from the *View* list and click *OK*.

   The *Open Configuration or Top Cellview* form is displayed.



4. Set both *Configuration* and *Top Cell View* to *yes* to open both the physical configuration view and the schematic view.

   You can set defaults for this form using the following environment variables:

```
envSetVal("ddserv.he" "hierEditor" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

```
envSetVal("ddserv.he" "schEditor" 'boolean t)
```

5. Click *OK*.

   The schematic view is opened in a Virtuoso Schematic Editor window and the Configure Physical Hierarchy window is displayed showing the physical configuration view you specified. The system checks out a Layout XL license if there is not one currently checked out.

6. Start Layout XL. For information, see Layout XL Launched from a Schematic View.

Related Topics

Open File Form

Configuring the Physical Hierarchy

Layout XL Constraint Group

The Layout XL connectivity extractor requires a constraint group containing information on valid layers, valid vias, and error layers.

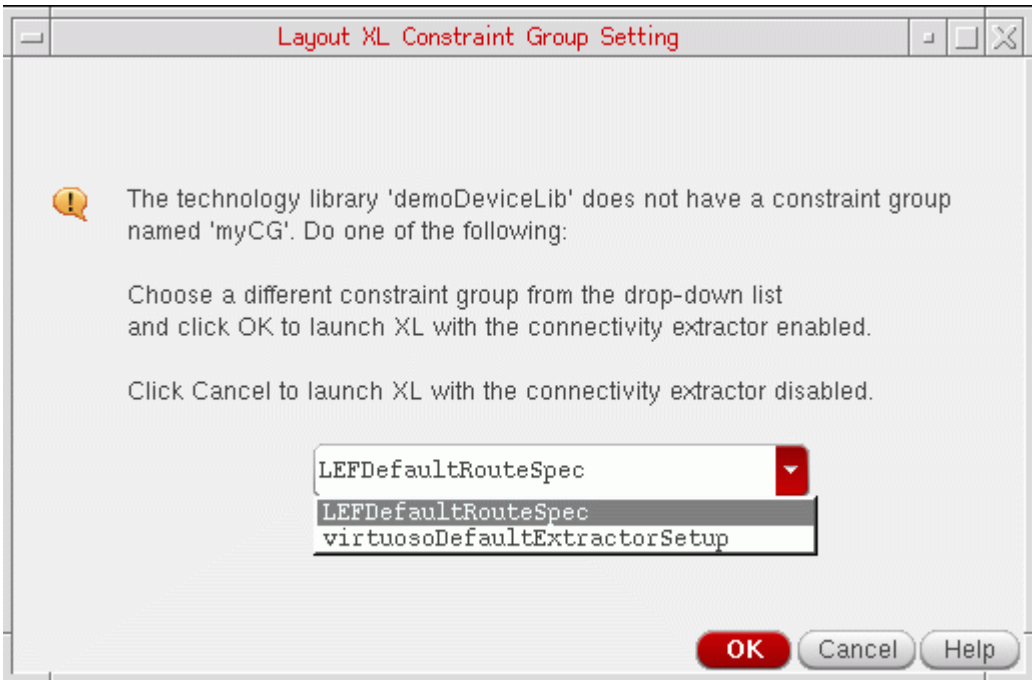
On startup, Layout XL transfers all the constraints defined in the schematic view to the layout view; correctly mapping the constraints and their members between the two views. Constraints in the schematic are transferred to the top-level layout view. Constraints that have been created or changed in the schematic but not yet saved are also transferred. Device correspondence information is maintained during the transfer. Physical name mappings are tracked and constraints are updated appropriately to take account of folded instances in the layout.

You can specify the default constraint group using the layout environment variable setupConstraintGroup. If the specified constraint group does not contain the information required by Layout XL, during initialization the software displays the Layout XL Constraint Group Setting form prompting you to do one of the following:

- Choose the constraint group you want from the ones specified in the technology file.

© 2023 Cadence Design Systems, Inc.

- Launch XL with the connectivity extractor disabled.



Related Topics

Starting the Design Environment for Layout XL

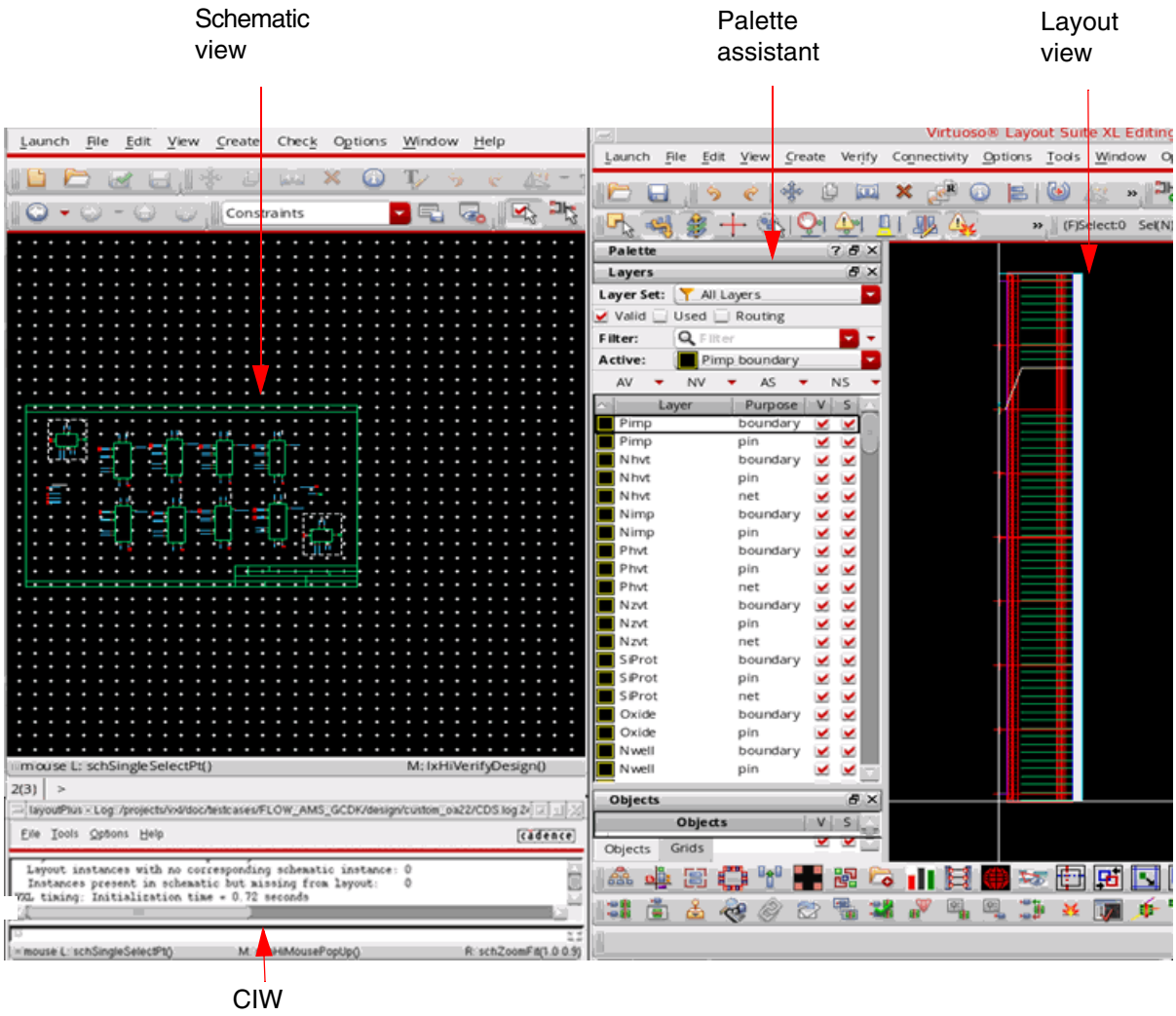
Layout XL Desktop

When you launch Layout XL, four windows are rearranged into the default window configuration on your desktop: command interpreter window (CIW), schematic window, layout window, and the Palette assistant.

The layout window is on the right with *Virtuoso® Layout Suite XL* displayed in the banner; the schematic that corresponds to the layout is in the schematic window on the left. The

© 2023 Cadence Design Systems, Inc.

Palette assistant and the command interpreter window (CIW) are placed to the left and bottom of the screen, respectively. You can move, resize, and iconify any of the four windows.



The layout window comprises a menu bar, a canvas, in which you can display and manipulate your design graphically, and a number of toolbars and dockable assistants to help you with your design tasks.

If you customize the desktop layout, you can save and reuse the customized desktop layout in one of the following ways:

- Using the *Options – Save Defaults* command from the CIW.

  When you use the *Save Defaults* command, Layout XL saves the size and position of the schematic and layout windows as a property on the current layout cellview. It also

© 2023 Cadence Design Systems, Inc.

sends the Palette assistant and the CIW window size and location information to the `.cdsenv` file.

When you reopen the layout and schematic cellviews, they appear in the same configuration as they had when you used the *Save Defaults* command.

- By adding the appropriate environment variables to your `.cdsenv` file. For example, to prevent Layout XL from rearranging or resizing windows, add the following line to the `.cdsenv` file:

  ```
  envSetVal("layoutXL" "autoArrange" 'boolean nil)
  ```

  If you save the current window positions and form settings to a file other than `.cdsenv`, for example, `.envFileName`, you can restore the settings by adding the following line to your `.cdsinit` file or typing it in the CIW.

  ```
  envLoadFile("layoutXL" ".envFileName")
  ```

For an overview of the connectivity-driven features supported by Layout XL and a lab database and workshop instructions to help you explore how these features can bring productivity gains for a design, view or download the rapid adoption kit, *Introduction to Connectivity-Driven Design in VLS XL*.

Related Topics

Saving and Recalling Default Settings

Interacting with the User Interface

Command Interpreter Window

Palette Assistant

autoArrange

Working with Designs

A design can be opened for reading or editing. To open a design, you select *Open* from the *File* menu in the command interpreter window (CIW). A form is displayed for you to specify the library, cell, and view name of the design and the access mode, which can be *edit* or *read.*

- Edit mode excludes other users from editing the same file, although they can still view the old copy saved to the disk.
- Read mode allows users to share design files.

© 2023 Cadence Design Systems, Inc.

Whether a design is opened in edit or read mode, all designs under the hierarchy of that design are opened in read mode. The only exception is edit-in-place, where both the top design and the design being edited can be modified; the other designs maintain their non-modifiable status.

After a design is opened, you can change the mode from *read* to *edit.* If a design has already been modified, changing the mode from *edit* to *read* makes the system to prompt you to save or discard the changes made. If you choose the *discard* option, the changes are ignored and the original design is brought back from the disk. You cannot make a cell editable if you do not have write permission. The name of the current design and its mode are displayed in the window banner.

Working with your designs involves the following basic tasks:

- Opening a New Design
- Opening an Existing Cellview
- Making a Design Read Only or Editable
- Saving a Design
- Closing a Design

Related Topics

Working with Cellviews

Opening a New Design

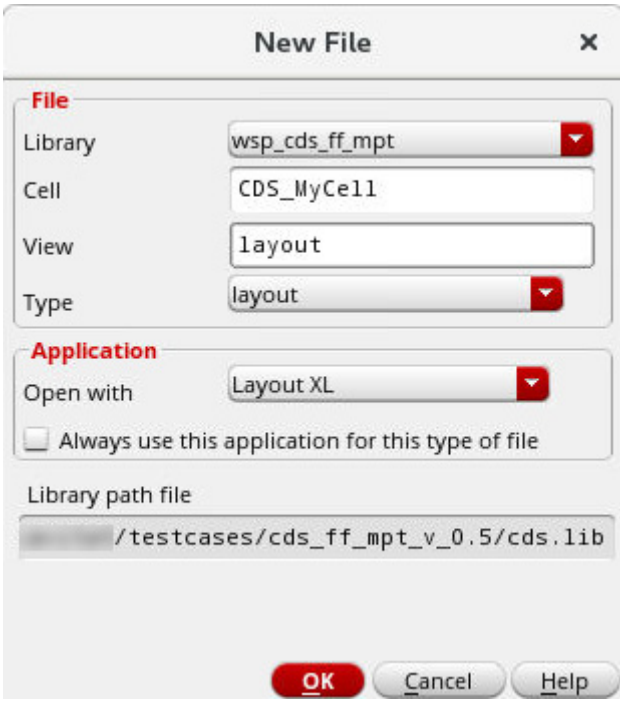
To open a new cellview from the layout editor:

1. Choose *File – New* from an editor window or the CIW.

   The New File form appears.
2. From the *Library* list, choose the library in which to store the new cellview.
3. In the *Cell* field, enter a new cell name.
4. In the *View* field, enter a name for the new cellview.
5. From the *Type* list, choose the view type, such as *layout*, that is to be associated with the new cellview.

© 2023 Cadence Design Systems, Inc.

6. In the *Open with* list, choose the application that the new cellview should automatically open with.



7. Optionally, select the *Always use this application for this type of file* check box to associate the selection made in the *Open with* field with the current *View* type selected. This will save you in future from having to select what application should be opened with a selected cellview *Type*.

   The *Library path file* field displays the location where the new cellview will be stored.

8. If the *Open in* option is displayed, choose to open the cellview in a *new tab* in the current session window, the *current tab* in the current session window, or a *new window* in a new session.

9. Click *OK*.

You can use an existing layout cellview as a template to auto-create width spacing pattern (WSP) and row region specifications in the design by using the autoCopyTemplateCellViewList environment variable. Auto-creation works only while creating a new cellview.

Related Topics

Working with Designs

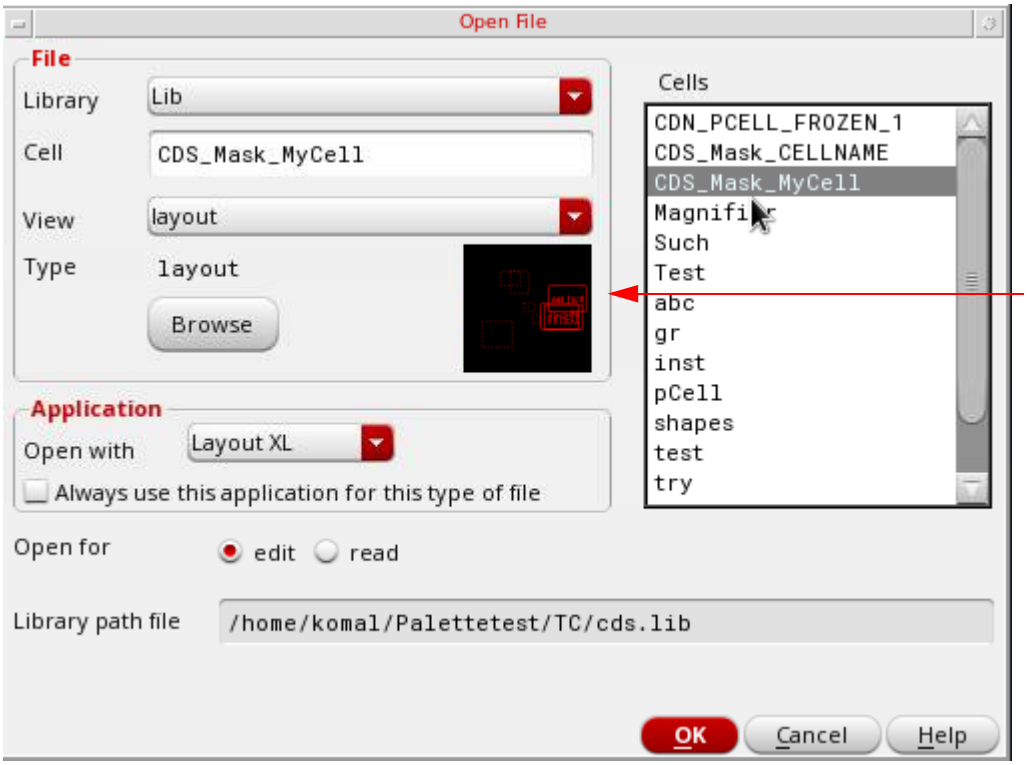
© 2023 Cadence Design Systems, Inc.

Opening an Existing Cellview

To open an existing cellview in the layout editor:

1. Choose *File – Open* from an editor window or the CIW.

   The Open File form appears.



2. From the *Library* list, choose the library that contains the cellview you want to open.
3. In the *Cell* field, specify a cell name using one of the following methods:
   - Type the cell name.
   - Click a cell name in the *Cells* list box on the right.
   - Click *Browse* to select the cell in *Library Browser*.
4. Choose the *View* to be opened.

   The *Type* field automatically updates to display the type of view of the selected cell view.

   If a thumbnail of the selected cellview is available, it is displayed in the form as an aid to selection. If no thumbnail is available, the space appears blank. For more information about the thumbnail of a cellview, see

© 2023 Cadence Design Systems, Inc.

5. From the *Open with* list, choose the application that the cellview should open with.
6. Optionally, select the *Always use this application for this type of file* check box to associate the selection made in the *Open with* list with the current *View* type selected. This will save you in future from having to select what application should be opened with a selected cellview *Type*.
7. Choose whether to open the cellview in *edit* or *read* mode.

   The *Library path file* field displays the location where the cellview is stored.
8. If the *Open in* option is displayed, choose to open the cellview in a *new tab* in the current session window, the *current tab* in the current session window, or a *new window* in a new session.
9. Click *OK*.

Related Topics

Open File Form

Thumbnail Images of Cellviews

Working with Cellviews

Working with Designs

Making a Design Read Only or Editable

To make a design open in read-only mode:

➡ Choose *File – Make Read Only* from the editor window.

If you have made any changes to the current cellview but have not saved them, the *Change Edit Mode* dialog box appears and prompts you to either save the changes or discard them.

After you click *Yes* or *No*, the mode becomes read-only. *Yes* saves edits, and *No* discards your edits. You can click *Cancel* to let the design remain in edit mode.

This command toggles with *File – Make Editable*.

To make a design opened in read-only mode as editable:

➡ Choose *File – Make Editable* from the editor window.

© 2023 Cadence Design Systems, Inc.

If you have made no change in read-only mode, the design switches to the editable mode. However, if you have made any changes in read-only mode, and reopen the design in edit mode, then:

- If the design has changed on the disk either because of data management or because it was edited and saved by another user, a warning message is displayed in the CIW indicating that the design has been refreshed. As a result, changes you have made in read-only mode will be lost.
- If the design has not changed on the disk, a dialog box is displayed informing you that you have changed the design in read-only mode, and you can choose *Yes* to keep the changes or *No* to discard them.

Related Topics

Working with Designs

Saving a Design

To save a design:

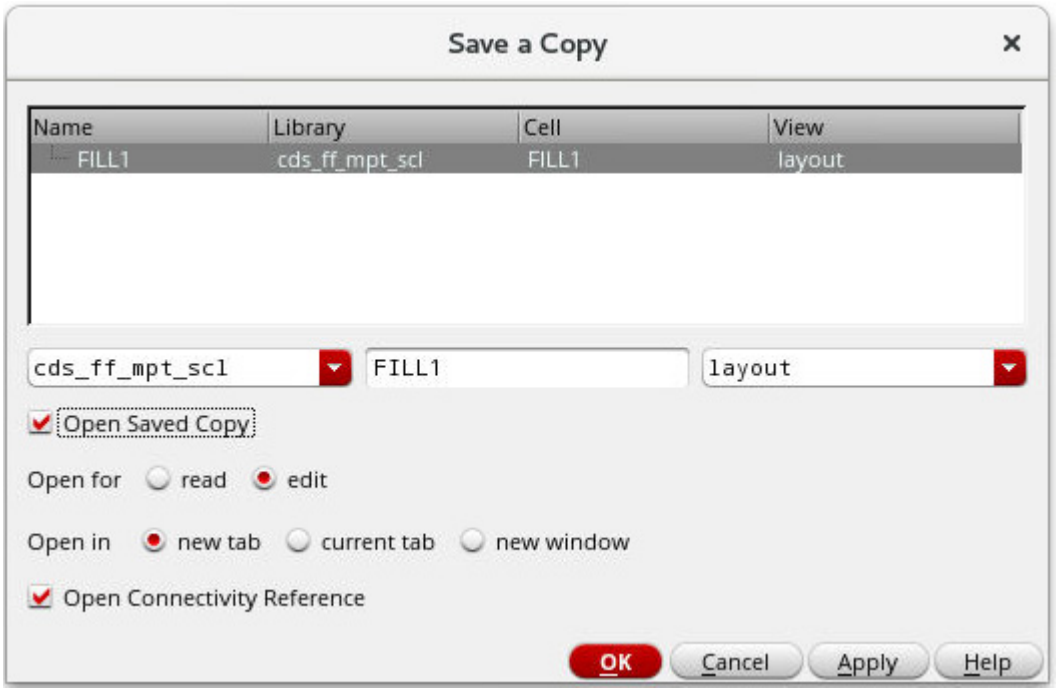
➡ In the layout window, choose *File – Save*.

To save a design with a different name:

1. Choose *File – Save a Copy*.

© 2023 Cadence Design Systems, Inc.

The *Save a Copy* form appears.



2. Select a library from the first drop-down list. This should be the library in which you want to save the cellview.
3. Specify a new cell name in the text field.
4. Select the cellview type from the last drop-down list.
5. Select the *Open Saved Copy* check box if you want to open the saved copy of the design.
6. Select *read* or *edit* from the *Open for* option.
7. Select the location where you want to open the design: *new tab*, *current tab*, or *new window*.
8. Select the *Open Connectivity Reference* check box if you want the schematic view for connectivity reference to be opened when the saved copy of the design is opened.
9. Click *OK*.

You can access the *Save a Copy* command if a design is opened in read mode. The command checks out the license and checks it back in after the save operation is complete.

Thumbnail images, where they have been set up in the current view for pre-viewing cellviews, are updated automatically when you choose to *Save* or *Save a Copy* of that cellview. This

© 2023 Cadence Design Systems, Inc.

happens through an automatic call to the `hiGenerateThumbnails` SKILL command. To disable the auto-generation of thumbnails, you can set the following environment variable in the CIW:

```
envSetVal("ui.thumbnails" "generate" 'boolean nil)
```

To save hierarchical instances that have been modified without running the *Edit In Place* or *Descend Edit* command in a design, choose *File – Save Hierarchically*.

Saving Hierarchical Cellviews under a Different Name

Use the *Save As* command to save your cellviews hierarchically under new names.

To save cellviews under a new name:

1. From the layout window menu bar, choose *File – Save As.*

   The Save As form is displayed, listing all the cellviews in the design hierarchically.

2. Select one or more rows from the list in the table.

   The fields at the bottom of the form are populated based on the selection in the table. If multiple rows are selected, then the field displays the value if all the values in the selected rows are same. Otherwise it shows *AS IS*.

3. Type new *Library*, *Cell*, and *View* names you want to use. In the example above, the library and cell names are unchanged, but the view name for all the cells has been changed from layout to layout.edited.

4. Click *OK* or *Apply* to save the selected cellviews.

Related Topics

Save a Copy Form

Thumbnail View Functions

Thumbnail Images of Cellviews

Hierarchical Designs

Working with Designs

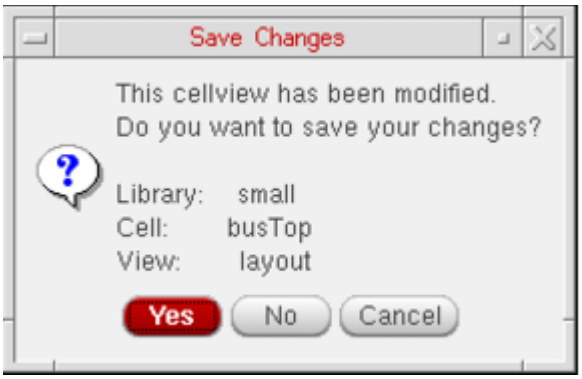
© 2023 Cadence Design Systems, Inc.

Closing a Design

To close an editor window:

➡ Choose *File – Close*.

If you edited the current cellview since the last time you saved your data, the *Save Changes* dialog box appears and prompts you to save the changes, discard them, or cancel the close operation.



- *Yes* saves your edits and closes the window.
- *No* discards your edits and closes the window.

  You can use the discardCVEditOnWindowClose environment variable to preserve the edits in the memory when you click *No*.
- *Cancel* cancels the *Close* command but does not discard the changes. It returns you to the cellview you were editing.

Related Topics

Working with Designs

© 2023 Cadence Design Systems, Inc.

Basic Command Functions in Layout XL

The following table describes how you can run some basic command functions in Layout XL.

Command Function	Description
Start a command	To start a command, you can do any of the following: ■ Choose the command from the menu. ■ Use the toolbar icon. ■ Choose the command from the layout editor shortcut menu. ■ Press the bindkey associated with the command. ■ Draw a stroke in the design window.
Cancel a command	To cancel a command without changing your data or to stop a command that automatically repeats, press `Esc` or click *Cancel* on the corresponding form. The *Modify Corner* and *Size* commands are exceptions. To cancel these commands, click *Cancel* on the form.
Undo a command	To undo a command, choose *Edit – Undo* or press `U`. You can undo up to 128 previous commands. You can set the undo limit to either 0 or 128. You can set the number of commands by choosing *Options – User Preferences* in the Command Interpreter Window (CIW). The system keeps an undo history or check points in a single, global stack. When you choose *Undo* in a cellview, the last edit made to any open cellview is undone. Once you have saved a cellview, the edits made prior to saving the cellview cannot be undone.
Redo a command	To reinstate a change you canceled with *Undo*, choose *Edit – Redo* or press `Shift+U`.

© 2023 Cadence Design Systems, Inc.

Command Function	Description
Repeat a command	Many layout editor commands automatically repeat. For example, each time you finish drawing a rectangle, the *Create Rectangle* command prompts you to draw another one. By default, all commands on the *Create* menu and the *Create Measurement* command are set to repeat. To set the automatic repeat on or off, choose *Options – Editor* and use the *Repeat Commands* check box to enable or disable the repeat operation.
Nest a command	You can pause before completing one command to perform a second command. This is called nesting a command. For example, if you move an object and then zoom out, the zoom command would be nested, allowing you to use another command such as moving another object. You can nest any display command but you cannot nest edit or create commands. As exceptions, edit and create commands can be nested within the *Modify Corner* and *Size* commands. You can nest up to 20 commands. To set the number of commands you can nest, choose *Options – User Preferences* from the CIW and set the *Nest Limit* field to the number you want.
Discard all edits	To cancel your edits, choose *File – Discard Edits*. A Discard Edits dialog box appears and prompts you to confirm whether you want to discard your edits. This command discards all edits you made since the last time you saved the cellview.
Use infix mode for the first point location	You can set the layout editor to automatically use the current pointer location as the starting point whenever you start commands with bindkeys. This is called infix mode. To set infix mode on or off, from the CIW, choose *Options – User Preferences*. In the User Preference form, turn the *Infix (No Click Necessary for First Point)* check box on or off, as required.

© 2023 Cadence Design Systems, Inc.

Command Function	Description
Use menu access keys	Menu access keys provide keyboard access to application menus and the functionality they contain without the need to use mouse selections. For example, if you press `Alt+F` on your keyboard, the software displays the contents of the *File* menu. The menu access key for a Layout XL menu, submenu, or command is the underlined letter in the menu command label. For example, to display the Connectivity form using only menu access keys, with the cursor in the Layout XL window, press `Alt+O`. This displays the contents of the *Options* menu. Then, press `O`.

Related Topics

Shortcut Menus

Bindkeys in Virtuoso Layout Suite

Layout Editor Strokes

User Preferences Form

Object Creation in Layout XL

Object Editing in Layout XL

Bindkeys in Virtuoso Layout Suite

Many layout editor commands can be run by using bindkeys or keyboard shortcuts. If bindkey settings are loaded, bindkeys appear to the right of the command name on the menu.

The list of default bindkeys for Layout XL is available at the following location, where *your_install_dir* is the directory where your Cadence software is installed.

your_install_dir/samples/local/lxBindKeys.il

To load a file of bindkey definitions every time you run Layout XL, add the following line to your `.cdsinit` file.

```
load(prependInstallPath("path_to_bindkey_file"))
```

For example, the following line loads the list of default bindkeys for Layout XL.

© 2023 Cadence Design Systems, Inc.

```
load(prependInstallPath("samples/local/lxBindKeys.il"))
```

The following table lists some of the Layout XL bindkeys defined in `lxBindKeys.il` and the corresponding SKILL functions and menu commands.

Key	SKILL Function	Menu
`F10`	nclToggleCAEMode()	*Options – Layout XL – General* tab; *Constraint-aware editing* option
`Ctrl+H`	geDSAFreeze(hiGetCurrentWindow() hiGetPoint(hiGetCurrentWindow()))	*Freeze DSA* and *Unfreeze DSA* options on the shortcut menu
`Ctrl+L`	lxProbeRemoveAll()	
`Ctrl+M`	lxHiSwapComps()	*Edit – Advanced – Swap Components*
`Shift+L`	lxHiProbe()	*Connectivity – XL Probe*
`RMB`	lxCmdOptions()	
`Shift+RMB`	lxCmdShiftOptions()	
`Shift+Ctrl+H`	lxToggleShowAllIncompleteNets()	*Connectivity – Incomplete Nets – Show/Hide All*
`H`	lxToggleShowIncompleteNets()	*Connectivity – Incomplete Nets – Show/Hide Current*
`Shift+H`	lxShowHideIncompleteNets()	*Connectivity – Incomplete Nets – Show/Hide Selected*

To view a detailed list of bindkeys, see Bindkeys in Virtuoso Layout Suite.

Related Topics

Bindkeys and Access Keys

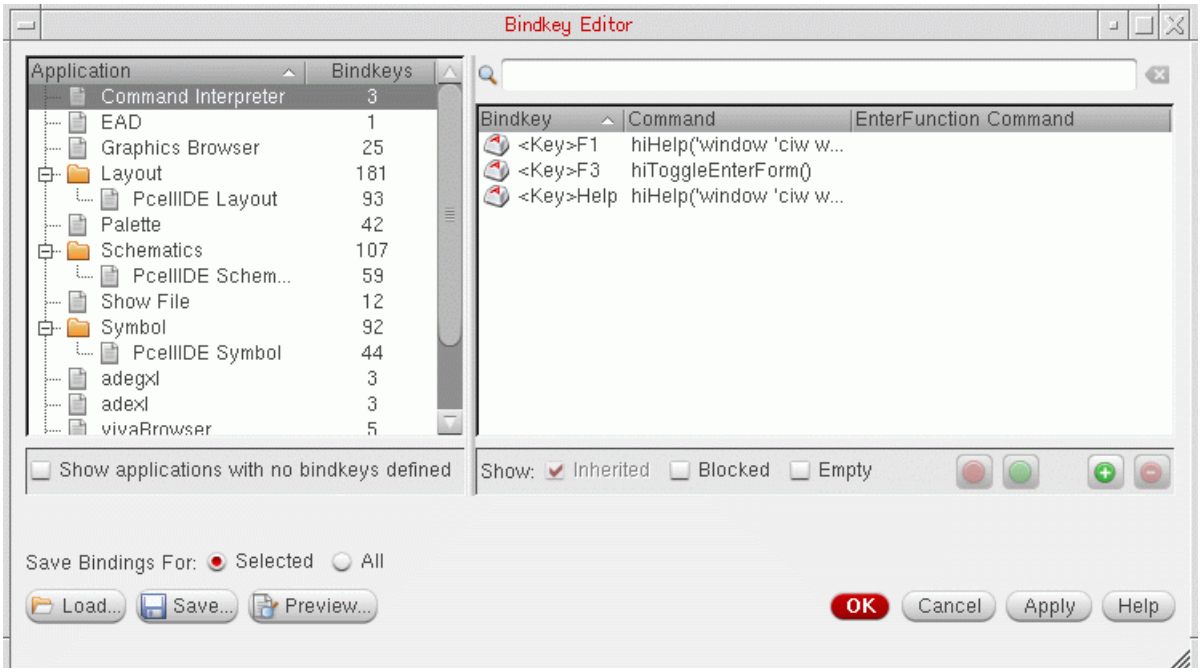
Displaying Bindkeys for Layout XL

To display the list of bindkeys for Layout XL:

1. From the CIW, choose *Options – Bindkeys*.

© 2023 Cadence Design Systems, Inc.

The Bindkey Editor form is displayed. The *Application* tree on the left of the form, by default, initially displays all the applications for which bindkeys have been defined.



2. In the *Application* tree, select the application whose bindkeys you want to view.

   The bindkey table on the right updates to list the bindkeys associated with the currently selected application.

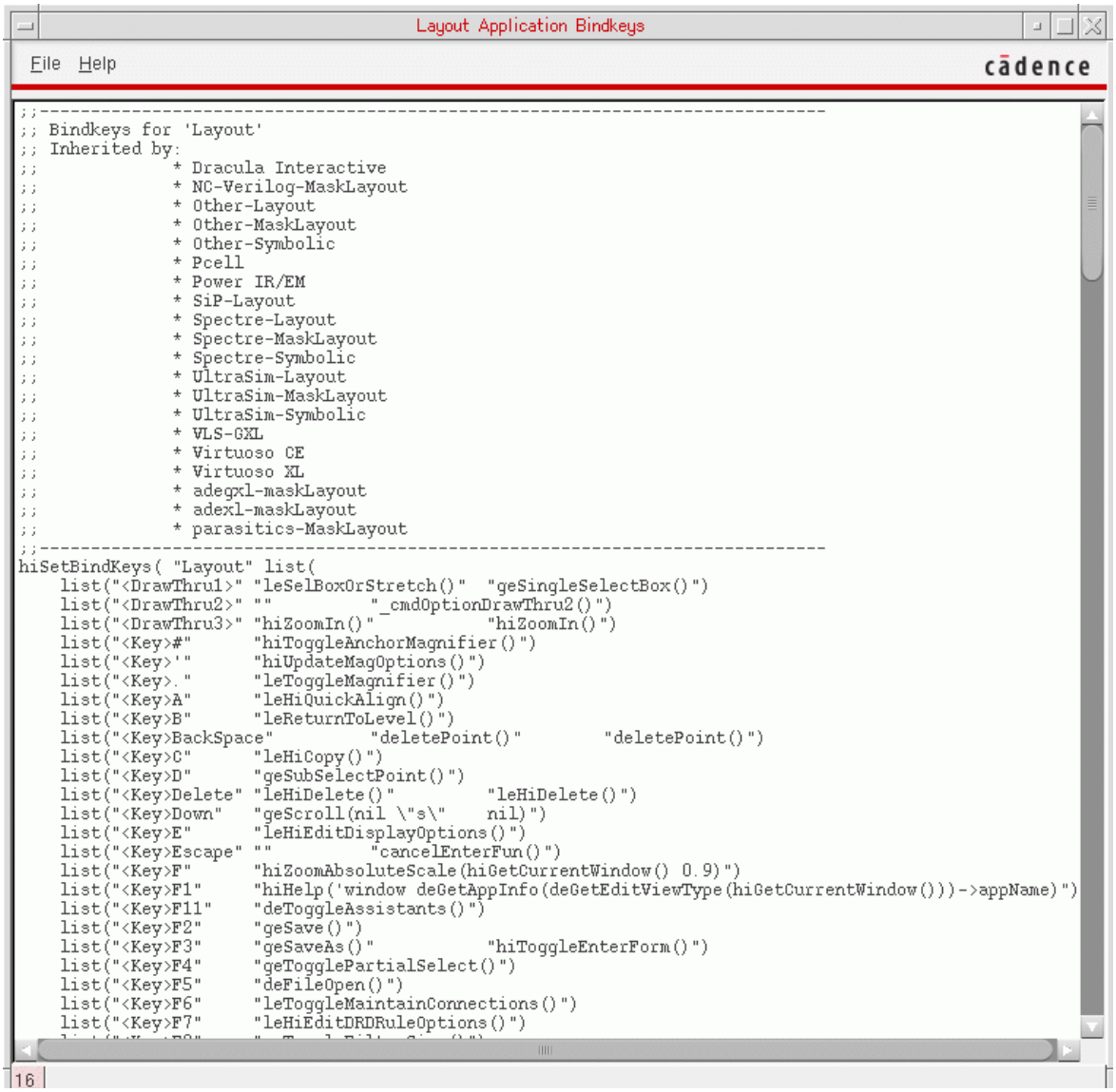
3. Optionally, use the *Search* field to filter the bindkey listing.

4. To view the bindkey settings for the currently selected application, do one of the following:

   - Scroll through the listed bindkeys in the bindkey table on the right to view the current settings for one or more bindkeys.

© 2023 Cadence Design Systems, Inc.

- Select the *Preview* button to display a listing of all the bindkeys associated with the selected application in a text window.



Related Topics

Bindkeys in Virtuoso Layout Suite

© 2023 Cadence Design Systems, Inc.

Layout Editor Default Setup

You can customize your design environment by setting the default values using the startup files and certain common forms in the layout editor.

Startup Files

Before you can start working in the layout editor, several startup files are initiated. Some of the things these files do include setting up your environment, pointing to libraries, and defining your plotters. The following table lists the startup files that the layout editor uses. You can use these files to customize your design environment settings.

File Name	More Information
`.Xdefaults`	Modifying the .Xdefaults or Equivalent File
`.cdsinit`	Specifying Environment Settings
`.cdsenv`	Specifying Environment Settings
`.cdsplotinit`	Plotting Your Design
`display.drf`	Display Resource File
`cds.lib`	Cadence Library Definition File

Forms for Setting Layout XL Default Values

The following layout editor forms control most of the default values for your design environment:

- Display Options Form: Controls display options in layout windows
- Layout Editor Options Form: Controls the options for layout editor behavior. You use this form to define your layout editing environment settings.
- User Preferences Form: Controls your Cadence software environment. These settings affect all Cadence applications, not just the layout editor.
- Display Resource Editor (DRE): Controls the appearance of layers.

© 2023 Cadence Design Systems, Inc.

Default Application Used for a Cellview Type

You can automate the launching of an application when a cellview is opened rather than choosing the application from the *Launch* menu each time. For information, see Specifying New Default Values for a Virtuoso Session and Setting the Default Application for a Cellview Type.

You can change the background color of the layout editor window using the environment variable, defaultEditorBackgroundColor. You can change the background color of the current window using the following command:

```
hiGetCurrentWindow()->backgroundColor = "red"

"red"
```

Related Topics

Layout Editor Basics

Set Default Application Form

Technology File

Each design library must be associated with a technology library that is compiled from a technology file. The layout editor requires the technology file to contain complete definitions of the layers used in your design.

A technology file defines the following:

- Design layers and layer properties
- Constraint and constraint groups
- Via definitions
- Site definitions
- Technology devices

Related Topics

The ASCII Technology File

Virtuoso Technology Data

© 2023 Cadence Design Systems, Inc.

Workspaces in Layout XL

You can rearrange the canvas, assistants, and toolbars to suit your individual work preferences. A customized configuration of the toolbars and assistants makes a *workspace*.

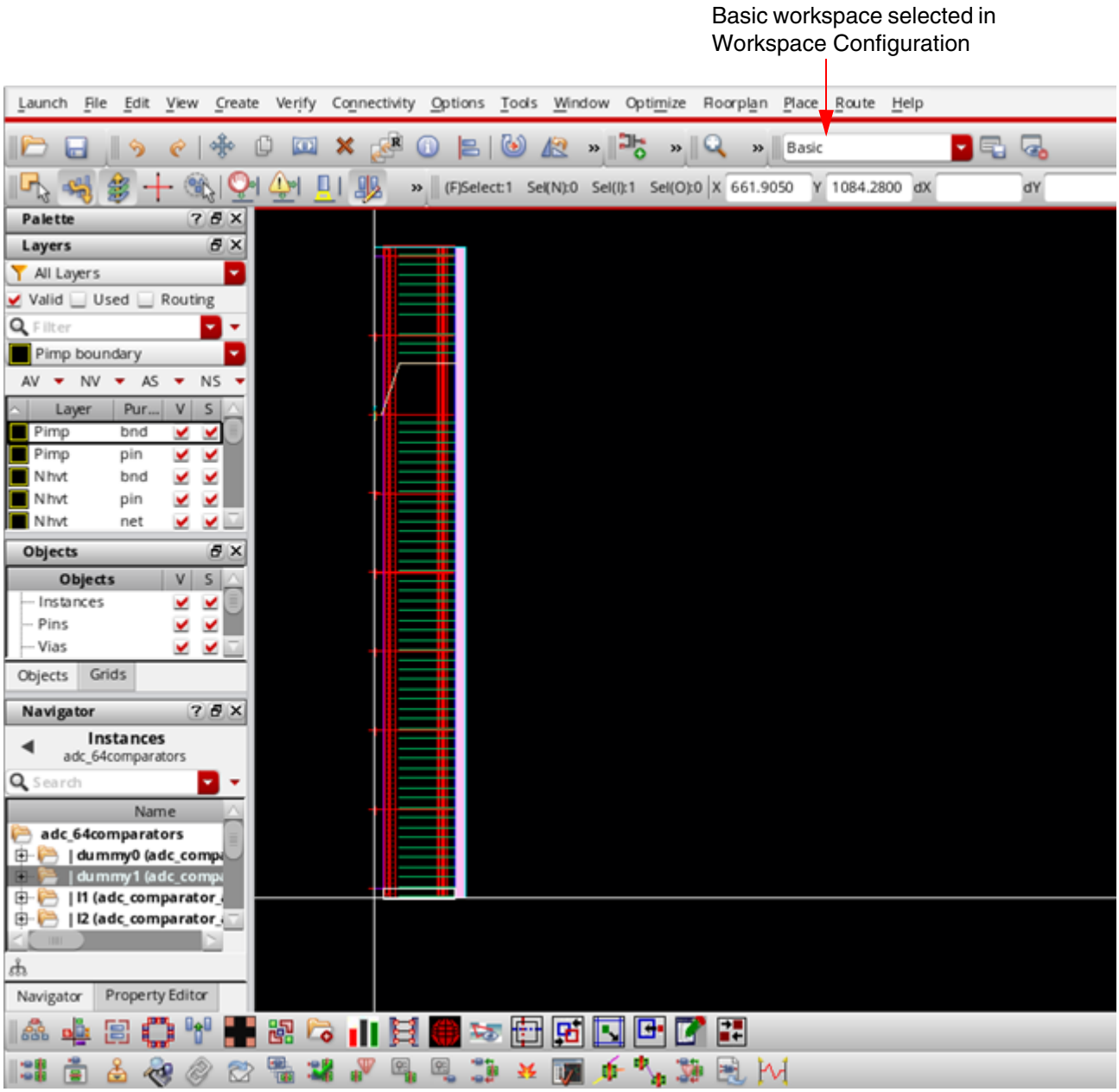
Layout XL ships with the following default workspaces:

- Basic
- CAS
- Classic
- Constraints
- EAD
- Floorplan
- Modgen
- Multipatterning
- VSR

© 2023 Cadence Design Systems, Inc.

Basic

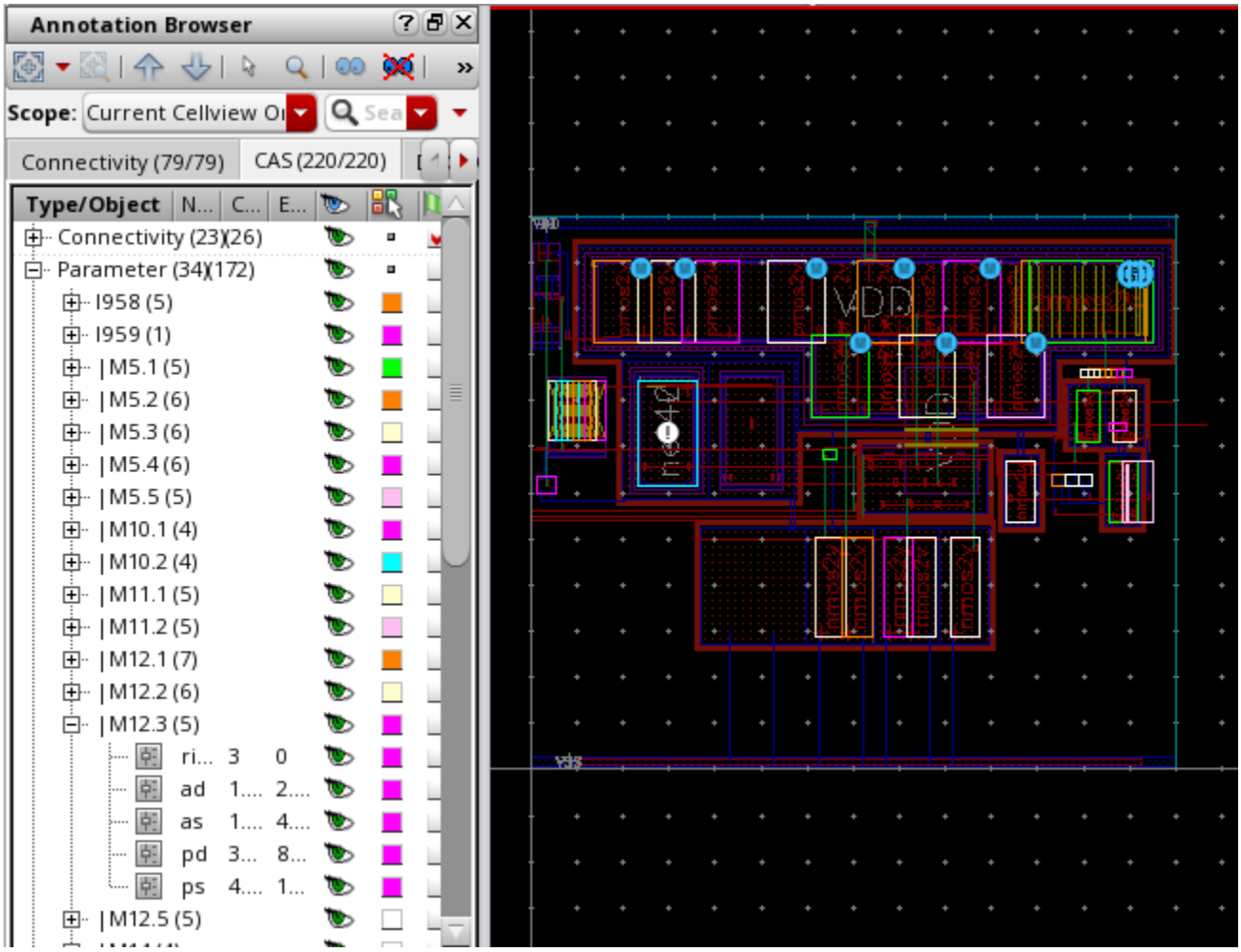
The *Basic* workspace shows the canvas, the selected toolbars, and the Palette, Navigator, and Property Editor assistants. Notice that the Navigator and the Property Editor assistants are available as tabs at the bottom on the left panel.



© 2023 Cadence Design Systems, Inc.

CAS

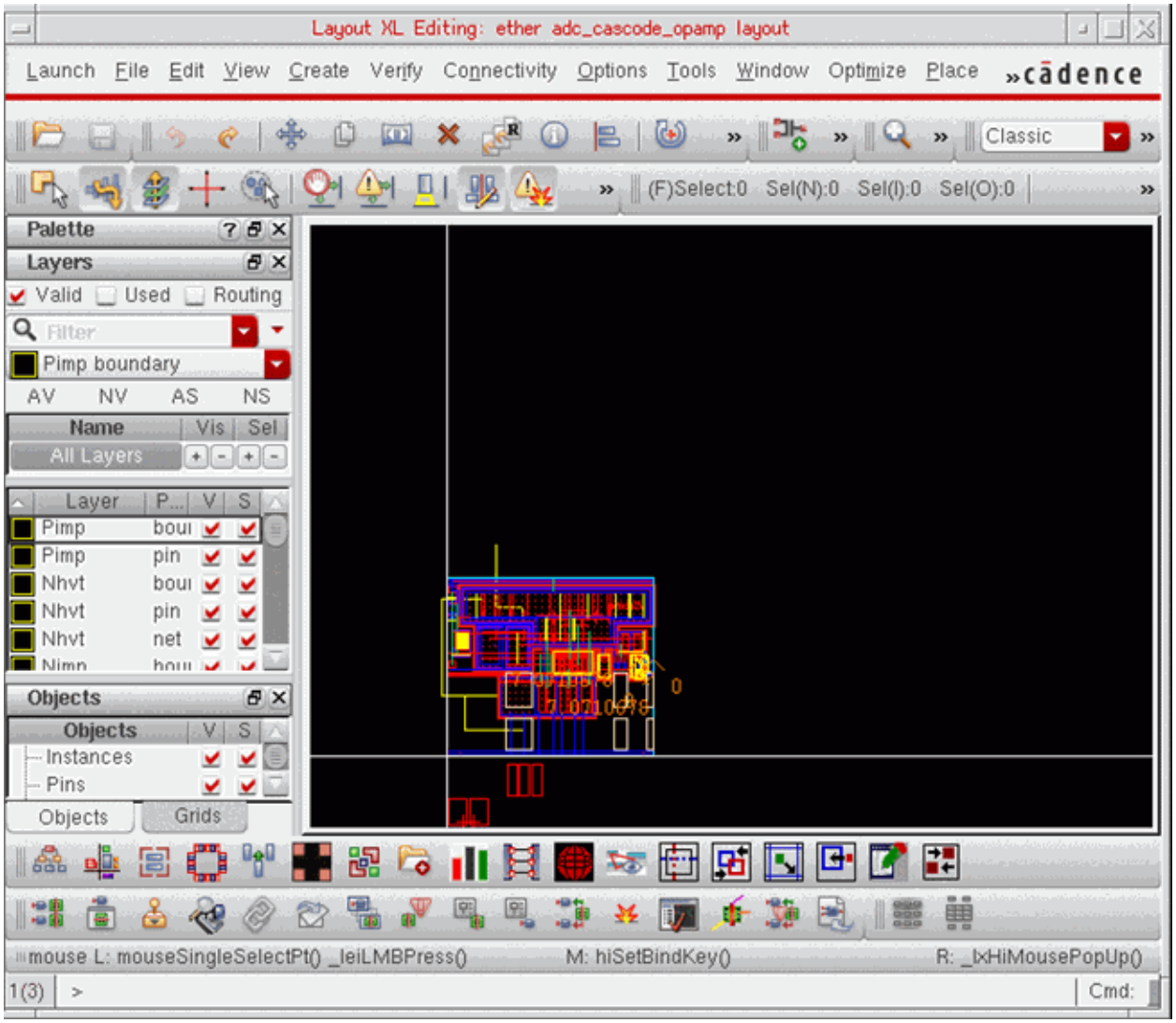
The *CAS* workspace shows the CAS tab in the Annotation Browser assistant, in addition to displaying the canvas and the selected toolbars. The CAS tab lists the markers that report the differences between the schematic and the layout in terms of instances, masters, connectivity, parameters, and instance or net names.



© 2023 Cadence Design Systems, Inc.

Classic

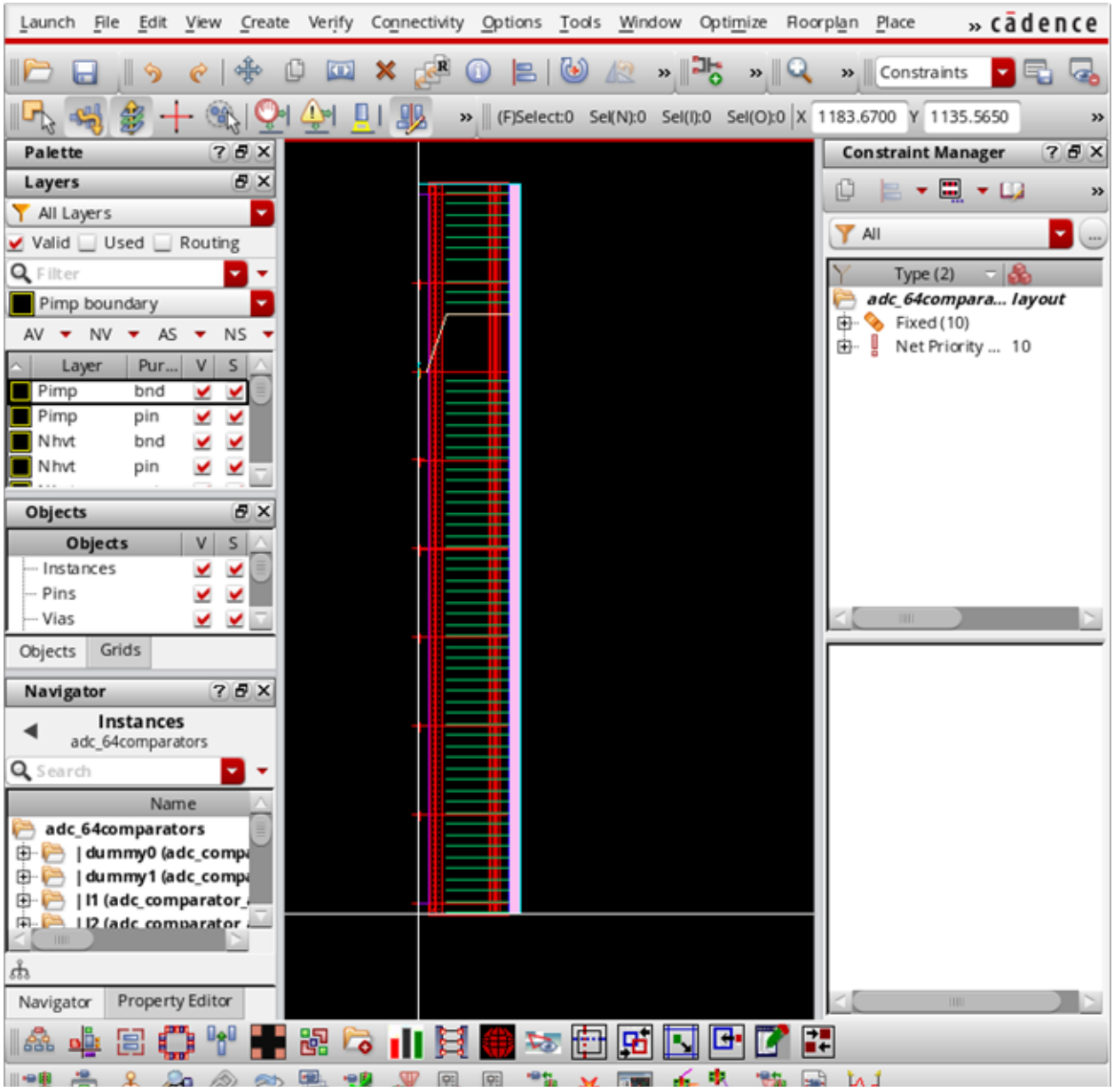
The *Classic* workspace is the default workspace. It shows the Palette assistant in addition to the canvas and the selected toolbars.



© 2023 Cadence Design Systems, Inc.

Constraints

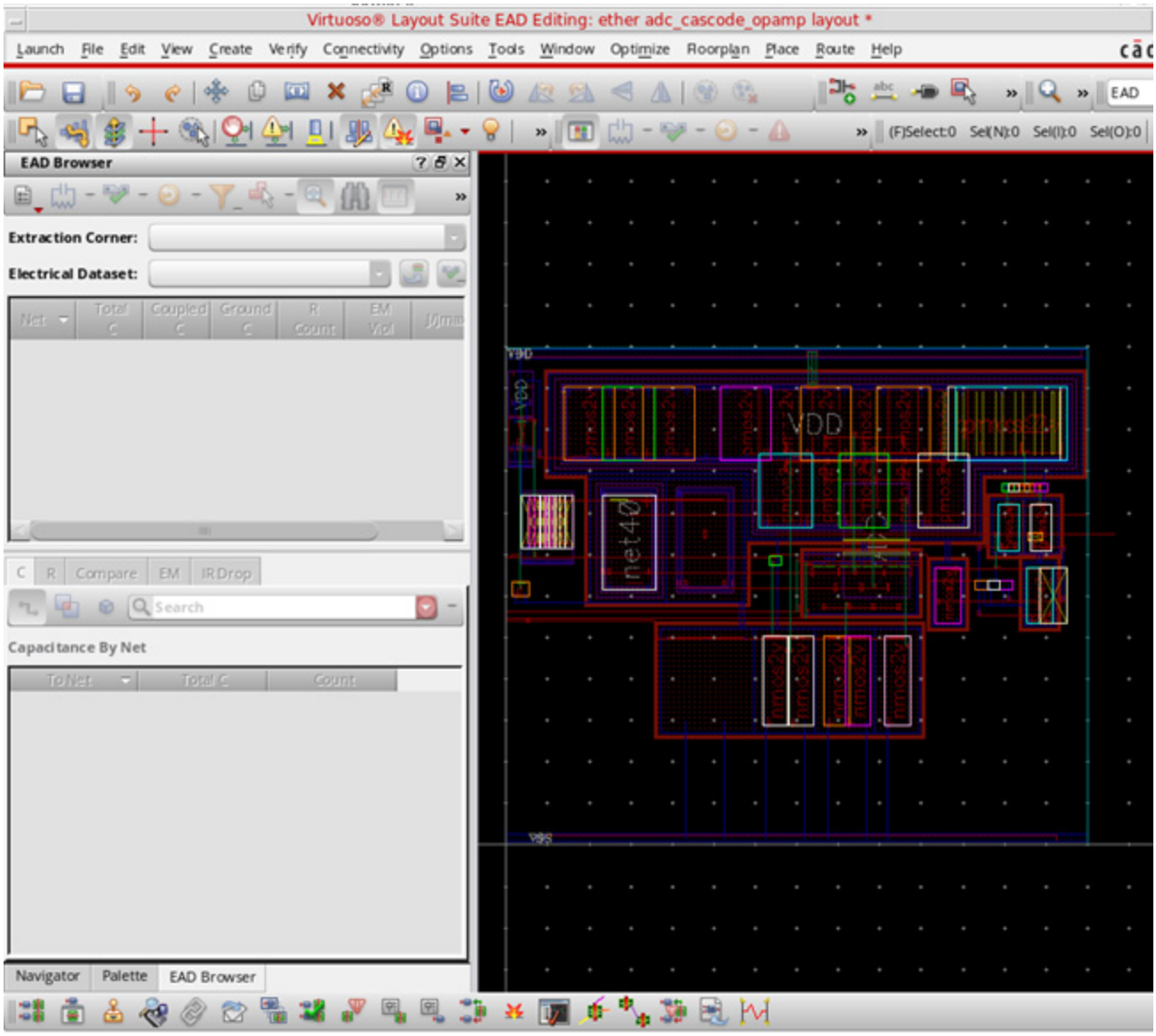
The *Constraints* workspace shows the canvas, the selected toolbars, and the Palette, Navigator, Property Editor, and Constraint Manager assistants. Notice that the Navigator and the Property Editor assistants are available as tabs in the left panel.



© 2023 Cadence Design Systems, Inc.

EAD

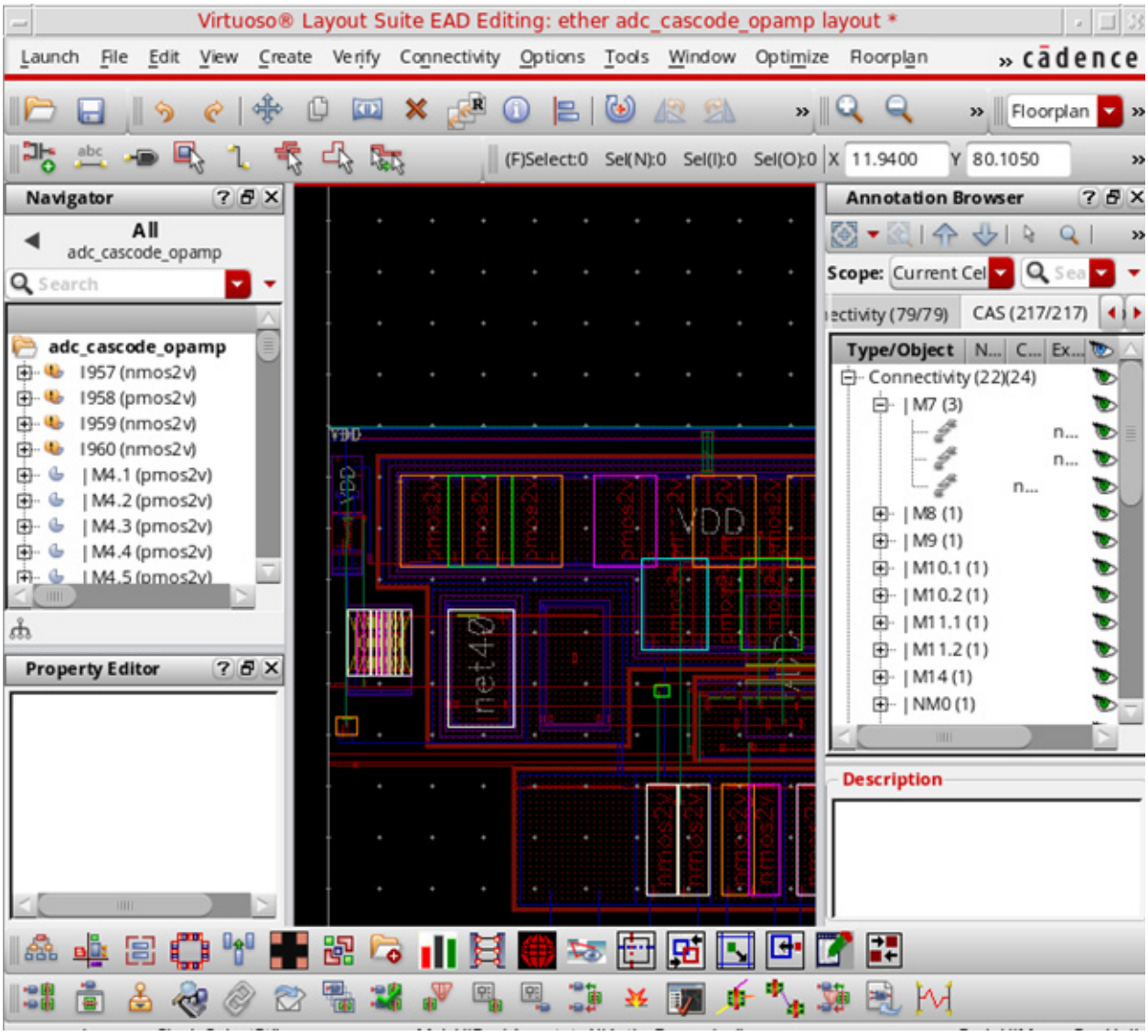
The *EAD* workspace shows the canvas, the selected toolbars, and the EAD Browser, Navigator, and Palette assistants.



© 2023 Cadence Design Systems, Inc.

Floorplan

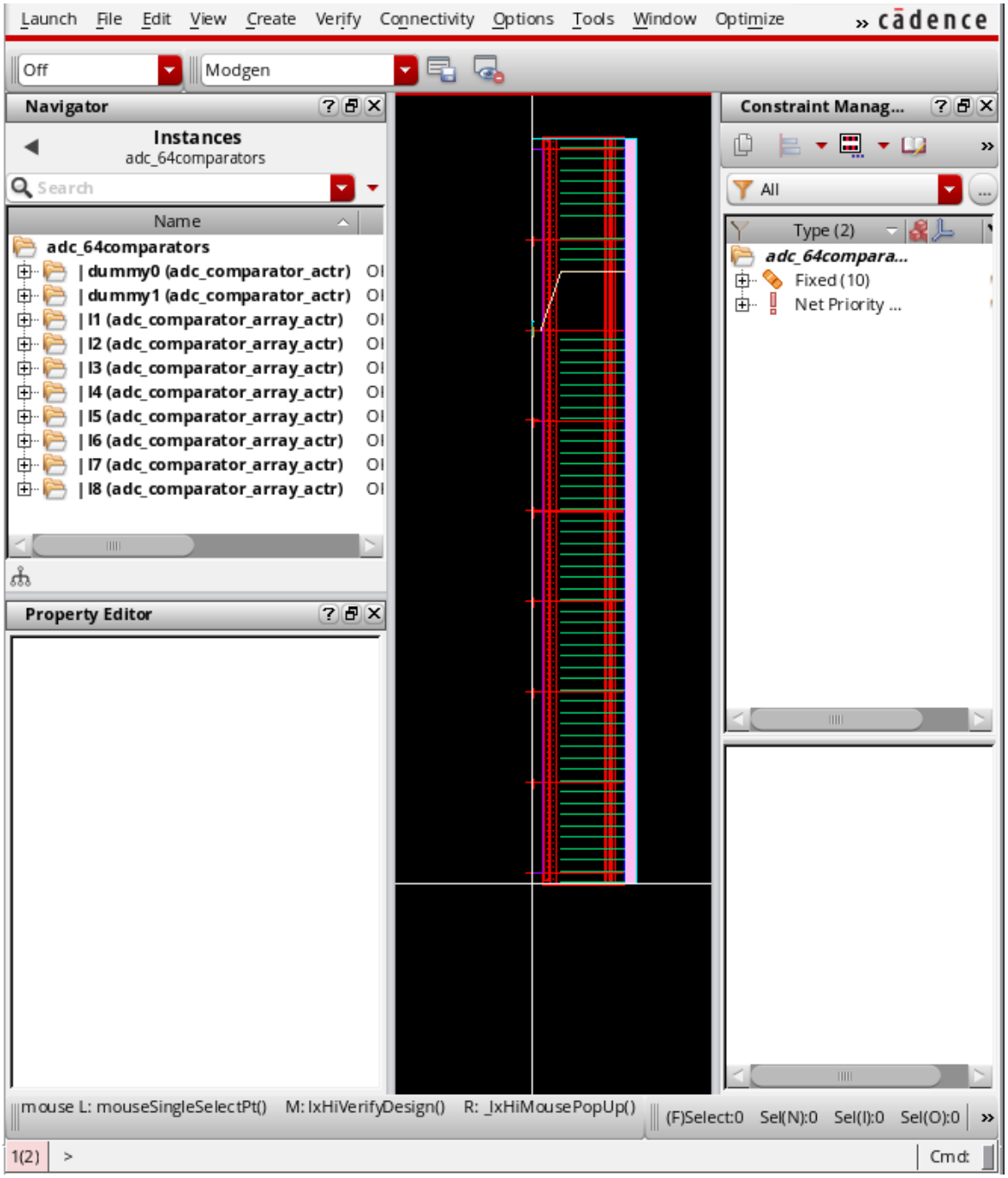
The *Floorplan* workspace shows the canvas, toolbars, and the Navigator, Property Editor, and Annotation Browser assistants.



© 2023 Cadence Design Systems, Inc.

Modgen

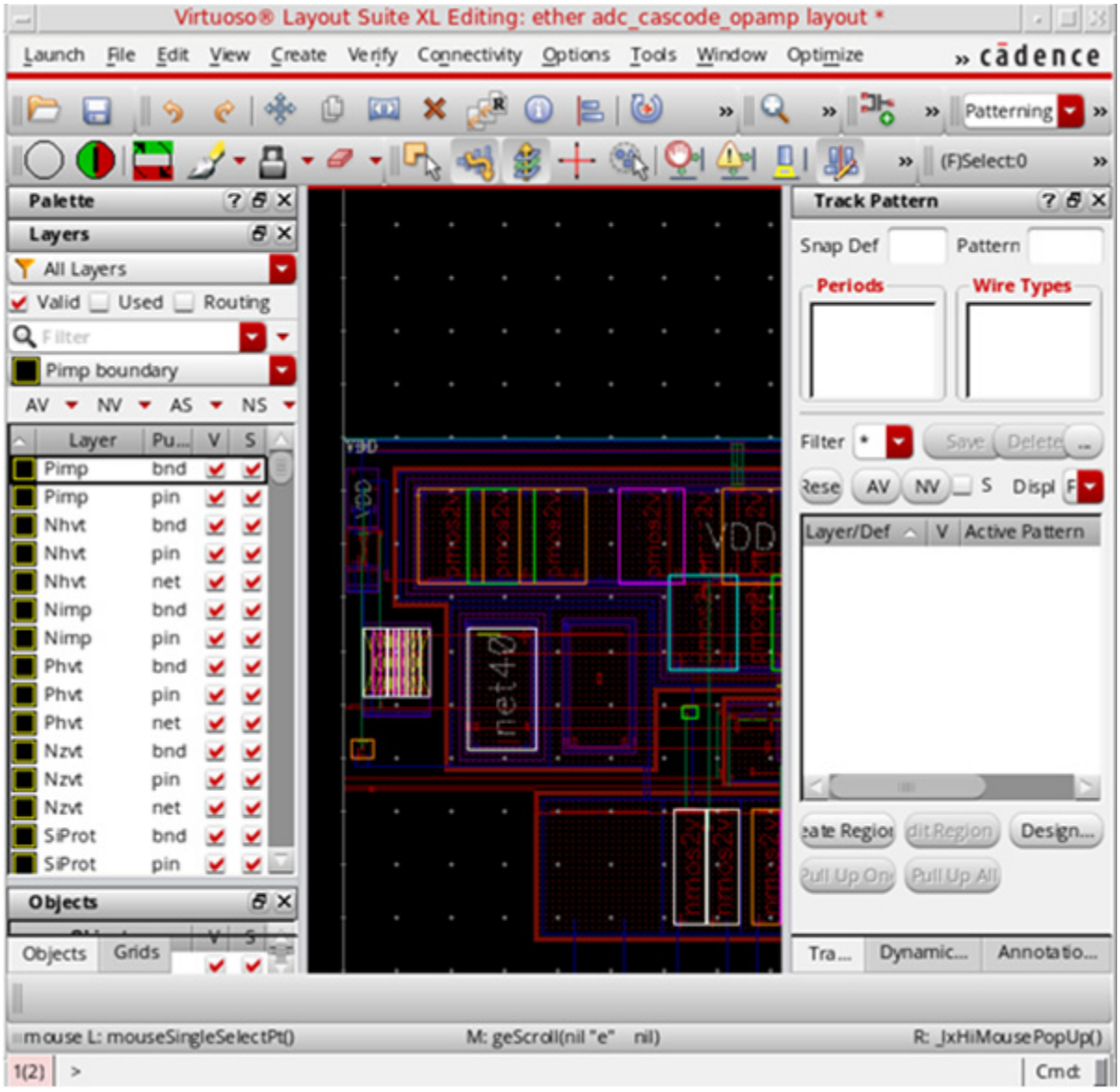
The *Modgen* workspace shows the canvas, toolbars, and the Navigator, Property Editor, and Constraint Manager assistants.



© 2023 Cadence Design Systems, Inc.

Multipatterning

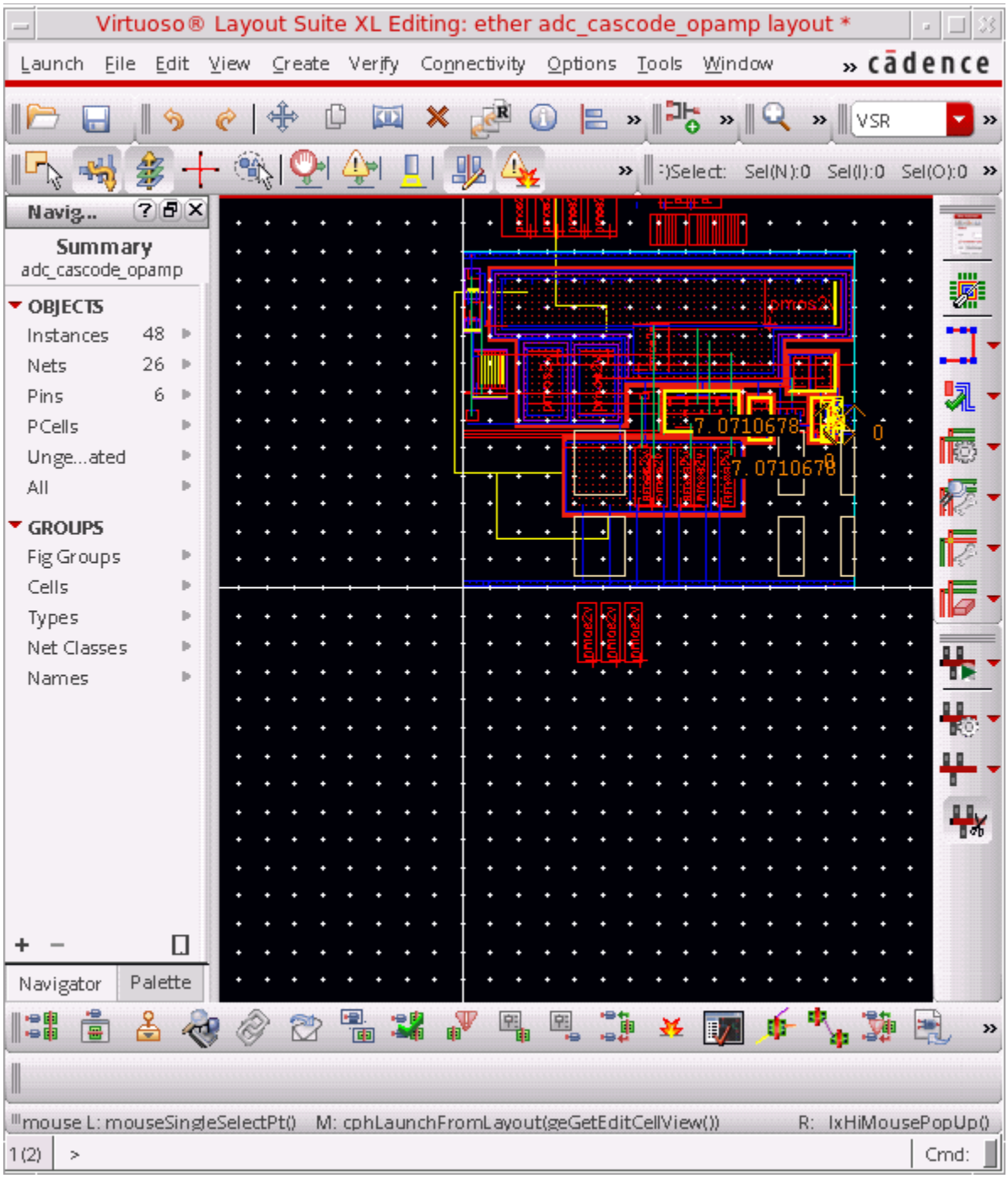
The *Mul*tipatter*ning* workspace shows the canvas, toolbars, and the Palette and Track Pattern assistants.



© 2023 Cadence Design Systems, Inc.

VSR

The *VSR* workspace shows the canvas, toolbars, the Navigator and Palette assistants, and the Virtuoso Space-Based Router toolbar.



© 2023 Cadence Design Systems, Inc.

Related Topics

Palette assistant

Navigator Assistant

Property Editor Assistant

Constraint Manager Assistant

Annotation Browser

Track Pattern Assistant

EAD Browser

Virtuoso Space-Based Router Toolbar

Getting Started with Workspaces

Working with Workspaces

Display and Selection Controls

Layout XL provides various display and selection controls to make it easier for you to work with your designs. For example, you can choose to print process information to the Command Interpreter Window (CIW). You can use shortcut menus to access commands quickly. You can select objects either before or after you start a command.

You can customize the display settings for the layout canvas. You can set the options forms to be opened automatically and change layer display colors.

Printing in the CIW

To print process information to the CIW instead of to a separate Layout XL Info window:

➤ Add the following line to your `.cdsenv` file or type it in the CIW:

```
envSetVal("layoutXL" "infoWindow" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

Multiple Cellviews Opened in Layout XL

You can open more than one pair of schematic-layout windows in Layout XL and work on each pair independently. However, when you open multiple cellviews in Layout XL, the second and subsequent sets of windows are not automatically configured in the same way as the first set.

- If you open a schematic and two copies of the same layout, the *XL Probe* command and the commands in the *Check* and *Update* submenus apply to both layouts.
- If you open a schematic and two different layouts, the *XL Probe* command and the commands in the *Check* and *Update* submenus apply only to the layout from which you selected the command.

Related Topics

Shortcut Menus

Object Selection Modes in Layout XL

Displaying Options Forms Automatically

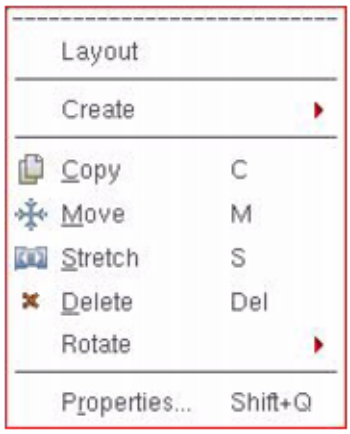
Layer Display Colors

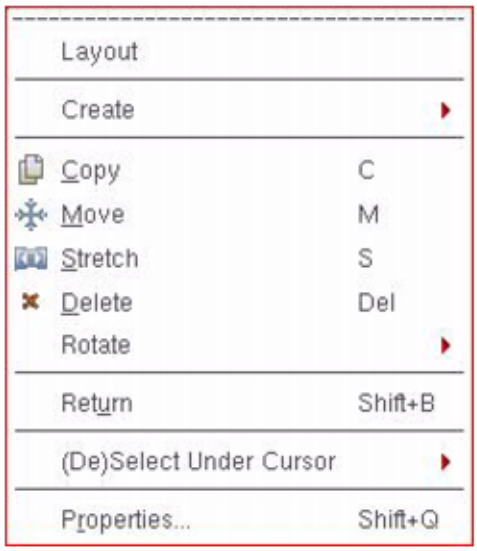
Shortcut Menus

Shortcut menus display context-sensitive options, based on the object under the mouse pointer at the time of the right-click operation. Shortcut menus provide a quick access to object-specific commands. For example, the layout editor shortcut menu lets you start a few commonly used layout commands. Right-click any free space in the design canvas to display the *Layout* shortcut menu and select the required command.

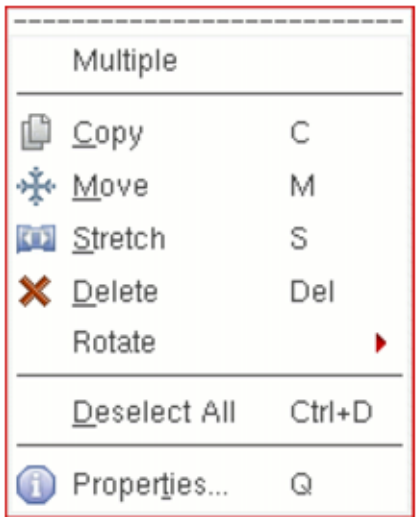
© 2023 Cadence Design Systems, Inc.

If you use *Edit In Place*, *Descend Edit*, or *Descend Read* commands, the *Layout* shortcut menu also displays the *Return* command to ascend the hierarchy.



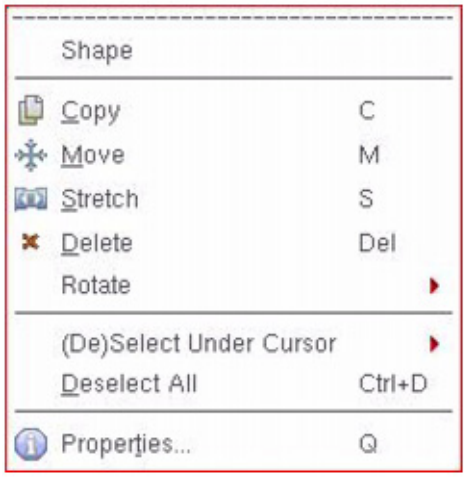


If multiple objects of different types are selected when you right-click, the *Multiple* shortcut menu appears, which allows you to perform basic editing operations on all the objects selected.



© 2023 Cadence Design Systems, Inc.

However, if all selected objects are of the same type, the object-specific shortcut menu appears. For example, if you select a group of rectangles and right-click, the *Shape* shortcut menu appears.



Related Topics

Display and Selection Controls

Object Selection Modes in Layout XL

You can select objects either before or after you start a command.

Preselect Mode

If you select objects before starting a command, the command is said to be operating in *preselect mode*. In this mode, the command operates only on the objects that were selected when the command was started. When the command is completed, the selected objects remain in the selected set.

In preselect mode:

- Editing commands do not automatically repeat, even if the *Repeat Commands* option is switched on in the Layout Editor Options form.
- Some editing commands prompt you for a starting point, called the reference point, for the edit operation.

© 2023 Cadence Design Systems, Inc.

Postselect Mode

If you select the command and then the objects, the command is said to be operating in *postselect mode*. In this mode, you make your selection within the command and when the command is completed, nothing remains selected.

In postselect mode:

- Editing commands automatically repeat if *Repeat Commands* is on (the default).
- Editing commands do not necessarily prompt you for a reference point. If you click to select the first object, the editor uses that point as the reference point.

Object Deselection

To deselect an object, right-click the object and select *(De)Select Under Cursor* from the shortcut menu. On the De(select) submenu, hover over the name of the object you want to deselect. The object gets highlighted on the canvas. Click the object name to deselect the object on the canvas.

Related Topics

Object Selection

Displaying Options Forms Automatically

There are two ways to display forms in Layout XL, depending on whether a command has a standard form or an options form.

- Standard forms: These forms have three dots (...) following the command name on the menu. These forms appear automatically.
- Option forms: These forms let you change command settings while you are running commands. These forms do not appear automatically. You can display an options form by double-clicking the right mouse button or pressing `F3` when a command is active.

To display options forms automatically when you choose a command:

1. In the CIW, choose *Options – User Preferences.*

   The User Preferences form opens.

2. Select the *Options Displayed When Commands Start* check box and click *OK* or *Apply*.

© 2023 Cadence Design Systems, Inc.

Many forms do not have a *Defaults* button because there are no appropriate default settings for that command.

Related Topics

Display and Selection Controls

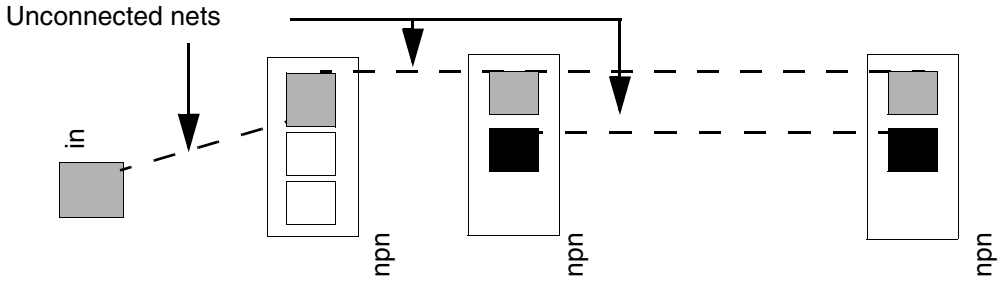
Layer Display Colors

Layout XL uses certain layers for command functionalities such as to show flight lines and highlight components. You can change the colors and characteristics of the display layers that Layout XL uses for such functionality.

- The Annotation Browser assistant uses the `y0` through `y9` layers to show flight lines.
- The *XL Probe* command uses the `hilite drawing` through `hilite drawing9` layers to probe pins, nets, and devices.
- The *Generate Selected From Source* command uses the `hilite drawing1` layer to highlight schematic symbols.
- The *Check Against Source* command uses the `hilite drawing9` entry layer to identify missing devices.
- The *Highlight* command for placement constraints uses the `annotate drawing3` entry layer to highlight in the layout the components bound by a particular placement constraint.

Incomplete Net Highlighting

To highlight incomplete connections with flight lines, Layout XL cycles through the `y0` through `y9` layers.



© 2023 Cadence Design Systems, Inc.

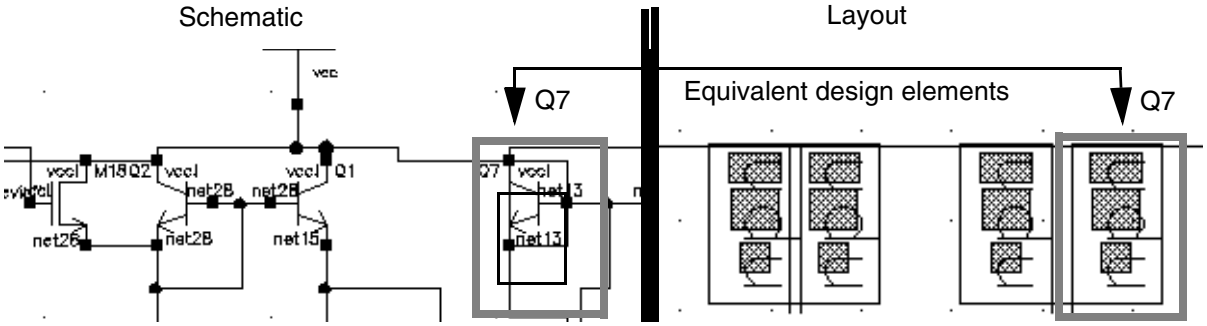
Layout XL displays the flight lines of each net in a different color. If there are more than 10 flight lines, the cycle is repeated unless you have specified certain layers for specific nets.

When assigning new colors to the flight lines:

- Use colors that do not closely resemble each other or the colors you use for paths and components.
- Avoid solid fill so you can see what is underneath.

Probe Highlighting

Layout XL uses the `hilite drawing` through `hilite drawing9` entry layers for probes, which identify equivalent design elements on the schematic and the layout.



The *Check Against Source* command uses the `hilite drawing9` layer to identify components in the schematic that are not in the layout, and vice versa.

If you change the colors of the `hilite drawing` layers, consider the following:

- Use thick lines so that you can see device pins.
- Use bright colors to make probes and highlights easy to find.

Do not make the `hilite drawing` layer and the `hilite drawing2` layer solid fill because these layers are used to indicate selected components and to manipulate shapes.

Related Topics

Annotation Browser

Palette assistant

© 2023 Cadence Design Systems, Inc.

XL Probe Form

Placement Modes for Generating Selected Components from Source

Layout Editor Strokes

A stroke is a unique shape you draw in a cellview by pressing and holding the right mouse button. Cadence ships a set of predefined strokes for the layout editor that you can load.

To start a command with a stroke, using the right mouse button you can draw one of the strokes listed in the following table. The arrows indicate the direction in which you draw the strokes. Note that there are two ways to start *Zoom In* with strokes.

Stroke	Drawing Direction	Command
		Copy
		Delete
		Window Fit
		Move
		Properties

© 2023 Cadence Design Systems, Inc.

Stroke	Drawing Direction	Command
		Undo
		Stretch
		Zoom In
		Zoom In
		Zoom Out
		Pan Right
		Pan Left
		Pan Up
		Pan Down

© 2023 Cadence Design Systems, Inc.

Related Topics

Loading and Unloading Strokes

Loading and Unloading Strokes

Before you can use the predefined layout editor strokes, you must load them from the installation hierarchy. Note that after you load strokes, you cannot use the right mouse button for any other functions.

To load layout editor strokes:

➤ Type the following in the text line of the CIW:

```
load(prependInstallPath( "etc/sted/stroke.il"))
load(prependInstallPath( "etc/sted/defstrokes.il"))
hiLoadStrokeFile("def.strokes" "Layout")
```

Where:

- `prependInstallPath` is a SKILL function that uses the value set for the UNIX environment variable `CDS_INST_DIR`, which usually points to *`install_dir`*`/tools/dfII`.
- The `stroke.il` file defines mouse key bindings and the applications that recognize strokes.
- The `defstrokes.il` file lists the SKILL language functions used by the predefined strokes.
- The `def.strokes` file defines the stroke shapes.

You can also add these commands to your `.cdsinit` file, so the strokes are loaded whenever you start the software.

To unload layout editor strokes:

1. Quit the current session and restart the software.
2. Remove the load strokes commands from your `.cdsinit` file if added them to the file.

Related Topics

Layout Editor Strokes

© 2023 Cadence Design Systems, Inc.

2

Design Display and Search Controls

You can use various display controls to quickly navigate around the canvas window and view your designs. For example, you can zoom, pan, redraw, or magnify a design, save and restore a view, add a background cellview, and view the coordinates of a shape.

You can use the robust search functionality that makes design data search both simple and productive. You can choose from a wide range of search criteria to search for objects with attributes and properties that match your search criteria. Also, you can easily replace search results of an object with a new object, attribute value, or property value.

Related Topics

Design Display

Find Functionality

Replace Functionality

Object Search Criteria

© 2023 Cadence Design Systems, Inc.

Design Display

The following table describes the commands available on the *View* menu of a layout window that you can use to view and manage your designs.

Command	Description
Zoom	Controls magnification of a cellview through the following zoom commands: ■ *Zoom In*: Magnifies the image in the cellview window by a factor of 2. To zoom in, you can also scroll up the mouse scroll wheel. ■ *Zoom Out*: Reduces the image in the cellview window by a factor of 2. To zoom out, you can scroll down the mouse scroll wheel. ■ *Zoom To Area*: Prompts you to create a box on the canvas and then magnifies the image defined by the box to fit in the cellview window. ■ *Zoom To Grid*: Reduces the image to the smallest magnification at which the grid is visible. ■ *Zoom To Selected*: Increases the image to the largest magnification at which the selected objects can be viewed in the cellview window. ■ *Zoom To Fit All*: Redraws the window so that all objects in the cellview appear in the window. This saves time after you zoom or pan and want to see the entire cellview again. ■ *Zoom To Fit Edit*: Places the cell that you are editing in the center during edit-in-place. If you are not editing in place, *Zoom To Fit Edit* has the same result as *Zoom To Fit All*. For more information, see Zooming a Design.
Magnifier	Lets you magnify or zoom a specific part of the design display area, rather than the entire design display area. For more information, see Magnifying the Design Area.
Dynamic Zoom	Pans or zooms the view automatically on the canvas based on the objects selected in the Navigator assistant. For more information, see Using Dynamic Zoom.

© 2023 Cadence Design Systems, Inc.

Command	Description
Pan	Lets you pan a point in the center of the cellview window. For more information, see Panning a Cellview.
Redraw	Refreshes the cellview display. This is helpful after you have made edits to a cellview and the design area sometimes shows portions of lines or objects that you moved or deleted.
Area Display	Sets the display detail for an area. For more information, see Setting an Area Display.
Show Coordinates	Displays the coordinates of the selected shape. For more information, see Viewing Coordinates.
Show Angles	Lets you display the angle enclosed at the intersection of two edges. For more information, see Viewing Angles.
Show Selected Set	Displays information about a selected set of objects. For more information, see Show Selected Set Window.
Save/Restore	Saves a zoomed or panned image to a file that lasts for the current session. You can later restore a saved view or cycle through the last three views. For more information, see Saving and Restoring a View.
Background	Lets you place an existing cellview in the background so you can place your new elements in alignment or by an offset based on the previously created cellview. For more information, see Background Cellview.

Setting an Area Display

To set the display detail for an area:

1. In the layout window, choose *View – Area Display – Set*.

   The Set Area View Level form opens.

2. Set the *From* and *To* display levels for the area.
3. On the canvas, create a box around the area in which you want to view the detail of the lower levels of the design.

To remove one detail area at a time:

1. Choose *View – Area Display – Delete*.

© 2023 Cadence Design Systems, Inc.

2. Click the detail area you want to remove.

To remove all detail areas:

➤ Choose *View – Area Display – Delete All.*

Related Topics

Design Display

Viewing Coordinates

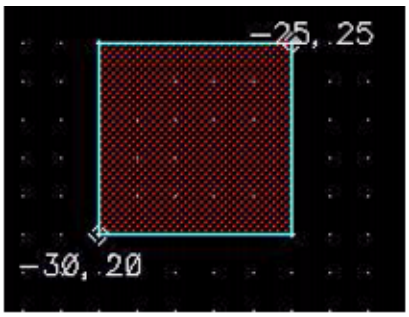
The *Show Coordinates* command displays the coordinates of the selected shape. To view coordinates of a shape, select it fully or partially. You can select a shape either before (pre-selection mode) or after (post-selection mode) you run the *Show Coordinates* command.

- Viewing Coordinates of a Shape
- Viewing Coordinates of All Vertices of a Shape
- Viewing Coordinates of Vertices of Multiple Shape
- Examples of Coordinates of Shapes

Viewing Coordinates of a Shape

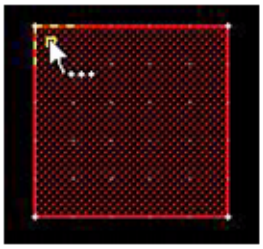
To view the coordinates of a shape:

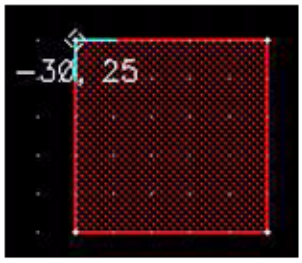
1. In the layout window, choose *View – Show Coordinates.*
2. Select a rectangle as explained below:
   - Point at or select the complete shape: The coordinate information is displayed for the lower-left and top-right vertices of the rectangle, as shown in the figure below. Notice the diamond-shaped markers identifying the coordinate locations.



© 2023 Cadence Design Systems, Inc.

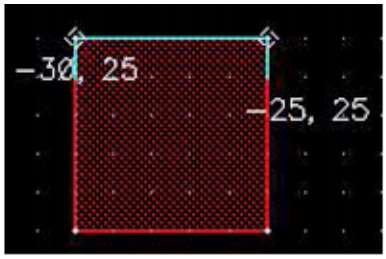
- Point at or select a vertex: The coordinate information is displayed at the selected vertex of the rectangle, as shown in the figure below.





- Point at or select an edge: The coordinate information is displayed at the two vertices of the selected edge of the rectangle, as shown in the figure below.



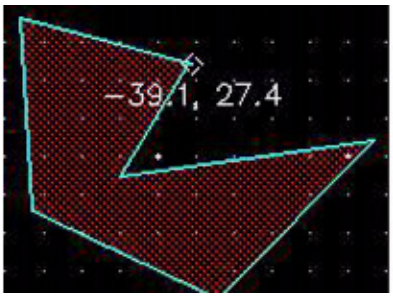


Viewing Coordinates of All Vertices of a Shape

To view the coordinates of all the vertices of a selected shape:

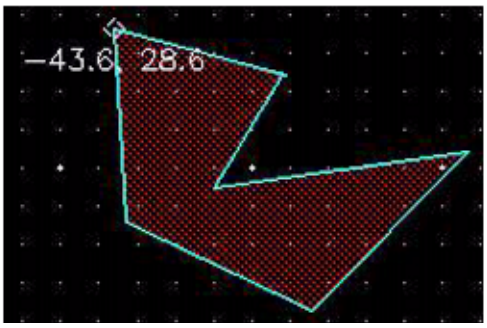
1. Select a shape, say a polygon.

   The coordinate information is displayed for one of the vertices, as shown in the figure below.



© 2023 Cadence Design Systems, Inc.

2. Press the spacebar to display the coordinates of the next vertex.



You can continue to press the spacebar to view the coordinates of all the vertices in succession.

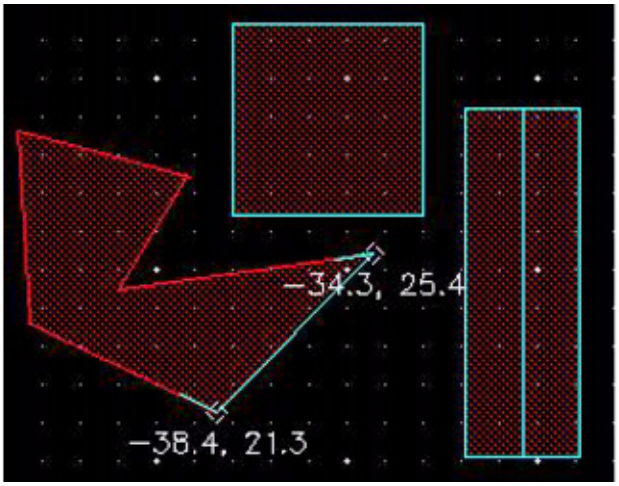
If you select a shape partially, only the coordinates of the selected vertices are displayed.

Viewing Coordinates of Vertices of Multiple Shape

To view the coordinates of the vertices when multiple shapes are selected:

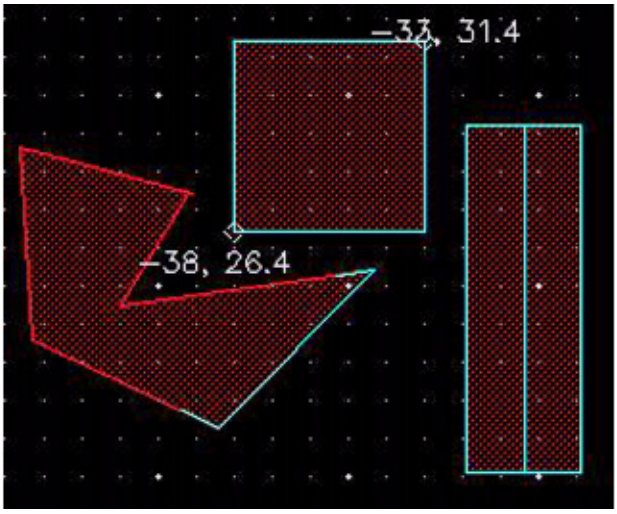
1. Select multiple shapes, say a rectangle, a path, and the edge of a polygon.

   The coordinates are displayed for each shape in the selected set one by one, for example, for the selected edge of the polygon, as shown in the figure below.



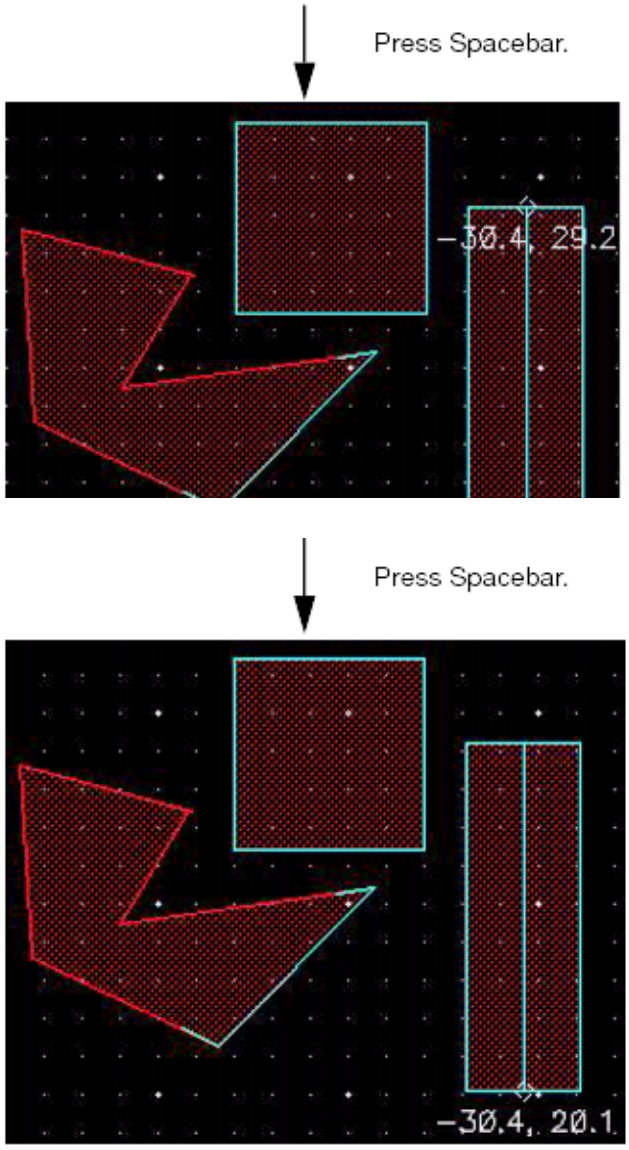
© 2023 Cadence Design Systems, Inc.

2. Press the spacebar again to view the next set of coordinates—lower-left and upper-right vertices of the rectangle, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

3. Continue to press the spacebar to view the remaining sets of coordinates.



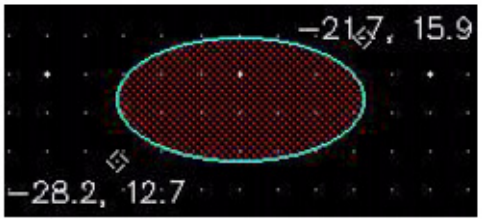
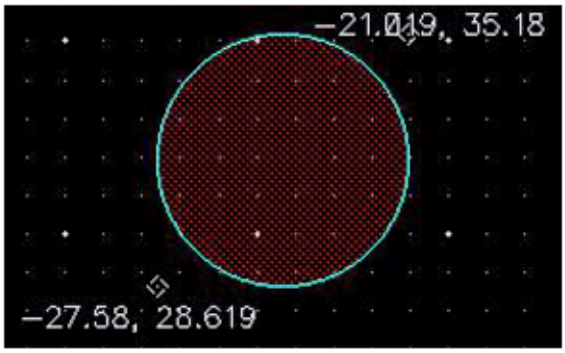
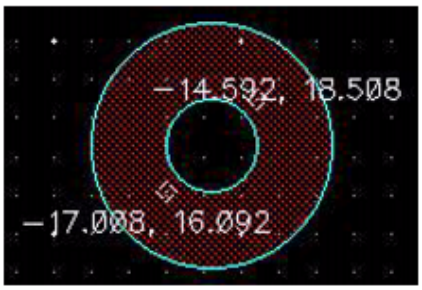
4. Press `Esc` to finish the command.

If you first run the *Show Coordinates* command and then select the required shapes, you can repeatedly cycle over the coordinates by pressing the spacebar. On the other hand, if you first select the required shapes and then run the *Show Coordinates* command, the command exits after cycling through all the coordinates once.

© 2023 Cadence Design Systems, Inc.

Examples of Coordinates of Shapes

The following table shows examples of coordinates displayed for various shapes:

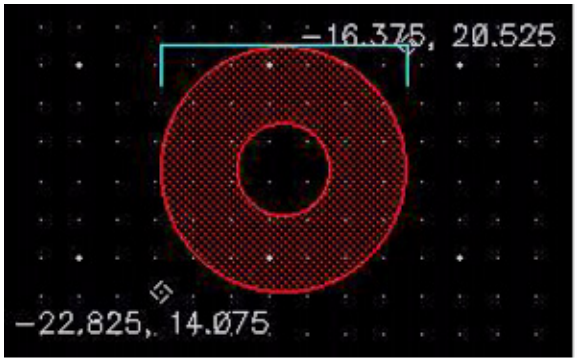
Path 	Ellipse 
Circle 	Donut 

Circles and ellipses have two sets of coordinates, one for the bounding box and one for the center point.

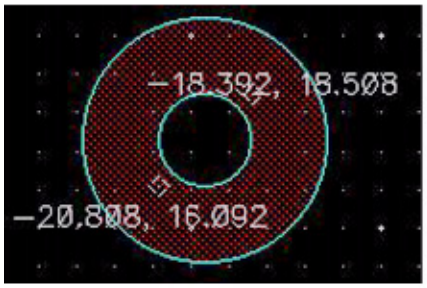
Rectangles, paths, labels, and vias too have only one set of coordinates. Selecting a label, or via fully or partially displays the same set of coordinates, that of their respective origins. Selecting an instance displays the lower-left and upper-right coordinates of the bounding box, and after you press the spacebar the origin is displayed.

© 2023 Cadence Design Systems, Inc.

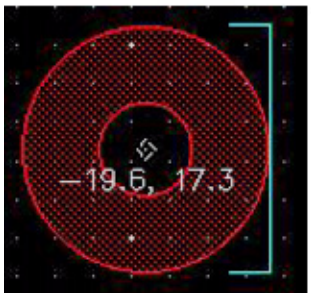
A donut has three sets of coordinates—one each for its two bounding boxes and one for the center point—as shown in the figures below.



Outer Bounding Box Coordinates



Inner Bounding Box Coordinates



Center Point Coordinates

You can control the font size and color of the coordinates displayed by using the *Dynamic Measurement* tab of the Dynamic Display form.

Related Topics

Design Display

Dynamic Display Form

© 2023 Cadence Design Systems, Inc.

Find Functionality

Use the find functionality to search for objects that have an attribute or property that matches your search criteria. You can choose from a list of conditions and logic to form the search criteria. For example, you can search for ROD objects, blockages, and boundaries based on the following criteria:

- ROD objects: You can search for ROD names on labels, paths, polygons, rectangles, any shape, and text display objects.
- Blockages: You can search for blockages by blockage type, layer name, and owner name. You can use the Zoom to Figure and Previous and Next buttons to navigate through all blockages. However, you cannot replace any of the values based on search results.
- Boundaries: You can search for different types of boundaries in the following manner:
  - PR and snap boundary by object type
  - Area boundary by boundary name and property
  - Cluster boundaries by boundary name, cluster name, and property

  You cannot replace any of the values based on the search results.

For information about the complete list of conditions and logic to use when searching for objects, see Object Search Criteria.

Related Topics

Search Criteria

Finding Objects

Replace Functionality

Search Criteria

While defining the search criteria, you can specify a text entry or choose from a cyclic field, depending on the type of value you want to search for. To search for multiple objects, you can use wild card characters such as asterisk (`*`) and dot (`.`).

- The asterisk (`*`) searches for the entry that you provide, with any number of unspecified characters before, after, and/or between any characters in your entry. For example, to

© 2023 Cadence Design Systems, Inc.

search for all instances that have `PInst` at the beginning of their names, you specify the entry as "`PInst*`".

- The dot (`.`) searches for the entry that you provide, with any single unspecified character at any position in your entry. For example, to search for all instances that have `Inst` in their names irrespective of the first character, you specify the entry as "`.Inst*`".

The use of a wild card character to search for multiple instances returns *all* results that meet the specified criterion, even if you are looking only for fewer instances. For example, if you are searching for only `PInst1` and `PInst2` out of all the instances that have `PInst` at the beginning of their names, specifying the search criterion as one of the following does not give the desired results:

- inst name == `PInst1 PInst2`
- inst name == `PInst1, PInst2`
- inst name == `PInst1`
  inst name == `PInst2`

The following table lists the wildcard characters that you can include in a search string.

Wildcard Character	Description	Example
`*`	Matches any number of characters.	`po*` finds `point`, `points`, `polygon`, but not opposite, `export`.
`.`	Matches any alphabetical character.	`p.n` finds `pan`, `pin` but not `polygon`, `phon`.
`[ ]`	Matches any character within the brackets.	`t[io]p` finds `tip`, top but not `tap`
`-`	Matches any one of a range of characters.	`t[d-p]` finds `tip`, `top` but not `tap`.
`\`	Allows characters `[`, `*`, and `\` to be considered literally.	Finds `poly*1`

The following table lists some more examples of search strings that include wildcard characters.

Expression	Result
`Ci*e.`	Finds `Circle1`, `Circlea`, `Cie2` but not `circle1`, `circle12`, `Cie`, and `Circle*1`.

© 2023 Cadence Design Systems, Inc.

Expression	Result
`[a-z,A-Z]ire.[0-9]`	Finds `wire`, `Wire`, `Fire`, `fire` but not `rewire` and `afire`.
`poly\*. or poly[*].`	Finds `poly*1`, `poly*2` but not `polygon`.
`A.B[0-9]C[0-9]`	Finds `A1B1C1`, `A2B2C3` but not `ABC` and `AB2C4`.

Related Topics

Finding Objects

Find Functionality

Object Search Criteria

Replace Functionality

Finding Objects

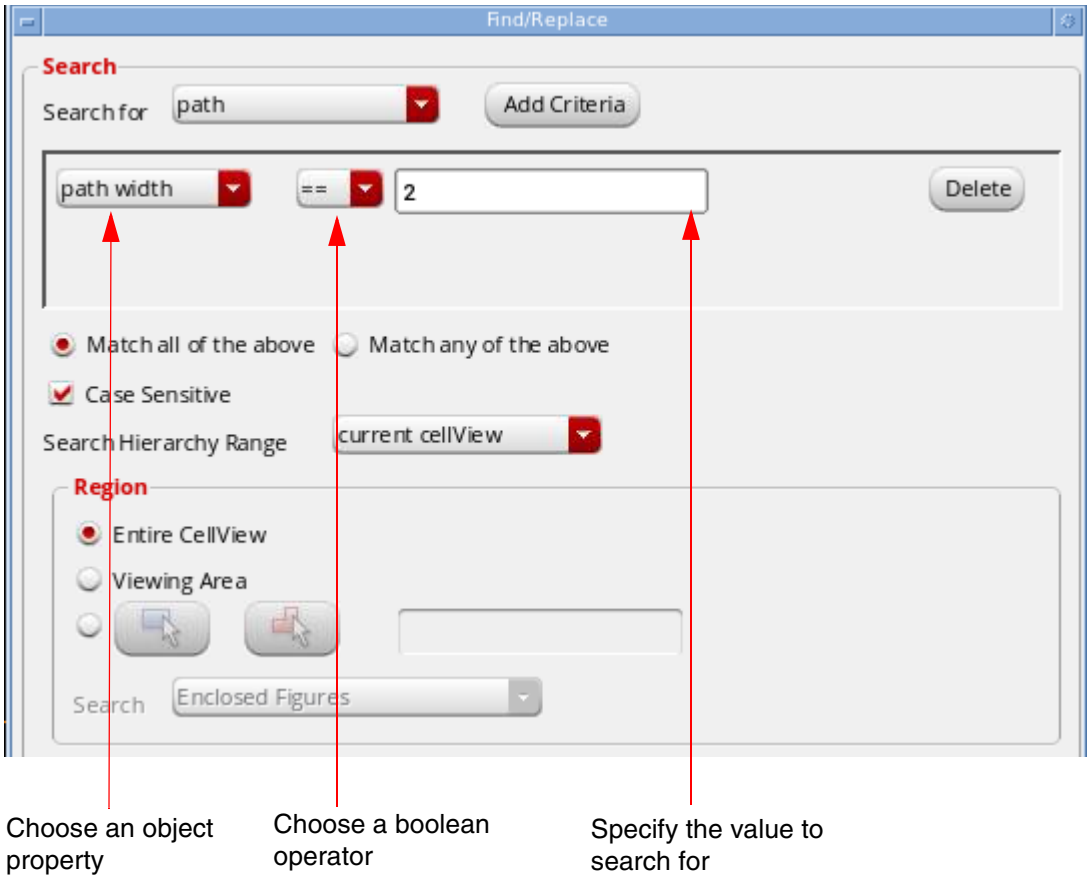
To find objects:

1. In the layout window, choose *Tools – Find/Replace*. Alternatively, press `Shift+S`.

   The Find/Replace form appears.
2. Select the type of object you want to find from the *Search for* list.
3. Select the location where you want to perform the search by choosing from the *Search Hierarchy Range* list. By default, the search is performed in the current edit cellview in the layout window.
4. Click *Add Criteria* to add boolean expressions, which control the search criteria.
5. Set the search criteria:
   a. From the first list, choose the object property that you want to search.
   b. From the boolean operator list, choose the operator you want to use, such as equal (==) or not equal (!=).

© 2023 Cadence Design Systems, Inc.

c. Type or choose the value that you want to search.



- Click *Delete* at the end of the row if you want to delete a row of search criteria.
- Click *Add Criteria* again if you want to add more search criteria.

6. Click *Find*.

All the objects matching the specified search criteria are highlighted on the canvas and are put in a search group. The first or current object in the group is highlighted in a different color.

7. Optionally, click *Previous* or *Next* to search through the group.

8. Optionally, click *Add Select* to select the current object in the search group.

9. Optionally, click *Select All* to select all the highlighted objects.

10. Click *Close* when you are done selecting objects.

© 2023 Cadence Design Systems, Inc.

Related Topics

Find Functionality

Search Criteria

Replace Functionality

Object Search Criteria

Replace Functionality

The search operation finds the objects that have a property that matches the search criteria you specify. You can replace the search results of an object with a new object, attribute value, or property value. In the Replace section of the Find/Replace form, you can specify the value of the property with which you want to replace the original value. The value specified replaces the property values of all the objects matching the search criteria. For example, to replace parameter values for a parameterized cell, set the replace criterion to *property* and specify the parameter name in the replace field. You can use *Add Criteria* to add a line of fields to the form that lets you make the replace operation more specific.

You can use partial strings and wildcard characters in the replace field. The following table describes their usage.

Replace String-Tagged Characters	Description
`{`*n*`}`	Interprets *n* as an integer from 1 to 9. Tagged fields correspond to `*` or `.` or `[c..]` or `.*` in the search string. The first nine wildcard characters in the search pattern are tagged. If a tagged field that is used in the replace string does not exist, it is replaced by a NULL string.
`*nth "*"`	Interpreted as {*n*}, *n* ranging from 1 to 9. If more than nine wildcard characters exist in the search string and more than nine `*`'s in the replace string, the system substitutes the first nine `*`'s in the replace string with the corresponding nine tagged fields from the search string. The remaining `*`'s in the replace string are substituted with a NULL string.

The following table lists some examples of find and replace expressions using partial strings and wildcard characters.

© 2023 Cadence Design Systems, Inc.

Note: If wildcard characters appear together in a sequence in the find string, each corresponds to a different tag except when `.` and `*` appear together, in that order. In such cases, wildcard characters are treated together as one tag. If the find string does not have a wildcard character and the replace string does, a substitution is not achieved.

Example	Find Expression	Replace Expression
Replace `I0`, `I1`, `I2`, …`I10` with `M1`, `M2`, …`M10`	`I*`	`M*` or *M{1}* The `{1}` tag or `*` in the replace string corresponds to the characters represented by the first asterisk in the find string.
Swap the start and end values of strings with prefix `ABC`	`ABC<*:*>`	`ABC<{2}:{1}>` The `{2}`and `{1}` tags in the replace string correspond to the characters represented by the second and first asterisk in the find string, respectively.
Replace `PN`, `PN_`, `PN_IN` with `PIN`, `PIN_`, `PIN_IN`	`PN*`	`PIN*` or `PIN{1}`
Replace `AfooB` , `AblahB` with `PfoofooQ`, `PblahblahQ`, respectively.	`A*B`	`P{1}{1}Q`
Replace `AxyzB` with `CxD`	`A.*[x-z]B`	`C*D` The `*` in the replace string is interpreted as the first tagged field, which is the "`.`" wildcard. `C***D` or `C{1}{2}{3}` will yield `CxyzD`.
Replace `AxyzB` with `CxyD`	`A.*[x-z]B`	`C*D` The * in the replace string is interpreted as the first tagged field in the find string, which is the sequence " `.*` ". `C**D` or `C{1}{2}` yields `CxyzD`.
Replace `A.B` with `A*B`	`A*B` or `A.B`	`A\*B`

© 2023 Cadence Design Systems, Inc.

Example	Find Expression	Replace Expression
Replace `A[1]`, `A[2]`, `A[3]` with `A{1}`, `A{2}`, `A{3}`	`A\[*] or A\[[0-9]]`	`A\{{1}}` or `A\{*}` **Note:** The `\` before the first `{` in the replace string treats the `{` as a literal and not as a tagged field.

Video

To learn about find and replace functionality enhancements, watch the video, *Find/ Replace Enhancements*.

Related Topics

Replacing Instances

Finding and Replacing Text of Labels

Object Search Criteria

Find Functionality

Replacing Instances

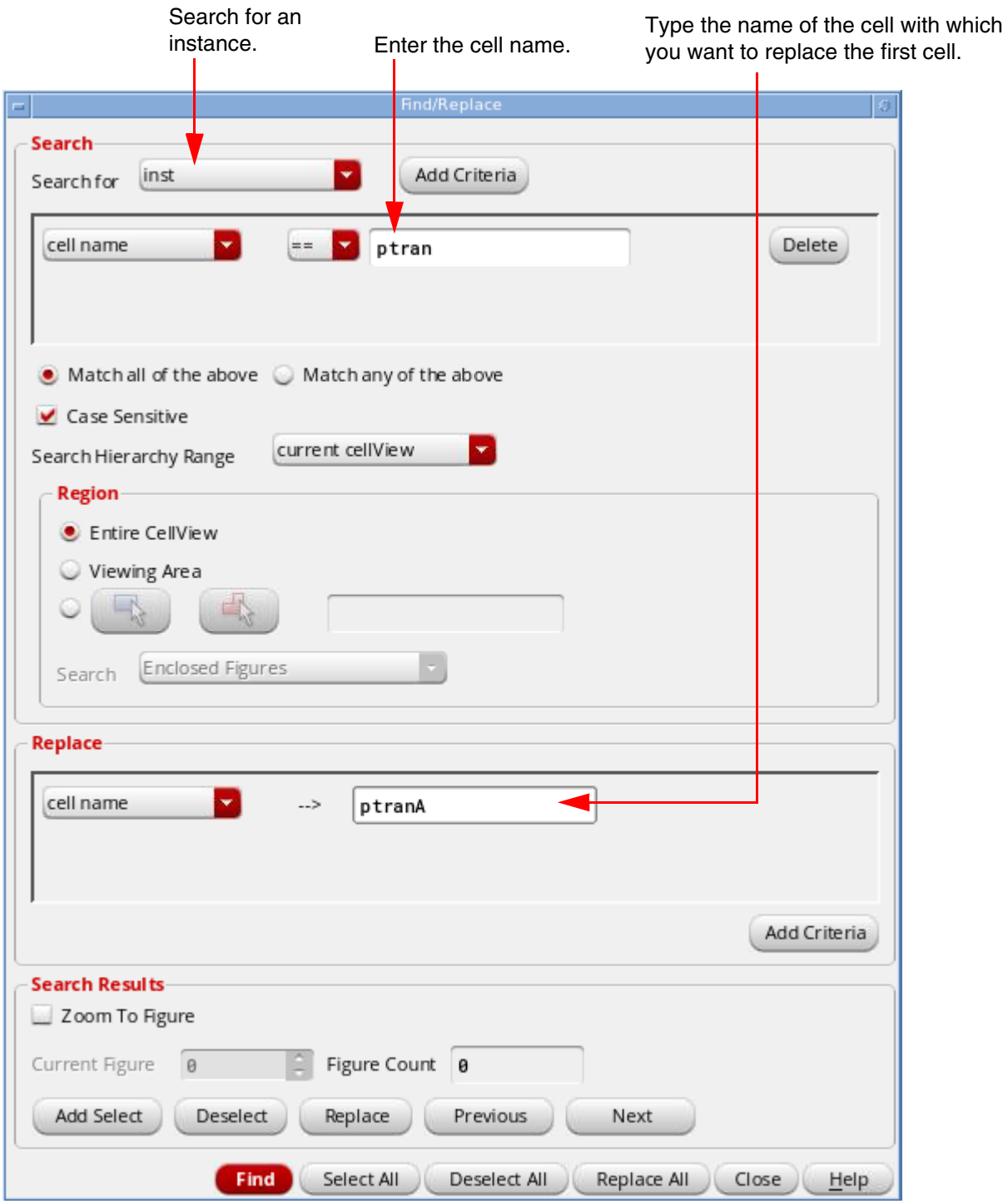
To replace instances of one master cell with instances of a different master cell:

1. In the layout window, choose *Tools – Find/Replace*. Alternatively, press `Shift+S`.

   The Find/Replace form appears.
2. Specify the criteria to search for the required instances.
3. In the *Replace* section, choose *cell name* from the replace list.

© 2023 Cadence Design Systems, Inc.

4. Type the name of the new master cell.



© 2023 Cadence Design Systems, Inc.

Similar to the find criteria, you can optionally specify multiple replace criteria using the *Add Criteria* option in the Replace section if you want to replace multiple attributes together.

5. Click *Find*.

   The layout editor highlights all the instances of the master cell.

6. To replace only one instance of the cell, click *Previous* or *Next* until the instance you want is highlighted.

7. Click *Replace* to replace the current matching instance.

8. Optionally, click *Replace All* to replace all instances of the highlighted cell.

Similar to the Find option, you can specify multiple replace criteria using the *Add Criteria* option in the *Replace* section of the Find/Replace form. This enables you to replace multiple attributes together.

Related Topics

Search Criteria

Finding and Replacing Text of Labels

Object Search Criteria

Find Functionality

Finding and Replacing Text of Labels

To search and replace labels in the current cellview:

1. In the layout window, choose *Tools – Find/Replace* or press `Shift+S`.

   The Find/Replace form appears.

2. Choose *label* from the *Search for* list.

3. Click *Add Criteria* to add the search criteria.

   The first field in the criterion line shows *text* by default.

4. To search for all the labels in the cellview, type `*` in the text field. You can specify multiple criteria to search for the required labels.

© 2023 Cadence Design Systems, Inc.

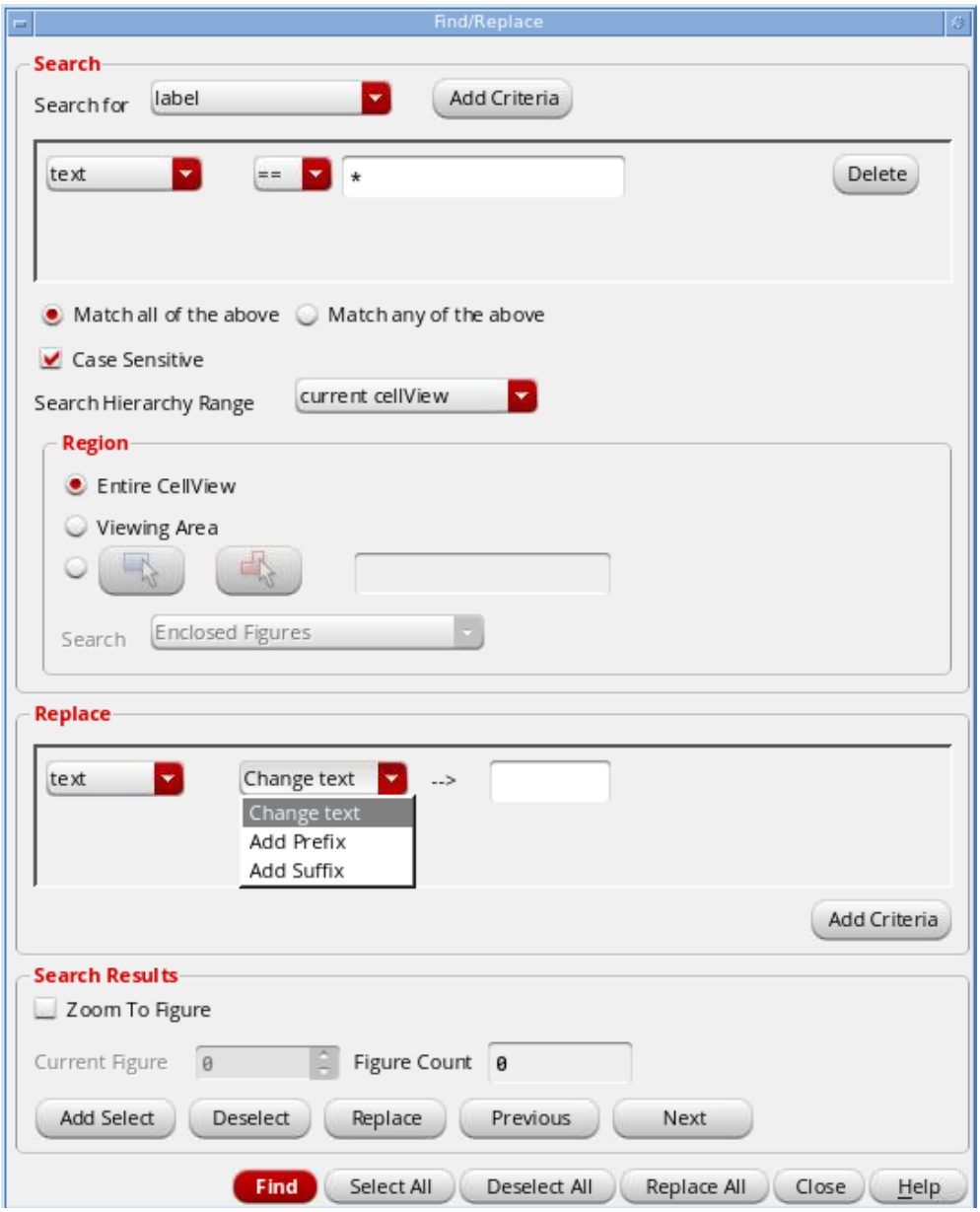
5. Select the location where you want perform the search by choosing from the *Search Hierarchy Range* list. By default, the search is performed in the current edit cellview in the layout window but you can choose to perform the operation across different hierarchy levels.
6. Click *Find*.

   The layout editor highlights all the labels in the cellview.
7. In the *Replace* section, choose *text* from the first list.
8. Choose one of the following from the second list:

   - *Change text*: Select this option and specify the text to replace the label text of all the labels matching the search criteria.
   - *Add Prefix*: Select this option and specify the text to be prefixed to all the labels matching the search criteria.

© 2023 Cadence Design Systems, Inc.

- *Add Suffix*: Select this option and specify the text to be added at the end of all the labels matching the search criteria.



9. To replace only one label, click *Previous* or *Next* until the label you want is highlighted and then click *Replace*.
10. To replace all the highlighted labels, click *Replace All*.

© 2023 Cadence Design Systems, Inc.

Related Topics

Replacing Instances

Replace Functionality

Object Search Criteria

Find Functionality

Object Search Criteria

You can choose from a wide range of search criteria to search for objects and replace the search results of an object with a new object, attribute value, or property value.

The following table describes the properties, conditions, and types of values that you can use to form the criteria to search for an object and replace the search results.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
PR boundary	No *Add Criteria*			
any conic	*any radius*		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.
any microstrip	*property*	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	net name		*==* and *!=*	Text
	no net name			No input required.
any shape	*layer*	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	ROD name		*==* and *!=*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.
	bbox height		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	bbox width		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	area		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	color		*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	color state		*==* and *!=*	Cyclic field. Choices are *unlock*, *lock*
area boundary	*boundary name*		*==* and *!=*	Text
	property		*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
array	*cell name*	*cell name*	*==* and *!=*	Text
	lib name	*lib name*	*==* and *!=*	Text
	rows	*rows*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	columns	*columns*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	deltaX	*deltaX*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	deltaY	*deltaY*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	arrayname	*array name*	*==* and *!=*	Text
	view name	*view name*	*==* and *!=*	Text
bend	*bend style*	*bend style*	*==* and *!=*	Cyclic field. Choices are *bend*, *chamfer*, and *radial*.
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	net name		*==* and *!=*	Text

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	no net name			No input required.
blockage	*blockage type*		*==* and *!=*	Cyclic field. Choices are *placement*, *routing*, *fill*, *slot*, *pin*, *feedthru*, and *screen*.
	layer name		*==* and *!=*	Combination field. Choices are none, Unrouted, Canplace, Cannotoccupy, Group, Row, marker, changedLayer, designFlow, and unset.
	owner name		*==* and *!=*	Text
	property		*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	halo		*==* and *!=*	Cyclic field. Choices are *True* and *False*.
	color	*color*	*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.
cluster boundary	*boundary name*		*==* and *!=*	Text
	cluster name		*==* and *!=*	Text
	property		*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
donut	*any radius*		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	inner radius	*inner radius*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	outer radius	*outer radius*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.
	area		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	color		*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.
	color state		*==* and *!=*	Cyclic field. Choices are *unlock* and *lock*.
ellipse	*any radius*		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	radiusX	*radiusX*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	radiusY	*radiusY*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.
	area		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	color		*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.
	color state		*==* and *!=*	Cyclic field. Choices are *unlock* and *lock*.
group	*group name*		*==* and *!=*	Text
	property		*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
inst	*cell name*	*cell name*	*==* and *!=*	Text
	lib name	*lib name*	*==* and *!=*	Text
	inst name	*inst name*	*==* and *!=*	Text
	view name	*view name*	*==* and *!=*	Text
	orient	*orient*	*==* and *!=*	Cyclic field. Choices are *R0*, *R90*, *R180*, *R270*, *MY*, *MYR90*, *MX*, and *MXR90*.

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	cell type	*cell type*	*==* and *!=*	Cyclic field. Choices are *none*, *block*, *blockRing*, *cover*, *coverBump*, *pad*, *padSpacer*, *core*, *coreSpacer*, *coreAntenna*, *corner*, *softMacro*, *via*, *blockBlackBox*, and *padAreaIO*.
	cluster name	*cluster name*	*==* and *!=*	Text
	physical only		*==* and *!=*	Cyclic field. Choices are *TRUE* and *FALSE*.
	placement status	*placement status*	*==* and *!=*	Cyclic field. Choices are *unknown*, *unplaced*, *placed*, *locked*, and *firm*.
inst or array	*cell name*	*cell name*	*==* and *!=*	Text
	lib name	*lib name*	*==* and *!=*	Text
	view name	*view name*	*==* and *!=*	Text
label	*text*	*text*	*==* and *!=*	Text
	font	*font*	*==* and *!=*	Cyclic field. Choices are *euroStyle*, *gothic*, *math*, *roman*, *script*, *stick*, *fixed*, *swedish*, and *milSpec*.
	VDR	*none*	*none*	Choices are *Created by VDR*, *Created by user*, and *Both.*

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	justify	*justify*	*==* and *!=*	Cyclic field. Choices are *lowerLeft*, *centerLeft*, *upperLeft*, *lowerCenter*, *centerCenter*, *upperCenter*, *lowerRight*, *centerRight*, and *upperRight*.
	height	*height*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	orient	*orient*	*==* and *!=*	Cyclic field. Choices are *R0*, *R90*, *R180*, *R270*, *MY*, *MYR90*, *MX*, and *MXR90*.
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	ROD name		*==* and *!=*	Text
	area		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
label or text display	*text*		*==* and *!=*	Text
	font	*font*	*==* and *!=*	Cyclic field. Choices are *euroStyle*, *gothic*, *math*, *roman*, *script*, *stick*, *fixed*, *swedish*, and *milSpec*.

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	justify	*justify*	*==* and *!=*	Cyclic field. Choices are *lowerLeft*, *centerLeft*, *upperLeft*, *lowerCenter*, *centerCenter*, *upperCenter*, *lowerRight*, *centerRight*, and *upperRight*.
	height	*height*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	orient	*orient*	*==* and *!=*	Cyclic field. Choices are *R0*, *R90*, *R180*, *R270*, *MY*, *MYR90*, *MX*, and *MXR90*.
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	ROD name		*==* and *!=*	Text
path	*path width*	*path width*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	path style	*path style*	*==* and *!=*	Cyclic field. Choices are *truncate*, *extend*, *round*, *variable*. (Virtuoso RF Option) *curve*

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	begin ext	*begin ext*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	end ext	*end ext*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	ROD name		*==* and *!=*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.
	area		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	length		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	color		*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.
	color state		*==* and *!=*	Cyclic field. Choices are *unlock* and *lock*.
pathSeg	*width*	*width*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	begin style	*begin style*	*==* and *!=*	Cyclic field. Choices are *truncate*, *extend*, *variable*, *chamfer*, and *custom*.
	end style	*end style*	*==* and *!=*	Cyclic field. Choices are *truncate*, *extend*, *variable*, *chamfer*, and *custom*.
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.
	area		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	length		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	color		*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.
	color state		*==* and *!=*	Cyclic field. Choices are *unlock* and *lock.*

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
path or pathSeg	*layer*	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.
	width	*width*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	area		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	length		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	color		*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.
	color state		*==* and *!=*	Cyclic field. Choices are *unlock* and *lock*.
pin	*term name*	*term name*	*==* and *!=*	Text
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	pin type		*==* and *!=*	Cyclic field. Choices are *input*, *output*, *inputOutput*, *switch*, *jumper*, and *unused*.
	connected to net		*==* and *!=*	Cyclic field. Choices are *signal*, *ground*, *supply*, *clock*, *testLatch*, *analog*, *tieOff*, *tieHi*, *tieLo*, *scan*, *reset*, *optical*, *singleModeOptical*, and *multiModeOptical*.
	color		*==* and *!=*	Cyclic field. Choices are *grayColor*, *mask1Color*, *mask2Color*, and *mask3Color*.
	color state		*==* and *!=*	Cyclic field. Choices are *unlock* and *lock*.
polygon	*layer*	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	ROD name		*==* and *!=*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	bbox height		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	bbox width		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	area		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	color		*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.
	color state		*==* and *!=*	Cyclic field. Choices are *unlock* and *lock*.
	(Virtuoso RF Option) *curved*	*none*	*==* and *!=*	*TRUE*
rectangle	*layer*	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	width	*width*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	height	*height*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	ROD name		*==* and *!=*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	area		*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	color		*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.
	color state		*==* and *!=*	Cyclic field. Choices are *unlock* and *lock*.
row	*row name*		*==* and *!=*	Text
	site def name		*==* and *!=*	Text
	site def type		*==* and *!=*	Cyclic field. Choices are *pad* and *core*.
	property		*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
snap boundary	No *Add Criteria*	*none*		
taper	*taper style*	*taper style*	*==* and *!=*	Cyclic field. Choices are *linear* and *exponential*.
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
text display	*text*		*==* and *!=*	Text
	font	*font*	*==* and *!=*	Cyclic field. Choices are *euroStyle*, *gothic*, *math*, *roman*, *script*, *stick*, *fixed*, *swedish*, and *milSpec*.
	justify	*justify*	*==* and *!=*	Cyclic field. Choices are *lowerLeft*, *centerLeft*, *upperLeft*, *lowerCenter*, *centerCenter*, *upperCenter*, *lowerRight*, *centerRight*, and *upperRight*.
	height	*height*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	orient	*orient*	*==* and *!=*	Cyclic field. Choices are *R0*, *R90*, *R180*, *R270*, *MY*, *MYR90*, *MX*, and *MXR90*.
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property		*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	ROD name		*==* and *!=*	Text
trl	*trl width*	*trl width*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	bend style	*bend style*	*==* and *!=*	Cyclic field. Choices are *bend*, *chamfer*, and *radial*.
	chamfer fac	*chamfer fac*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	radius fac	*radius fac*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	layer	*layer*	*==* and *!=*	Cyclic field. Choices are *Any Valid Layer*, *Any LSW Layer*, *Any Techlib Layer*, and a list of layer-purpose pairs.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.
via	*color*		*==* and *!=*	Cyclic field. Choices populated from the technology file. For example, *grayColor*, *mask1Color*, and *mask2Color*.
	color state		*==* and *!=*	Cyclic field. Choices are *unlock* and *lock*.

© 2023 Cadence Design Systems, Inc.

Objects	Properties (Find)	Properties (Replace)	Condition/Logic	Value Type/Values
	via definition	*via definition*	*==* and *!=*	Cyclic field. Choices are populated from the technology file. If you select *Others* from the list, the text field next to it gets enabled. You can then enter a via name and combine it with a wildcard character to get multiple matches.
	property	*property*	*<*, *<=*, *>*, *>=*, *==*, *!=*, *EXIST*, and *!EXIST*	Text
	net name		*==* and *!=*	Text
	no net name			No input required.
	width	*width*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	length	*length*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	rows	*rows*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	columns	*columns*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	row spacing	*rowspacing*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	column spacing	*column spacing*	*<*, *<=*, *>*, *>=*, *==*, and *!=*	Number
	cutClass	*cutClass*	*==* and *!=*	Cyclic field. Choices are *Others*, *Default*, *Rect*, *W30*, and *W40*.

© 2023 Cadence Design Systems, Inc.

Related Topics

Find Functionality

Replace Functionality

© 2023 Cadence Design Systems, Inc.

3

Design Tools

Layout XL provides various design tools to enable you to work on your designs efficiently. The following are some of the tasks that you can perform using the design tools:

- Area and Density: Calculate the area and density of shapes on specified layers
- Rulers: Use rulers to measure objects and the distance between objects
- Instance Remastering: Remaster cellviews based on the view names, cell names, and the library
- Object Tapping: Tap attributes of existing objects to create new objects
- Layer Generation: Generate a layer to change objects on the layer without modifying existing objects
- Markers: Find errors and warnings in your design by using markers
- Layout Integrity: Perform integrity checks on your layout design
- Pad Shapes: Generate reports and labels for pad shapes in your design
- Status Toolbar Customization: Customize the information displayed on the status toolbar
- Capture and Replay Your Design Actions: Record a series of actions in your design and later replay those actions as a macro.
- Cross Section Viewer: View cross sections of physical layers in a layout.

Calculating Area and Density

You can calculate the area and density of shapes on specified layers within a region. You can calculate the area of metal lines, resistor networks, and MOS devices placed together. You can also calculate the layer density in a specified region.

Area is defined as the total area of the specified layers within the region that you select. Density is defined as the total area divided by the area of the region that you specify. You can

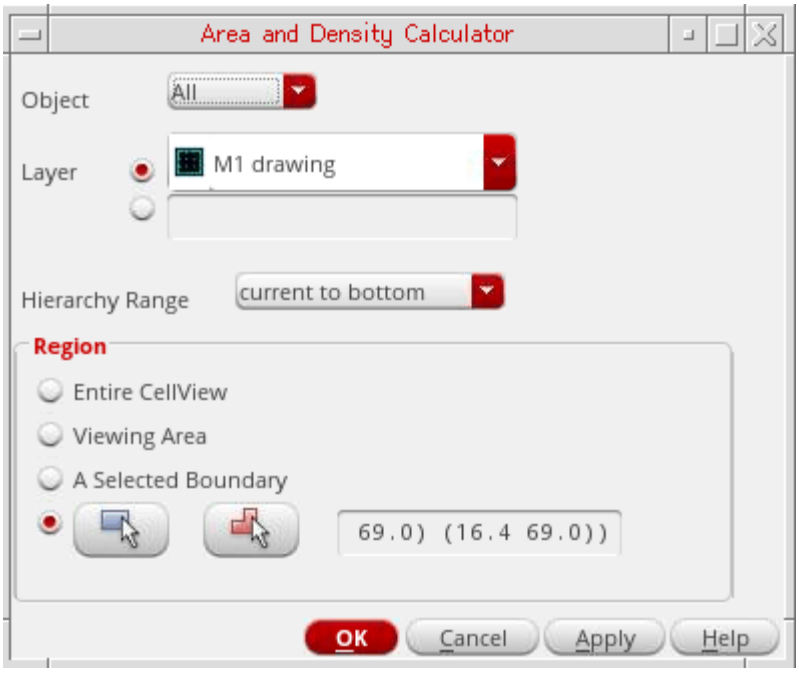
© 2023 Cadence Design Systems, Inc.

calculate the area and density by using the *Area and Density Calculator* command. You can also use the SKILL function `leComputeAreaDensity`.

To calculate the area and density of a region:

1. In the layout window, choose *Tools – Area and Density Calculator*.

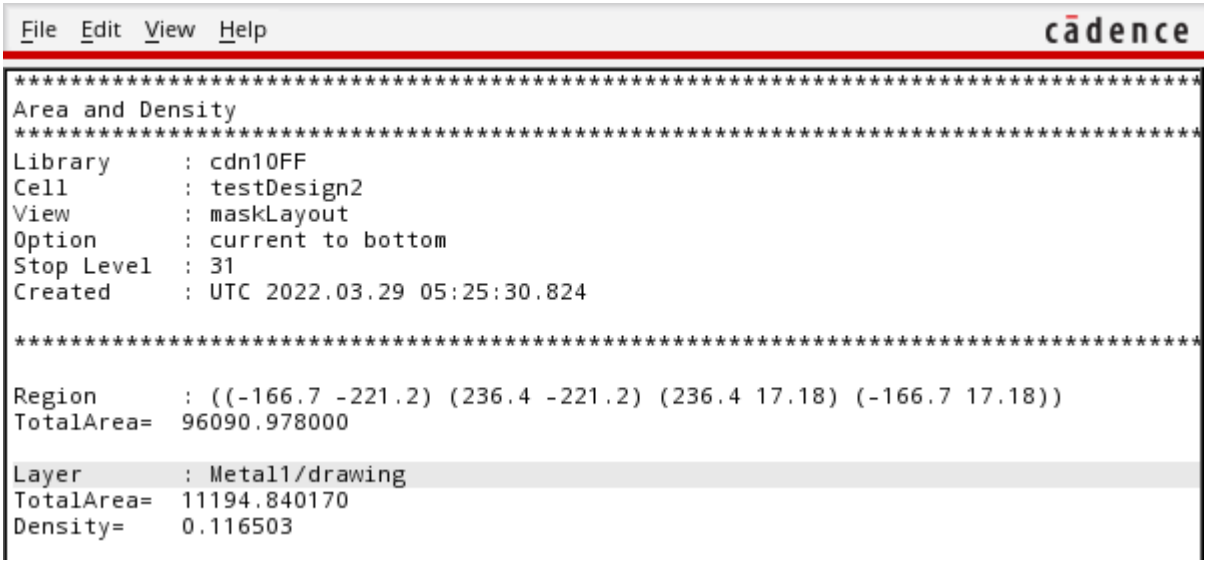
   The Area and Density Calculator form is displayed. You can specify the object, layer, hierarchy range, and region for which you want to calculate the area and density.



2. From *Object,* select the object range for which you want to calculate the area. You can consider all objects or only selected objects for area calculation.
3. From *Layer*, select a single layer or specify multiple layers for area calculation.
4. From *Hierarchy Range*, select the search depth. You can select the search depth as *current cellView*, *current to bottom*, *current to stop level*, or *current to user level*.
5. In the *Region* section, select the region for which you want to calculate the area and density. You can select *Entire CellView*, *Viewing Area*, *A Selected Boundary*, *Draw a Rectangle Region on the Canvas*, or *Draw a Polygon Region on the Canvas*.
6. Click *OK*.

© 2023 Cadence Design Systems, Inc.

The area and density calculation report is displayed.



Video

For an overview of how to use the area and density calculator, see the video *Using the Area and Density Calculator*.

Related Topics

Area and Density Calculator Form

Design Tools

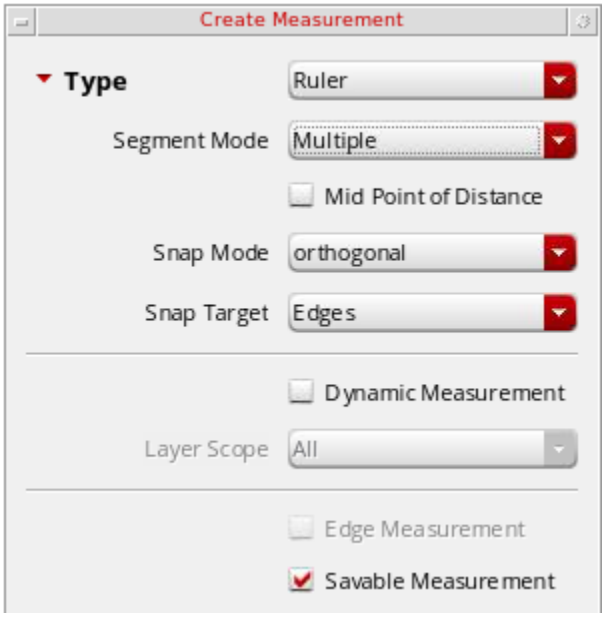
Rulers

You can use rulers to measure objects or the distance between objects in a cellview. You can also measure the distance between two edges or vertices.

Use the *Create Measurement* command to create a ruler object. You can specify ruler settings in the Create Measurement form. You can create persistent or transient rulers. A

© 2023 Cadence Design Systems, Inc.

persistent ruler can be saved, whereas a transient ruler cannot be saved. By default, persistent rulers are created.



Saving and Deleting Rulers

By default, rulers are saved when you save a cellview. If you do not want to save the rulers, ensure that the *Savable Measurement* option on the *Create Measurement* form is not selected.

You can clear all rulers from the design window by selecting *Tools – Clear All Measurements,* or from all levels of the hierarchy by selecting *Tools – Clear All Measurements In Hierarchy.* This deletes all persistent and transient rulers.

Related Topics

Create Measurement Form

Ruler Segment Modes

Mouse and Key Actions in Ruler Segment Modes

Ruler Snapping Support

Creating Rulers with Snapping On

© 2023 Cadence Design Systems, Inc.

Creating Rulers with Smart Snapping Off

Ruler Segment Modes

You can create a ruler using the following segment modes: single, multiple, or auto.

Single Segment Mode

By default, the *Create Measurement* command creates single-segment rulers. In this mode, when you add two points to the canvas, the corresponding measurement is displayed on the canvas. If *Repeat Commands* is on, the *Create Measurement* command continues, else the command completes.

By default, the placement of the measurement is calculated automatically. You can view the placement when you drag the pointer. You have the flexibility to specify the location of the measurement segment either before or after adding the second point.

- If you want to specify the location of the measurement segment before the second point is added to the canvas, move the pointer to the desired location, and then press `Ctrl` and click the middle mouse button. The location of the measurement segment is now fixed. You can then select the second point.
- If you want to specify the location of the measurement segment after the second point is added to the canvas, press `Ctrl` and click on the canvas. When you move the pointer, the measurement segment also moves. You need to click on the canvas to fix the location of the measurement segment and end the command.

You can move the measurement segment only if one of the points is snapped to a vertex or an edge.

Single measurement mode supports edge measurement, which allows you to click an edge of an object and a measurement is drawn with the two vertices as the two ends. The vertex of the edge closest to the point where you click is considered as the first vertex, and the other edge is considered as the second vertex. When you move the pointer on the canvas, the measurement also moves. You can then click on the canvas to place the measurement at the desired location.

Edge measurement allows you to measure the width of an edge of an object. The first click lets you select the edge and the second click lets you place the measurement on the canvas. A double-click lets you create and place the edge measurement at the edge itself. When the pointer snaps to a vertex, the edges connected to the vertex get measured.

© 2023 Cadence Design Systems, Inc.

Edge measurement is available only in single segment mode and works only when smart snapping is on.

Multiple Segment Mode

In multi-segment mode, a segment is added to the ruler with every click on the canvas. If you double-click or press `Enter`, the last point is added to the ruler and the *Create Measurement* command completes.

You can create a multi-segment measurement while you are in single-segment measurement mode. In single-segment mode, after the first click, press `Shift` and then click to add a point to the measurement. Press `Shift` and click to add subsequent points to the measurement. If you click on the canvas or press `Enter`, the last point is added to the ruler and the *Create Measurement* command completes.

Similar to single segment mode, in multi-segment mode, you can place the measurement segment by using the `Ctrl` and click combination.

Auto Segment Mode

Auto segment mode creates a multi-segment ruler where points are automatically calculated. A ruler is created in auto mode in the following ways:

- Auto ruler is created between two specified points on the canvas.
- Auto ruler is created when you press the `Shift` key in auto mode. In this case, an orthogonal ruler is created with all edges visible on the canvas in vertical or horizontal direction. The direction of the ruler is based on the edge closest to the pointer.

The *Mid Point of Distance*, *Dynamic Measurement*, and *Edge Measurement* options are not available for auto segment mode in the Create Measurement form.

Related Topics

Rulers

Create Measurement Form

Mouse and Key Actions in Ruler Segment Modes

Ruler Snapping Support

Creating Rulers with Snapping On

© 2023 Cadence Design Systems, Inc.

Creating Rulers with Smart Snapping Off

Mouse and Key Actions in Ruler Segment Modes

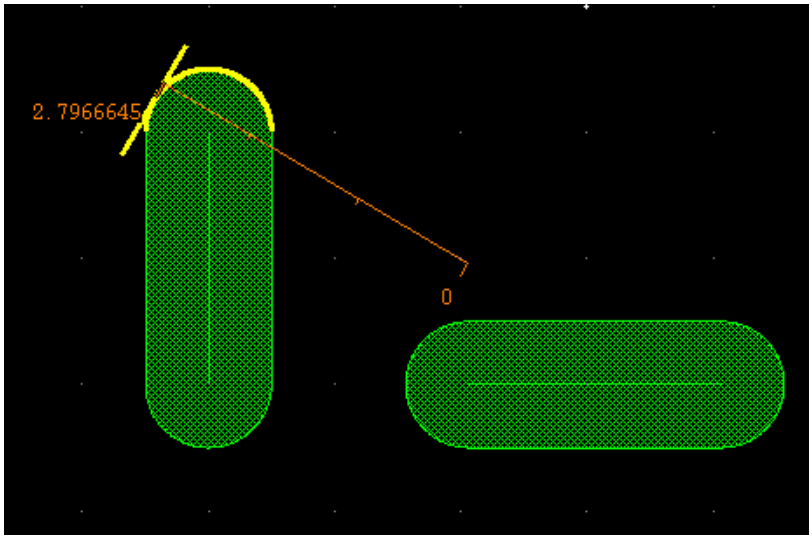
The following table summarizes the effect of mouse actions and key bindings during the *Create Measurement* command for single, multiple, and auto measurement modes. In the table, LMB refers to the left mouse button click and MMB refers to the middle mouse button click.

Mouse Action/ Key	Single	Multi	Auto
`LMB`	Adds a point or positions the measurement segment, where the second point completes the measurement.	Adds a point.	Adds a point or positions the measurement segment, where the second point completes the measurement.
`MMB`	Completes the ruler without drawing.	Completes the ruler without drawing.	Completes the ruler without drawing.
`Ctrl+LMB`	Enters second point but waits to complete the measurement by allowing you to place the segment by clicking the left mouse button. This works only when the end points are snapped.	NA	Enters second point but waits to complete the measurement by allowing you to place the segment by clicking the left mouse button. This works only when one of the end points is snappable.
`Shift+LMB`	Inserts a point so that the current measurement becomes multi-segment. Does not complete the measurement. You must enter one more point by using `LMB` or `Enter`.	NA	Inserts another point in the measurement. Does not complete the measurement. You must enter one more point by using `LMB` or `Enter`.

© 2023 Cadence Design Systems, Inc.

Mouse Action/ Key	Single	Multi	Auto
`Ctrl+MMB`	Moves the measurement segment to the clicked location, after the first point.	Moves the measurement segment to the clicked location, after the first point	NA
`Enter`	Completes the current measurement.	Completes the current measurement.	Completes the current measurement.
`Esc`	Cancels the *Create Measurement* command.	Cancels the *Create Measurement* command.	Cancels the *Create Measurement* command.

(Virtuoso RF Option) For curved shapes. the measurement is drawn at a tangent to the point where you have snapped, as shown in the following figure. Edge measurement is disabled for measurements on curved shapes.



Related Topics

Rulers

Ruler Segment Modes

© 2023 Cadence Design Systems, Inc.

Ruler Snapping Support

Creating Rulers with Snapping On

Creating Rulers with Smart Snapping Off

Ruler Snapping Support

You can create a ruler with the smart snapping feature on or off. You can also create a ruler by using a combination of both – define the start or the end point with smart snapping on and define the other point with smart snapping off.

The ruler snapping support differs for different objects.

- Shapes: Snapping of ruler to all shapes is supported.
- Blockages: Snapping of rulers to blockages is supported. All snapping modes, except snapping to nexus, are supported for blockages.
- Labels: Snapping is supported only to the origin of the labels.
- Rows and boundaries: Snapping of ruler to rows and boundaries is not supported.

To quickly define the start and end points of a ruler or the various click points in the case of a multi-segment ruler without the need for zooming in to the exact click locations, you can select the snap target as edges, points, or both. This identifies objects in vicinity of the mouse pointer and attaches to them.

When snapping is enabled for edges, points, or both, the ruler object is created with the following features:

- Snap to vertex, nexus, edge midpoint, edge, or centerline.
- Cycle over potential targets within a specified distance from the pointer.
- Cycle over orthogonal end targets prior to parallel or non-orthogonal ones.
- Snap to non-orthogonal edges as well as edges and vertices of instance boundary.
- Snap to targets within a hierarchy, in Edit in Place (EIP) mode, and inside a group. Rulers at level 0 remain visible when you EIP to any other hierarchical level.

You can set snap mode to specify how a ruler snaps to the grid: diagonal, orthogonal, horizontal, vertical, or at any angle. When you create rulers with snapping on, if the snap target is points or points and edges, the snap mode becomes *anyAngle* for ruler and distance.

© 2023 Cadence Design Systems, Inc.

In EIP mode, the ruler can be snapped to top-level shapes.

By default, window snap modes are used when you create a ruler. You can use snap modes that are specific to the ruler command when you invoke the command. To use ruler snap modes, you need to enable the useRulerSnapModes environment variable and set the snap modes using the rulerSnapMode and rulerSegSnapMode environment variables. Window snap modes are restored when you exit the ruler command.

Related Topics

Rulers

Creating Rulers with Snapping On

Creating Rulers with Smart Snapping Off

Ruler Segment Modes

Creating Rulers with Snapping On

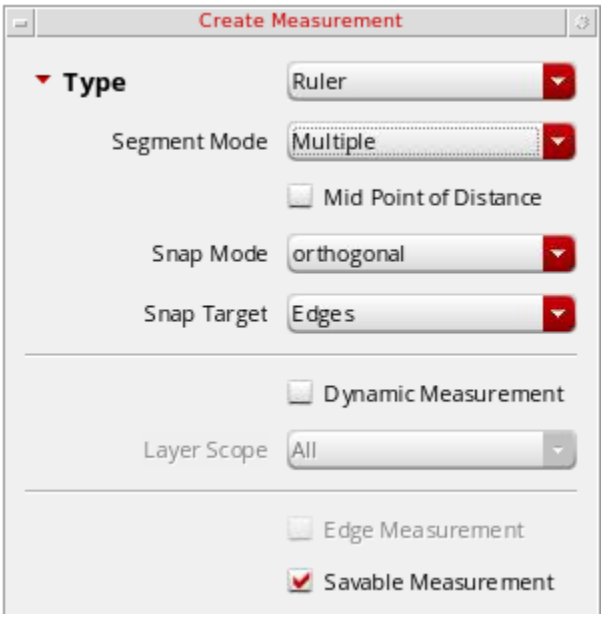
To create a ruler when snapping is on:

1. If required, in the Display Options form in the layout window, adjust the display level for displaying targets within the hierarchy.
2. Choose *Tools – Create Measurement*.
3. Press `F3` or select *Options* from the ruler shortcut menu.

   The Create Measurement form appears.
4. Specify the ruler settings. For example:
   a. From *Segment Mode*, select the ruler mode as single, multiple, or auto.
   b. From *Snap Mode*, select how the ruler snaps to the grid: *anyAngle*, *diagonal*, *orthogonal*, *horizontal*, or *vertical*.
   c. From *Snap Target*, select whether the ruler snaps to edges, points, or both.

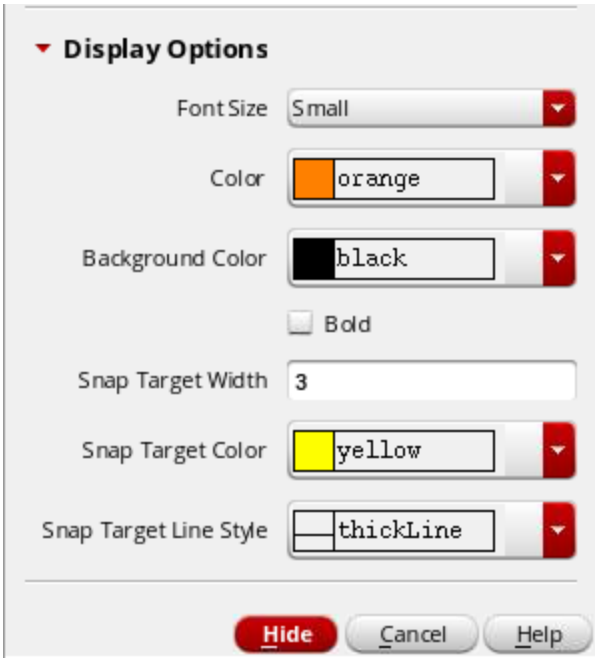
© 2023 Cadence Design Systems, Inc.

d. Select the *Savable Measurement* check box if you want to create a persistent ruler. By default, rulers are saved when you save the cellview. Clear the selection if you do not want to save rulers.



e. In the *Display Options* section, specify the ruler and snap target display settings.

f. Click *Hide*.



© 2023 Cadence Design Systems, Inc.

5. Move the mouse pointer on the canvas.

   As you move the pointer, a vertex, nexus, edge midpoint, edge, or centerline highlights dynamically based on the snapping option you have selected.

6. Click near one of the highlighted targets.

   The pointer snaps to the highlighted target.

7. Press the spacebar if there are multiple potential targets within the distance from the pointer.

   The highlight cycles over the targets. With each spacebar key press, the next possible target within the distance from the pointer is highlighted. You can highlight up to five targets in the proximity of the pointer by pressing the spacebar in succession.

8. Click to snap the pointer to the highlighted target.

   - If the target is an edge or a center line, the pointer drops orthogonally from the point of click to the target edge or center line, respectively.
   - In case of points, the pointer snaps to the target vertex, nexus, or edge midpoint. If the selected snapping option is *Mid Point of Distance*, click twice for the pointer to snap to the starting point of the smart ruler.
   - If you select *Mid Point of Distance* on the ruler shortcut menu, by default, the *Snap Target* to *Edges* option appears selected on the shortcut menu.

   Then, click to snap the pointer to that edge. After you define the two targets, the pointer immediately snaps to the midpoint of the distance between the two targets on the selected edges.

9. Press `Esc` to finish creating rulers.

You can clear all rulers from the design window by choosing *Tools – Clear All Measurements.* This deletes all persistent and transient rulers.

Related Topics

Create Measurement Form

Display Options Form

Ruler Snapping Support

Creating Rulers with Smart Snapping Off

© 2023 Cadence Design Systems, Inc.

Rulers

Ruler Segment Modes

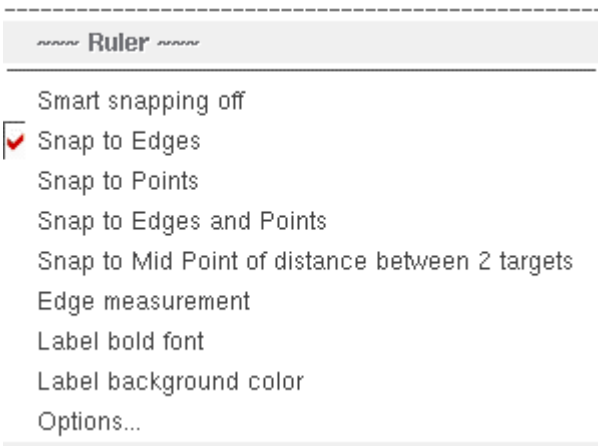
Creating Rulers with Smart Snapping Off

In *Smart snapping off* mode, the snap targets are not visually highlighted. You cannot cycle through the highlight over potential targets within the distance from the pointer. The ruler object follows the gravity controls settings specified in the Layout Editor Options form. If *Gravity On* is selected, ruler snapping to the objects in the vicinity of the mouse pointer is enabled. If *Gravity On* is not selected, you need to create the ruler by specifying the exact start and end points.

In *Smart snapping off* mode, with *Gravity On*, the pointer snaps to the specified target type within the aperture distance.

To create a ruler in *Smart snapping off* mode:

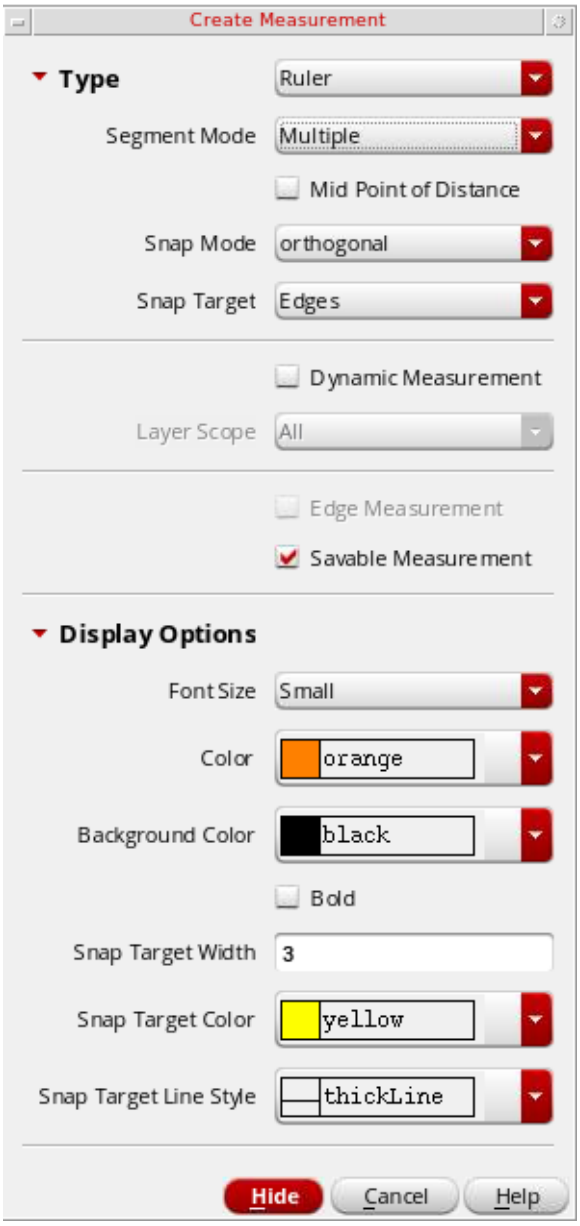
1. In the layout window, choose *Tools – Create Measurement*.
2. Right-click on the canvas to display the ruler shortcut menu and choose *Smart snapping off*.



3. Press `F3` or select *Options* from the ruler shortcut menu.

© 2023 Cadence Design Systems, Inc.

The Create Measurement form opens.



4. Select the ruler settings you want to use, such as the ruler type and segment mode. Then, click *Hide*.

   For information about the ruler settings, see Create Measurement Form.

5. Click on the canvas to specify the starting point of the ruler.

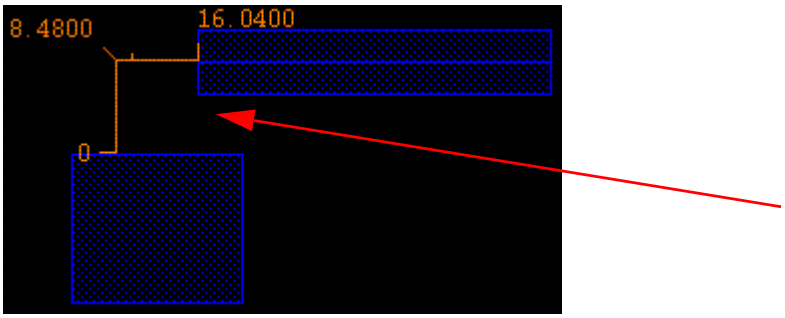
   After specifying the starting point, you can move it around on the canvas depending on the snap mode selected.

© 2023 Cadence Design Systems, Inc.

6. Click again on the canvas to specify the end point of the ruler.

   If *Gravity On* is selected in the Layout Editor Options form, you can snap the ruler to the required target edge or point.

   If you are creating a multi-segment ruler, click at the end of each segment and double-click to define the end point of the ruler.



A multi-segment ruler shows the total distance, in user units, between each point of the ruler.

7. Press `Esc` to finish creating rulers.

Related Topics

Create Measurement Form

Ruler Shortcut Menu

Layout Editor Options Form

Gravity Controls

Rulers

Creating Rulers with Snapping On

Instance Remastering

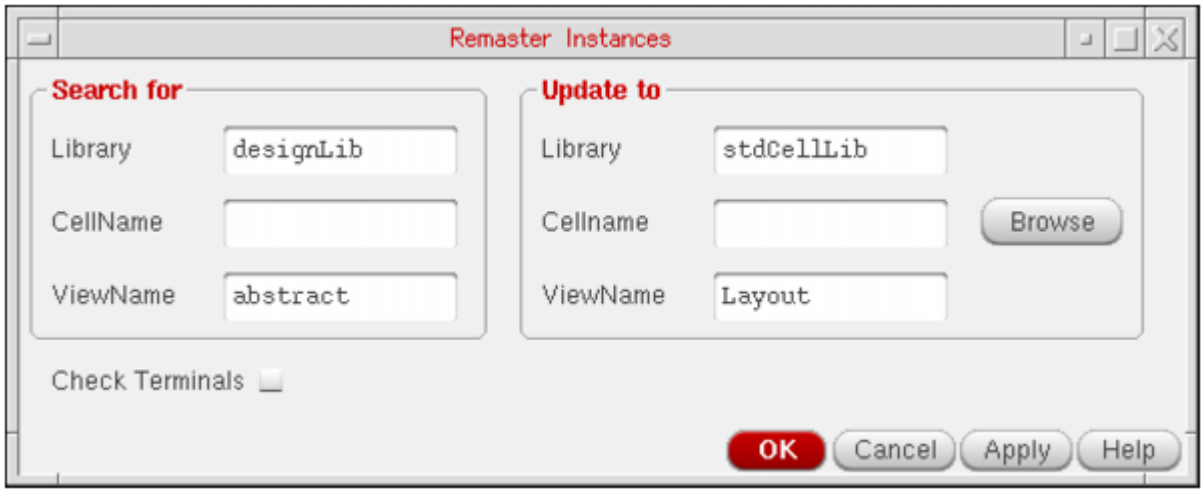
You can hierarchically remaster cellviews based on view names, cell names, and the library in which they reside by using the *Remaster Instances* command. For example, remastering the abstract views with layout views directly before translating the final design to Stream format provides optimal performance. Replacing cellviews before editing the design can add unnecessary complexity to the design for routing and editing purposes.

There are various options for remastering cellviews. By leaving text fields blank in the Remaster Instances form, entire libraries and cellviews can be searched for matching

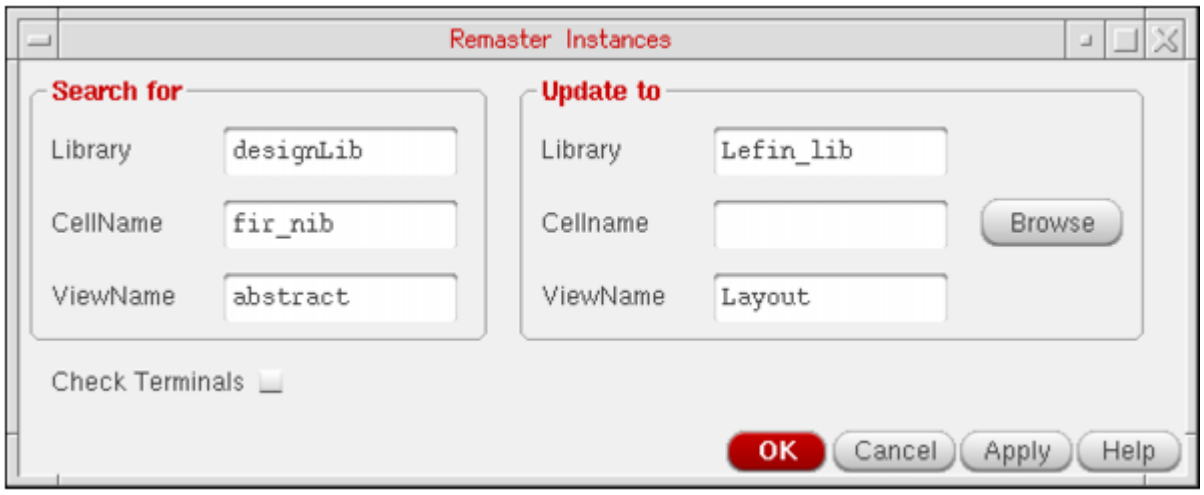
© 2023 Cadence Design Systems, Inc.

cellviews and view names. You can use the asterisk (*) wildcard to replace any number of characters prior to or following the asterisk.

The following are some of the common types of searches and remastering of cellviews you can perform using the Remaster Instances form. The following figure shows that all instances in the current cellview that have masters from the library `designLib` and the view `abstract` are remastered to the library `stdCellLib` and the view `Layout`, and their master cell names are not changed.

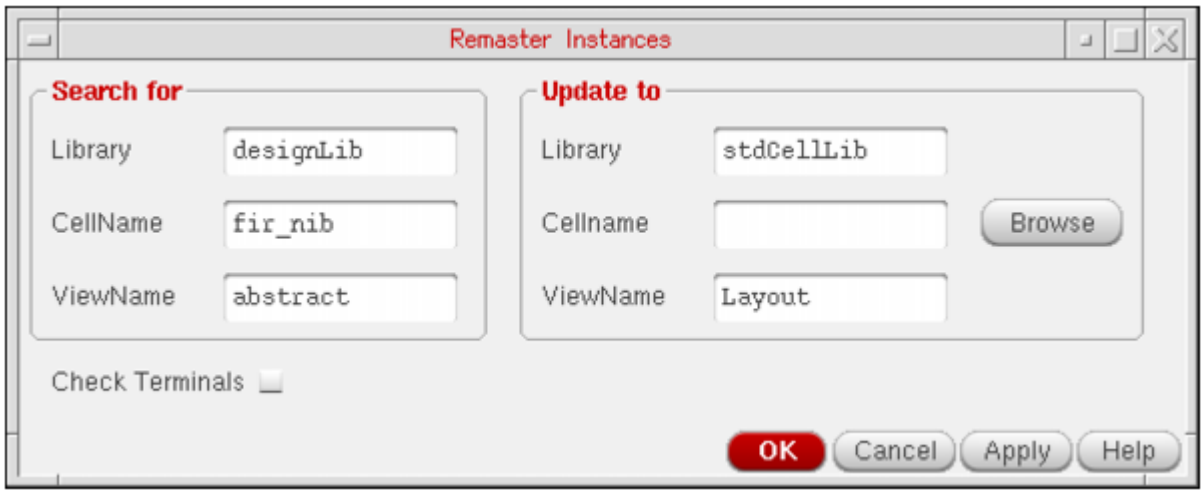


The following figure shows that all instances of `designLib/fir_nib/abstract` in the current cellview are remastered to `Lefin_lib/fir_nib/Layout`.

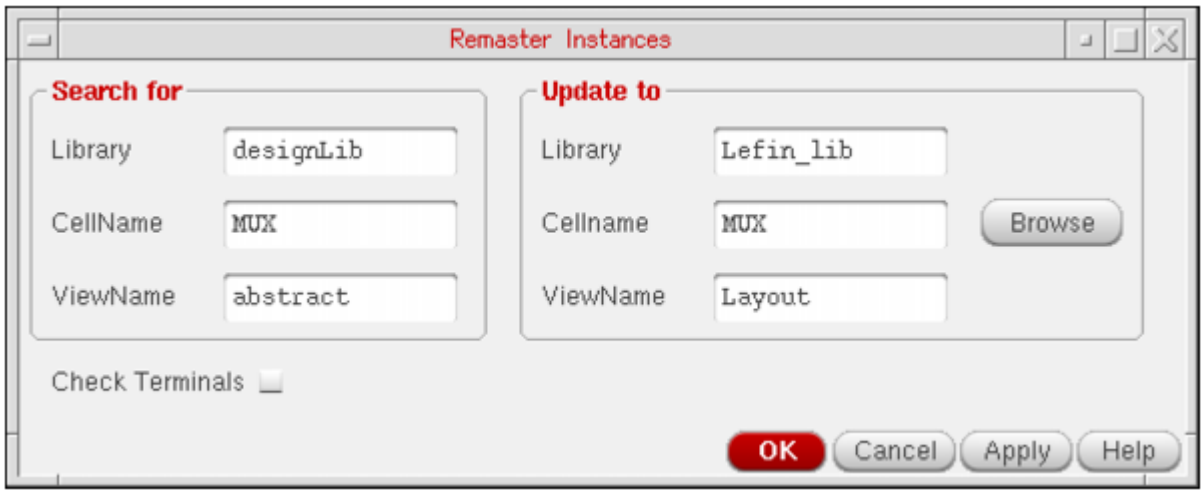


© 2023 Cadence Design Systems, Inc.

The following figure shows that all instances of `designLib/fir_nib/abstract` in the current cellview are remastered to `stdCellLib/fir_nib/Layout`.



The following figure shows that all instances of `designLib/MUX/abstract` in the current cellview are remastered to `Lefin_lib/MUX/Layout`.



Related Topics

Remaster Instances Form

Terminal Checking During Instance Remastering

Replacing Instance Abstract Views with Layout Views

© 2023 Cadence Design Systems, Inc.

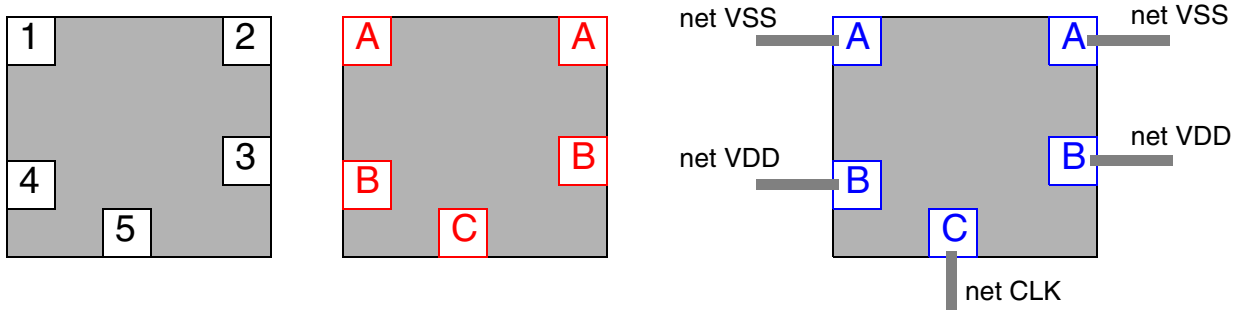
Terminal Checking During Instance Remastering

While remastering cellviews, you can choose to check instTerms of cellviews for matching or a subset of matching instTerms. When the *Check Terminals* check box is enabled in the Remaster Instances form, and the instTerms of the cellview in the *Search for* section are equivalent to or are a subset of the instTerms of the cellview in the *Update to* section, the cellviews are remastered.

Terminals expose a net up to the next level of the hierarchy. For example, a master might have 10 nets and 4 terminals, associated with 4 of the nets.

When you create an instance of a master, it may have objects called instTerms associated with it. These instTerms map terminals in the instance's master to the nets of the cellview in which the instance has been placed. The Layout XL editor creates and assigns the instTerms automatically. To expose the nets to the next level of the hierarchy, you create terminals for the nets, which are then mapped using instTerms at the next level.

For example, the following figure represents an instance that contains five pins: two on net A, two on net B, and one on net C. These nets need to be available at the next level of the hierarchy, so you create three terminals named A, B, and C.



Pins and Pin Numbers | Terminal and Net Names | instTerms and Nets

Related Topics

Remaster Instances Form

Replacing Instance Abstract Views with Layout Views

Instance Remastering

© 2023 Cadence Design Systems, Inc.

Replacing Instance Abstract Views with Layout Views

You can use the *File – Discard Edits* command to restore data to the last saved design in the event you make an error when replacing data. However, due to the extensive changes available when using this command, it is recommended that you make a backup copy of your design to guard against errors when replacing data.

To search and replace instance abstract views with layout views:

1. Open the cellview in which you want to remaster instances.

   Instances are remastered only in the currently open cellview.

2. Choose *Tools – Remaster Instances*.

   The Remaster Instances form appears.

3. Specify the following in the *Search for* section:

   a. In *Library*, type the name of the library that contains the abstract cellviews you want to remaster.

   b. In *CellName*, type the name of the block or cellview that contains the cellviews you want to remaster.

      If the *CellName* field is left blank, the *Remaster Instances* command searches throughout the entire library in the *Update to* section for matching cellview names.

   c. In *ViewName*, type the name of the view (abstract) to be remastered.

4. Specify the following in the *Update to* section:

   a. In *Library*, type the name of the library that contains the layout cellviews.

   b. In *CellName*, type the name of the block or cellview that contains the layout cellviews.

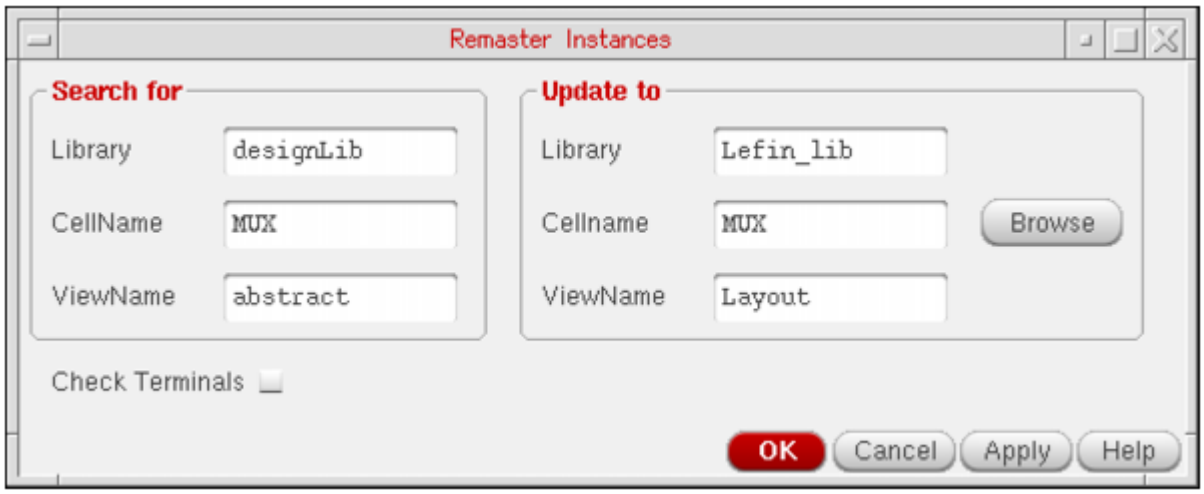
      If the *CellName* field is left blank, all cellviews that have matching cellview names specified by *Library* name and *ViewName* in the *Update to* section are remastered.

   c. In *ViewName*, type the name of the view (layout) that will remaster the current abstract cellviews specified by the *Search for* library.

   Alternatively, you can use the *Browse* button to open the Library Browser to specify the details. The Library Browser lets you look though the libraries specified in your *cds.lib* file and set the library, cell, and view names for the form.

© 2023 Cadence Design Systems, Inc.

5. Optionally, turn on *Check Terminals* to check all corresponding cellviews for matching or a subset of matching instTerm information.



6. Click *OK* or *Apply*.
7. Optionally, choose *File – Summary* to check the results of the *Remaster Instance* command.

   In the *Instance Statistics* section of the summary, the master cellview name, view name, and library are listed for each cellview. Verify the results of the command by checking the master cellview names and views names for the specified cellview.

   Due to the complexity of different types of errors, the CIW does not display a confirmation of the command or warning messages if the *Remaster Instance* command is not successful.

Related Topics

Remaster Instances Form

Instance Remastering

Terminal Checking During Instance Remastering

Object Tapping

You can tap objects, such as paths, instances, and vias, to capture their attributes.

© 2023 Cadence Design Systems, Inc.

- Blockages: Tapping blockages captures the blockage type and the maximum density of slot and fill blockages, and the effective width, if any. It captures directional offsets if the blockage has a halo. Tapping blockages does not change the active layer.
- Path, wire, and wire segments: Tapping these captures the beginning and ending extension in user units, the path end type, and the path width.
- Instances: Tapping instances captures the instance master's cell type, list of halos on the instance, as well as any clusters associated with the instance.
- Rows: Tapping rows captures the number of sites, whether the row is vertical or horizontal, and the orientation of the row.
- Trls, bends, and tapers attributes capture the following:
  - Bend factor, the maximum width before they are chamfered
  - Taper style, linear or exponential
  - Number of points to taper in user units
  - Bend style, bend chamfer or radial
  - Width of a chamfered bend in a transmission line as a multiple of its standard width in user units
  - Number of segments in a radial bend in a transmission line as a function of the number of segments in a 180-degree radial bend
  - Radius of a radial bend in a transmission line as a multiple of the transmission-line width in user units
  - Width of the first end of a transmission line bend in user units
  - Width of the second end of a transmission line bend
  - Width of the second end of a transmission line bend in user units

You can specify tap settings while using the *Tap* command on the *Tools – Tap* menu and in the *Tap* group box in the *Layout Editor Options* form.

Related Topics

Tapping Objects Using the Tap Command

Customizing Tap Settings Using the Layout Editor Options Form

Tap Form

© 2023 Cadence Design Systems, Inc.

Tapping Objects Using the Tap Command

The *Tap* command accesses information from the following objects: paths, wires, rectangles, polygons, circles, ellipses, donuts, microwave objects, blockages, instances, vias, rows, labels and text displays. In the Tap form, you can specify how the Tap command affects the current layer, net, attributes, and create commands when each type of object is tapped.

To tap objects by using the *Tap* command:

1. In the layout window, choose *Tools – Tap*.
2. Press `F3` to open the Tap form.
3. Select the required tap types. You can select *Layer*, *Net*, *Attributes*, and *Command*. You can also choose to display a selection window if there are overlapping objects in the clicked area. The selection window allows you to choose the object to be tapped.
4. Click an object on the canvas that you want to tap.

   The specified attributes are picked from the selected object.
5. Choose a create command. For example, you can choose to create a rectangle. If you selected *Command* in the Tap form, the command to create a rectangle is automatically started if you tapped a rectangle.

   The create form is populated with the information obtained from the original object that you tapped and seeds the form with the information you selected in the Tap form.

Related Topics

Tap Form

Object Tapping

Customizing Tap Settings Using the Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

Customizing Tap Settings Using the Layout Editor Options Form

You can customize the settings for tapping objects by using the *Tap* section of the Layout Editor Options form. For example, you can enable auto tapping and specify the purposes to be tapped.



You can auto tap wires and shapes. When tapping a wire or path, you can tap the layer-purpose pair, extension type, wire width, net name, and constraint group. *Wire Auto Tap*, *Shape Auto Tap*, and *Tap Purpose List* are automatic during the creation of certain objects when these *Tap* fields are selected in the Layout Editor Options form. They are used during certain create commands when you want to automatically tap an existing object that is located at the first point of your new object.

By default, the `drawing` layer purpose is tapped. To tap specific layer purposes, you can define a prioritized list in the *Tap Purpose List* field. For example, consider that the following valid layer-purpose pairs exist in the technology file: `metal1 drawing`, `metal1 pin`, `metal2 pin`, and `metal2 net`. If *Tap Purpose List* is set to `drawing net`:

- The default purpose for `metal1` is `drawing`, based on the ordered list of specified purposes, and for `metal2` is `net`.
- Tapping an object on the LPP `metal1 net` changes the active layer to `metal1 net` because `net` is specified in *Tap Purpose List*.
- Tapping an object on the LPP `metal1 pin` changes the active layer to `metal1 drawing`. This is because `pin` is not specified in *Tap Purpose List*, so the default purpose for `metal1` is used.
- Tapping an object on the LPP `metal2 pin` changes the active layer to `metal2 net`. This is because `pin` is not specified in *Tap Purpose List*, so the default purpose for `metal2` is used.

© 2023 Cadence Design Systems, Inc.

You can also define layer purposes by using the useTapPurposeList and tapPurposeList environment variables.

To customize the settings for tapping objects:

1. In the layout window, choose *Options – Editor*.

   The Layout Editor Options form opens. You can define the settings in the *Tap* section of the form.

2. Specify *Auto Tap* settings.

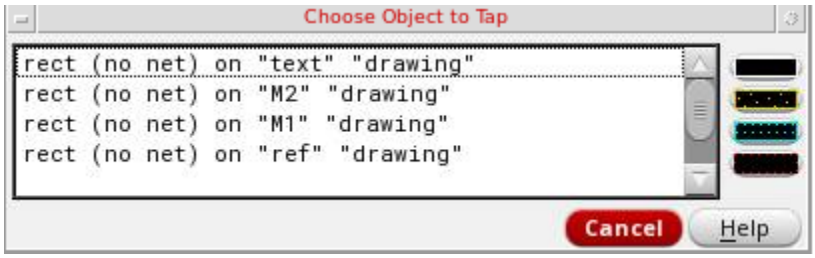
   - Select *Shape* to enable auto tap for rectangles, polygons, paths, circles, ellipses, and donuts.
   - Select *Wire* to enable auto tap for wires.

   If *Auto Tap* is selected, a *No Tap* button is available on the Choose Object to Tap form to select objects where the `Tap` command should not be performed.

3. In *Tap Purpose List*, specify layer purposes if you want to tap specific layer purposes. The default purpose is `drawing`.

4. Select the *Select from Overlaps* check box if you want the Choose Object to Tap form to appear if the clicked point has overlapping objects.

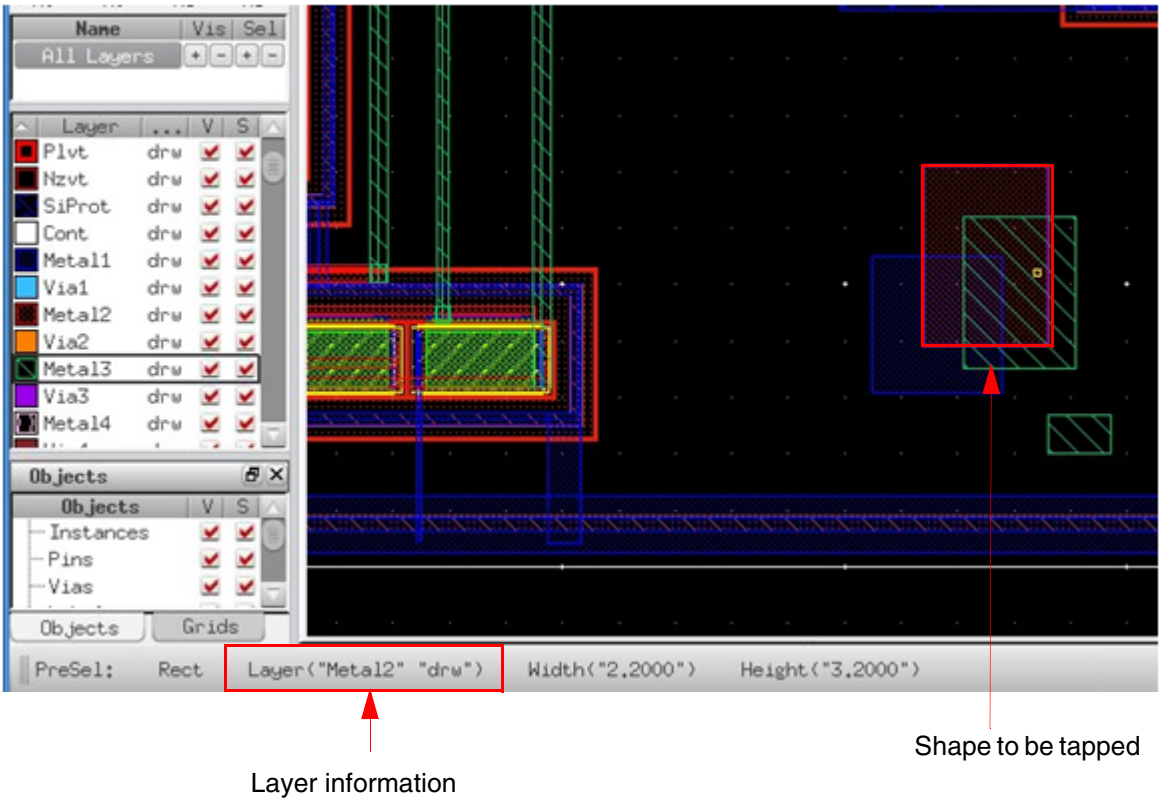
   The following image shows a sample Choose Object to Tap form.



   If *Select from Overlaps* is disabled, the object with an edge closest to the selected point is tapped. Also, the shape to be tapped is highlighted. You can use the spacebar to

© 2023 Cadence Design Systems, Inc.

toggle the highlight for shapes under the pointer. The layer information of the shape is displayed on the Selection toolbar.



5. Click *OK*.

Related Topics

Layout Editor Options Form

Tap Form

Tapping Objects Using the Tap Command

Object Tapping

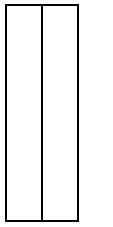
Layer Generation

By using the *Layer Generation* command, you can perform Boolean edits such as resizing and merging objects on a layer. *Layer Generation* edits the copies of objects; original objects remain unchanged.

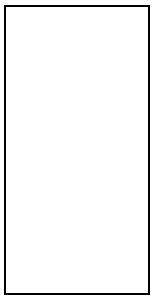
© 2023 Cadence Design Systems, Inc.

When you resize an object such as a path by using size, the object is converted to a polygon. When you use the Layer Generation form, the original path remains and the new object on the new layer is converted to a rectangle or a polygon.

For example, if you enlarge a path by size, the resulting object is a polygon.



A path before resizing



After resizing, the path converted to a polygon

You can also create an abstract of your design and use it for place and routing. You can generate an abstract by using the *Tools - Abstract Generation* command, which opens the Virtuoso Abstract Generator tool.

Related Topics

Overview of Abstract Generator

Making Changes to Objects on Layers

Making Boolean Edits by Layer

Layer Generation Form

Making Changes to Objects on Layers

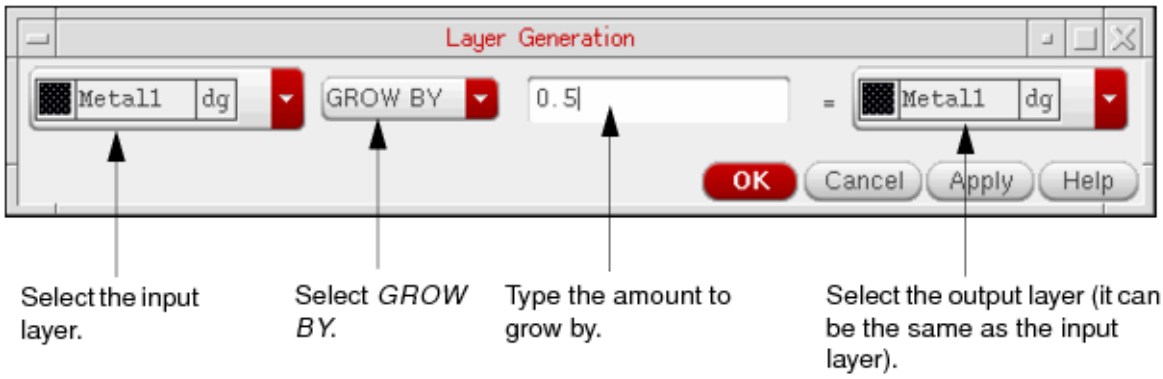
To enlarge copies of the objects selected:

1. On the layout canvas, select the objects that you want to resize.
2. Choose *Tools – Layer Generation*.

   The Layer Generation form opens.

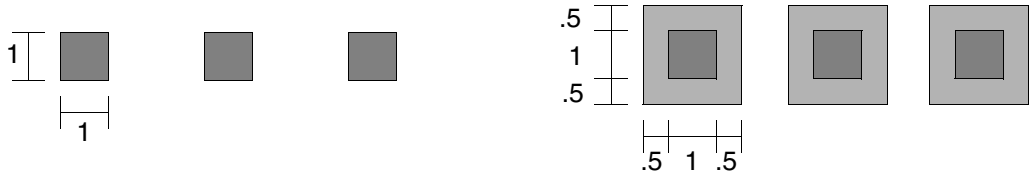
© 2023 Cadence Design Systems, Inc.

3. Fill out the Layer Generation form. Specify the input layer, Boolean operator, the resize value, and the output layer. The output layer can be same as the input layer.



4. Click *OK*.

   Copies of the selected objects are enlarged.



Original objects

After *GROW BY*, enlarged copies of originals appear on the *metal1* layer

Related Topics

Layer Generation Form

Making Boolean Edits by Layer

Layer Generation

Making Boolean Edits by Layer

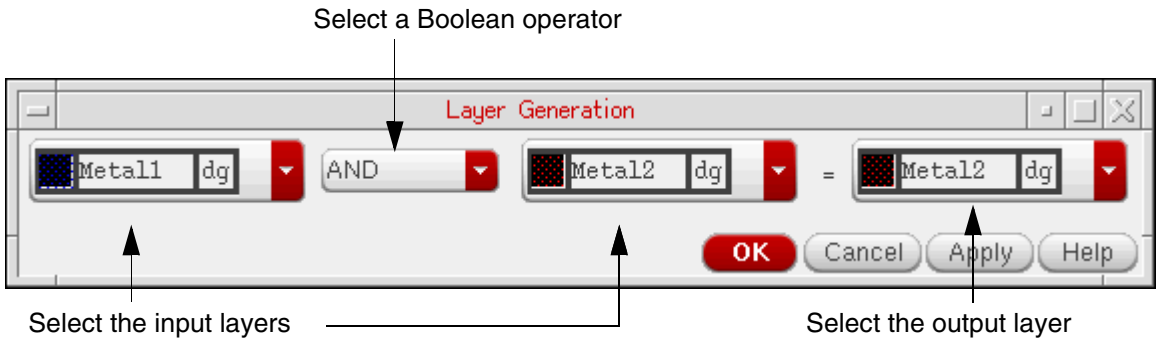
You can perform Boolean logical operations on objects on a single layer or two layers. You edit copies of the objects, so the originals are unchanged.

1. On the layout canvas, select the objects you want to edit.
2. Choose *Tools – Layer Generation*.

© 2023 Cadence Design Systems, Inc.

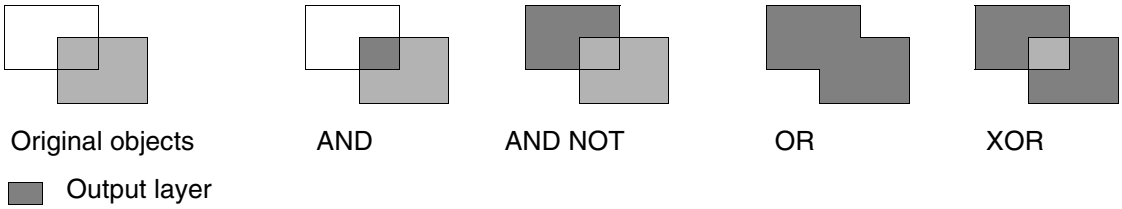
The Layer Generation form opens.

3. Select the layers that you want to edit, the Boolean operator you want to use, and the output layer.



4. Click *OK*.

The following figure shows the original objects and the results of different Boolean operations.



Related Topics

Layer Generation Form

Making Changes to Objects on Layers

Layer Generation

Markers

You must check your designs for any connectivity and rule violation errors and warnings. A flashing highlighted marker is displayed on each object that generated the error or warning. You can search for markers, review marker information, and fix and delete them as required.

To work with markers in your design, use the following commands on the *Verify – Markers* menu:

© 2023 Cadence Design Systems, Inc.

- *Explain*: Displays the reason for the error or warning marker in a text window.
- *Find*: Searches for and highlights each error or warning marker.
- *Delete*: Removes a specific marker.
- *Delete All*: Removes all markers.

Related Topics

Displaying Marker Information

Finding Markers in a Design

Deleting Markers

Displaying Marker Information

Use the *Verify – Markers – Explain* command to view the information about a marker. You can select a marker either before or after starting the command.

To view the reason or explanation associated with markers:

1. On the layout canvas, select a marker. To select multiple markers, draw a selection box around them.
2. Choose *Verify – Markers – Explain*.

   The marker text window appears. It displays the information about the error.
3. Continue clicking the markers for which you want information.

   The text scrolls in the marker text window so you can see it.
4. Press `Esc` to end the command when you are finished viewing the errors and warnings.

   You can correct the errors and then run the command again for verification.

Related Topics

Marker Text Window

Markers

Finding Markers in a Design

© 2023 Cadence Design Systems, Inc.

Deleting Markers

Finding Markers in a Design

Use the *Verify – Markers – Find* command to search for markers in a design. You can search for markers anywhere in a design hierarchy.

To find markers in a design:

1. In the layout window, choose *Verify – Markers – Find*.

   The Find Marker form appears.

2. Change the options on the form as required. You can narrow down your search or expand the scope.

3. Click *Apply*.

   The first error or warning marker is highlighted and the explanation for the marker is displayed in the marker text window.

4. To find the next marker, click *Next*.

   The marker text window displays the reason for the marker.

   Use the *Next* and *Previous* buttons to navigate through the markers.

Schematic editor provides a similar *Find Marker* command. The command options are a little different than the layout editor version and provide additional functionality. You can run the schematic editor *Find Marker* command in a layout window by using the geHiCommonFindMarker SKILL function.

Related Topics

Find Marker Form

Markers

Marker Text Window

Resolving Check Errors

Deleting Markers

You can delete specific markers individually or delete all markers at one go.

© 2023 Cadence Design Systems, Inc.

- Deleting Specific Markers
- Deleting All Markers

Deleting Specific Markers

To delete specific markers in a design window:

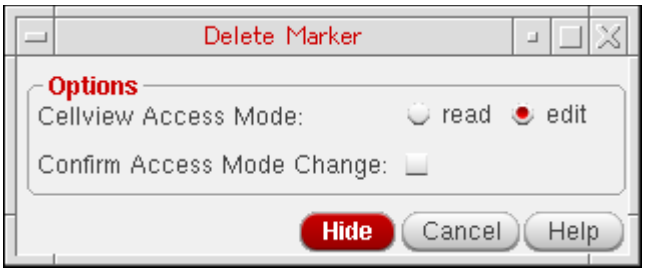
1. Select the marker that you want to delete. To select multiple markers, you can draw a selection box around them.

   You can also use the Find Marker form to find the marker you want to delete.

2. Choose *Verify – Markers – Delete*.

   The markers are deleted. You can continue deleting the markers by clicking on them.

3. Optionally, press `F3` to open the Delete Marker form.

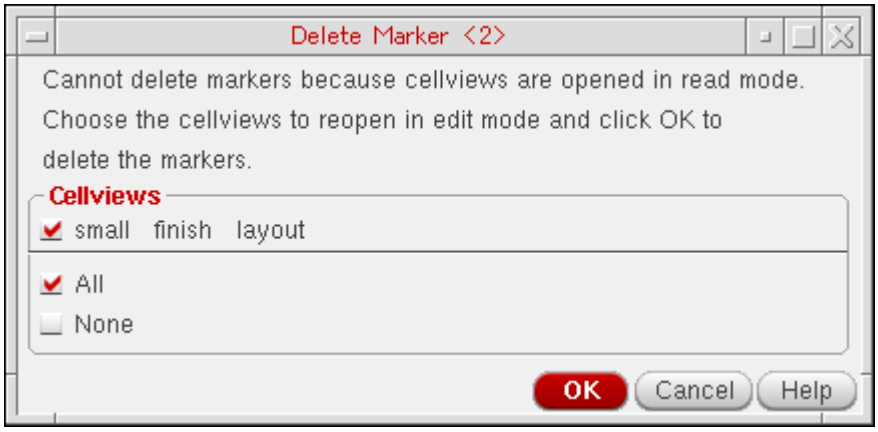


4. Set the cellview access mode as read or edit in *Cellview Access Mode* and delete the marker.

   - If you set cellview access mode to *edit,* select the *Confirm Access Mode Change* check box. Then, point at a marker on the canvas to delete it. The *Delete Marker* dialog box appears for you to confirm opening the selected cellviews in edit mode.

© 2023 Cadence Design Systems, Inc.

Select the cellviews that you want to open in edit mode and click *OK*. If you do not want to open the cellviews in edit mode, select *None* and click *OK*.



- If you set the *Cellview Access Mode* to *read*, then point at a marker on the canvas to delete it. The marker is deleted and the cellview remains in read mode.

5. Continue deleting the markers as required. When done, click *Cancel* in the *Delete Marker* form.

Deleting All Markers

To delete all markers in a design window:

1. Choose *Verify – Markers – Delete All*.

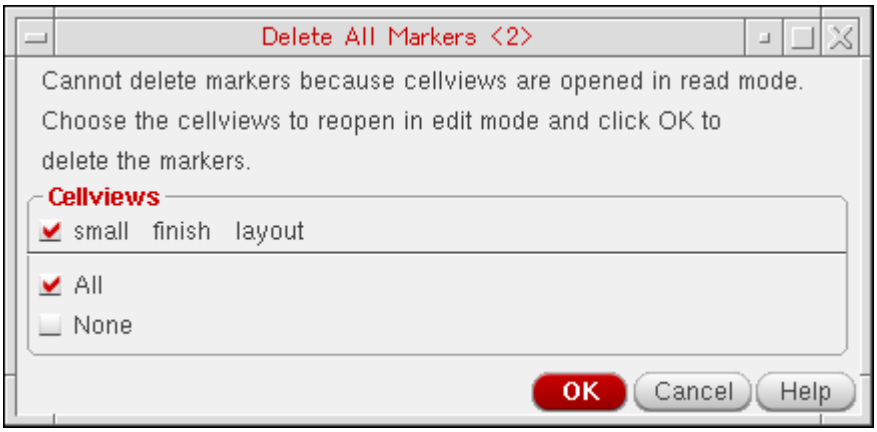
© 2023 Cadence Design Systems, Inc.

The Delete All Markers form opens.



2. Select the *Severity* types, *Search Scope*, and *Source*.
3. In the *Options* section, select the *Cellview Access Mode*.
4. Click *OK* or *Apply*.

   If you set *Cellview Access Mode* to *edit*, select the *Confirm Access Mode Change* check box, and then click *OK* or *Apply*. The *Delete All Markers* dialog box appears for you to confirm opening the selected cellviews in edit mode.



© 2023 Cadence Design Systems, Inc.

Select the cellviews that you want to open in edit mode and click *OK*. If you do not want to open the cellviews in edit mode, select *None* and click *OK*.

If you set the *Cellview Access Mode* to *read,* the *Confirm Access Mode Change* check box is disabled. Click *OK* or *Apply*. The markers are deleted and the cellview remains in read mode.

Related Topics

Delete Marker Form

Delete All Markers Form

Markers

Displaying Marker Information

Finding Markers in a Design

Checking Layout Integrity

You can perform integrity checks in a layout and view a summary of results. You can check for the following:

- Cells without library
- Cells without master
- Vias without definition
- Shapes without layer and purpose definition
- Pcell evaluation failure

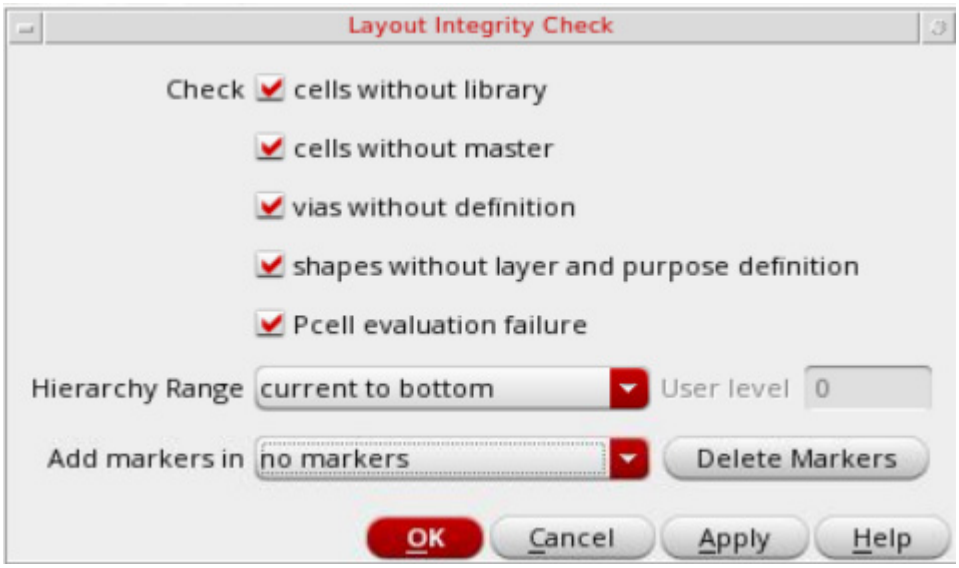
You can specify the hierarchy range to be considered for integrity checks. You can also specify whether markers should be added to the design.

To perform integrity checks in a layout:

1. In the layout window, choose *Verify – Integrity*.

© 2023 Cadence Design Systems, Inc.

The Layout Integrity Check form opens.



2. Select the checks that you want to perform. You can select *cells without library*, *cells without master*, *vias without definition*, s*hapes without layer and purpose definition*, and *Pcell evaluation failure*.

   The s*hapes without layer and purpose definition* option does not work when the `addDesignLppsToTech` environment variable is set to `t`.

3. From *Hierarchy Range*, select how many levels of hierarchy are to be considered for the layout integrity check.

4. From *Add markers in*, select where you want to add the markers by selecting from *no markers*, *current cellview*, or *cellviews in the hierarchy* options.

5. Click *OK* or *Apply*.

   The total number of warnings and the layout integrity checks report is displayed.

The objects with issues are tagged with markers in the design. You can use the *Markers* commands to find, view marker information, and delete the markers. The markers are also listed in the Annotation Browser assistant. Using Annotation Browser, you can select the columns to be displayed, sort the markers, and customize how the markers are grouped and displayed.

Related Topics

Layout Integrity Check Form

© 2023 Cadence Design Systems, Inc.

addDesignLppsToTech

Markers

Annotation Browser

Pad Shapes and Labels

In hierarchical designs, a pad shape and its corresponding pad label might not be at the same hierarchical level.

The *Pad Opening Info* command helps you find pad shapes at lower levels of your hierarchical cellview, generate reports and labels for the pad shapes that are found, and promote the generated labels to the top level of the cellview.

The *Pad Opening Info* command supports all shapes, except labels and text displays.

You can generate a standard report or a customized report for information about the pad shapes in a hierarchical cellview.

Related Topics

Generating a Standard Report for Pad Shapes

Generating a Customized Report for Pad Shapes

Setting Filters for a Customized Report

Generating Labels for Bond Pads

Promoting Labels to the Top Level

Generating a Standard Report for Pad Shapes

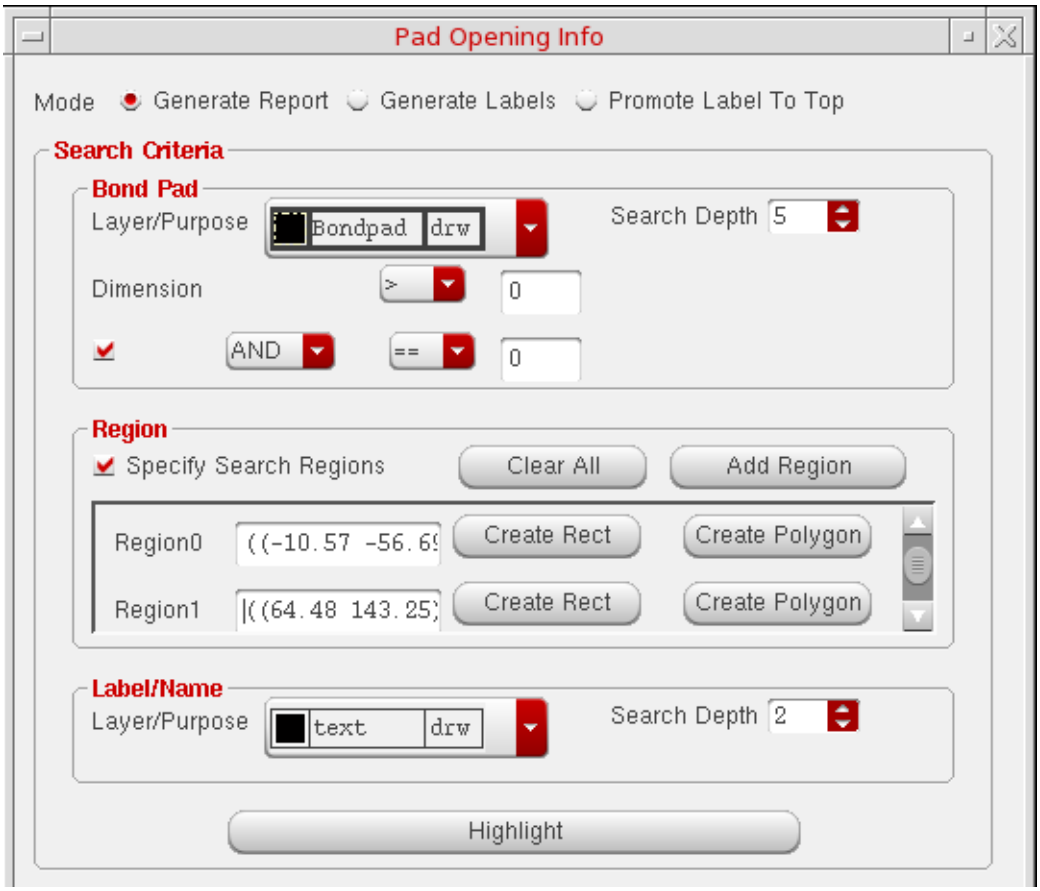
To generate a standard report containing information about the pad shapes found in all or part of a hierarchical cellview:

1. In the layout window, open a hierarchical cellview that contains bond pad shapes.
2. Choose *Tools – Pad Opening Info*.

   The Pad Opening Info form appears.

© 2023 Cadence Design Systems, Inc.

3. Select the mode as *Generate Report*.
4. Specify *Search Criteria* for the pad shapes and labels you are interested in.



   a. In the *Bond Pad* section, select the appropriate *Layer/Purpose* to search for pad shapes.

   b. Set *Search Depth* to limit the number of hierarchy levels to be searched. For example, specifying 5 means that levels 0 through 5 are searched.

   c. Set the X and Y *Dimension* for the pad shapes you are interested in, if required.

   d. In the *Region* section, optionally, use *Create Rect* or *Create Polygon* to limit the region in the cellview that is searched. Either draw the region that you require on the canvas or edit the coordinates manually in the field.

   e. In the *Label/Name* section, specify the *Layer/Purpose* and *Search Depth* to search for existing labels. For example, you can choose to search the layer purpose `text drw` at level 2 only.

© 2023 Cadence Design Systems, Inc.

5. Specify the coordinates for the first pad to be generated by clicking the *1st PAD coords* button and then clicking on the desired location on the canvas.

   The coordinates are populated in the field automatically. You can edit them manually if required.

6. Choose whether subsequent pad shapes should be labeled in the *Clockwise* or *Counter Clockwise* direction.



7. In the *Reporting* section, select the *Make Standard Report* check box and use the *Browse* button to specify the directory and file name for the report file.
8. Specify the file format for the report as *Plain text* or *CSV*.
9. Click *OK* or *Apply*.

   The report is displayed, listing the name, width, height, and area of bond pads. If the design contains duplicate bond pads, the duplicate bond pads are listed in the report only once.

   If there are multiple labels, the pad opening name is the label that has minimum distance between the pad origin and the label origin.

```
*****************************************************************************************
Pad Opening Locator
*****************************************************************************************
Created : Sep  8 12:18:28 2021
Libname : gpdk090_test
Cellname : test_leTbPadInfo_toplevelpads_stagger


Pad Label Hierarchy : 0
65 bond pad elements found
64 labeled bond pad elements found
1 non-labeled bond pad elements found
*****************************************************************************************

*****************************************************************************************
Pad Opening Name      X               Y               Area            Width:Height
*****************************************************************************************
PAD 66                -1030.723000    33.690000       3850.000000     70.000000 : 55.00
PAD 69                -1233.593000    131.920000      3850.000000     70.000000 : 55.00
PAD 68                -1030.723000    224.440000      3850.000000     70.000000 : 55.00
PAD 71                -1233.593000    322.670000      3850.000000     70.000000 : 55.00
PAD 70                -1030.723000    415.190000      3850.000000     70.000000 : 55.00
PAD 7                 -118.925000     1075.038000     3850.000000     55.000000 : 70.00
```


© 2023 Cadence Design Systems, Inc.

Related Topics

Pad Opening Info Form

Generating a Customized Report for Pad Shapes

Generating Labels for Bond Pads

Promoting Labels to the Top Level

Generating a Customized Report for Pad Shapes

To create a customized report containing information about the pad shapes found in a hierarchical cellview:

1. In the layout window, open a hierarchical cellview that contains bond pad shapes.
2. Choose *Tools – Pad Opening Info*.

   The Pad Opening Info form appears.
3. Ensure the mode is set to *Generate Report*.

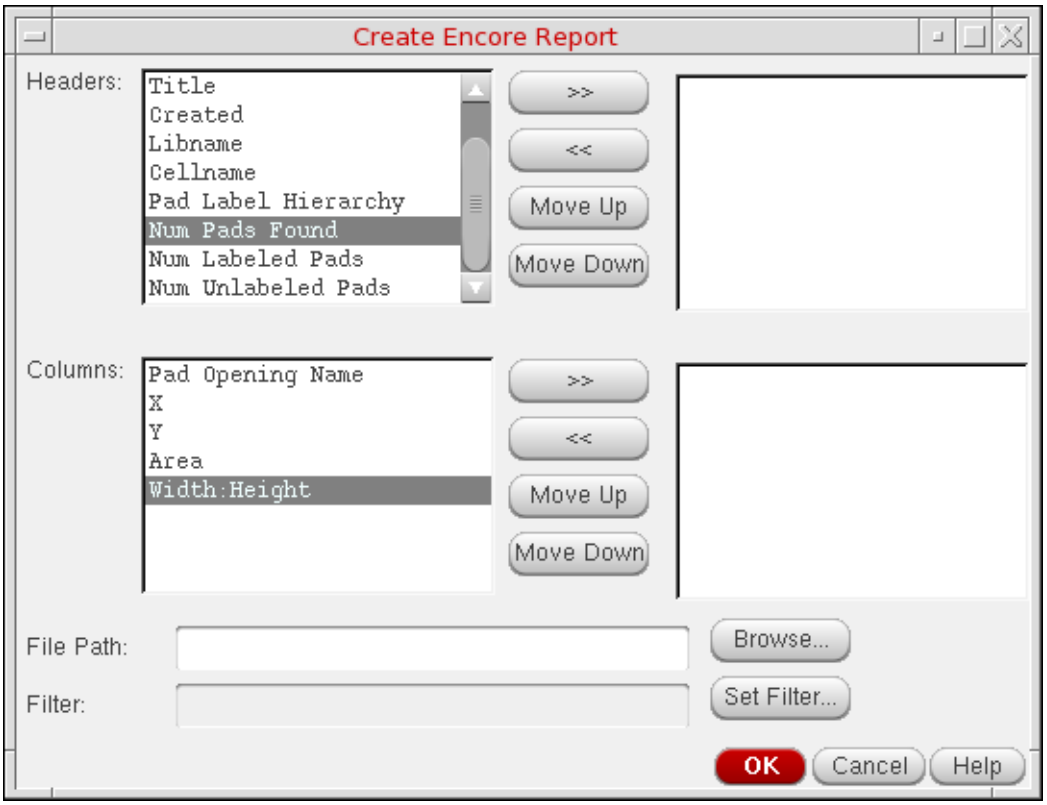


4. Set the *Search* and *Sort* criteria as required.
5. In the *Reporting* section, choose *Make Customized Report*.

   The *Add*, *Del*, and *Edit* buttons are enabled.
6. Click *Add* to create a customized report.

© 2023 Cadence Design Systems, Inc.

The Create Encore Report form is displayed.



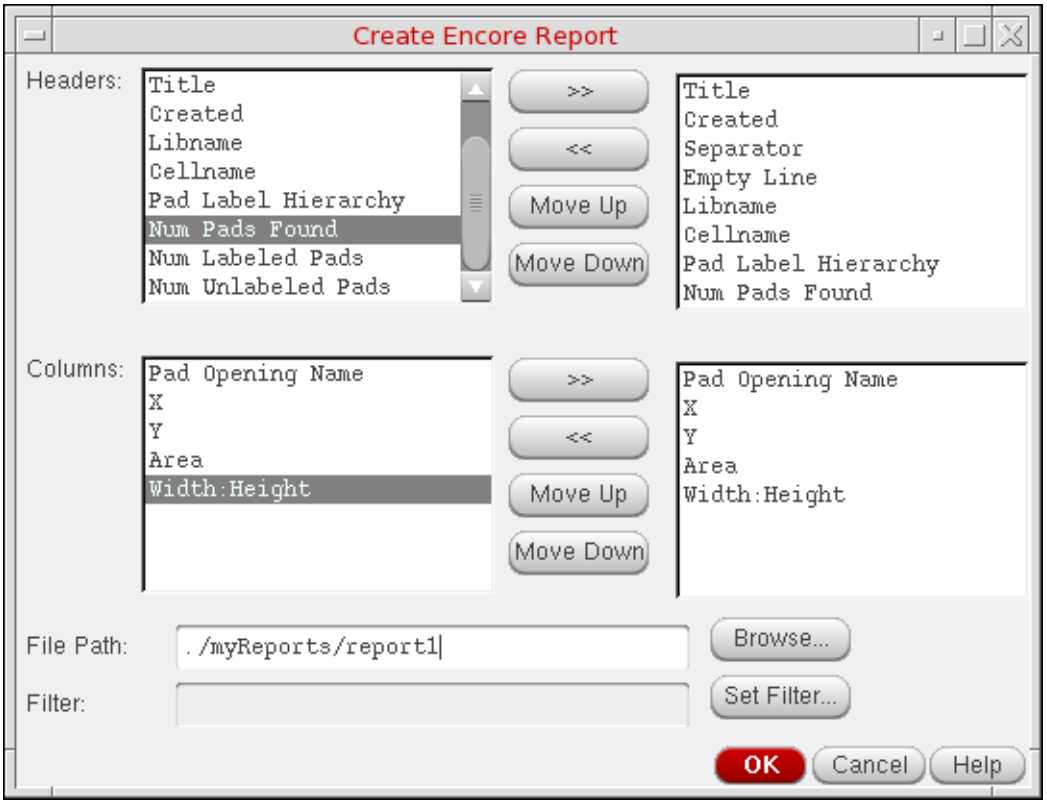
7. Specify the headers to be included in the report by selecting an option from the *Headers* list on the left and clicking the *>>* button. The selected header name is moved to the box on the right.

   To remove a header from the report, select the header in the box on the right and click the *<<* button.

8. Specify the columns to be included in report by selecting an option from the *Columns* list on the left and clicking the *>>* button. The selected column name is displayed in the box on the right.

© 2023 Cadence Design Systems, Inc.

To remove a column from the report, select the column label in the box on the right and click the << button.



9. In the *File Path* field, specify where you want the report to be saved. You can use the *Browse* button to navigate to the desired location.
10. Optionally, use *Set Filter* to filter the data in your customized report.
11. Click *OK* to close the Create Encore Report form.
12. Specify the file format for the report as *Plain text* or *CSV*, in the Pad Opening Info form.
13. Click *OK* or *Apply.*

    The custom report is generated and saved to the specified location.

Related Topics

Pad Opening Info Form

Create Encore Report Form

Generating a Standard Report for Pad Shapes

© 2023 Cadence Design Systems, Inc.

Generating Labels for Bond Pads

Promoting Labels to the Top Level

Setting Filters for a Customized Report

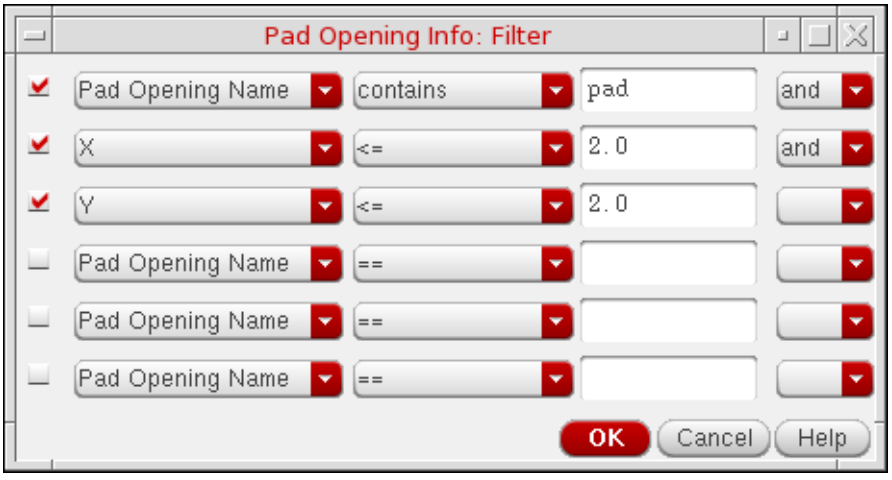
You use the Pad Opening Info: Filter form to filter the data shown in a customized report.

To filter the data for a customized report:

1. Open the Create Encore Report form from the Pad Opening Info form.
2. Click *Set Filter*.

   The Pad Opening Info: Filter form is displayed.
3. To enable a filtering criterion, select a check box, and from the corresponding list, select the mode as *Pad Opening Name*, *X*, *Y*, or *Area*.
4. Choose the appropriate operator from the second list.

   For example, to specify the criterion as equal to the content in the left field, select the == operator. The other valid operators in *Pad Opening Name* are *!=*, *contains*, and *doesn't contain*. For *X*, *Y*, and *Area*, the valid operators are ==, !=, <, <=, >, and >=.
5. Specify the required values in the input field.
6. From the last list for a criterion, select the operator for logical operation with the subsequent filters defined. The valid operators for logical operation are *and* and *or*.



7. Click *OK*.

© 2023 Cadence Design Systems, Inc.

The filters are set for the customized report and displayed in the *Filter* field in the Create Encore Report form.

Related Topics

Pad Opening Info Form

Create Encore Report Form

Generating a Customized Report for Pad Shapes

Generating Labels for Bond Pads

Use *Generate Labels* mode in the Pad Opening Info form to specify the label names that are generated and how they are drawn on the canvas.

To generate labels for bond pads:

1. Open a hierarchical cellview that contains bond pads.
2. Choose *Tools – Pad Opening Info*.

   The Pad Opening Info form is displayed.
3. Select the *Generate Labels* radio button at the top of the form.



4. Set the *Search* and *Sort* criteria as required.
5. In the *Label Control* section, set the following:
   a. Set the pattern for pad naming in the *Label (Pattern)* field. The label names include the label prefix and the pad numbers, for example, PAD1, PAD2, and so on.

      Alternatively, choose *From File* to specify a file from which the names are to be read.
   b. Select the appropriate *Layer/Purpose*. The labels are printed on the specified layer.
   c. Set the *Font*, *Height*, and *Justification* for the labels as required.

© 2023 Cadence Design Systems, Inc.

6. If required, click *Delete Bond Pad Labels* to remove all the labels from the top level.
7. Click *Apply*.

   You can now zoom in to any bond pad and see the label information placed on the pad opening.

Related Topics

Label Control

Pad Opening Info Form

Generating a Standard Report for Pad Shapes

Generating a Customized Report for Pad Shapes

Promoting Labels to the Top Level

Promoting Labels to the Top Level

Use *Promote Label to Top* mode in the Pad Opening Info form to promote generated labels to the top level of a cellview. You can also control how the labels are drawn on the canvas and whether the corresponding lower-level labels should be removed.

1. Open a hierarchical cellview that contains bond pads.
2. Choose *Tools – Pad Opening Info*.

   The Pad Opening Info form is displayed.
3. Select the *Promote Label to Top* radio button at the top of the form.



4. Set the *Search* and *Sort* criteria as required.
5. In the *Promote Label* section:
   a. Choose the object type as *Label*, *Text Display*, or *Both*.
   b. Select the appropriate *Layer/Purpose* where the labels need to be printed.

© 2023 Cadence Design Systems, Inc.

c. Set the *Font*, *Height*, and *Justification* for the labels as required.

d. Choose whether to *Remove Original Labels*.

6. Click *Apply*.

The required labels are promoted to the top level of the cellview.

Related Topics

Promote Label

Pad Opening Info Form

Generating a Standard Report for Pad Shapes

Generating a Customized Report for Pad Shapes

Generating Labels for Bond Pads

Status Toolbar Customization

You can customize the information that is displayed on the status toolbar by using the statusToolbarFields environment variable. You can display information such as the cellview size, number of selected objects, and values of environment variable, shell environment variables, and window variables.

The following table describes the types of information that you can display on the status toolbar.

Type of Information	Description	Example
Predefined selectors	Display the value of predefined selectors like `X`, `Y`, `dX`, `dY` on the toolbar. You need to specify each selector in parenthesis. On the toolbar, the selector is replaced by its actual value. The toolbar displays the current information for the current window.	`layout statusToolbarFields string "(X) (Y)"`

© 2023 Cadence Design Systems, Inc.

Type of Information	Description	Example
Text strings	Display a fixed text string on the toolbar. The fixed string is displayed in all windows.	`layout statusToolbarFields string "(T \"GPDK45\" \"Batch\")"`
CDS environment variables	Display the value of a CDS environment variable on the toolbar. You need to specify the variable as a string. The toolbar displays the current value of the variable in all windows.	`layout statusToolbarFields string "(E \"layout\" \"snapMode\")"`
Window variables	Display the value of a window variable on the toolbar. You need to specify the variable as a string or symbol. Values are fetched for the current window.	`layout statusToolbarFields string "(W stopLevel)"`
Shell variables	Display the value of a Shell variable on the toolbar. You need to specify the variable as a string. Values at window creation are shown for the current window.	`layout statusToolbarFields string "(SH \"USER\")"`
SKILL variables	Display the value of a SKILL variable on the toolbar. Specify the variable as a symbol or string. The current value is displayed in all windows.	`layout statusToolbarFields string "(S SomeVersion)"`
Separator	Display a separator between two items on the toolbar.	`layout statusToolbarFields string "(X) (Y) \\| (dX) (dY)"`

You can choose to display any number of items on the status toolbar. However, too many items might not fit in the window horizontally and the indicator >> appears at the end of the toolbar when the toolbar size is larger than the available width. As the data displayed on the toolbar is dynamic, Cadence recommends that you dock the toolbar only horizontally, either to the top or bottom of the design window.

© 2023 Cadence Design Systems, Inc.

Related Topics

Status Toolbar Information Formatting

Status Toolbar Information Formatting

You can also customize the way information is displayed on the status toolbar by using tooltips, styles, and conditional styles. When using the statusToolbarFields environment variable, the format of the string is a series of selectors. Use the following format for defining each selector in the series.

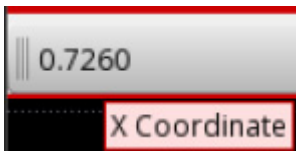
(*selector tooltip style_format*)

Where, *selector* refers to the selector that you want to display on the toolbar, *tooltip* refers to the tooltip for the selector, and *style_format* refers to the formatting style you want to apply.

Tooltips

Tooltips help you provide the description of items on the status toolbar. For example, if you want to show the X coordinate on the toolbar, then:

- `(X)` or `(X t)`, that is no second string or `t`, means that the tooltip is same as the first string; that is `X`.
- `(X "X Coordinate")` means that the tooltip for X is `X Coordinate`.



Style

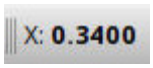
When you specify a selector in parenthesis, it is replaced by its value on the status toolbar. For example, the following figure shows a sample value for the argument `"(X) (Y)"`, depending on the mouse pointer coordinates.



© 2023 Cadence Design Systems, Inc.

You can display additional text and formatting for a selector on the toolbar. For example, you can choose to display the information in bold and in a different font color. For the `X` selector, the following sample argument displays the fixed text `X` and the coordinate in bold format.

```
"(X t \"X: <b>%s</b>\")"
```



For the `Select` selector, the following sample argument displays the number of selections in red and bold format.

```
"(Select t \"<span style=\\\"color:red;font-weight:bold\\\">Select:
%s</span>\")"
```



Conditional Styles

You can also use an expression format that allows conditional formatting of information based on values.

```
"(selector t (V operator compare_value format))"
```

Where:

- *`selector`* refers to the selector you want to display on the toolbar.
- `V` refers to any character that you may choose to add for readability.
- *`operator`* refers to any of the following operators:
  - == equal to the given value
  - != not equal to the given value
  - > greater than the given value
  - < less that the given value
- *`compare_value`* refers to the value based on which conditional formatting is to be applied.
- *`format`* refers to the conditional formatting you want to apply to the information.

For example, the following argument displays the selection count in bold and red when the count is greater than zero.

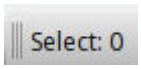
© 2023 Cadence Design Systems, Inc.

```
"(Select t (V==\"0\" \"Select: %s\")

(V!=\"0\" \"<span style=\\\"color:red;font-weight:bold\\\">

Select: %s</span>\"))"
```

The following figures show how the information appears on the toolbar when the selection count is zero or greater than zero.





Related Topics

statusToolbarFields

Status Information

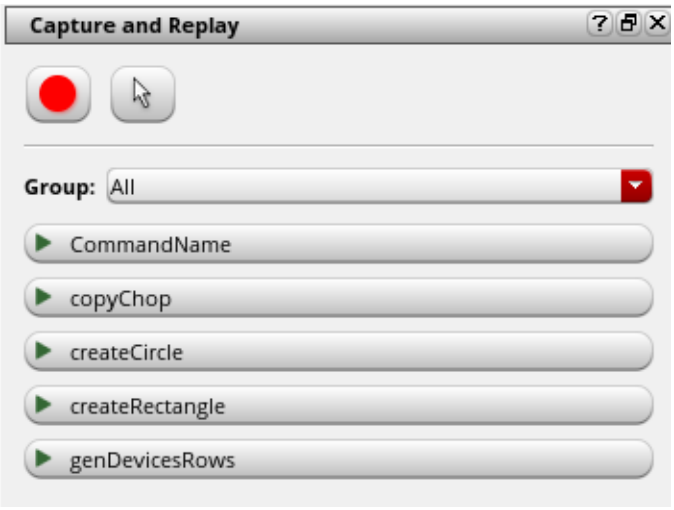
Status Toolbar Customization

Capture and Replay Your Design Actions

With the Capture and Replay assistant, you can record a series of actions that you perform in your design and later replay those actions as a macro. This helps you to perform repetitive tasks quickly and consistently across cellviews and sessions. For example, you can record your actions to generate devices and insert rows. You can then replay the actions in your designs as and when required. The capture and replay functionality is available in Layout EXL and higher tiers.

© 2023 Cadence Design Systems, Inc.

The following image shows the Capture and Replay assistant, with some sample recordings listed.



In addition to the assistant, you can use the following SKILL functions to capture and replay your design actions: leCARStartCapture, leCARStopCapture, and leCARCommand.

Related Topics

Capture and Replay Assistant

Recording Your Actions in a Design

Replaying Your Recorded Actions in a Design

Deleting a Recording in the Capture and Replay Assistant

Capture and Replay Assistant

Use the Capture and Replay assistant to record a series of actions and later replay them in your designs easily as required. The capture and replay functionality is available in Layout EXL and higher tiers.

© 2023 Cadence Design Systems, Inc.

The following table describes the Capture and Replay assistant buttons and options.

Icon	Command	Description
	*Start recording*	Lets you records the action that you perform in the current window. Once you start recording, the term *Recording* is displayed on the status banner. Also, the button toggles to *Stop recording*  .
	*Stop recording*	Stops recording your actions and displays the Save Recording form.
	*Replay recording with saved points*	Performs the actions from a recording at the exact same locations as saved in the recording.
	*Replay recording with user-entered points*	Performs actions from a recording at the custom point that you specify as the starting point for the replay.
	Group	Lists all the groups available for a set of recordings. You can specify a new group name or add the recording to an existing group using the Save Recording form. If you select a group, it displays a list of recordings related to the group. The default is `All`, which shows a list of all the recordings. You can change the default group using the carDefaultGroup environment variable. When you open the Capture and Replay assistant in multiple windows, the *Group* field in one window remains synchronized with the *Group* field of another window. This means that the list of groups and the list of recordings are the same in multiple windows. However, you can select different groups for different windows.

© 2023 Cadence Design Systems, Inc.

Icon	Command	Description
	Details	Displays the following details about a recording: ■ File name and location ■ Creation date and author ■ Command name and description ■ Name of the group under which the recording is categorized ■ SKILL function that you can use to replay the selected recording You can access the *Details* option by right-clicking a recording.
	Delete	Lets you delete a recording. You can access the *Delete* option by right-clicking a recording.
	Rename	Lets you change the name, description, and group of a recording using the Rename Command Name form. You can access the *Rename* option by right-clicking on a recording.

Related Topics

Save Recording Form

Capture and Replay Your Design Actions

Recording Your Actions in a Design

Replaying Your Recorded Actions in a Design

Deleting a Recording in the Capture and Replay Assistant

Renaming a Recording in the Capture and Replay Assistant

Rename Command Name Form

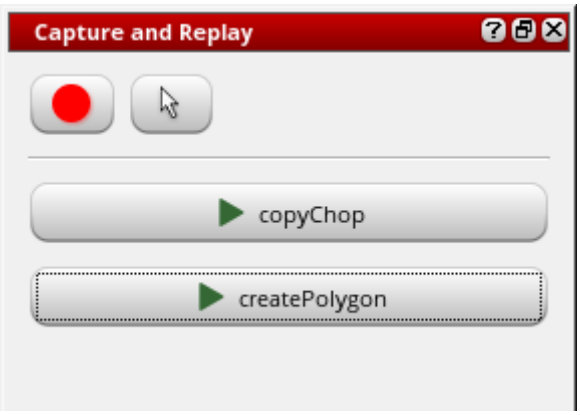
© 2023 Cadence Design Systems, Inc.

Recording Your Actions in a Design

To record your actions in the current design window by using the Capture and Replay assistant in Layout EXL or a higher tier:

1. Before you can start recording, ensure that you exit any currently active command.
2. From the menu bar in your layout design, choose *Windows – Assistants – Capture and Replay*.

   The Capture and Replay assistant is displayed.

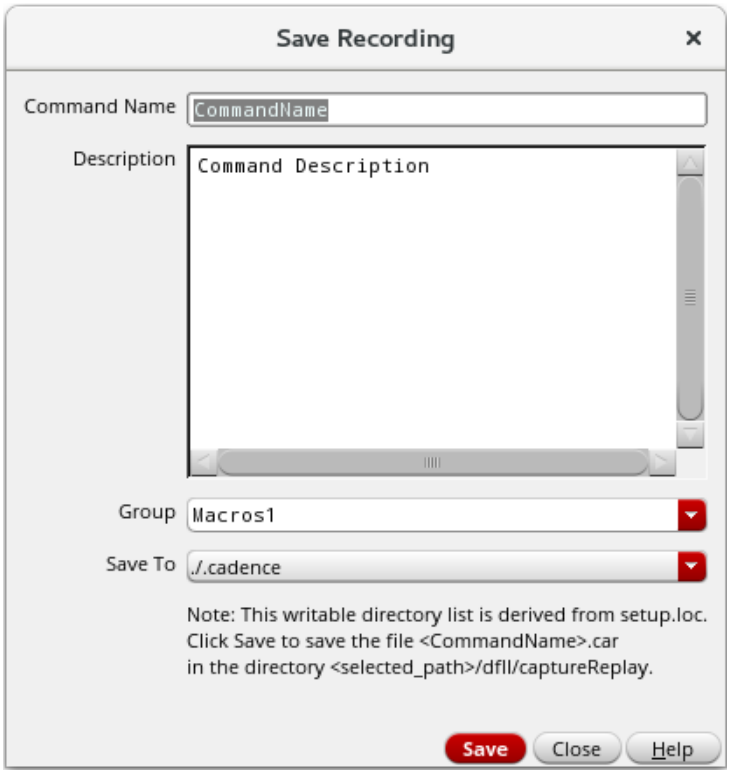


3. Click *Start recording* .

   This is a toggle button, which now switches to *Stop recording* . The term *Recording* appears on the status banner.
4. Perform the actions you want to capture.
5. Click S*top recording* .

© 2023 Cadence Design Systems, Inc.

The recording is stopped and the Save Recording form appears.



6. Specify the name of the recording in the *Command Name* field.
7. Add a description of your recording in the *Description* field.
8. Select a group or enter a group name in *Group*.

   Groups allow you to categorize your recordings.
9. From the *Save To* list, select a location to save your recording.

   The list of locations is derived from `setup.loc`. Your recording is saved with the `.car` extension under the `dfII/captureReplay` directory at the selected location.
10. Click *Save*.

    The saved recording is listed in the assistant.

    A recording operation is canceled if you close the current layout window during the recording.

    If Capture and Replay assistant gets closed during the recording due to any reason, for example while switching to another workspace, you need to invoke the assistant again or use the leCARStopCapture SKILL function to stop and save the recording.

© 2023 Cadence Design Systems, Inc.

Related Topics

Save Recording Form

Capture and Replay Your Design Actions

Capture and Replay Assistant

Replaying Your Recorded Actions in a Design

Deleting a Recording in the Capture and Replay Assistant

leCARStartCapture

Replaying Your Recorded Actions in a Design

You can play a recording of your actions to repeat those actions in a design. You can replay the actions at the points saved in the recording or you can specify your starting point on the design canvas.

- Replaying a Recording with Saved Points
- Replaying a Recording with User-Entered Points

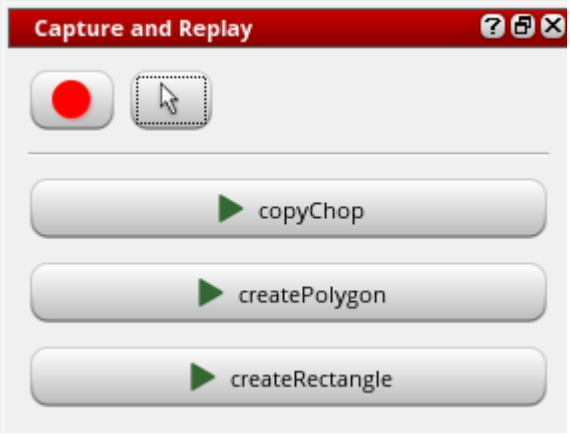
Replaying a Recording with Saved Points

To replay the actions at the points saved in a recording, in Layout EXL or a higher tier:

1. In your layout design, from the menu bar, choose *Windows – Assistants – Capture and Replay*.

© 2023 Cadence Design Systems, Inc.

The Capture and Replay assistant is displayed, listing existing recordings. You can hover the mouse pointer over a recording to see its brief description.



2. Optionally, view the details of the recording that you want to replay.
   a. Right-click the recording and select *Details.*

      Recording details such as the file name, creation date, and group name are displayed.

   b. Click *Close*.
3. Click the recording that you want to replay.

   The actions are performed in your design at the points that are saved in the recording.

Replaying a Recording with User-Entered Points

To replay the actions from a recording at a different starting point in a design, in Layout EXL or a higher tier:

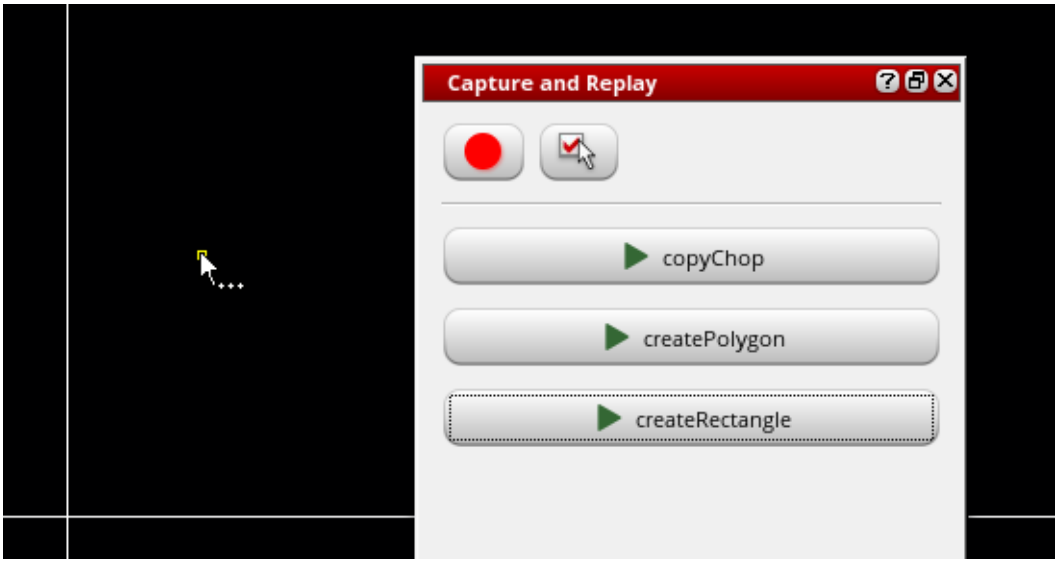
1. In the Capture and Replay assistant, click the *Replay recording with saved points* button  .

© 2023 Cadence Design Systems, Inc.

This is a toggle button, which now changes to *Replay recording with user-entered points* .

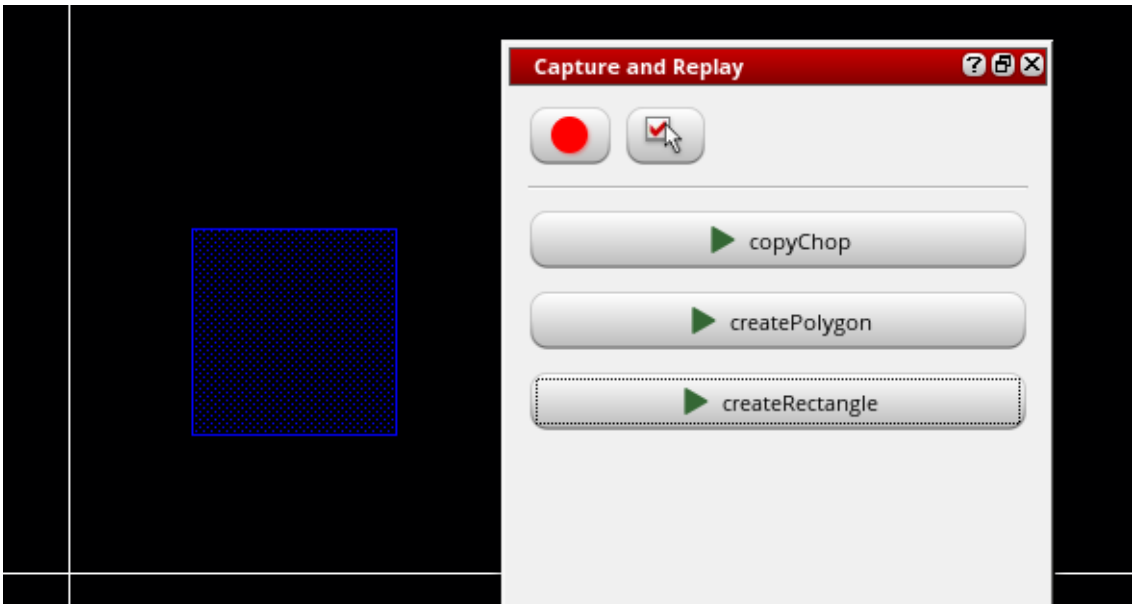


2. Click the recording that you want to replay.
3. Click at the point on the canvas where you want the recorded actions to be performed.



© 2023 Cadence Design Systems, Inc.

The recorded actions are performed starting with the specified point in your design.



Related Topics

Capture and Replay Your Design Actions

Capture and Replay Assistant

Recording Your Actions in a Design

Deleting a Recording in the Capture and Replay Assistant

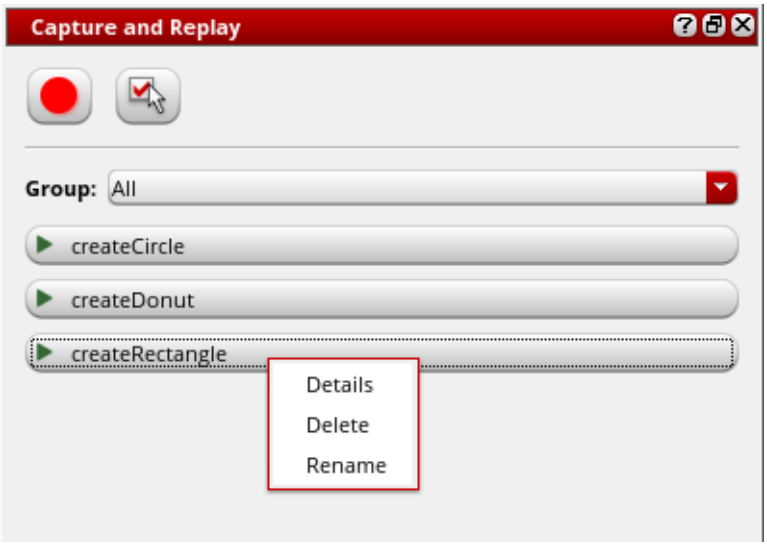
leCARCommand

Deleting a Recording in the Capture and Replay Assistant

To delete a recording from the Capture and Replay assistant in Layout EXL or a higher tier:

© 2023 Cadence Design Systems, Inc.

1. In the Capture and Replay assistant, right click on the recording that you want to delete.



2. Select *Delete* from the shortcut menu.
3. Click *Yes* to confirm deletion.

   The recording is removed from the list of recordings displayed in the assistant.

Related Topics

Capture and Replay Your Design Actions

Capture and Replay Assistant

Recording Your Actions in a Design

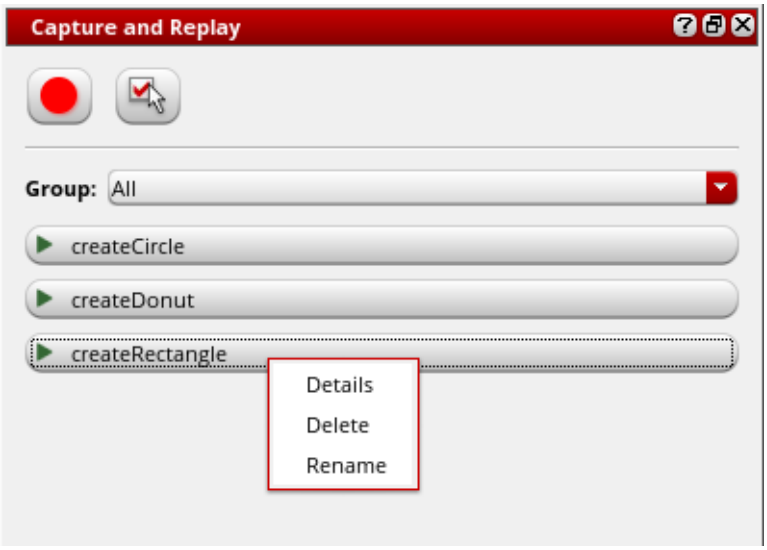
Replaying Your Recorded Actions in a Design

Renaming a Recording in the Capture and Replay Assistant

To rename a recording in the Capture and Replay assistant of Layout EXL or MXL:

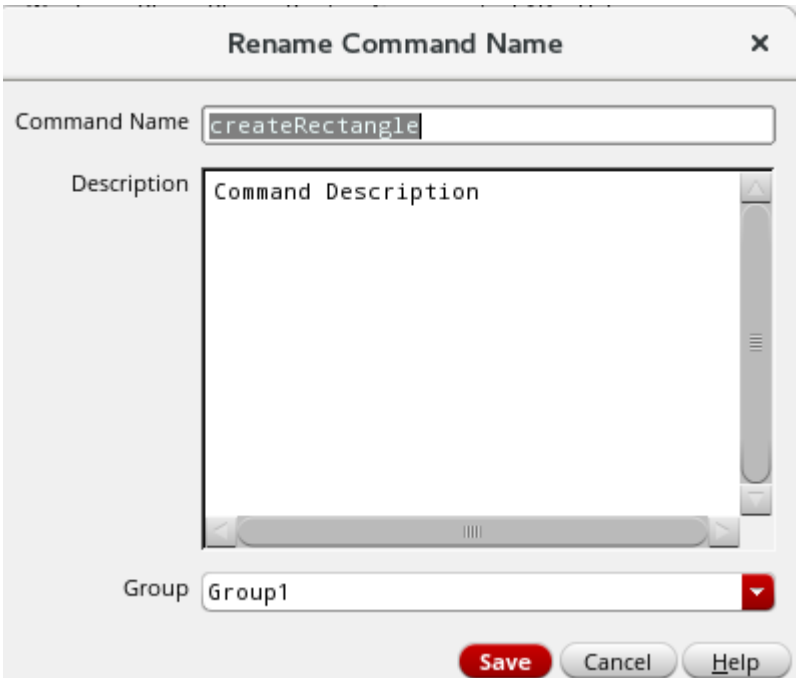
© 2023 Cadence Design Systems, Inc.

1. In the Capture and Replay assistant, right click the recording that you want to rename.



2. Select *Rename* from the shortcut menu.

   The Rename Command Name form appears.



3. Specify the name of the recording in the *Command Name* field.
4. Specify the information about the recording in the *Description* field.
5. Specify the group under which you want to save your recording in the *Group* field.

© 2023 Cadence Design Systems, Inc.

6. Click *Save*.

   The recording is renamed and listed in the specified group category in the assistant.

Related Topics

Capture and Replay Your Design Actions

Capture and Replay Assistant

Recording Your Actions in a Design

Replaying Your Recorded Actions in a Design

Rename Command Name Form

Cross Section Viewer

(Layout MXL Only) You can view cross sections of physical layers in a layout by using the cross section viewer. The viewer displays the different layers, from top to bottom, present in a cross section, enabling you to view each layer clearly. This is helpful to identify any missing or wrongly placed layers and to debug connection issues across layers. The layers displayed in a cross section are read from the technology file. The cross section viewer displays only the shapes that are visible on the canvas; it follows the settings specified for the filter size, stop level, layer visibility, and so on.

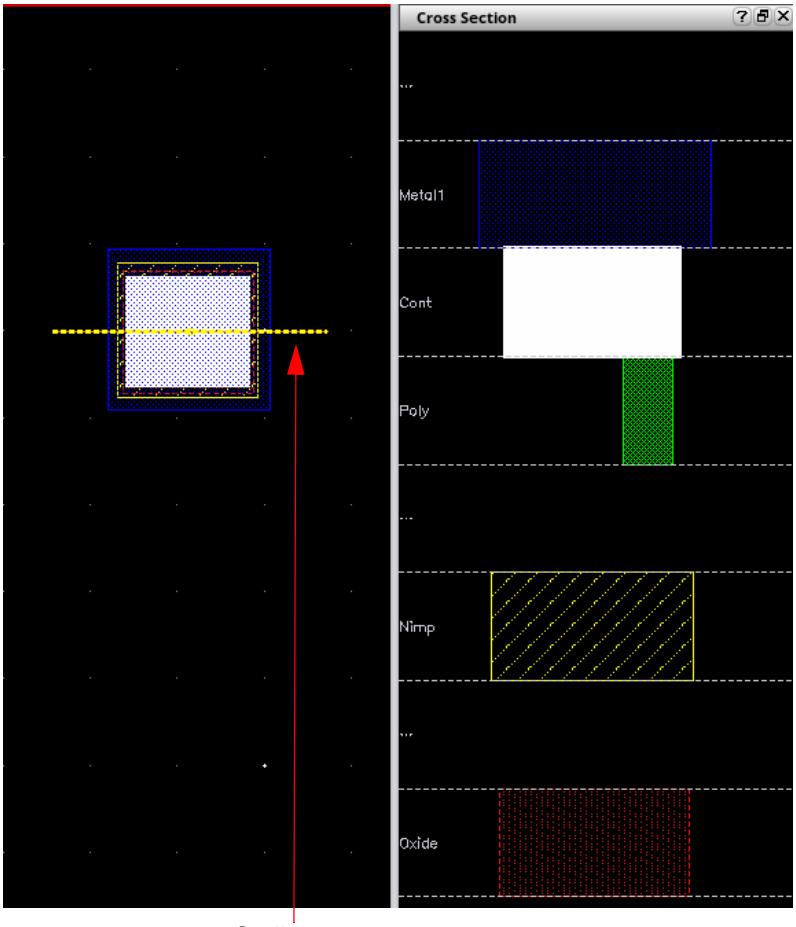
When you invoke the cross section viewer, a cutline appears on the canvas that follows your mouse pointer. As you hover the cutline over the design, the cross section of the area underneath the cutline is displayed in the viewer. When you hover the mouse pointer over a shape in the cross section viewer, it displays a tooltip showing the “layer / purpose” of that shape.

You can view a cross section in the following modes:

- Assistant mode: Displays the cross section in the Cross Section assistant. The following image shows that the area underneath the yellow cutline is displayed in the assistant

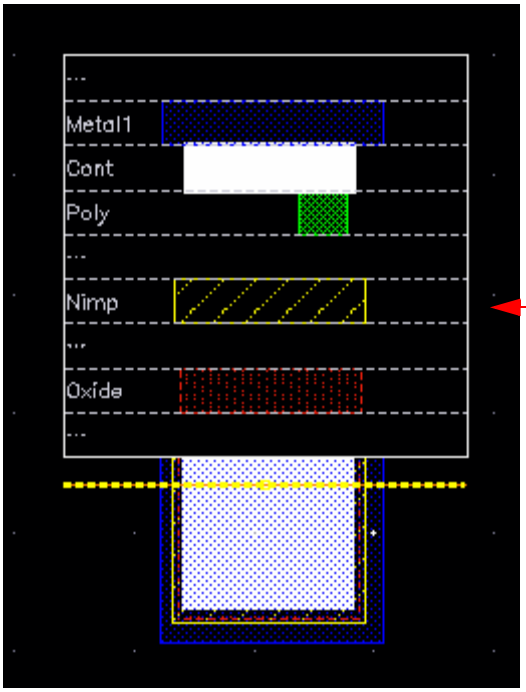
© 2023 Cadence Design Systems, Inc.

window. All the layers, from top to bottom, present in the cross section are listed in the assistant.



© 2023 Cadence Design Systems, Inc.

- Floating window mode: Displays the cross section in a floating window. In the following image, the area underneath the yellow cut line is displayed in a floating window that follows the mouse pointer.



Related Topics

Viewing the Cross Section of a Layout

Specifying Cross Section Viewer Settings

Cross Section Viewer Configuration File

Cross Section Viewer Form

Viewing the Cross Section of a Layout

To view the cross section of a layout in Layout MXL:

1. From the menu bar in your layout, do one of the following to invoke the *Cross Section* command:
   - Choose *Windows – Assistants – Cross Section*.
   - Choose *Tools – Cross Section*.

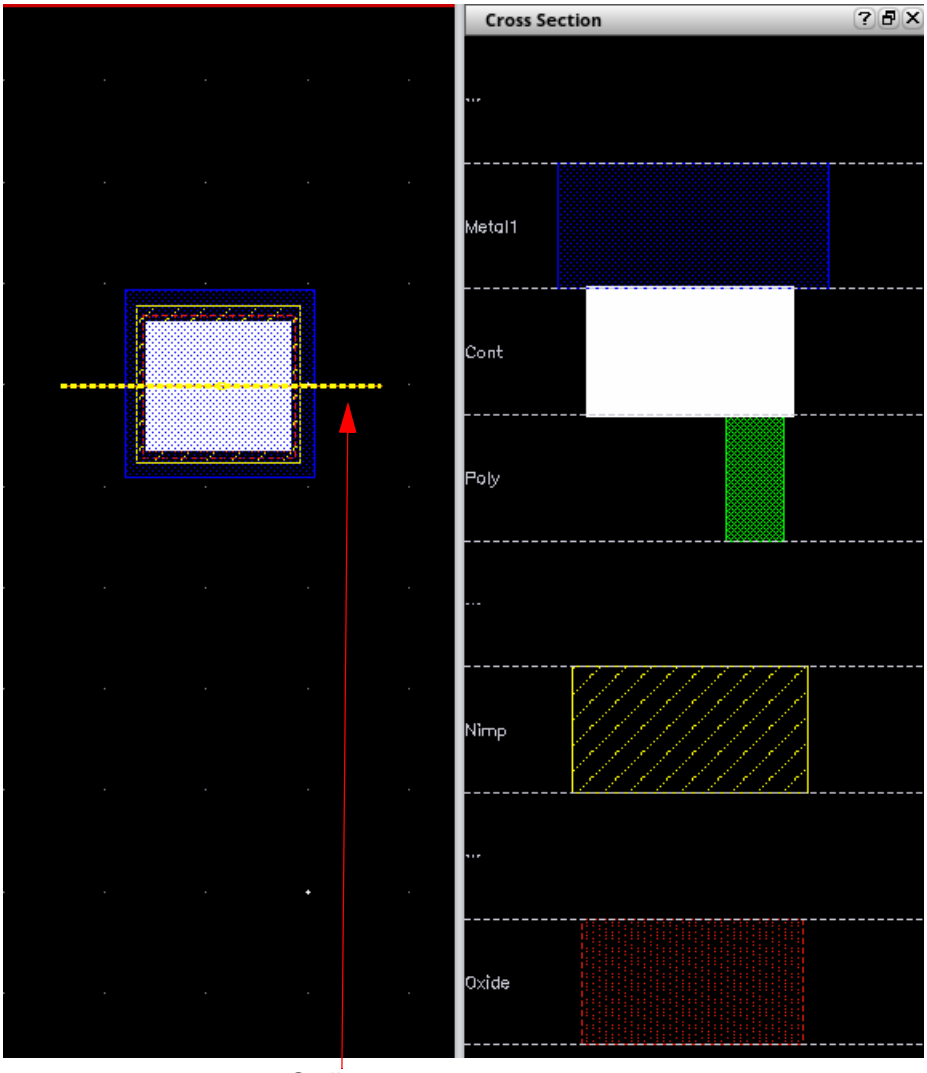
© 2023 Cadence Design Systems, Inc.

You can also press `Control` + `Q`.

A cutline corresponding to the mouse pointer is displayed on the canvas.

2. Move the mouse pointer to the area where you want to view the cross section of the layout.

   The cutline follows the pointer and displays the cross section in the assistant or a floating window, depending on the view mode setting. The following image shows the area underneath the yellow cutline displayed in the assistant window. All the layers, from top to bottom, present in the cross section are listed in the assistant.



3. To freeze the cross section of an area, click on that area on the canvas. Alternatively, select *Freeze* from the shortcut menu.

© 2023 Cadence Design Systems, Inc.

The selected cross section continues to be displayed on the canvas and the *Cross Section* command is exited.

4. To unfreeze the cross section, right-click in the floating window or the Cross Section assistant, as applicable, and select *Unfreeze* from the shortcut menu.

   The *Cross Section* command is active again and the cutline appears on the canvas, which you can move to view the cross section of the desired area.

Toggling Between Cross Section View Modes

You can view the cross section of a layout either in the Cross Section assistant window or in a floating window.

To toggle between the assistant and floating window in Layout MXL:

1. Invoke the *Cross Section* command.
2. Do one of the following to change the view mode:
   - Press the spacebar.
   - Right-click on the canvas and select *Assistant/Floating* from the Cross Section shortcut menu.
   - Select *Options* from the Cross Section shortcut menu. This displays the Cross Section Viewer form. Select the *Assistant* or *Floating* view mode.

   The view mode is toggled.

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Configuration File

Cross Section Viewer Form

geCrossSection

Specifying Cross Section Viewer Settings

To specify the settings for viewing cross sections of a layout in Layout MXL:

© 2023 Cadence Design Systems, Inc.

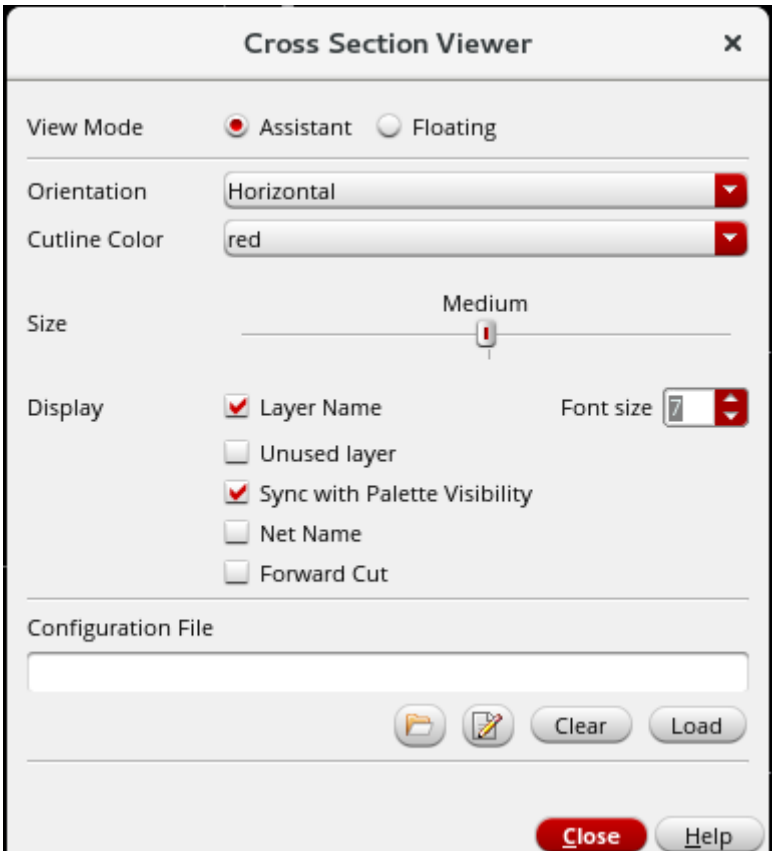
1. From the menu bar in your layout design, do one of the following:

   - Choose *Windows – Assistants – Cross Section*.
   - Choose *Tools – Cross Section*.

   The cross section viewer is displayed. Also, a cutline is displayed on the canvas corresponding to the mouse pointer.

2. Right-click on the canvas and select *Options* from the Cross Section shortcut menu.

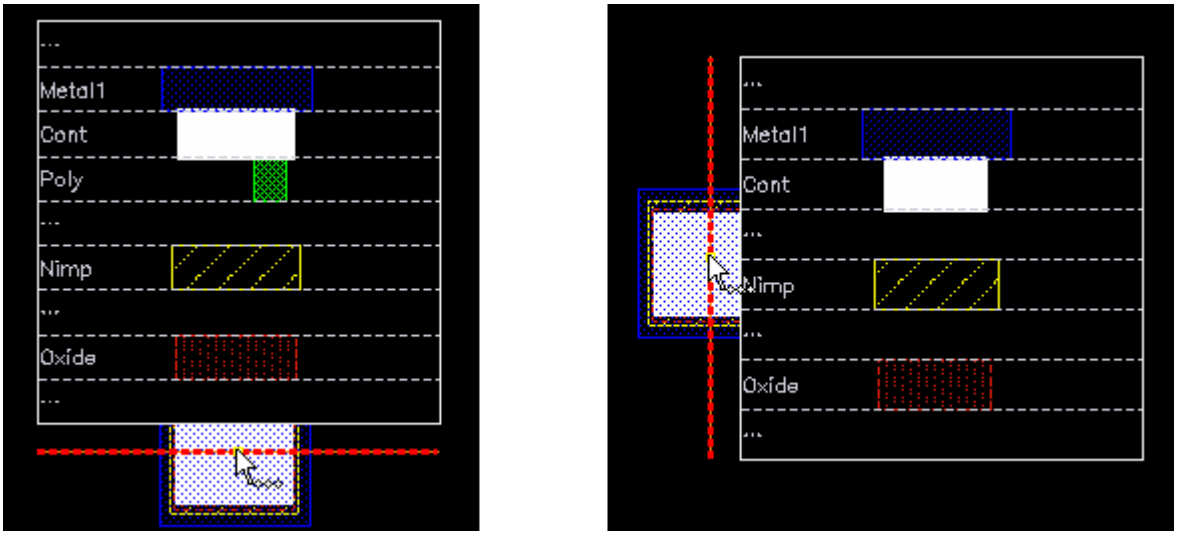
   The Cross Section Viewer form is displayed.



3. Select the preferred *View Mode* as *Assistant* or *Floating*.
4. Select the *Orientation* of the cutline as *Horizontal* or *Vertical*.

© 2023 Cadence Design Systems, Inc.

The following image shows how the cutline appears depending on the orientation, when you view a cross section.



5. Select a color of your choice for the cutline, from *Cutline Color*.
6. Drag the slider to set the size for the cutline as *Small*, *Medium*, or *Large*, from *Size*.
7. Specify the display settings, whether you want the names of used layers, unused layers, and nets to be displayed in the cross section. You can enable *Forward Cut* to display the vias present between two adjacent layers even if those vias are not directly under the cutline.
8. If you want to customize the cross section display, select the configuration file that contains the settings for displaying a cross section.
   - Use *Select configuration file on disk* to select an existing configuration file.
   - Use *Create or edit configuration file* to edit the specified file or generate a new one and edit it.

     You can customize details such as the metal thickness, layers or materials skipped, or layers merged in the cross section viewer. For more information, see Cross Section Viewer Configuration File.
9. Click *Load* to load the specified configuration file.

   The specified settings are used for displaying the cross section.

Related Topics

Cross Section Viewer

© 2023 Cadence Design Systems, Inc.

Viewing the Cross Section of a Layout

Cross Section Viewer Form

geCrossSectionLoadConfig

geCrossSection

geCrossSectionClearConfig

© 2023 Cadence Design Systems, Inc.

Cross Section Viewer Configuration File

You can use a configuration file to customize the settings that you want to use for displaying the cross section of a layout. If a configuration file does not exist, you can create one by using the sample auto-generated configuration file named `generatedConfig.il`. It contains certain default entries that you can edit as per your design requirements. The following image shows a sample configuration file. The layers listed in the file are read from the technology file.

./generatedConfig.il

```
; Define the thickness for metal layer, missing layer will be interpolate
geCrossSectionAddLayerThickness(`(
( "BuriedNWell" "1" )
( "NWell" "1" )
( "ThickOx" "1" )
( "Active" "1" )
( "Nsvt" "1" )
( "Nlvt" "1" )
( "Nhvt" "1" )
( "Psvt" "1" )
( "Plvt" "1" )
( "Phvt" "1" )
( "Poly" "1" )
( "PPitch" "1" )
( "M1" "1" )
( "CutM1" "1" )
( "V1" "1" )
( "M2" "1" )
))

; Define which layer need to be skipped
geCrossSectionAddLayerToSkip(`(
; ("BuriedNWell")
))

; Define which material need to be skipped
; All layer with this material will be skipped
geCrossSectionAddMaterialToSkip(`(
; ("recognition")
))

; Define which layers need to be merged on the same line
geCrossSectionAddLayersToMerge(`(
; ("BuriedNWell" "NWell")
))
~
```

Configuration File Information

The sample configuration file contains sections where you can define the following information:

© 2023 Cadence Design Systems, Inc.

- Metal thickness: Lets you define the thickness of each physical layer in the cross section viewer. By default, all layers are displayed using the same thickness.

  This information is defined using the geCrossSectionAddLayerThickness function.

  The cross section viewer adapts the thickness of layers to show electrical connections between different layers. For example, in the technology diagnostic file if we have under ‘Electrically connected layers’:

  (Metal1) -> (Cont) -> (Oxide & Nimp)(Oxide & Pimp)(Poly)

  The thickness of the "Cont" (Contact) layer would adapt in the viewer to show that it is connected to a shape on Oxide and Nimp (for example).

- Skipped layers: Lets you specify the layers that you do not want to be displayed in the cross section viewer. This information is defined using the geCrossSectionAddLayerToSkip function.

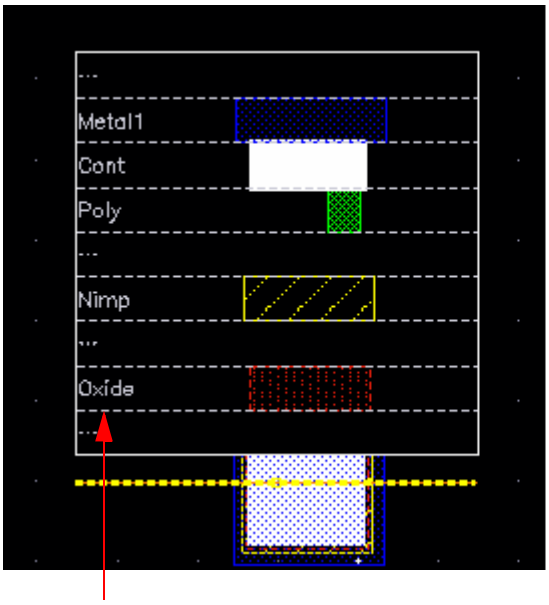
- Skipped material: Lets you specify the layers that are made of a specific material not to be displayed in the cross section viewer. For example, you can filter the material `passivationCut` and `recognition` if you do not want to display them in the viewer.

  This information is defined using the geCrossSectionAddMaterialToSkip function.

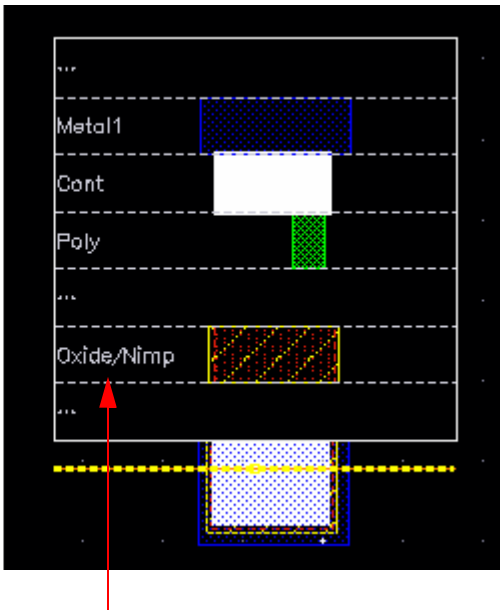
- Layers merged on the same line: Lets you specify the layers that you want to be listed on the same line in the cross section viewer. For example, the following image shows how the cross sections look when the `Nimp` and `Oxide` layers are listed on separate lines and on the same line.

© 2023 Cadence Design Systems, Inc.

This information is defined using the geCrossSectionAddLayersToMerge function.



Nimp and Oxide layers displayed on separate lines in the cross section viewer



Nimp and Oxide layers displayed on the same line

Loading the Configuration File

If you want the configuration file to be loaded automatically when you launch Virtuoso, you can add the geCrossSectionLoadConfig SKILL function call to your `.cdsinit` file. For example, the following entry loads the configuration file named `crossConfig.il` associated with the technology library `gpdk045`, every time you start Virtuoso.

```
geCrossSectionLoadConfig("gpdk045" "./crossConfig.il")
```

Related Topics

Cross Section Viewer

Viewing the Cross Section of a Layout

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

Cross Section Viewer Functions

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

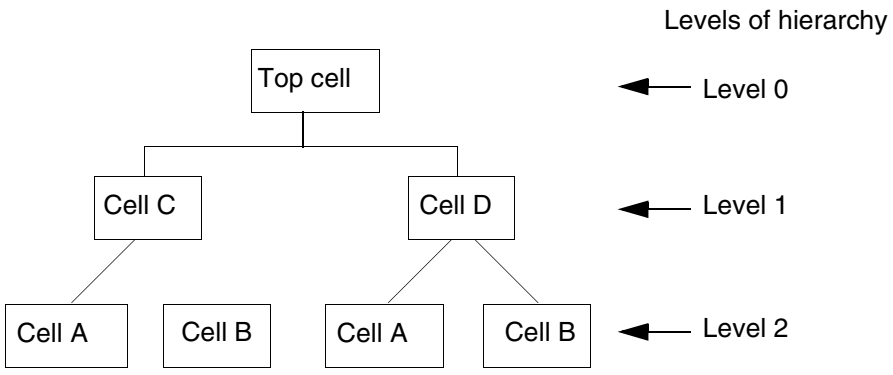
4

Hierarchical Designs

A design hierarchy comprises many levels of a design. Each time you descend into the design hierarchy, you view a smaller division of the larger design in greater detail. Hierarchical designs are created by placing instances of other cellviews inside a cellview. Levels of hierarchy are numbered from top to bottom, with the top cellview having the lowest number. The top cell is the cellview you are currently viewing in a design window, unless you have descended into the hierarchy or are using *Edit In Place*.

The following figure illustrates a design hierarchy. When you view Top cell, it is level 0. When you view Cell C or Cell D, it is considered level 1, and Cell A and Cell B are considered level 2.

You can descend into any level of an instance hierarchy and view the contents. You can also choose the view type or view name when descending into instances. By default, you are not prompted for a view type.



When you place a cellview within another cellview, the placed reference of the cellview is called an instance. The placed instance now has a master cellview. In the figure, Cell A has been placed twice in the design. If you edit the master of Cell A, all instances of that cellview are changed.

You can view the hierarchy tree of a design by using *Edit – Hierarchy – Tree* and *CIW – Tools – Print Hierarchy Tree*. The *Edit – Hierarchy – Tree* option prints the hierarchy as displayed in the *Navigator* assistant. It considers edits in the virtual memory. However, *CIW*

© 2023 Cadence Design Systems, Inc.

– *Tools – Print Hierarchy Tree* does not consider edits in the virtual memory which are not committed. For more information, see Viewing the Hierarchy Tree.

Related Topics

Checking for Instance Recursion

Changing the Origin of a Cellview

Descending into a Cellview

Returning to a Different Hierarchy Level

Editing a Cell in Place

Freezing and Unfreezing Pcell Instances

Making Hierarchy Editable

Making Cells

Instance Hierarchy Flattening

Copying and Cutting Through the Hierarchy

Uniquifying a Cellview

Viewing Cellview Summary

Refreshing Cellviews to Reflect Updated Masters

Checking for Instance Recursion

You can prevent commands such as *Create Instance*, *Make Cell*, and *Search/Replace* from creating an instance of itself in a cellview. You can enable recursion checking by using the *Layout Editor Options* form.

To automatically check for recursion when placing instances:

1. In the layout window, choose *Options – Editor*.

   The Layout Editor Options form opens.

2. Select *Recursion Check* and click *OK*.

© 2023 Cadence Design Systems, Inc.

Related Topics

Layout Editor Options Form

Changing the Origin of a Cellview

You can use the *Move Origin* command to move the origin of a cellview. Moving the origin of a master cellview changes the location of all placed instances of that cellview.

The *Move Origin* command fully supports blockage objects. When you move the origin of blockage objects without owners, the coordinates are updated. When you move the origin of owners of blockage objects, the blockage coordinates are also updated. The *Move Origin* command also fully supports boundary objects.

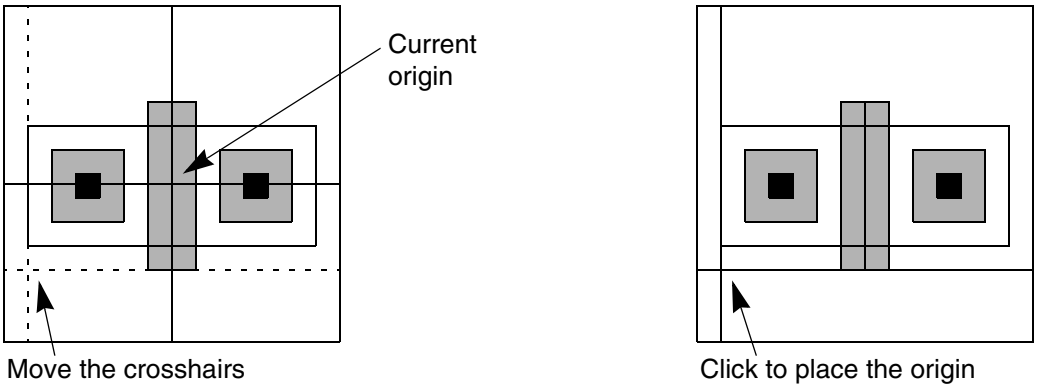
To change the origin of a cellview:

1. In the Layout window, choose *Edit – Advanced – Move Origin*.

   The *Move Origin* command is disabled if you are editing a group.

2. Move the pointer to the location on the canvas where you want the new origin and then click at that location.

   As you move the pointer, two crosshairs showing the new X and Y axes follow the pointer.



Related Topics

Hierarchical Designs

© 2023 Cadence Design Systems, Inc.

Descending into a Cellview

You can descend into any level of an instance hierarchy by using the *Descend* command. You can choose to be prompted for the view type when descending into a hierarchy. Once you have chosen to be prompted for the view type, you can pre-select or post-select an instance when using the *Descend* command.

If the top cellview is opened in Layout Viewer, the descend into a new tab or window always opens the master in the Layout Viewer, regardless of the default application settings.

To descend into a cellview:

1. Choose *Edit – Hierarchy – Descend Edit* or *Descend Read.*

   You can choose whether you want to edit using *Descend Edit*, as long as the parent is opened in edit mode, or read only using *Descend Read* that allows you to descend into an instance and view the contents without the ability to edit the instance.

2. Optionally, press `F3` to bring up the Descend Options form if you want to be prompted for the view type every time you descend into the hierarchy.

   a. Select the *Prompt For View Name* check box.

      If *Prompt For View Name* is not enabled and the master cell of the instance has more than one view type available, the command will determine which cellview to descend into. If *Prompt For View Name* is set and the master cell of the instance has more than one view type available, a form is displayed listing all the possible view names. You can then select the cellview to descend into. By selecting *Prompt for View Name*, each time you descend into a cellview, you will be prompted for the view type.

   b. Specify whether you want to open the cellview in the current tab, new tab, or new window.

   c. Click *OK*.

3. Click the instance whose cell you want to view. You can use the spacebar to toggle between the overlapping objects under the pointer.

   To descend into an instance, alternatively, you can select the instance, right-click it to display the instance shortcut menu, and then choose *Descend Edit* or *Descend Read.*

You can set in advance the descend mode you prefer by using the `leHiDescend` SKILL function. It has an optional text string *`t_mode`* specifying the mode in which to open the selected instance. When `nil`, the instance is opened in the same mode as its parent. When `read`, the instance is opened in read mode. When `edit`, the instance is opened in edit mode

© 2023 Cadence Design Systems, Inc.

if its parent is open in edit mode; otherwise, a window pops up stating that the instance cannot be opened in edit mode and that you can open it in read mode. When the `t_mode` argument is not specified, edit mode is automatically used when descending into the hierarchy.

The layout editor does not use the *Create New Window When Descending* option in the User Preferences form. Selecting or deselecting this option will not affect your work in the layout editor.

Related Topics

User Preferences Form

Returning to a Different Hierarchy Level

Hierarchical Designs

Descend Options Form

Returning to a Different Hierarchy Level

When you descend into an instance hierarchy by using *Edit in Place* or *Descend* commands, you can return to a higher level in the hierarchy.

To return to a specific level in the hierarchy:

1. In the layout window, ensure you have descended into an instance hierarchy.
2. Return up the hierarchy by choosing any of following, as applicable, from *Edit – Hierarchy*:
   - *Return*: Returns one level up in the instance hierarchy.
   - *Return To Top:* Return to the top level of the instance hierarchy.
   - *Return To Level*: Return to a specific level of the instance hierarchy. You need to select the level you want to return to.
3. If using *Return To Level*, choose the cell level to return to and click *OK*.

   If you are in *Edit In Place* mode, the window border changes to show the cell you selected. If you previously used *Descend,* the window contents change to show the cell you selected.

© 2023 Cadence Design Systems, Inc.

Related Topics

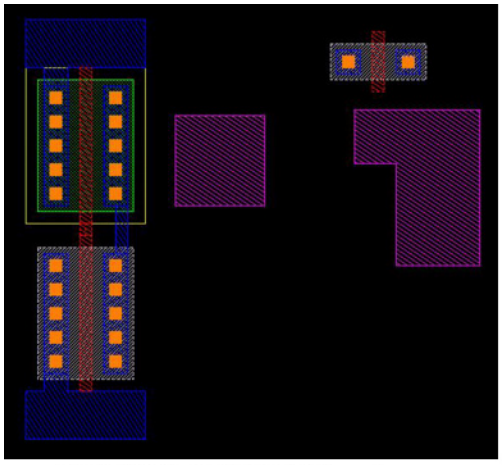
Descending into a Cellview

Hierarchical Designs

Editing a Cell in Place

You can edit the master cell of an instance while viewing the instance in the current cellview. This is called editing in place (EIP).

The behavior of the *Edit In Place* command differs based on the selection mode. Consider the following design, where cellview `cv1` contains the instance `ILevel1`. Using the spacebar, you can toggle between the overlapping objects under the pointer.



Cellview `cv1` Containing the Instance `ILevel1`



To edit in place in pre-selection mode:

© 2023 Cadence Design Systems, Inc.

1. In the cellview, select the instance. For example, in cellview `cv1`, select the instance `ILevel1`.
2. Choose *Edit - Hierarchy - Edit In Place* or right-click the selected instance `ILevel1` and select *Edit In Place* from the shortcut menu.

   The master cellview of `ILevel1` displays for editing in place. Because you can edit only top-level instances, the selection mode lets you EIP only one level in the hierarchy.
3. To edit in place other levels of the hierarchy, select the instance again and run the *Edit In Place* command each time to further descend into the hierarchy.
4. To ascend the hierarchy, choose *Edit – Hierarchy – Return To Level* or *Return*.

When you create a group and you select the *Edit In Place* command, you can use the *Remove from Group* option from the shortcut menu to remove a shape from the group.

When editing in place into an instance array, you can use the Select Array Member form to enter the row and column information for the array member you want to edit.

Related Topics

Edit In Place in Post-Selection Mode

Select Array Member Form

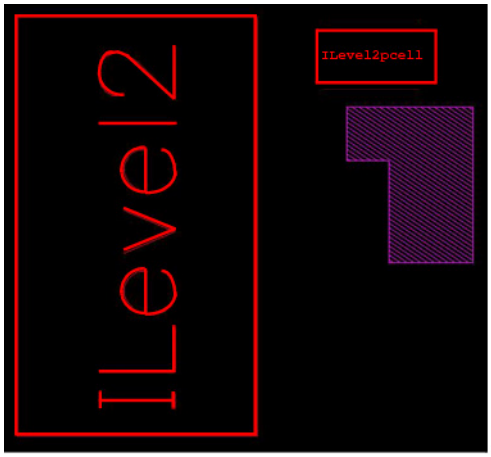
Hierarchical Designs

Edit In Place in Post-Selection Mode

The behavior of the *Edit In Place* command in post-selection mode depends on the area clicked for selection. Consider a sample design where cellview `cv1` contains the instance

© 2023 Cadence Design Systems, Inc.

`ILevel1`. The instance `ILevel1` contains a polygon, the Pcell instance `ILevel2pcell`, and the instance `Ilevel2`.



Instance `ILevel1`
Containing the Instance `ILevel2`,
the Pcell Instance `ILevel2`pcell,
and a polygon

When you invoke the *Edit - Hierarchy - Edit In Place* command and then click the bounding box of the instance `ILevel1`, the following situations are possible in the sample design:

- If the click point is on the bounding box of `ILevel2`, the master cellview of `ILevel2` becomes the cellview to be edited in place. The banner at the top of the layout window changes to show that the master cell for that instance is open for edit. If the *EIP Surround* is off in the Display Options form, only the cell to be edited in place appears.
- If the click point is in the area of `ILevel2pcell`, the following message displays: *Cannot edit in place a parameterized cell*. This is because the cellview at the lowest level in the hierarchy at the point of click is a Pcell. If the click point is on an empty area within the `Ilevel2pcell` bounding box, there is a beep sound and the following prompt displays: *Point at a shape in the cellview to be edited-in-place*.
- If the click point is in the area of the polygon, the master cellview of `ILevel1` becomes the cellview to be edited in place.
- If the click point is in an empty area in `ILevel1`, there is a beep sound and the following prompt displays: *Point at a shape in the cellview to be edited-in-place*.

In post-selection mode, when the pointer is over a shape of the cellview to be edited in place, the shape is highlighted. You can edit in place ‘n’ levels; the destination cellview (the master of the lowest instance in the hierarchy) must not be a Pcell.

When you are done editing with editing in place, and as you return up the hierarchy, each level you pass is accessible in edit mode if you started in a cell that is in edit mode. If a lower-level cell was originally in read-only mode, it will return to read only once you return up the hierarchy. Additionally, if you have changed anything in a lower cell, you will be prompted to save that master cell.

© 2023 Cadence Design Systems, Inc.

Video

For an overview on how to toggle shapes in *Edit in Place*, *Descend Edit*, and *Descend Read* modes, see the video *Toggling shapes in EIP and Descend Edit / Read mode*.

Edit In Place and *Descend Edit* options are available for virtual hierarchies.

Related Topics

Editing a Cell in Place

Display Options Form

Freezing and Unfreezing Pcell Instances

For Pcell editing, all geometric updates are performed through the Pcell parameters. Also, stretch handles provide you with a graphical way to update Pcell geometries. However, these graphical updates are also internally managed through Pcell parameters. For user-defined Pcells, there may be a situation when the desired Pcell editing cannot be performed by the supported Pcell parameters. In such cases, you can use the freeze and unfreeze Pcells options. The freeze and unfreeze commands enable you to perform Pcell editing that is not supported by the Pcell parameters.

The freeze Pcell feature enables you to convert a Pcell instance to a non-Pcell instance of a new master created on the disk. Similar to any other instance, once you freeze the Pcell instance, you can use the *Edit In Place* command to perform geometric updates to the design master.

The unfreeze Pcell feature enables you to convert the frozen non-Pcell instance to its original Pcell instance.

Note: The freeze Pcells option creates a new design master on the disk. So, it should be used only when the Pcell parameters are not sufficient to create the required geometry.

You can use either of the following ways to freeze and unfreeze Pcells:

- Use SKILL functions:
  - Use the `leFreezeInst` function to convert a Pcell instance to a non-Pcell instance of a new master created on the disk. This function preserves the geometry, connectivity, and cross-probing with schematic information of the instance.

© 2023 Cadence Design Systems, Inc.

- Use the `leUnfreezeInst` function to convert the frozen non-Pcell instance to its original Pcell instance. Any editing that has been performed on the non-Pcell instance is not preserved.

- Use the *Freeze Pcell* and *Unfreeze Pcell* shortcut menu commands to freeze and unfreeze a Pcell instance.

  Use the Freeze Pcell Instance form to specify the library, cell, and view of the non-Pcell instance of the new master created on the disk. Alternatively, you can use the supressFreezePcellUI environment variable to freeze a Pcell instance without displaying the Freeze Pcell Instance form.

Related Topics

Freeze Pcell Instance Form

reuseExistingFrozenMaster

Making Hierarchy Editable

Use the *Make Editable* command to switch to edit mode for updating a hierarchy.

To make a hierarchy editable:

1. In the layout window, choose *Edit – Hierarchy – Make Editable*.

   The Make Readonly/Editable form opens, listing master cellviews of all the instances in the current cellview.

2. Use the *Filter* options if you want to specify a search criteria for listing only specific master cellviews.

   You can display the master cellviews of all or only the instances selected in the cellview. You can specify a name and type criteria for filtering master cellviews.

3. Select a *Readonly* master cellview in the left pane and click *Make Editable*.

   Alternatively, select an *Editable* master cellview in the left pane and click *Make ReadOnly*. You can select multiple master cellviews and change their mode.

Related Topics

Make Readonly/Editable Form

© 2023 Cadence Design Systems, Inc.

Making Cells

You can copy selected objects in your current cellview into a new cell. You can add levels of hierarchy by using the *Make Cell* command.

While creating a new cell from objects in your layout, consider the following about blockages and boundaries.

- Blockages:
  - When you select blockages without selecting their owner, a new cell with the selected blockages is created without any owner information. When *Replace selected figures* option is selected, *Make Cell* replaces all the selected set including blockage objects with the instance of the new cell in the current cellview. Once it is replaced, the ownership information is not retained.
  - When you select both the owner and the blockage, a new cell with the selected owner, all its children, and all associated blockages is created. When *Replace selected figures* option is selected, all selected objects are replaced in the current cellview with the instance of the new cell and its owner.
- Boundaries: The *Make Cell* command does not support snap and cluster boundary objects.
  - For area boundary objects, if the constraint group name is part of the same cellview, both the area boundary object and the constraint group are copied over. If the constraint group is not part of the same cellview, only the area boundary object is copied over.
  - For the PR boundary object, if its associated blockages are selected as well, the relationship between the PR boundary and blockage objects must be maintained. Even when the *Replace selected figures* option is selected, the PR boundary remains in the current cellview.

To create a cell from objects in your layout:

1. In the layout window, select the objects on the canvas that you want to place into a new cell.
2. Choose *Edit – Hierarchy – Make Cell*.

   The Make Cell form opens.
3. Specify the library, cell, and view name for the new cell.
4. Select the *Replace selected figures* check box to replace the selected objects with an instance of the new cell.

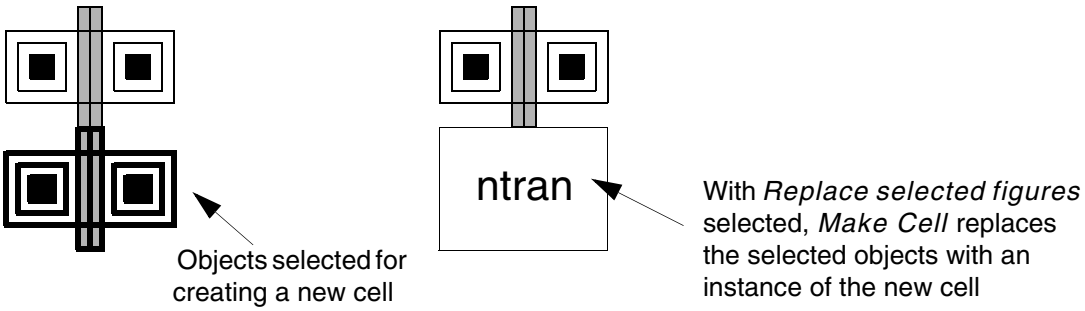
© 2023 Cadence Design Systems, Inc.

You can clear the check box to leave the objects unchanged.

5. Click *OK*.

   If you specified the name of an existing cell, a dialog box lets you choose to overwrite the existing cell with a new one.

   By default, the lower-left point of the bounding box of the newly created cell is placed at the origin (0,0).



Related Topics

Make Cell Form

Instance Hierarchy Flattening

You can flatten a design hierarchy by using the *Flatten* command. The command copies the contents of a cell or array up one or more levels in the hierarchy.

© 2023 Cadence Design Systems, Inc.

While flattening instances, consider the following about ROD objects, blockages, and boundaries.

Object	Flattening Support
ROD objects	The *Flatten* command fully supports ROD objects. You can preserve ROD objects attributes, such as object name, alignments, multipart path subparts, multipart rectangle sub-rectangles, and master rectangles in multipart rectangles, by selecting the *Preserve: ROD objects* option on the Flatten form. When this option is disabled, ROD objects become ordinary unnamed objects, the subparts of multipart paths become ordinary paths and rectangles, the master rectangles in multipart rectangles become ordinary, unnamed rectangles, and the sub-rectangles of multipart rectangles become ordinary rectangles. The system assigns the flattened object a name based on the hierarchical name of the ROD object by replacing slashes with dashes. For example, when you flatten the ROD object `I1/I4/rect3`, the resulting object is named `I1-I4-rect3`.
Blockages	Only instances contained within the blockage are flattened. The blockage object itself is not flattened. ■ If you select an instance which is the owner of a blockage to flatten based on the flattening level, only the instance is flattened, and the blockage remains without any ownership information. If the owner instance is not selected for flattening, the ownership relation remains unchanged. ■ When you flatten an instance with a placement halo, the placement halo will convert to a placement blockage with appropriate points. ■ You can flatten instances containing blockages and delete the remaining blockages by using the *Delete Detached blockages* option on the Flatten form.

© 2023 Cadence Design Systems, Inc.

Object	Flattening Support
Boundaries	The *Flatten* command does not support PR and snap boundary objects. ■ If you flatten an area boundary object which results in a non-unique name, an extension is added to the name, with a resulting message appearing in the CIW. If the associated constraint group no longer exists after the flattening operation, the object is stripped of its constraint group name, with a resulting information message appearing in the CIW. All normal transformations associated with flattening of an instance also apply to the area boundary object. ■ If you flatten a cluster with a cluster boundary and the cluster boundary becomes irrelevant, the cluster object set is updated and the cluster boundary is removed. If you flatten a cluster boundary object which results in a non-unique name, an extension is added to the name, with a resulting message appearing in the CIW. It is possible to have a hierarchical set of clusters based on the depth of the *Flatten* command. All transformations associated with the flattening of an instance also apply to the cluster boundary object. ■ If the flatten command results in a cellview that has no instances, you must remove all clusters and cluster boundary objects from the cellview.

Related Topics

Flattening Instances

Flattening Instances

To display the detail in all the instances that you want to flatten:

1. In the layout window, choose *Options – Display*.

   The Display Options form opens.

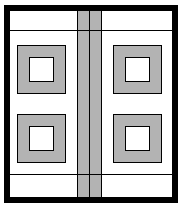
2. Set *Display Controls* to display the detail in all instances you want to flatten.

© 2023 Cadence Design Systems, Inc.

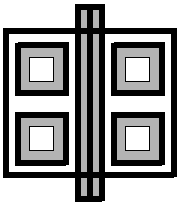
3. Click *OK*.
4. Choose *Edit – Hierarchy – Flatten*.

   The Flatten form opens.
5. Set the *Hierarchy Level* to *one level*, *displayed level*, or *user level*.
6. Use the Preserve options to preserve the required information and attributes.
7. Click *OK*.

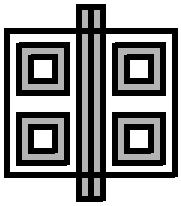
   The instance is flattened as per the specified settings.



The original instance (bold outline) contains four via instances.



One level flattened— geometries in the via instances are not flattened.



All levels flattened

When flattening Pcell instances, you can delete fluid shapes by enabling the deleteFluidShapes environment variable.

Related Topics

Display Options Form

Flatten Form

Copying and Cutting Through the Hierarchy

Yank and *Paste* can cut and copy all or part of instances and shapes in a designated area.

To copy parts of cell instances:

1. In the layout window, choose *Options – Display*.

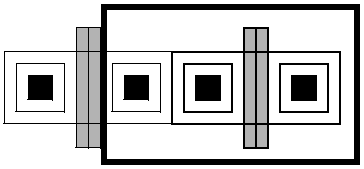
   The Display Options form appears.

© 2023 Cadence Design Systems, Inc.

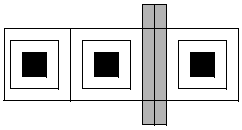
2. Set the *Start* and *Stop* levels in *Display Levels* to display details of all the instances that you want to copy. Then, click *OK*.
3. Choose *Edit – Basic – Yank*.

   The Yank form appears. Else, press `F3` to open the form.
4. Set the yank shape, yank level, and snap mode.
5. Click and drag on the canvas to create a box around the area you want to yank.
6. Choose *Edit – Basic – Paste*.
7. Click on the canvas where you want to place the copy.

   A copy of the selected part of the instance is placed at the specified reference point.



The yank box (in bold) cuts through the transistor instance on the left and encloses an instance on the right.



Paste places flattened copies of objects from the left instance plus the whole instance on the right.

Related Topics

Display Options Form

Yank Form

Uniquifying a Cellview

In cellviews that contain Embedded Module Hierarchy (EMH), if you want to edit a module that has been instantiated multiple times and you do not want all modules with the same name to be impacted, you need to first uniquify the module. Uniquifying a cellview involves renaming all modules with the same name to make them unique.

Cellviews containing EMH elements must be uniquified before making changes involving hierarchical objects, such as creating, deleting, or editing objects with hierarchical names. For example, you need to uniquify a cellview when creating an instance using a hierarchical path name.

© 2023 Cadence Design Systems, Inc.

You can open a non-uniquified design in Virtuoso Layout Suite. However, while performing an edit operation on the design that tries to change the hierarchy of the EMH, you receive warning messages regarding the non-uniquified instances. You then need to uniquify the cellview using the *Uniquify* command. The command checks out the Virtuoso Layout Suite XL or a higher license. The command works only on `oacMaskLayout` view types.

To uniquify a cellview:

1. In the CIW, select *Tools — Uniquify*.

   The Uniquify form appears.

2. Specify the library, cell, and view names in the *Library*, *Cell*, and *View* fields. Then, click *OK*.

   If uniquification is successful, a message displays after its completion. If the cellview is already uniquified, a message appears to inform you about it. If the specified cellview cannot be opened, the command exits displaying a message about the failure.

   The license is checked back in when the *Uniquify* form is closed.

The *Uniquify* command does not disable the *Undo* command, and it can, therefore, be used when you uniquify the cellview.

Related Topics

Uniquify Form

Viewing Cellview Summary

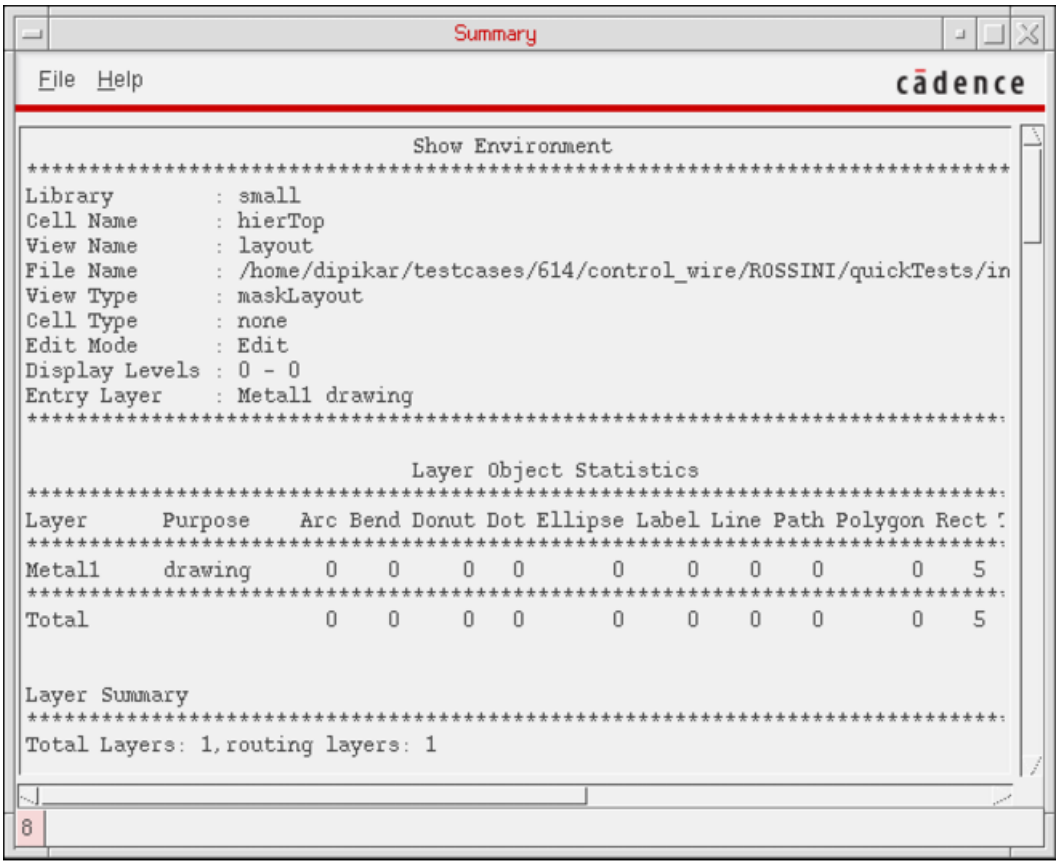
Use the Summary window to view statistics related to your current cellview. This can include statistics related to items such as layers, instances, markers, wires, and blockages.

To display a summary of the contents of a cellview:

➡ In the layout window, choose *File – Summary*.

© 2023 Cadence Design Systems, Inc.

The Summary window appears, displaying the cellview summary.



Related Topics

Summary Window

Refreshing Cellviews to Reflect Updated Masters

Another user might edit a master cellview that is instantiated in the cellview that you have currently open. You can update the memory of your computer to reflect the results of such edits by using the *Refresh* command.

To refresh a cellview to view other users' edits:

1. In the layout window, choose *Edit – Hierarchy – Refresh*.
   - If no other users have made edits that affect the cellview, a message displays that no cellviews were refreshed.

© 2023 Cadence Design Systems, Inc.

- If the master cells for any instances in the cellview have changed, a message prompts whether you want to update the instances.

2. In the Refresh Cellviews form, select the instances that you want to update. Then, click *OK*.

   The cellview is redrawn to show the changes made to the master cells.

Related Topics

Refresh Cellviews Form

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

Object Creation in Layout XL

You can add various types of objects to your design. You can add shapes such as rectangles, polygon, paths. You can also add instances, vias, pins, and labels. For placement and routing (PR), you can add objects such as blockages, rows, boundaries, and clusters.

- Object Creation by Entering Points
- ROD Support for Object Creation
- Creating a Rectangle
- Creating a Polygon
- Creating a Path
- Creating a Circle
- Creating an Ellipse
- Creating a Donut
- Creating Instances
- Pin Creation
- Label Creation
- Via Creation
- Creating Slots
- Creating a Group
- Placement and Routing Objects

Object Creation by Entering Points

When you create objects, you need to enter the points to create the shape you want. You can specify points in the following ways:

- By using the mouse pointer to specify the start and end points on the canvas

© 2023 Cadence Design Systems, Inc.

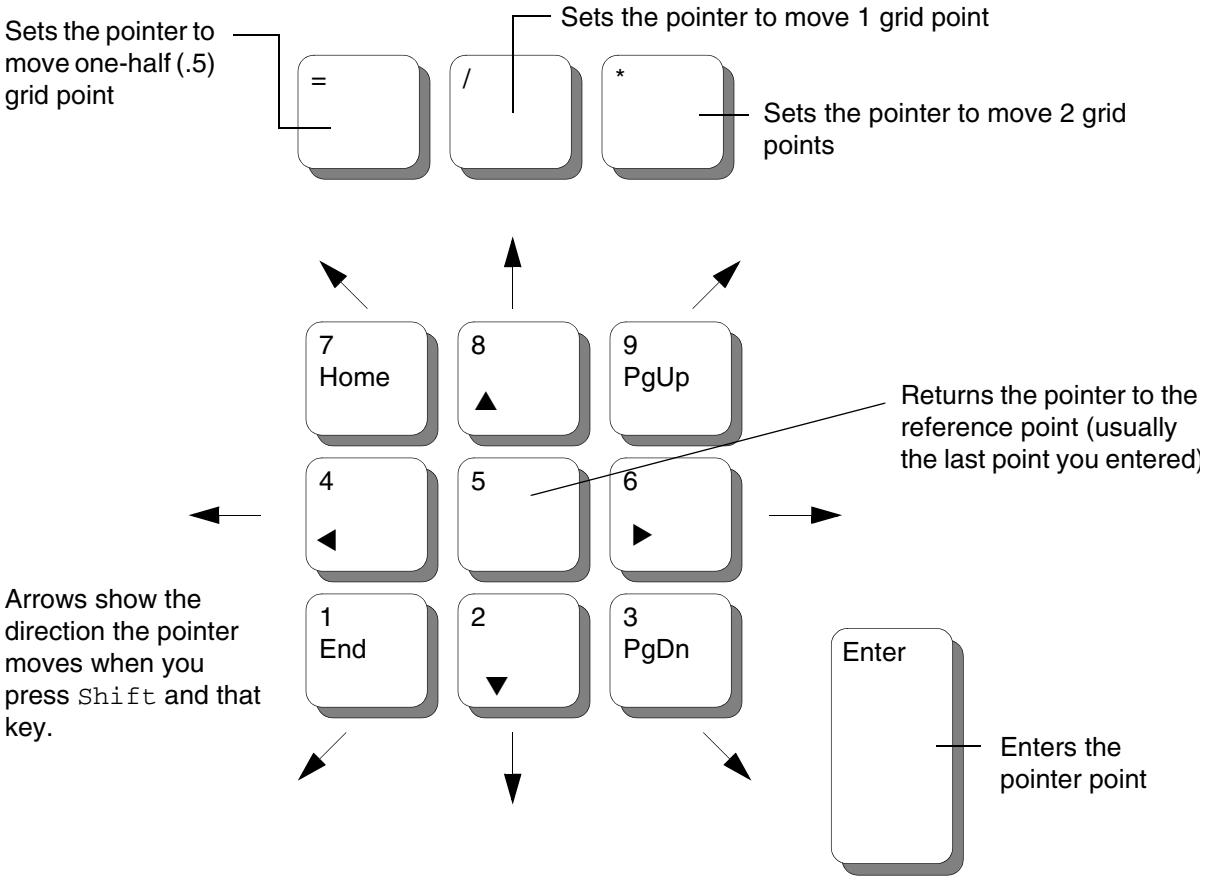
- By using the keypad to move the pointer on the canvas and enter the points
- By entering points in the CIW
- By entering points in the Enter Points form

You can use the following mouse actions for drawing a shape:

- To enter points for a shape or wire, click on the canvas at the starting point for the object.
- To complete a circle, rectangle, ellipse, or a donut, click again on the canvas to complete the last point of the shape.
- To complete a path, polygon, or wire, double-click on the canvas to enter the last point or click on the canvas and press `Return`.

To delete a point, you need to press `Backspace`.

You can use the keypad to move the mouse pointer and to enter points. This can be helpful if, for example, you want to enter a point just a few units away from the last point. The following figure illustrates a sample keyboard and the keys that you can use to enter the points.



© 2023 Cadence Design Systems, Inc.

Related Topics

Entering Points in the CIW to Create Objects

Entering Points Using the Enter Points Form

Points Formats Supported in the Enter Points Forms

Entering Points in the CIW to Create Objects

To create an object by typing coordinates in the CIW:

1. In the layout window, start the command for creating an object. For example, to create a rectangle, choose *Create – Shape – Rectangle*.

   You are prompted to enter the first corner of the rectangle.

2. In the CIW, type a coordinate pair and press `Enter`. Use a colon to separate the coordinates, for example `-0.2:-9.0`.

3. When prompted for the opposite corner of the rectangle, type another coordinate pair in the CIW and press `Enter`. For example, `0.5:20.0`.

   If you are creating a polygon, enter the same coordinates again to indicate the last point.

Related Topics

Object Creation by Entering Points

Entering Points Using the Enter Points Form

Points Formats Supported in the Enter Points Forms

Entering Points Using the Enter Points Form

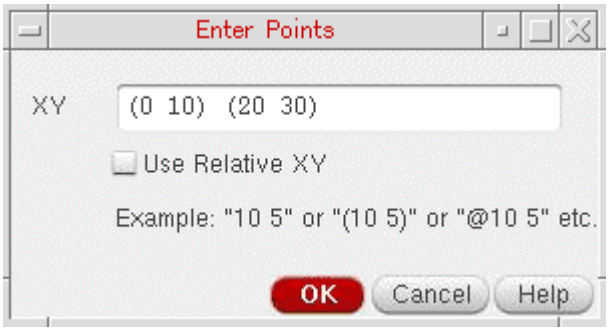
To specify the points to create an object using the Enter Points form:

1. In the layout window, start the command for creating an object. For example, to create a rectangle, choose *Create – Shape – Rectangle*.

2. Choose *Tools – Enter Points*.

© 2023 Cadence Design Systems, Inc.

The Enter Points form opens.



3. Enter points in the *XY* field. For example:

   ```
   (0 10) (20 30)
   ```

4. Click *OK*.

A rectangle is created with `(0 10)` as the first corner and `(20 30)` as the corner that is diagonally opposite to the first corner.

You can also use the *X*, *Y*, *dX*, and *dY* fields on the status toolbar to specify the points to create an object.

All point adjustment commands such as *DRD*, *Grid*, *Snap Mode*, *Acute Angle*, and *Gravity* are disabled when the points are entered using the Enter Points form. For example, the gravity settings are not applicable when you use the Enter Points form to specify points.

Video

For information about using the Enter Points form and the status toolbar to specify points for the create and edit commands, watch the video *Enter Points for Create and Edit Commands*. Access to the video depends on the availability of a web browser and a Cadence Online Support account.

Related Topics

Enter Points Form

Points Formats Supported in the Enter Points Forms

Object Creation by Entering Points

Entering Points in the CIW to Create Objects

© 2023 Cadence Design Systems, Inc.

Points Formats Supported in the Enter Points Forms

You can use the Enter Points form to specify the points to create an object. This is especially useful when you want to create an object or a list of objects at specific coordinates in the design window.

In the Enter Points form, you specify the points for an object in the *XY* field. The *Use Relative XY* field indicates that the points specified in the *XY* field are the points relative to the last entered points of the command (delta X and Y).

You can enter points in the *XY* field in the following formats. You can prefix or suffix `@` to a point to make it a relative point. The formats `@(x y)` or `@x y` or `x@ y` or `x @y` or `x y@` formats imply that `x y` are relative points.

Format	*Use Relative XY* **Field Deselected**	*Use Relative XY* **Field Selected**
`x y`	The points are added as absolute points.	The points are added as relative points.
■ `x1 y1 x2 y2 x3 y3` ■ `x1 y1,x2 y2, x3 y3` ■ `x1 y1;x2 y2; x3 y3` ■ `(x1 y1)(x2 y2)(x3 y3)` ■ `x1:y1 x2:y2 x3:y3` ■ `{x1 y1}{x2 y2}{x3 y3}` ■ `[x1 y1][x2 y2][x3 y3]` ■ `<x1 y1><x2 y2><x3 y3>`	The points are added as absolute points.	Points are added as per the following: ■ If a point has already been added to the command, the points in this list are added as relative points. ■ If a point has not been added to the command, a warning message is displayed in the CIW. No points are added to the canvas.

© 2023 Cadence Design Systems, Inc.

Format	*Use Relative XY* **Field Deselected**	*Use Relative XY* **Field Selected**
■ (*x1 y1*) @(*x2 y2*) (*x3 y3*) ■ *x1 y1 @x2 y2 x3 y3* ■ *x1 y1 x2@ y2 x3 y3* ■ {*x1 y1*} {*x2 @y2*} {*x3 y3*} ■ *x1 y1 x2 y2@ x3 y3*	(*x1 y1*) and (*x3 y3*) are added as absolute points and (*x2 y2*) are added as relative points.	All the points are added as relative points.

Caution

Combination of the formats listed in the table is not supported in Virtuoso Layout Suite. For example, {x1 y1}[x2 y2] is not supported as it uses a combination of curly and square brackets.

You can also enter points in the *XY* field for nested commands, for example, you run the `Rectangle` command and then run the `Ruler` command. Use the following format: (*x1 y1*)(*x2 y2*) (*x3 y3*) (*x4 y4*). The points (*x1 y1*) and (*x2 y2*) are used to create a ruler. The points (*x3 y3*) and (*x4 y4*) are used to create a rectangle.

Related Topics

Entering Points Using the Enter Points Form

Enter Points Form

Object Creation by Entering Points

Entering Points in the CIW to Create Objects

ROD Support for Object Creation

A relative object design (ROD) object contains information about an object, including its name and database ID.

© 2023 Cadence Design Systems, Inc.

Creating a ROD object lets you:

- Specify connectivity to associate the object with other objects on the same net by typing the net name in the *Net Name* field.
- Give an object a ROD name by either using the default database name or typing another name in the *ROD Name* field.

ROD names can contain spaces. For example, `rectangle 1a` is a valid ROD name. Only one ROD object can be created and assigned to a net at a time. You cannot enter a series of names in the *Net Name* and *ROD Name* fields. Once the object is created, you can edit the object by using ROD function commands in the CIW. These commands let you:

- Access ROD objects by name through all levels of hierarchy
- Access ROD objects' handle values through all levels of hierarchy
- Align ROD objects to each other or to specific coordinates
- Assign names to unnamed rectangles, polygons, lines, and paths
- Create complex parameterized cells

Using the commands that are not fully supported for ROD objects could cause the objects to lose the ROD information associated with them, changing the objects into ordinary shapes.

The following table summarizes the level of support for how creation commands work on ROD objects.

Create Command	**Level of ROD Support**
Rectangle	Create and name new rectangles as ROD objects using the *Create Rectangle* form.
Polygon	Create and name new polygons as ROD objects using the *Create Polygon* form.
Path	Create and name paths as ROD objects using the *Create Path* form.
Multipart Path	Create and name new multipart paths as ROD objects using the *Create Multipart Path* form. This form lets you choose a template from your technology file, load templates from an ASCII file, and save form values as a template in your technology file if you have write permission or in an ASCII file.

© 2023 Cadence Design Systems, Inc.

Label	You cannot create a label as a ROD object. However, you can make an existing label a ROD object by assigning it a name with the rodNameShape function.
Instance	An instance is automatically a ROD object because it has a unique name. The ROD object name is the same as the instance name.
Pin	Create and name new pins as ROD objects using the *Create Pin* form.
Pins From Labels	You cannot create a pin from a label as a ROD object. However, you can make an existing pin a ROD object by assigning it a name with the `rodNameShape` function.
Circle	Circles are supported as ROD objects. However, there is no user interface for naming a circle; use the `rodNameShape` function.
Ellipse	Ellipses are supported as ROD objects. However, there is no user interface for naming an ellipse; use the `rodNameShape` function. An ellipse ROD object has only bounding box handles.
Donut	Donuts are supported as ROD objects. However, there is no user interface for naming a donut; use the `rodNameShape` function. A donut ROD object has only bounding box handles.
Layer Generation	You cannot create a shape using a layer generation operation as a ROD object. However, you can make a generated shape a ROD object by assigning it a name with the `rodNameShape` function.

ROD functionality is not implemented for mosaics. Certain `dbCreate` functions, such as `dbCreateInst`, create named objects and thus associate ROD IDs with the objects. You can use the rodGetObj function to fetch the ROD ID of such objects.

You can use the environment variable rodAutoName to set system default values in the Create Pin, Create Rectangle, Create Path, and Create Polygon forms to turn on the *Create as ROD object* option so all shapes are created as ROD objects. In the *Name* field, you can then specify a name or use the default name.

You can also set the `rodAutoName` variable in your `.cdsenv` file, so the layout editor defaults to the options you want at startup. For example, the syntax to use in your `.cdsenv` file is:

```
layout rodAutoName string "rectangle pin polygon"
```

Related Topics

Relative Object Design Concepts

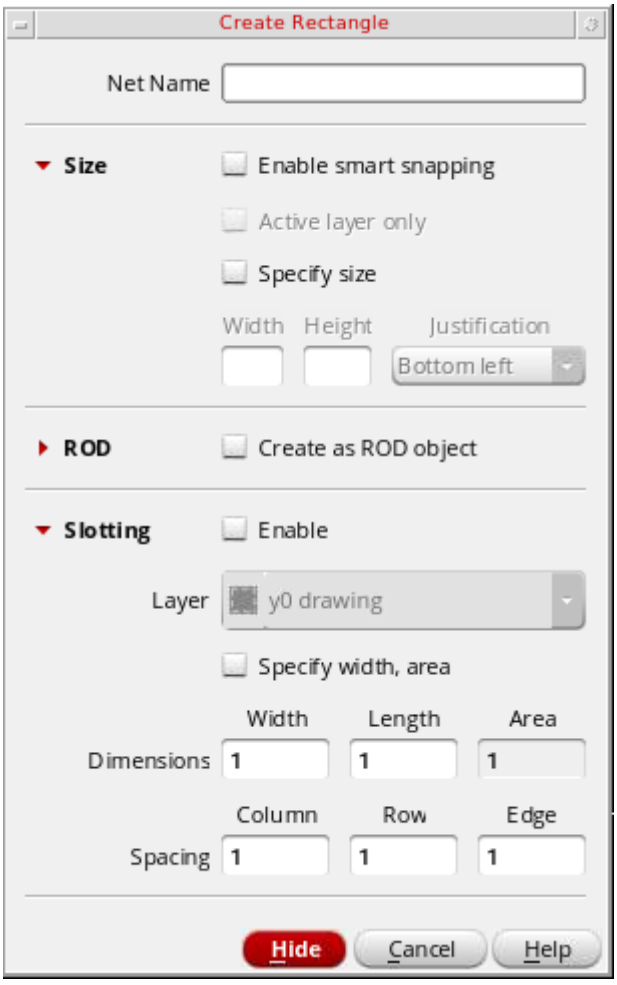
© 2023 Cadence Design Systems, Inc.

Creating a Rectangle

Using the *Create Rectangle* command, you can create rectangular shapes and rectangles with slots.

To create a rectangle:

1. In the layout window, from the Layers panel, select the layer on which you want to draw a rectangle.
2. Choose *Create – Shape – Rectangle*.
3. Press `F3` to open the Create Rectangle form.



4. In *Net Name*, type the name of the net with which you want to associate the rectangle.
5. In the *ROD* section, select the *Create as ROD object* check box if you want to create a ROD object rectangle; otherwise, the rectangle is created as a polygon.

© 2023 Cadence Design Systems, Inc.

In the *Name* field that now becomes editable, change the default rectangle name, if required.

6. In the *Slotting* section, select the *Enable* check box if you want to create the rectangle with slots.
7. Select a layer-purpose pair from the *Layer* list.

   This list is available only if the *Create as ROD object* check box is selected in the *ROD* section.
8. Specify the slot dimensions in the *Width* and *Length* or *Width* and *Area* fields.

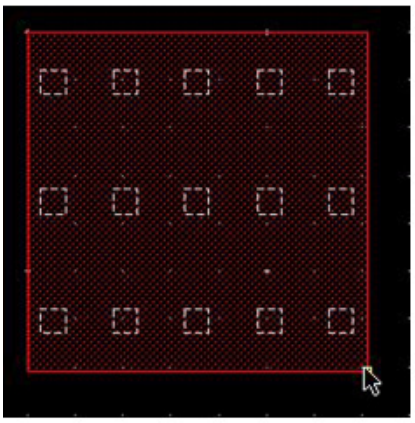
   If you deselect *Specify width, area*, the *Width* and *Length* text fields are editable. If you select *Specify width, area*, the *Width* and *Area* fields are editable.
9. Specify the spacing in the *Column*, *Row*, and *Edge* fields.
10. Click on the canvas to specify the first corner of the rectangle.

    Alternatively, you can specify the points of the rectangle using the Enter Points form.

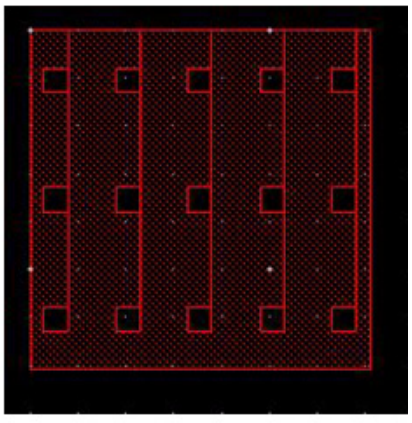
    Click again on the canvas to specify the corner of the rectangle that is diagonally opposite to the first corner that you specified.

    You cannot create a rectangle with zero area. If you click twice at the same point to specify the two corners of the rectangle that you are creating, a warning message is displayed in the CIW.

    If you are creating a rectangle with slots, ensure that you create a rectangle large enough to contain the slots with the specified dimensions; otherwise, slots are not created. Slots are added dynamically as you create the rectangle.



Dynamic Slot Generation While Creating the Rectangle



Resultant Rectangle with Slots

© 2023 Cadence Design Systems, Inc.

You can create a rectangle of a single track height when the snapping mode for a layer is set to `singleTrackCenter`. In this mode, a shape is snapped only if it can be centered on a single track while meeting the enclosure rule on both sides.

Related Topics

Create Rectangle Form

Enter Points Form

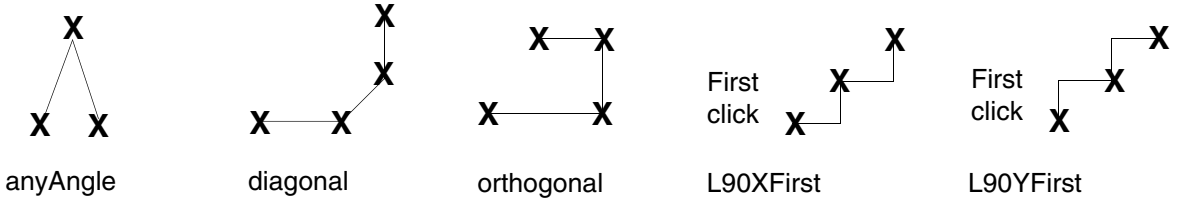
Rectangles Created on a Layer with singleTrackCenter Snapping Mode

Creating a Polygon

To create a polygon:

1. In the layout window, from the Layers panel, select the layer on which you want to draw a polygon.
2. Choose *Create – Shape – Polygon*.
3. Press `F3` to open the Create Polygon form.
4. Select a snap mode from the *Snap Mode* list to specify how the pointer snaps when you create a polygon.

   The following figure illustrates snap mode settings. If you chose an L90 mode, you can switch to the other L90 mode by clicking the middle mouse button. Keep clicking the middle mouse button to toggle between *L90XFirst* and *L90YFirst.*



4. In *Net Name*, type the name of the net with which you want to associate the polygon.
5. Select *Create as ROD object* if you want to create a ROD object polygon.

   The *Name* field becomes editable, displaying the default name assigned to the polygon. Change the default name if required.

© 2023 Cadence Design Systems, Inc.

6. Click on the canvas to enter the points. As you enter each point, a dotted line shows how the layout editor will close the polygon if you stop at that point.

   Alternatively, you can specify the points of the polygon using the *Enter Points* form.

7. Double-click on the canvas to finish drawing the polygon.

Related Topics

Create Polygon Form

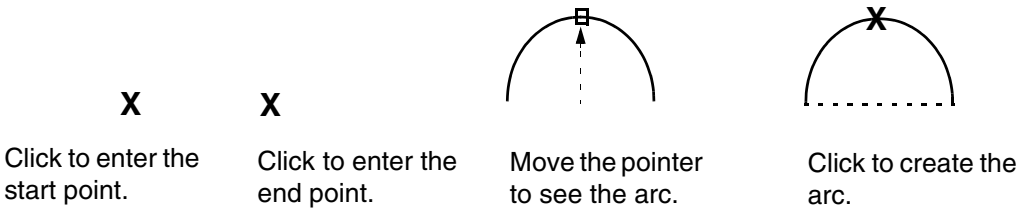
Enter Points Form

Creating an Arc in a Polygon

To create an arc in a polygon:

1. In the layout window, from the Layers panel, select the layer on which you want to create an arc.
2. Choose *Create – Shape – Polygon*.
3. Press `F3` to open the Create Polygon form.
4. Select the *Add an arc segment* check box.
5. Specify the net name and ROD settings as required.
6. Click on the canvas to indicate the start point, the end point, and a radius point on the arc.

   Alternatively, you can specify the points of the arc using the Enter Points form.



   If you have already entered some points of a polygon, the last point you clicked is the start point of the arc.

You can also use the *Modify Corner* command to curve the corner of a polygon.

© 2023 Cadence Design Systems, Inc.

Related Topics

Create Polygon Form

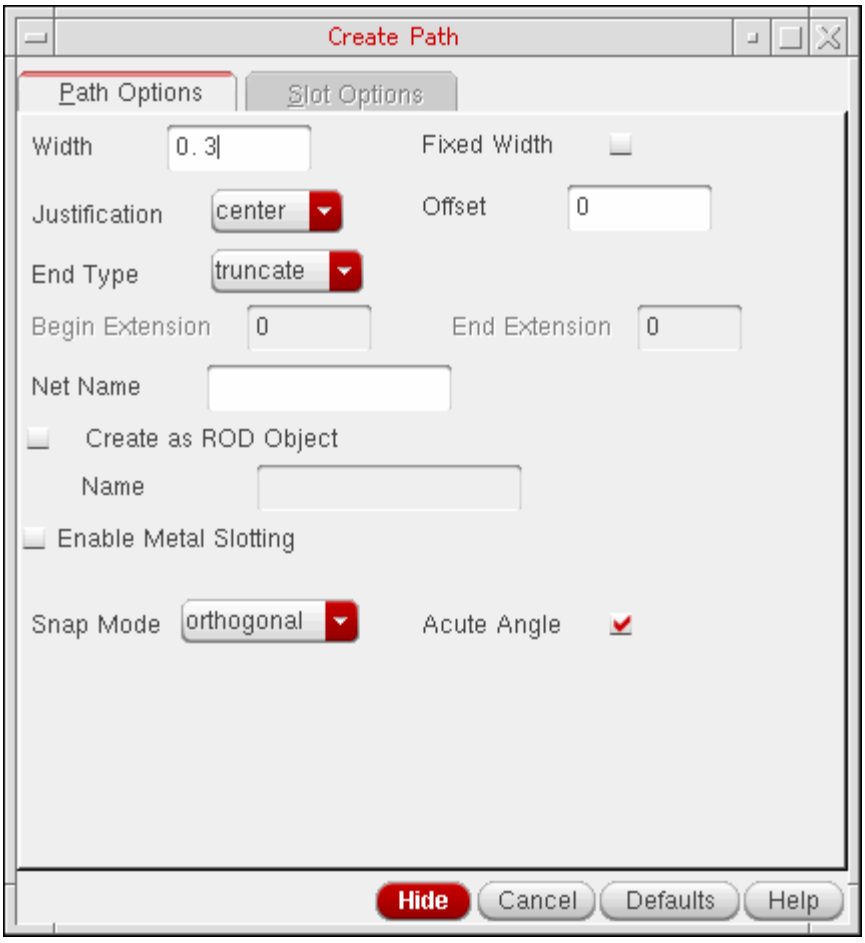
Enter Points Form

Modifying Corners of a Polygon

Creating a Path

To create a path:

1. In the layout window, from the Layers panel, select the layer on which you want to create a path.
2. Choose *Create – Shape – Path*.
3. Press `F3` to open the Create Path form.



© 2023 Cadence Design Systems, Inc.

4. On the Path Options tab, specify the *Width* for the path.

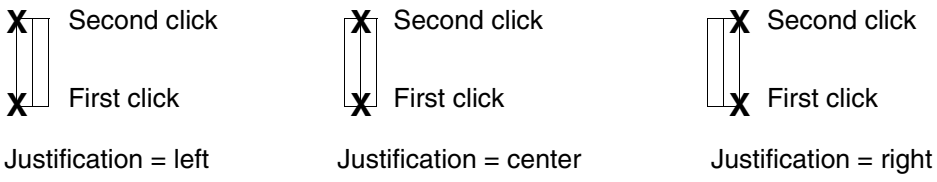
   The default path width is the `minWidth` for the entry layer defined in the technology file.

5. Select the *Fixed Width* check box if you want a fixed width for the path.

   If this check box is selected, the specified width is applied to all the segments of the path, even when you change layers.

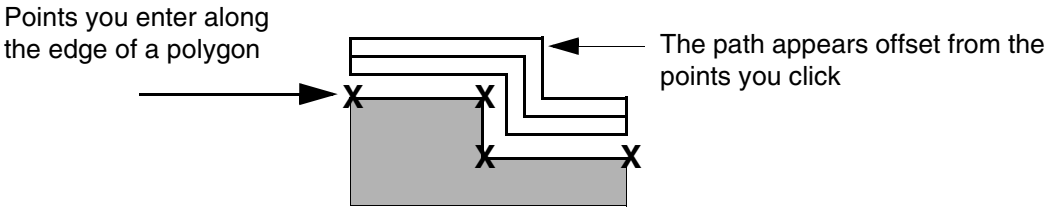
6. From *Justification*, select the edge of the path you want to offset.

   The path justification setting controls where you can enter the points—along the centerline or along the edge of a path.



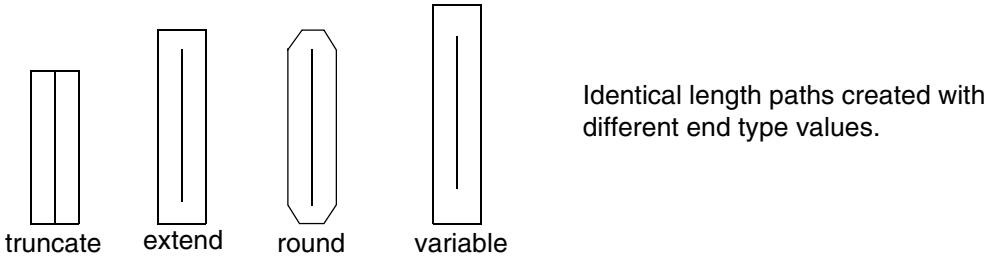
7. Type a value in the *Offset* field if you want to offset the path.

   For example, if you type a value of `0.5`, the path is created at a distance of `0.5` microns from the points you enter.



8. Select the required end type for the path.

   The path *End Type* setting changes the shape and spacing of path ends. If you set the end type as *variable*, specify the *Begin Extension* and *End Extension* values.



© 2023 Cadence Design Systems, Inc.

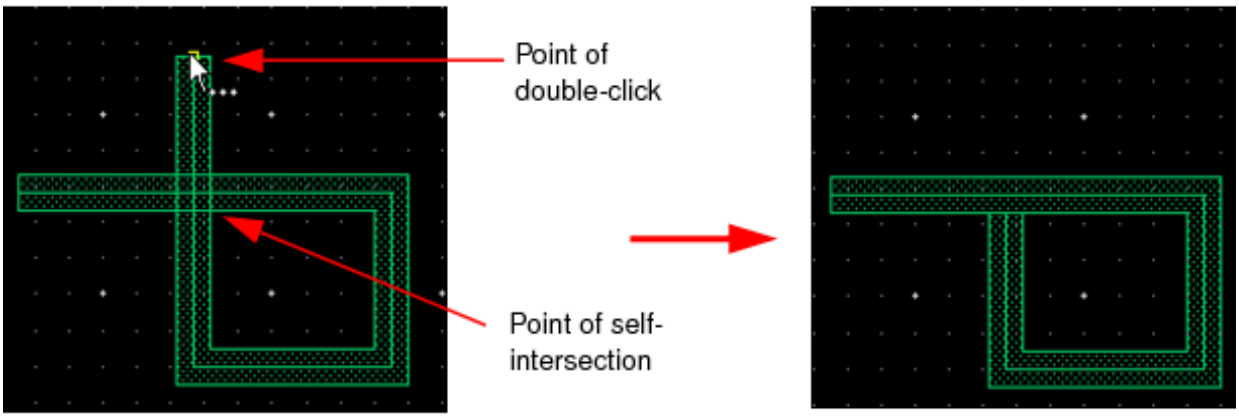
9. Specify the *Net Name* with which you want to associate the path.
10. Select the *Create as ROD Object* check box if you want to create a ROD object path; otherwise, the path is created as a polygon.

    In the *Name* field that now becomes editable, change the default path name, if required.
11. Select a value from the *Snap Mode* list.

    Set *Snap Mode* to *anyAngle* and select the *Acute Angle* check box if you want to create paths with acute angles.
12. Click on the canvas to specify the path endpoints. Segments are created when you change the direction of the path.

    Alternatively, you can specify the points of the path using the Enter Points form.
13. Double-click on the canvas or press `Enter` to finish creating the path.

    If the path you create is self-intersecting, every single-click after a self-intersection generates a message in the CIW, though you can continue to create the path. You can cancel the current and previous points by pressing the `Backspace` key or finish creating the path by double-clicking or pressing `Enter`. If you double-click or press `Enter`, the path is automatically truncated to a valid path. All the segments or points after the first intersection point are removed so that the remaining path shape is valid. A message appears in the CIW indicating the same.

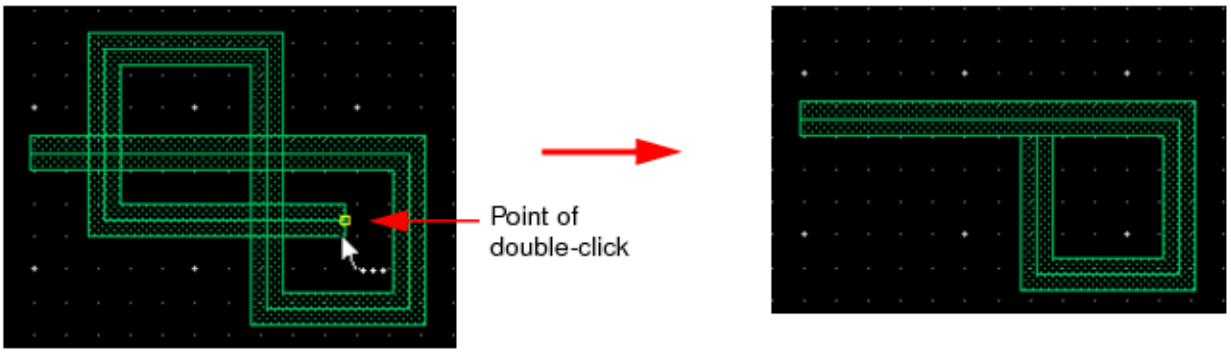


Self-intersecting path

Path truncated at the point of
intersection to create a valid pa

© 2023 Cadence Design Systems, Inc.

Even if there are multiple intersection points, the path is truncated to the first self-intersection point, as shown in the following figure.



Self-intersecting path

Path truncated at the first point of intersection to create a valid path

Related Topics

Create Path Form

Creating a Path with Slots

Controlling the Display of Paths

Enter Points Form

Creating a Path with Slots

To create a path with slots:

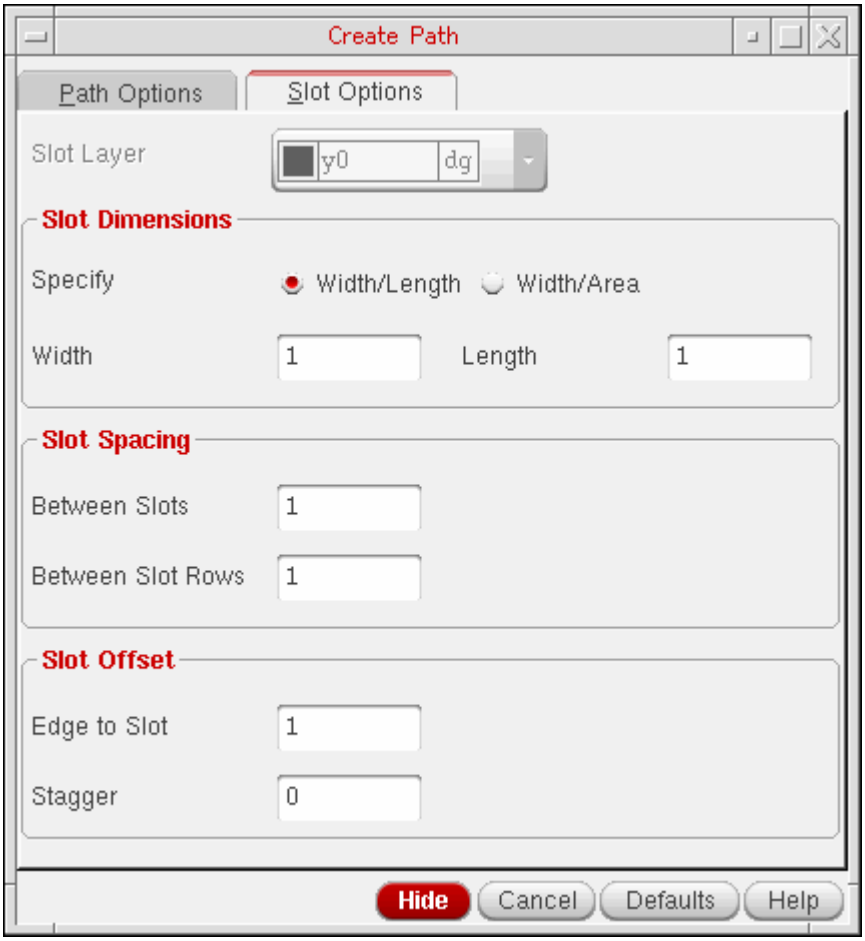
1. In the layout window, from the Layers panel, select the layer on which you want to create a path.
2. Choose *Create – Shape – Path*.
3. Press `F3` to open the Create Path form.
4. On the *Path Options* tab, specify the settings such as the path width, justification, end type, and net name, that you want to use to create the path.
5. Select the *Create as ROD Object* check box.

   In the *Name* field that now becomes editable, change the default path name, if required.
6. Select the *Enable Metal Slotting* check box to create a path with slots.

© 2023 Cadence Design Systems, Inc.

This makes the Slot Options tab available.

7. Open the *Slot Options* tab.



8. Select a layer-purpose pair from the *Slot Layer* list.

   This list is available only if the *Create as ROD Object* check box is selected on the *Path Options* tab.

9. Select *Width/Length* or *Width/Area* in the *Slot Dimensions* section.

   If you select *Width/Length*, the text fields available are *Width* and *Length*. If you select *Width/Area*, the *Length* field is replaced with the *Area* field.

10. Specify the slot dimensions in the *Width* and *Length* or *Width* and *Area* fields.

11. In the *Between Slots*, *Between Slot Rows*, *Edge to Slot*, and *Stagger* fields, specify values as required.

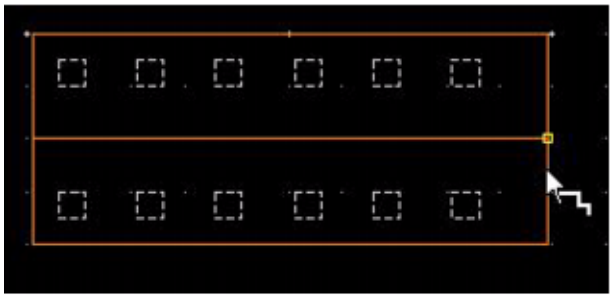
© 2023 Cadence Design Systems, Inc.

12. Click on the canvas to specify the path endpoints. Segments are created when you change the direction of the path.

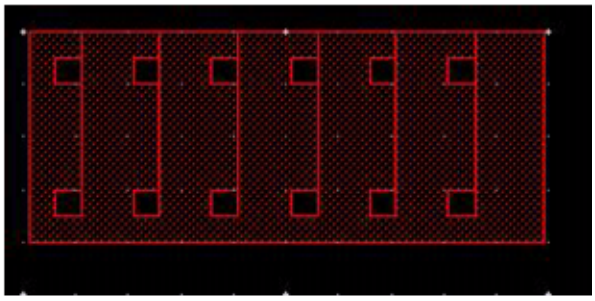
Alternatively, you can specify the points of the path using the Enter Points form.

13. Double-click on the canvas or press `Enter` to finish creating the path.

While creating a path with slots, ensure that you create a path wide enough to contain the slots with the specified dimensions; otherwise, slots are not created. Slots are added dynamically as you create the path.

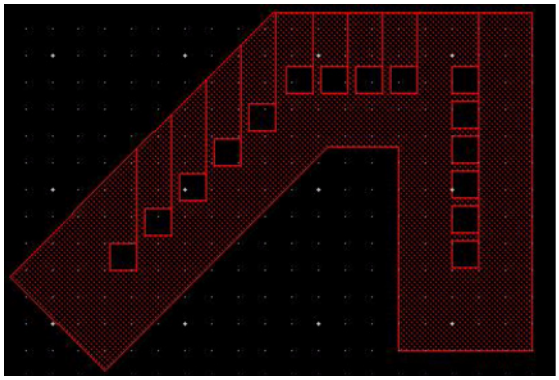


Dynamic Slot Generation While Creating the Path



Resultant Path with Slots
(No Stagger)

You can also create slots on *diagonal* paths, as shown in the following figure. Slots are not created on *anyAngle* paths.



Slots are not supported in geometric wires created from paths. For information about how to create wire interconnects, see Creating Wires in the *Virtuoso Interactive and Assisted Routing User Guide.*

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

Enter Points Form

Creating a Path

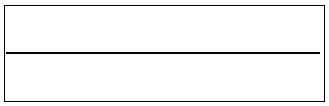
Controlling the Display of Paths

Controlling the Display of Paths

You can display path borders and centerlines, borders only, or centerlines only.

1. In the layout window, choose *Options – Display*.

   The Display Options form appears.
2. In the Display Control section, select one of the following from the *Path Display* list:
   - *Borders and Centerlines*: Displays both the path centerline and its outer edges.
   - *Borders only*: Displays only the path borders and eliminates the centerline display.
   - *Centerlines only*: Displays only the centerline.
3. Click *OK*.



A path displayed with *Borders and Centerlines* enabled



The same path with *Borders only* enabled

The same path with *Centerlines only* enabled

Related Topics

Display Options Form

Creating a Path

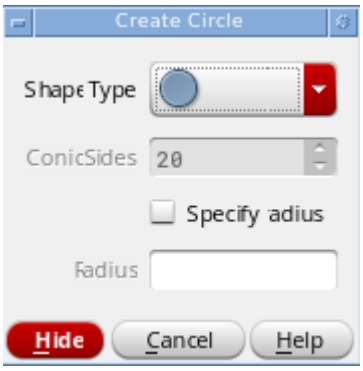
Creating a Circle

To create a circle:

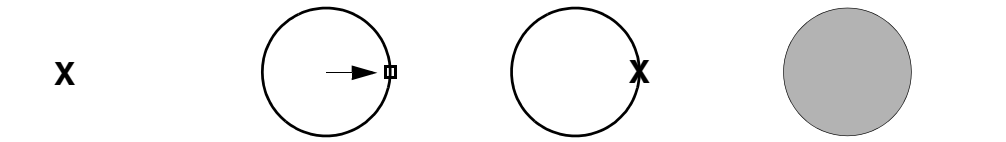
1. In the layout window, from the Layers panel, select the layer on which you want to create a circle.

© 2023 Cadence Design Systems, Inc.

2. Choose *Create – Shape – Circle*.
3. Press `F3` to open the Create Circle form.



4. Select the shape type for the circle that you want to create.
5. In *ConicSides*, specify the number of segments to be used if you are creating a half or quarter circle.
6. Enable the *Specify Radius* check box and specify the radius of the circle. Alternatively, you can specify the radius by clicking at the required point on the canvas.
7. Click on the canvas to define the edge of the circle.



Click the center | Move the mouse pointer | Click to finish the circle | The finished circle

Related Topics

Create Circle Form

Creating an Ellipse

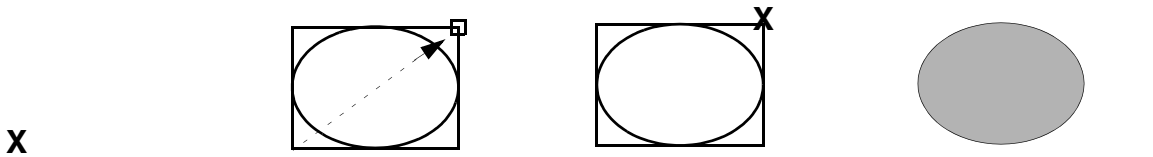
You create an ellipse by creating a rectangle that surrounds the ellipse.

© 2023 Cadence Design Systems, Inc.

To create an ellipse:

1. In the layout window, select the layer from the Layers panel on which you want to create an ellipse.
2. Choose *Create – Shape – Ellipse*.
3. Click at a point on the canvas where you want the first corner of the bounding box of the ellipse.

   Alternatively, you can specify the points of the ellipse using the Enter Points form.
4. Click at a point on the canvas where you want the opposite corner of the bounding box of the ellipse.



Click the first corner | Move the mouse pointer | Click the opposite corner | The finished ellipse

(Virtuoso RF Option) The *Create Ellipse* command does not support curved shapes.

Related Topics

Enter Points Form

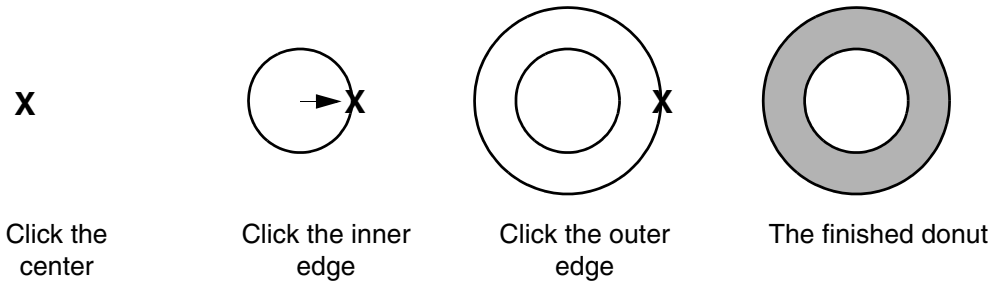
Creating a Donut

To create a donut:

1. In the layout window, from the Layers panel, select the layer on which you want to create a donut.
2. Choose *Create – Shape – Donut*.
3. Click at a point on the canvas where you want the center of the donut. Alternatively, you can specify the points of the donut using the Enter Points form.
4. Click at a point where you want the inner edge of the donut.

© 2023 Cadence Design Systems, Inc.

5. Click at a point where you want the outer edge of the donut.



Related Topics

Enter Points Form

Creating Instances

An instance is a database object that represents a master cellview. You can have several instances of the same cellview in a design. Use the *Create Instance* command to place an instance of a cell into the current cellview.

By default, the layout editor automatically assigns names to instances, starting with the first instance as `I1` and continuing with `I2`, `I3`, and so forth. If the instances are mosaics, the default names are `M1`, `M2`, and so forth. If you override the assigned name, the first name in the form is assigned to the first instance you place. The next name is assigned to the next instance you place, and so forth. You can use the Display Options form to display instance names in the cellview.

In case the design has Embedded Module Hierarchy (EMH), you can specify the complete hierarchical path name, starting from the top module to the new instance to be created using the hierarchy delimiter in CDB name space, for specifying the location in EMH where the instance is to be created. An example of such a name is `modA'modB'15`, where `A` and `B` are the names of the module instances and `15` is the name of the newly created instance.

To place a cell inside the current cellview:

1. In the layout window, choose *Create – Instance* or click the *Create Instance* icon  on the *Create* toolbar.

   Alternatively, right-click at any free space on the design canvas to display the *Layout* shortcut menu and select *Create – Instance.*

   The Create Instance form appears.

© 2023 Cadence Design Systems, Inc.

2. Fill out the *Library*, *Cell*, and *View* fields to specify the name of the master cell you want to place as an instance in the cellview.
3. To override the automatic naming of instances, type one or more instance names in the *Names* field, leaving a space between names.

   Do not use the character forward slash (`/`) in instances names. If used, it gets replaced with its octal representation `#2f`. Therefore, do not use a name such as `I1/I2` for an instance.
4. Select the *Physical Only* check box to create the instance in such a way such that it exists only in the block domain and not in the module domain.
5. Optionally, you can create a mosaic or array of instances by typing values in the *Rows* and *Columns* fields in the *Mosaic* section.
6. To change the spacing between the rows or columns, type values for *Delta Y,X*.

   By default, the fields are set to the width and length of the original cell, so that the edges of each instance touch but do not overlap.
7. To rotate the instance, use the *Rotate Left* or *Rotate Right* icons in the form, or click the middle mouse button as you place the instance.
8. To mirror the instance, use the *Flip Vertically* or *Flip Horizontally* icons in the form. Alternatively, as you place the instance, press `Ctrl` and click the middle mouse button to mirror the instance horizontally or press `Shift` and click the middle mouse button to mirror the instance vertically.
9. Click on the canvas to place the instance. Alternatively, you can specify the points where you want to place the instance using the Enter Points form.

If you specify the *Description* field in the Edit CDF form for a cell with Component Description Format (CDF) parameters, the description is displayed as a tooltip on the mouse over of that field.

Related Topics

Create Instance Form

Enter Points Form

Edit CDF Form

Display Options Form

© 2023 Cadence Design Systems, Inc.

Pin Creation

You can create pins in a cellview to connect the cellview to another cell in your design hierarchy. Use the *Create Pin* command to create pins and define the connectivity of pins as strong or weak.

For defining or editing the connectivity of existing pins in a design, use the *Connectivity – Pins* command. You can define the connectivity as strong connect, weak connect, must-join connect, or pseudo parallel connect.

You can apply net expression properties to pins.

Related Topics

Creating Pins

Placing a Pin Array

Creating Pins from Labels

Pin Connectivity

Inherited Connections

Creating Pins

To create pins:

1. In the layout window, choose *Create – Pin*.

   Alternatively, click the *Create Pin* icon  on the *Create* toolbar or right-click any free space on the design canvas to display the *Layout* shortcut menu and select *Create – Pin.*

   The Create Pin form appears.

2. Set *Mode* as *Manual*.
3. Set *Connectivity* to specify the type of pins you want to create.
   - Select *Strong* to create multiple pins as strongly connected.
   - Select *Weak* to create multiple pins as weakly connected.

© 2023 Cadence Design Systems, Inc.

Select *Must Connect All Pins* to apply the `LEF58_MUSTJOINALLPORTS` property to all newly created pin terminals in the layout. This implies that the router must connect all pins of the selected terminal.

4. Specify one or more terminal names in the *Terminal Names* field. This is the name of the net to which the pins will connect. Separate multiple names by using spaces.

   The behavior of the *Terminal Names* field depends on the selection of the *Keep First Name* check box. The following scenarios may occur when *Keep First Name* is off:

   a. If you specify a single name in *Terminal Names*, a single new terminal and pin are created. If the specified terminal already exists, a new pin is added to the terminal.

   b. If you leave *Terminal Names* blank, you are prompted to either specify a terminal name or to click an existing pin. If you click an existing pin, the terminal name of the pin populates the *Terminal Names* field in the form and a pin figure is added to the existing pin.

   c. If you specify multiple names in *Terminal Names*, new terminals with the specified names are created if they do not already exist. New pins are created on each of the new terminals.

5. Select *Physical Only* to create the terminal only in the block domain.

6. Select *Keep First Name* for adding multiple pin figures to the same pin shape on the same terminal. This option works in conjunction with the *Terminal Names* field. The following scenarios may occur when *Keep First Name* is on:

   a. If you specify a single name in the *Terminal Names* field, the behavior is the same when *Keep First Name* is off, that is, a new terminal and pin are created, if they do not already exist. Multiple pin figures can be added to this new pin shape.

   b. If you specify multiple names in the *Terminal Names* field, only a single terminal is created with the first specified name. Multiple pin figures are added to the same pin shape and they are connected strongly or weakly, depending on the *Connectivity* type you selected.

7. Edit the X and Y pitch values to set the distance between the center points of the pins, if required.

8. Select the *Create Label* check box to attach a label to the pin showing the terminal name.

   For more information, see Setting Pin Label Text Style Options.

9. Select *Create as ROD Object* to create the pin figure as a ROD object, else the pin figure is created as a polygon.

© 2023 Cadence Design Systems, Inc.

The *Name* field becomes editable, allowing you to update the name from the default that is displayed. The default name is derived based on the *Mode* you select.

10. Set *Pin Figure* to specify the shape of the pin.

When creating a rectangle, dot, polygon, or circle, the system prompts you to enter a point for the shape. To create a zero-size dot pin, click twice on the same point.

11. Select *Attach Pin to Overlapping Shape on Same Layer* to attach the pin shape to an overlapping shape present on the same layer, the purpose can be different. This option is useful to ensure that the attached pin moves when the parent shape is moved and gets deleted when parent shape is deleted. This option works only when there is a single overlapping shape. If there are more than one overlapping shapes, none of the shapes are attached and a warning message is displayed.

12. Select an *I/O Type* to indicate the direction of the signal into or out of the pin.

13. If you are creating a polygon pin, set *Snap Mode* to specify how the pointer snaps to the pin.

14. Select *Access Direction* to specify the direction in which the routing tools can connect routing to the pin.

15. Set the *Signal Type* of the net that is used for creating the pin.

16. Click on the canvas to create the pin.

17. If you selected the *Create Label* check box, click again to place the label on the pin figure.

18. If you are creating multiple pins, continue to click on the canvas to create and place pins and labels. If *Keep First Name* is not selected or you specified a single name in the *Terminal Names* field, specify a terminal name in the form or click an existing pin to pick the terminal name.

19. Click *Cancel* in the *Create Pin* form or press `Esc` when you are done.

If you are creating multiple pins, pin figures are added to the same first specified terminal and the pin figures share the pin name. If you are creating a single pin, the pin is added to the specified terminal name. Pin names are auto-generated.

Note: When the connectivity is set to strong, and a terminal name is entered in the create pin form, a new pin is created. This new pin is strongly connected to itself but weakly connected to other pins on the same terminal. For creating a new pin strongly connected to another pin in the terminal, you need to click the existing pin shape on the canvas.

© 2023 Cadence Design Systems, Inc.

Related Topics

Create Pin Form

Placing a Pin Array

Setting Pin Label Text Style Options

Support for Orthogonal WSP Grid

Creating Pins from Labels

Placing a Pin Array

A pin array is a line of pins with a single origin. You might create such an array if you have many nets bundled together into a bus. The pins in the array are placed for every bit of the bus and are spaced according to the *XPitch* and *YPitch* coordinates.

To place a pin array:

1. In the layout window, choose *Create – Pin*.

   The Create Pin form appears.

2. Type the net bus name in the *Terminal Names* field as follows: *baseName*<*x*:*x*>

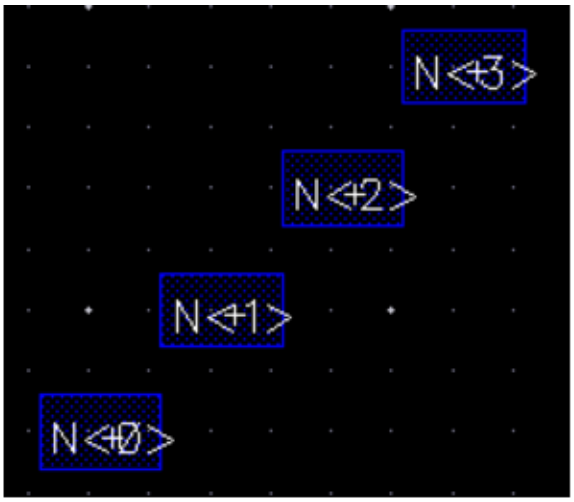
   For example, `NetA<0:5>`

3. Type the horizontal and vertical pitch for the array in the *X Pitch* and *Y Pitch* fields.

4. Move the pointer on the canvas and place the first pin where you want to start the array. The rest of the pins are automatically placed at the designated X and Y locations for every bit of the bus. They are measured from the center point of the previously placed pin.

© 2023 Cadence Design Systems, Inc.

The following figures show the examples of pin arrays with different X and Y pitch.



Terminal Names = N<0:3>
XPitch = 2
YPitch = 2



Terminal Names = A<0:3>
XPitch = 4
YPitch = 0

Related Topics

Create Pin Form

Creating Pins

Setting Pin Label Text Style Options

Support for Orthogonal WSP Grid

Creating Pins from Labels

Setting Pin Label Text Style Options

To specify pin name characteristics:

1. In the layout window, choose *Create – Pin*.

   The Create Pin form appears.
2. Select the *Create Label* check box.

© 2023 Cadence Design Systems, Inc.

3. Click the *Options* button.

   The Set Pin Label Text Style form appears.

4. Select the font height and style.



5. Select the text options.

   - Turning *Drafting* on prevents the label from being rotated more than 90 degrees.
   - The *Overbar* option determines how the underscore character is displayed.



6. Select a valid layer name.
7. Select a valid layer purpose.

   You must select a valid combination of layer name and purpose. In case, it is not a valid combination, an error message is displayed in the CIW while creating the text display.

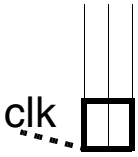
8. Select the justification for the pin name origin.

   The origin appears as a small square on the name when placed or selected.

9. Select the orientation of the pin name.

© 2023 Cadence Design Systems, Inc.

Once the pin is placed, an outline of the pin name appears.



The dotted line shows that the name will be attached to the pin

Related Topics

Create Pin Form

Set Pin Label Text Style Form

Creating Pins

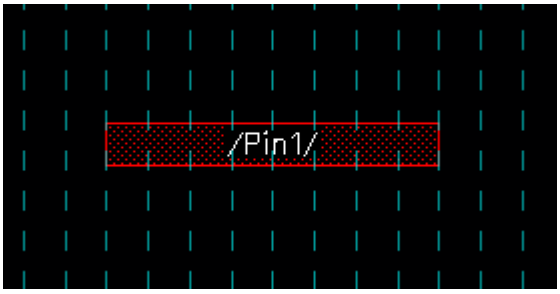
Placing a Pin Array

Support for Orthogonal WSP Grid

Creating Pins from Labels

Support for Orthogonal WSP Grid

While creating a pin, the orthogonal width spacing pattern (WSP) grid constraint defined in the technology file automatically adjusts the pin with a variable extension value so that the start and end points of a pin are on the grid.



The constraint group is specified in the technology file as follows:

```
constraintGroups(
;( group                [override]           [definition]          [operator] )
;( -----    ---------- ------------              ---------- )
( "orthogonalWSPGridCG"          nil     nil     'and
```


© 2023 Cadence Design Systems, Inc.

```
spacings(
     ( orthogonalWSPGrid          "Metal1"   "M2WSP" )
     ( orthogonalWSPGrid          "Metal2"   "M5WSP" )
     ( orthogonalWSPGrid          "Metal3"   "M4WSP" )
    ) ;spacings
  ) ;orthogonalWSPGridCG
) ;constraintGroups
```

If the orthogonal WSP grid constraints are part of a specific constraint group, you can select the constraint group from the *Default Wire Constraint Group* field in the *Wire Editing* section of the Layout Editor Options form.

Related Topics

Layout Editor Options Form

Pin Creation

Creating Pins from Labels

You can create pins from labels and text displays in your layout cellview by using the *Create Pins From Labels* command. You can specify the pin dimensions and the layer on which each pin is created. The generated pins are centered on the origin of the label or text display from which they are derived and have terminal names matching those on the corresponding labels or text displays.

If the text label contains a colon (:), then the colon and trailing text are dropped from the name of the generated pin. For example, for the text label `VDD:P,28`, the pin name is generated as `VDD`.

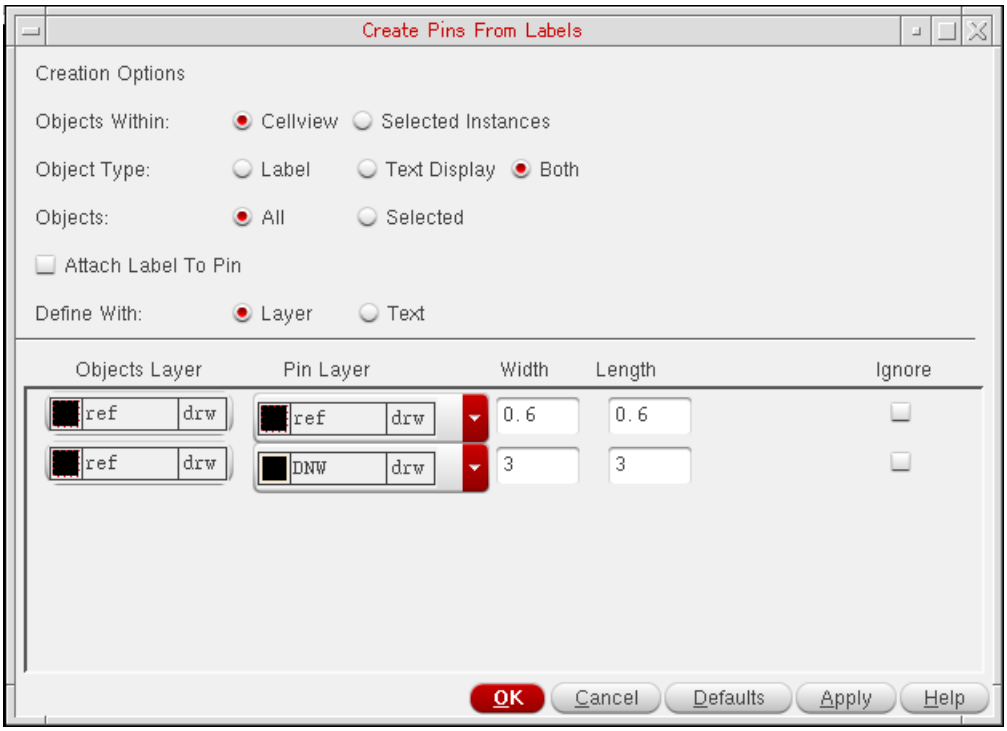
If a pin already exists for a selected label or text display, no new label is created for that object. The pin and the object are not attached to each other. If you move the pin, the corresponding label or text display do not follow. You can choose to attach an object to the pin.

To create a pin from a label or text display:

1. In the layout canvas, select the labels or text displays from which you want to create pins.
2. Choose *Tools – Create Pins From Labels*.

© 2023 Cadence Design Systems, Inc.

The Create Pins From Labels form opens.



3. Use *Objects Within* to specify whether to create pins at the top level of the cellview or within selected instances.
4. Use *Object Type* to specify whether to create pins from *Label*, *Text Displays*, or *Both*.
5. Use *Objects* to specify whether to create pins for all the objects of the specified type or only those that are currently selected.
6. Select *Attach Label to Pin* to attach a label to the pin.
7. From *Define With*, choose whether you want to define pin details using layer or text.
8. From *Pin Layer*, choose the layer on which each pin is to be created.

   If you selected labels on more than one layer, choose the appropriate pin layer for each label layer from the list of available layers.
9. Type the *Width* and *Length* of pins you require.
10. Click *OK* or *Apply*.

    The pins are created and placed at the center of the labels on the canvas.

© 2023 Cadence Design Systems, Inc.

Related Topics

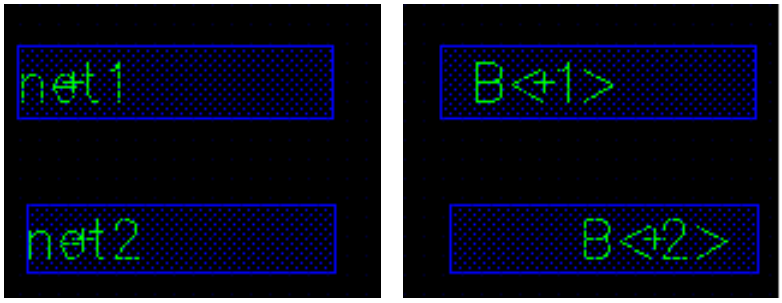
Parent-Child Relationship Between Objects

Create Pins From Labels Form

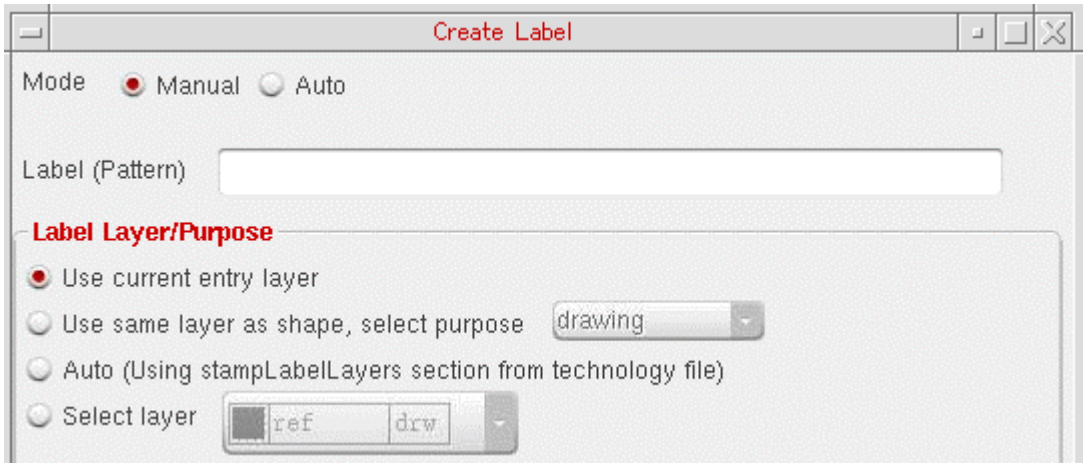
Pin Creation

Label Creation

You can identify objects or portions of your design by labeling the objects.



Use the *Create – Label* command to label objects. You can use the Create Label form to specify the settings for generating labels.



You can create labels in manual and auto modes. In manual mode, you can define the pattern that you want to use for generating labels. In auto mode, you can enable label creation from connectivity information and instances. You can also specify whether the object is created as a label or text display.

© 2023 Cadence Design Systems, Inc.

To place labels, you can use any of the following methods:

- Clicking the objects one by one to label them. This method is available in manual and auto modes.
- Drawing a scan line to label multiple objects in one go. This method is available in manual and auto modes.
- Using auto step to create labels automatically with the fixed, specified distance between consecutive labels. This method is available only in manual mode.
- Selecting shapes fully or partially, before or after invoking the *Create – Label* command. Labels are generated for fully or partially selected shapes that have connectivity information or are instances. This method is available only in auto mode.

Related Topics

Label Creation Patterns in Manual Mode

Configuring Label Creation Options

Generating Labels Using the Scan Line

Generating Labels Using Auto Step in Manual Mode

Label Creation in Manual and Auto Modes

Label Generation for Overlapping Shapes

Create Label Form

Label Creation Patterns in Manual Mode

In the Create Label form, use the *Label (Pattern)* field to specify the text or indexed pattern to be used for assigning label names. You can specify the pattern inside the < and > symbols. These symbols are retained in the generated labels and represent bus bits. You can also use the [and] symbols to enclose the pattern, in which case, the symbols are not part of the labels generated.

© 2023 Cadence Design Systems, Inc.

The following table provides some examples of the patterns that you can specify.

Pattern Definition	Example Pattern	Label Names Generated When *Expand* is On
By using < >		
Simple naming for each label	`J clk q c<1> c<2>`	`J clk q c<1> c<2>`
Simple sequential naming of the type <*n1:n2*>	`A<0:3>`	`A<0>, A<1>, A<2>, A<3>`
Simple sequential naming of the type <<*n1:n2*>>	`A<<0:3>>`	`A<0>, A<1>, A<2>, A<3>`
Mixed sequential naming	`A<2:3> clk B<1:0>`	`A<2>, A<3>, clk, B<1>, B<0>`
Interleaving using '-'	■ `A<1:4>-B<0:1>` ■ `A<1:2>-B<1:5:2>-C<1:2>.....`*`n`* ■ `A[[1:5:2]]-B<[0:1]>`	■ `A<1>, B<0>, A<2>, B<1>, A<3>, A<4>` ■ `A<1> B<1> C<1> .... A<2> B<3> C<2>.....`*`n`* ■ `A[1], B<0>, A[3], B<1>, A[5]`
Increment label index by more than 1 by using the format <*n1:n2:n+*> Where, *n+* is the increment value	`A<0:6:2>`	`A<0>, A<2>, A<4>, A<6>`
Decrease label index by more than 1 by using the format <*n1:n2:n-*> Where, *n-* is the decrement value	`B<7:0:2>`	`B<7>, B<5>, B<3>, B<1>`
By using [ ]		

© 2023 Cadence Design Systems, Inc.

Pattern Definition	Example Pattern	Label Names Generated When *Expand* is On
Simple sequential naming	■ `A[0:2]` ■ `A[[0:2]]` ■ `A[1:2]B[1:2]` ■ `A[1:2]C` ■ `A[1:5:2]C`	■ `A0, A1, A2` ■ `A[0], A[1], A[2]` ■ `A1B1, A1B2, A2B1, A2B2` ■ `A1C, A2C` ■ `A1C, A3C, A5C`
Interleaving using '–'	`A[1:2]-B[1:2].....n`	`A1 B1 A2 B2.....n`
Bus label	`A[1:2] B[1:2]`	`A1B1 A1B2 A2B1 A2B2`
By using both < > and [ ]		
Simple sequential naming	■ `A[<0:2>]` ■ `A<[0:2]>`	■ `A[0], A[1], A[2]` ■ `A<0>, A<1>, A<2>`

If you use an escape character (\) preceding a sequential pattern, the expansion is overridden.

You can include special characters in a label name. Spaces in a pattern are used to distinguish the different label names in the sequence. After you have placed all the labels of a sequence and if the *Keep Label* option is selected in the Create Label form, the labels in the sequence are re-generated and are available to be placed. If, however, *Keep Label* is off, you are prompted to specify the label text again in the *Label (Pattern)* field. With *Keep Label* selected, the labels are re-generated only if the *Create Label* command is in repeat mode.

The *Label (Pattern)* field is active only in the manual mode. In the auto mode, the label names are generated automatically from connectivity information of shapes or from instances.

Related Topics

Configuring Label Creation Options

Generating Labels Using the Scan Line

© 2023 Cadence Design Systems, Inc.

Generating Labels Using Auto Step in Manual Mode

Label Creation in Manual and Auto Modes

Label Generation for Overlapping Shapes

Create Label Form

Label Creation on LPPs Derived from stampLabelLayers

In the Create Label form, you can specify the layer purpose pair (LPP) on which you want to place labels. You can use the *Auto (Using stampLabelLayers section from technology file)* option to derive the layer and purpose on which to create the label from the `stampLabelLayers` section of the technology file. If the `stampLabelLayers` section does not exist in the technology file, a warning is displayed in the CIW. You cannot use this option to add labels to empty spaces in a cellview. With this option selected, you cannot label instances, vias, groups, and mosaics.

© 2023 Cadence Design Systems, Inc.

Consider the following example where the figure shows a sample `stampLabelLayers` section from a technology file.

```
;*********************************
; LAYER RULES
;*********************************
layerRules(

 functions(
 ;( layer        function                  [maskNumber])
 ;( -----        --------                  ------------)
  ( METAL1       "metal"                   1            )
  ( VIAS1        "cut"                     2            )
  ( METAL2       "metal"                   3            )
  ( VIAS2        "cut"                     4            )
  ( METAL3       "metal"                   5            )
  ( VIAS3        "cut"                     6            )
  ( METAL4       "metal"                   7            )
  ( VIAS4        "cut"                     8            )
  ( METAL5       "metal"                   9            )
  ( VIAS5        "cut"                     10           )
  ( METAL6       "metal"                   11           )
  ( CONT         "CUT"                     12           )
  ( POLY1        "poly"                    13           )
  ( POLY2        "poly"                    14           )
  ( NMOAT        "ndiff"                   15           )
  ( PMOAT        "pdiff"                   16           )
 ) ;functions
 stampLabelLayers(
  ( ("METAL2" "drawing") METAL1 METAL3 )                  //rule id1
  ( text METAL2 METAL5 )                                  //rule id2
  ( text ("METAL2" "drawing") METAL4 METAL5 )             //rule id3
  ( ("text" "drawing") ("METAL3" "drawing") METAL6 )//rule id4
  ( ("Poly" "drawing") METAL1 METAL5 )                    //rule id5
  ( pin VIAS1 VIAS2 CONT )                                //rule id6
 )

) ;layerRules
```

Each statement in the `stampLabelLayers` section defines a rule for creating a label on a particular layer-purpose pair. The first layer-purpose pair entry of a rule indicates the layer-purpose on which the label is created for a shape that exists on layer-purposes defined by the remaining entries of a rule. If a purpose is not specified in the first entry, the valid list of purposes for the specified layer is considered for creating the label. For example, in `rule`

© 2023 Cadence Design Systems, Inc.

`Id1`, the first entry `"METAL2" "drawing"` represents the layer-purpose on which the label will be created. The remaining entries `METAL1 METAL3` represent the layers on which the shape to be labeled should exist.

If a layer or layer-purpose pair of the shape occurs in multiple rules, the first encountered rule for that layer or layer-purpose is used for deciding the layer-purpose for the label. For example, if a shape exists on layer `METAL5`, `rule Id2` will be used even though `rule Id2`, `rule Id3`, and `rule Id5` are all `METAL5` rules. Similarly, for a shape on `METAL3`, `rule Id1` will be used. If you attempt to label a shape whose layer or layer-purpose pair are not part of any rule, an error message is issued in the CIW. In the sample `stampLabelLayers` section shown in the figure, there are rules for the layers `METAL1`, `METAL2`, `METAL3`, `METAL4`, `METAL5`, `METAL6`, `VIAS1`, `VIAS2`, and `CONT`. You cannot create a label for a shape that exists on layer `POLY` because no rule for the `POLY` layer has been defined in the `stampLabelLayers` section.

In the sample `stampLabelLayers` section, `rule Id1` specifies that if the shape to be labeled is on `METAL1`-any purpose or `METAL3`-any purpose, the label is created on `METAL2` purpose `drawing`. Similarly, `rule Id2` specifies that if a shape is on `METAL2`-any purpose or `METAL5`-any purpose, the label is created on `text` layer and on a valid purpose defined for the `text` layer in technology file. The `rule Id4` rule states that the label is created on `text` layer purpose `drawing` only if the shape exists on `METAL3` purpose `drawing` or `METAL6`-any purpose. If the shape does exist on `METAL3`, `rule Id1` is used for determining layer-purpose for the label.

The following table summarizes all the rules of the sample `stampLabelLayers` section. In the table, *Valid purpose* refers to any valid purpose defined in the technology file for that layer.

	If shape is on...		**Label is created on...**	
Rule	**Layer**	**Purpose**	**Layer**	**Purpose**
`rule Id1`	`METAL1`	Any	`METAL2`	`drawing`
	`METAL3`	Any		
`rule Id2`	`METAL2`	Any	`text`	Valid purpose
	`METAL5`	Any		
`rule Id3`	`METAL2`	`drawing`	`text`	Valid purpose
	`METAL4`	Any		
	`METAL5`	Any		

© 2023 Cadence Design Systems, Inc.

rule Id4	METAL3 METAL6	drawing Any	text	drawing
rule Id5	METAL1 METAL5	Any Any	Poly	drawing
rule Id6	VIAS1 VIAS2 CONT	Any Any Any	pin	Valid purpose

Related Topics

Create Label Form

Configuring Label Creation Options

Generating Labels Using the Scan Line

Generating Labels Using Auto Step in Manual Mode

Configuring Label Creation Options

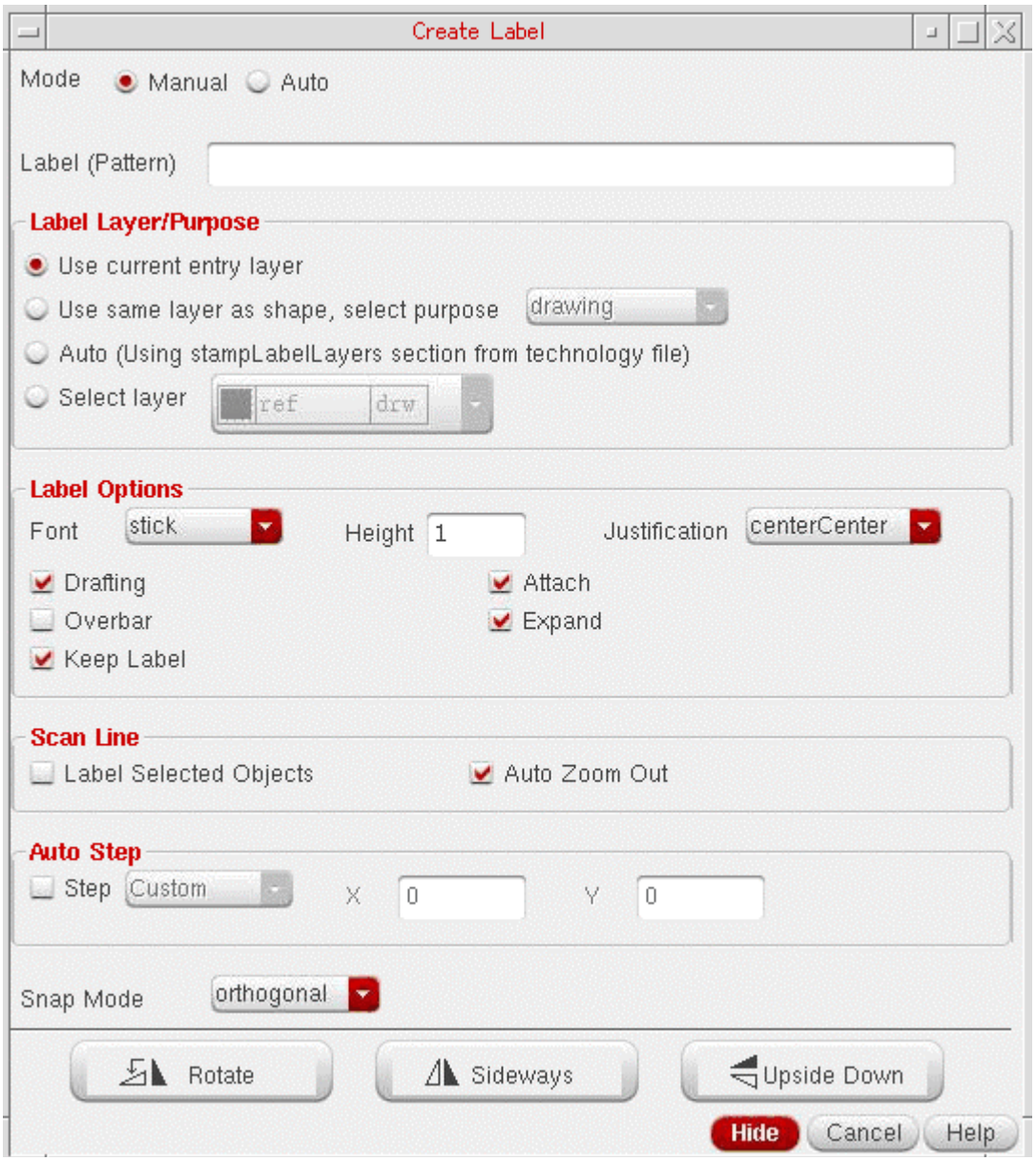
Use the Create Label form to customize the settings for creating labels.

To specify the settings for creating labels:

1. In the layout window, choose *Create – Label*.

© 2023 Cadence Design Systems, Inc.

The Create Label form appears. If the form is not displayed, press `F3`.



Note: When you run the *Create – Label* command, the *Create Label* form opens if the last setting in the form was *Manual*. If the last selected mode was *Auto*, the *Create Label* form does not open, but you are prompted to click or select a shape to create labels. The label is not dropped at the pointer location if you start the *Create Label* command in *Auto* mode because this command is not affected by the Infix mode setting.

2. Choose the mode of generating labels as *Manual* or *Auto*.

© 2023 Cadence Design Systems, Inc.

- If you select *Auto*, configure the required settings on the *From Interconnect* and *From Instance* tabs. If you want to use the *From Interconnect* tab, ensure that the selected objects have connectivity information.
- If you select *Manual*, in the *Label (Pattern)* field, type the text string or the pattern for the label names.

3. In the *Label Layer/Purpose* section, choose a layer and purpose for placing the label.
4. In the *Label Options* section:
   a. Choose a font for the label from the *Font* list.

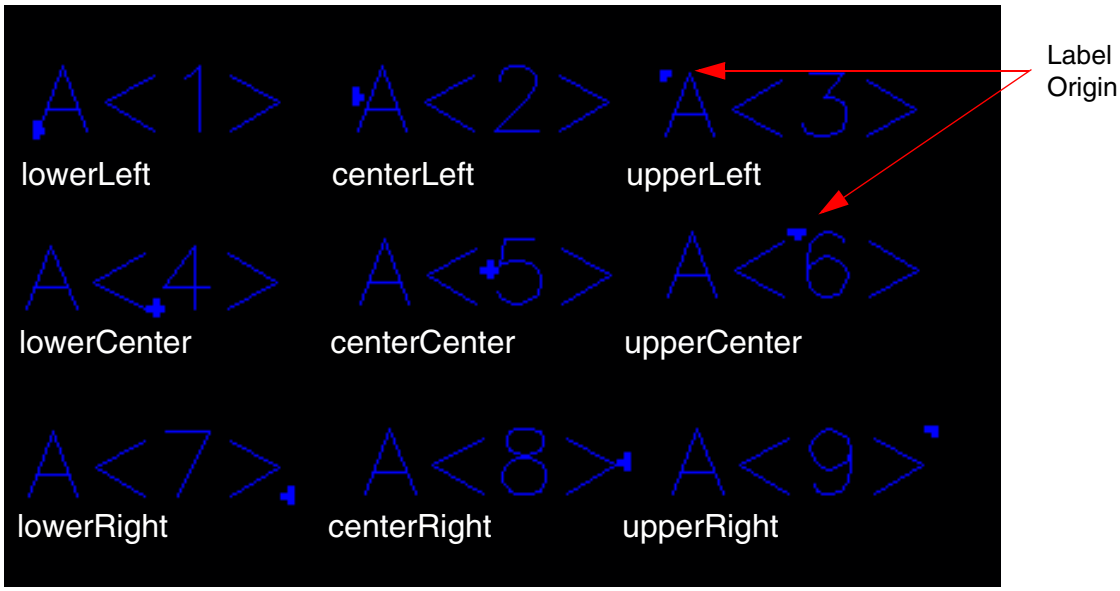
      The following figure shows the available fonts.



   b. Specify a height for the label text in the *Height* field.
   c. Select the position of the label origin with respect to the label position, from the *Justification* list.

© 2023 Cadence Design Systems, Inc.

The figure shows the available options that you can select.



d. Select the *Drafting* check box if you do not want the label text to be rotated by more than 90 degrees.

e. Select the *Overbar* check box if you want underscore characters in the label text strings or patterns to be displayed as overbars.

If the *Overbar* check box is off (default), the software displays underscore characters (_) as part of the text string. If the *Overbar* check box is on, the software interprets underscore characters in the text string name as toggle switches that control where overbars begin and end. Overbars appear above the text string, as shown in the following examples.

Property Name	Appears in Design Window As
_abcde	$\overline{\text{abcde}}$
ab_cde	ab$\overline{\text{cde}}$
_abc_de	$\overline{\text{abc}}$de
ab_cd_ef_gh_ij	ab$\overline{\text{cd}}$ef$\overline{\text{gh}}$ij

f. Select the *Keep Label* check box if you want the label or pattern specified in the *Label (Pattern)* field to be available after all the labels or the ones generated from a sequence have been used up.

© 2023 Cadence Design Systems, Inc.

g. Select the *Attach* check box if you want to attach the label to an object.

h. Select the *Expand* check box if you want to generate separate label names from the pattern specified in the *Label (Pattern)* field.

5. Set the *Scan Line* section if you want to create labels using the scan line:

a. Select the *Label Selected Objects* check box if you want only the selected objects to be labeled while using the scan line method.

b. Ensure that the *Auto Zoom Out* check box is selected to enable automatic zooming and panning if the scan line extends beyond the current display area.

6. In the *Auto Step* section, select the *Step* check box if you want labels to be created automatically with the fixed, specified distance between consecutive labels.

7. Choose a value from the *Snap Mode* list to control the direction in which you can move the scan line while using the scan line and the auto step methods.

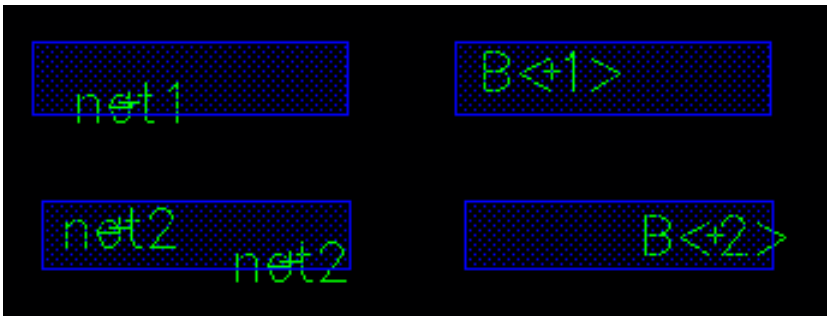
8. To rotate or mirror a label as you place it, set *Drafting* off, then click *Rotate*, *Sideways*, or *Upside Down* as required before placing the label.

Alternatively, you can click the middle mouse button to replicate *Rotate,* press `Ctrl` and click the middle mouse button to replicate *Sideways*, or press `Shift` and click the middle mouse button replicate *Upside Down.* Each time you click a button on the form or click the middle mouse button, the label is further rotated or mirrored.

After specifying various options in the *Create Label* form, as you move the pointer to the canvas, you see the ghost image of the label text in the *Manual* mode. This image represents how the label text will look when you place the label. The ghost image moves along with the pointer until you place the label. In the *Auto* mode, the ghost image of label text appears only when you hover the pointer over shapes that have connectivity information.

9. Click on the canvas to place the label. You can continue clicking the objects one by one and the label is placed at the point of click.

Auto Mode Manual Mode



© 2023 Cadence Design Systems, Inc.

As per the settings you specified, you can draw a scan line, use the auto step, and select shapes fully or partially to place the labels.

If there are multiple overlapping shapes in the manual and auto label creation modes, use the Choose Object for Label Creation form to choose a layer when clicking on a coordinate with objects on overlapping layers.

You can add a label to members of all types of figure groups. For modgen figure group, you can add a label to modgen figure group if you are editing in group for that modgen.

Related Topics

Create Label Form

Choose Object for Label Creation Form

Label Creation

Generating Labels Using the Scan Line

Generating Labels Using Auto Step in Manual Mode

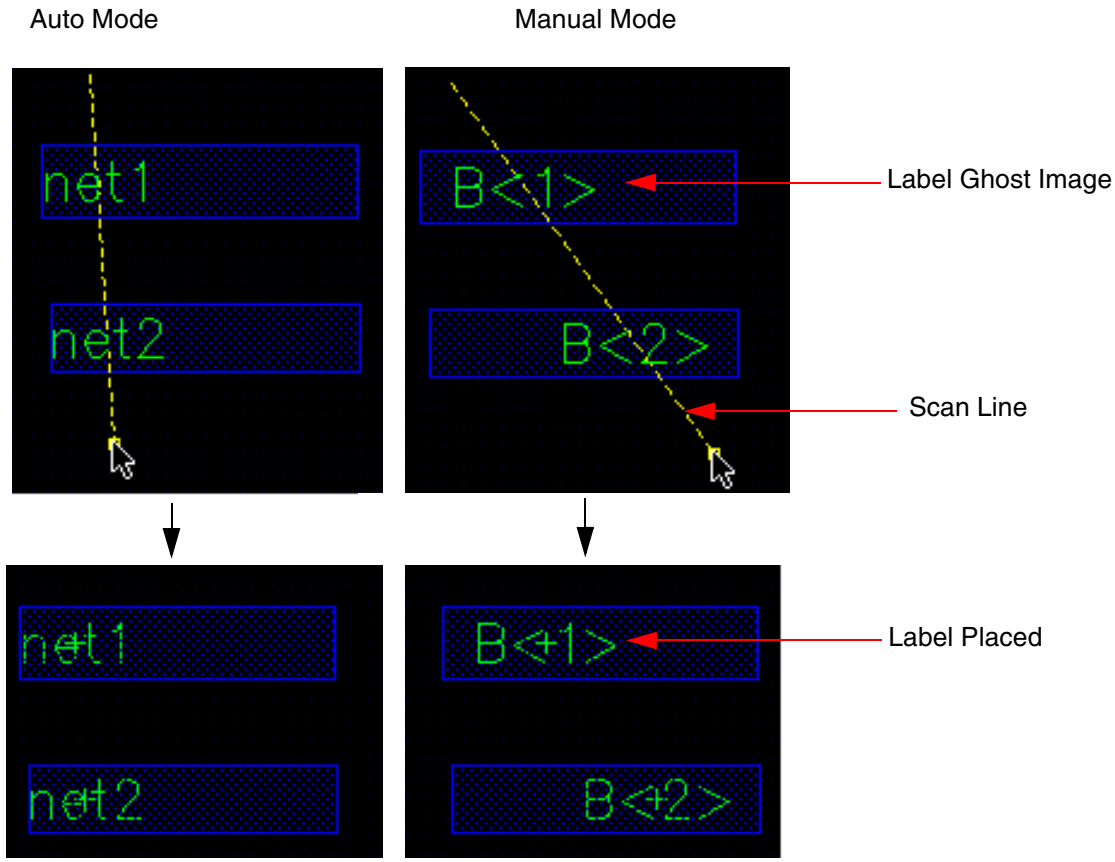
Generating Labels Using the Scan Line

By using the scan line, you can label multiple objects in one go. This method is useful if the specified pattern in the *Label (Pattern)* field generates multiple labels in a sequence. For example, the label `GDD` will label only the first object even though the scan line may intersect more objects. The scan line assigns labels to only as many objects as the number of labels generated by the specified pattern. Consider the pattern `A<0:2>` with the *Expand* option selected. Three labels are generated: `A<0>`, `A<1>`, and `A<2>`. If the scan line intersects five objects, only three of them are labeled.

To generate labels by using the scan line, with the middle mouse button, click and drag the pointer to draw a line scanning across the objects you want to label. Then, release the mouse button at the end point of the scan line. Labels are placed at the intersection of the scan line

© 2023 Cadence Design Systems, Inc.

and the objects. You can configure the direction in which you can move the scan line by setting the *Snap Mode* option, which is set to *orthogonal* by default.



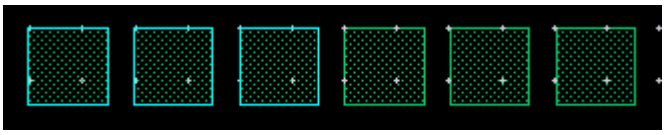
Labeling Selected Objects

In *Manual* mode, you can use the scan line method to label only the selected objects. You can select the objects before or after starting the *Create – Label* command.

In *Auto* mode, you can use the scan line method to label pre- or post-selected objects with connectivity information and pre- or post-selected instances.

To label selected objects using the scan line:

1. In the layout window, select the objects that you want to label.

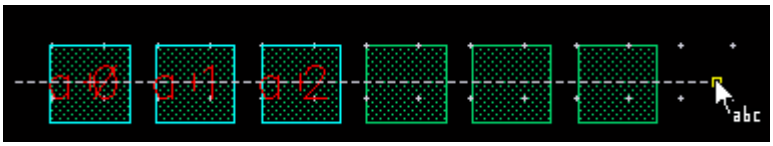


© 2023 Cadence Design Systems, Inc.

2. Choose *Create – Label*.

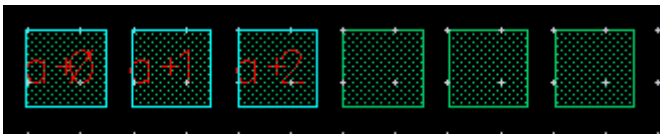
   The Create Label form opens.

3. Type the label pattern in the *Label (Pattern)* field.
4. Select the *Label Selected Objects* check box from the *Scan Line* section.
5. Ensure the *Auto Zoom Out* check box is selected.
6. With the middle mouse button, click on the canvas and drag a scan line that intersects the selected and unselected objects.



7. Release the mouse button.

   Only the selected objects are labeled.



8. Press `Esc` or click *Cancel* in the form to finish creating labels.

Only as many labels from the specified sequence in the *Label (Pattern)* field are used as there are selected objects.

Labeling Objects Out of the Design Display Area

In the scan line method, since you use the middle mouse button to draw the scan line, you cannot use the mouse scroll wheel to zoom in or zoom out of the design while creating the scan line. The `Tab` key is also unavailable to pan through the design. Therefore, in a scenario where there is a large number of objects to be labeled, some of which are out of the current display area, ensure that the *Auto Zoom Out* check box in the *Scan Line* section of the Create Label form is selected. As you click and drag the scan line beyond one of the edges of the design display area (to allow the scan line to intersect objects currently not in view), the *Auto Zoom Out* option automatically zooms out the design to make visible the other objects that need to be labeled. As you continue to drag the scan line, the design keeps zooming out and panning automatically. When you release the middle mouse button to complete the scan line, the design returns to the original zoom level and labels are added to all objects that were

© 2023 Cadence Design Systems, Inc.

intersected by the scan line. If you do not select this check box, no zoom out occurs, though the design pans as you drag the scan line beyond a design display area edge.

Related Topics

Create Label Form

Label Creation

Configuring Label Creation Options

Generating Labels Using Auto Step in Manual Mode

You can use the auto step method to create labels. This method is available only in *Manual* mode.

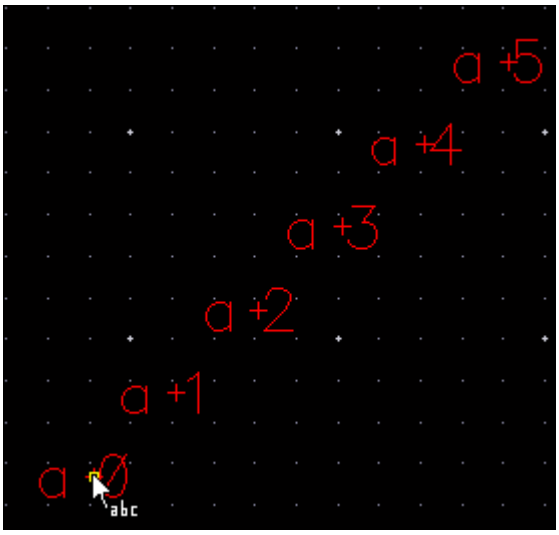
To create labels by using the auto step method:

1. In the layout window, choose *Create – Label*.

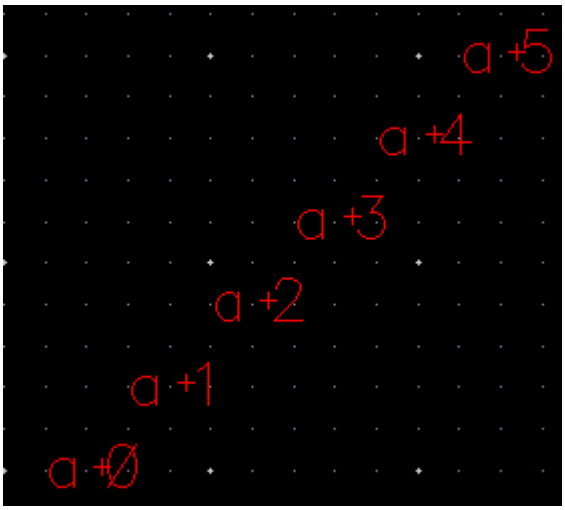
   The Create Label form opens.
2. Set the *Mode* as *Manual*.
3. Type the label pattern in the *Label (Pattern)* field.
4. Select the *Step* check box in the *Auto Step* section.
5. From the *Step* list, if you select *Input X&Y*:
   a. Specify values in the *X* and *Y* fields.

© 2023 Cadence Design Systems, Inc.

All the generated labels attach to the pointer and are visible as per the specified spacing.



b. Click on the canvas to place the labels.



If the *Keep Label* check box in the *Label Options* section is selected, the label sequence remains attached to the pointer after you click to place the label.

6. From the *Step* list, if you select *Custom*:

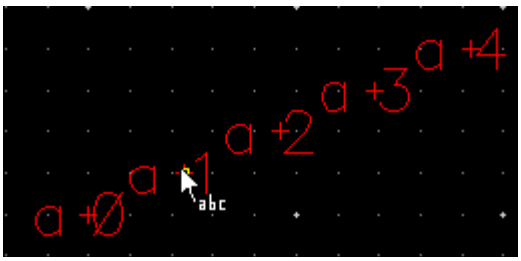
The first label attaches to the pointer.

a. Set up the *Snap Mode* setting.

b. Click to place the first label in the design display area.

© 2023 Cadence Design Systems, Inc.

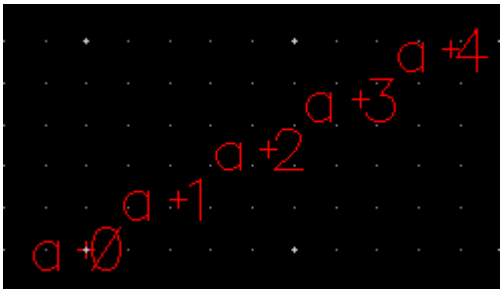
The remaining generated labels attach to the pointer, as shown in the figure.



You can move the pointer around to determine the position of the second label.

c. Click on the canvas to place the second label.

The remaining labels are placed by using the same spacing as between the first and second placed labels and along the same direction as the first two labels.



7. Press `Esc` or click *Cancel* in the form to finish creating labels.

Related Topics

Create Label Form

Label Creation

Configuring Label Creation Options

Label Creation in Manual and Auto Modes

You can create labels in manual and auto modes. The way labels are generated varies in the two modes.

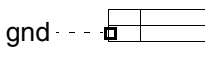
- Manual Mode
- Auto Mode

© 2023 Cadence Design Systems, Inc.

Manual Mode

In manual mode, each generated label is placed with a click, at the click location. For example, in the Create Label form if you specify `A<0:3>` in the *Label (Pattern)* field, the labels are incrementally generated from `A<0>` through `A<3>` and are placed one-by-one at the click location.

With the *Attach* option enabled, a label is attached to the object you click. If there is no object at the click point, you are prompted to click an object to attach the label to the object. When attaching a label to an object, a dashed line extends from the label to the object and disappears when you click the object.



A dashed line shows the object to which you attach the label

As you place a single label or multiple labels generated from a sequence, they get consumed from the *Label (Pattern)* field. For example, if `A1 B2 C<0:3>` is the specified label pattern, after placing `A1` and `B2`, the *Label (Pattern)* field displays only `C<0:3>`. The *Label (Pattern)* field shows `C<0:3>` until you place `C<0>` through `C<3>`. With the *Keep Label* option selected by default, the pattern specified is retained in the *Label (Pattern)* field and can be used to generate same set of more labels. If the *Keep Label* option is off, the *Label (Pattern)* field becomes empty after you place all the labels generated from the sequence, and you are prompted to specify a label text. You can specify the same or different label text.

Auto Mode

In auto mode, label is generated based on the connectivity information of the object you want to label or from the instance name or instance master you want to label. The label is placed at the click location and is automatically attached to the object you are labeling.

In the auto mode, you can pre- or post-select the shapes to be labeled. In both selection modes, you can select shapes either fully or partially. The full and partial scenarios described in this topic are the same for both pre- and post-selection modes.

If an object is non-selectable, the label is not created on the object in both manual and auto label creation modes.

Full Selection

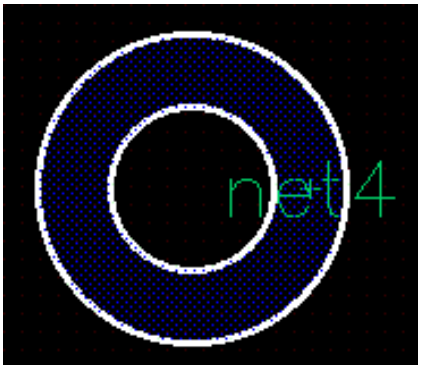
The label (origin) is automatically placed at the center of the shape. You can enable full selection mode by enabling the Full Select icon  on the Options toolbar or by pressing `F4`. Full selection of a shape in Auto mode labels the shapes in the following ways:

© 2023 Cadence Design Systems, Inc.

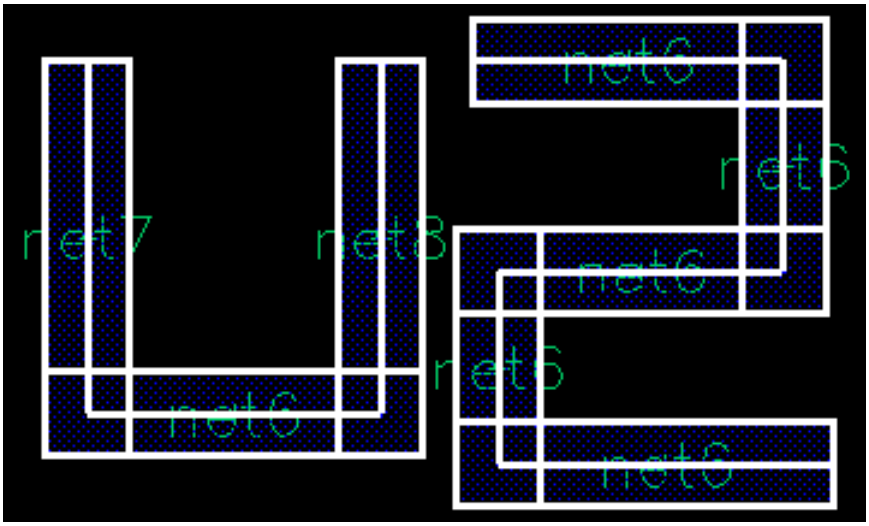
- In the case of shapes such as rectangles, circles, and polygons, the label is placed at the center of the shape. In the case of a fully selected polygon, if the center of the polygon is outside the polygon, the label is placed on the midpoint of the longest edge of the polygon.



- For a donut, the label is placed at the midpoint of the longest edge of a donut-equivalent polygon.



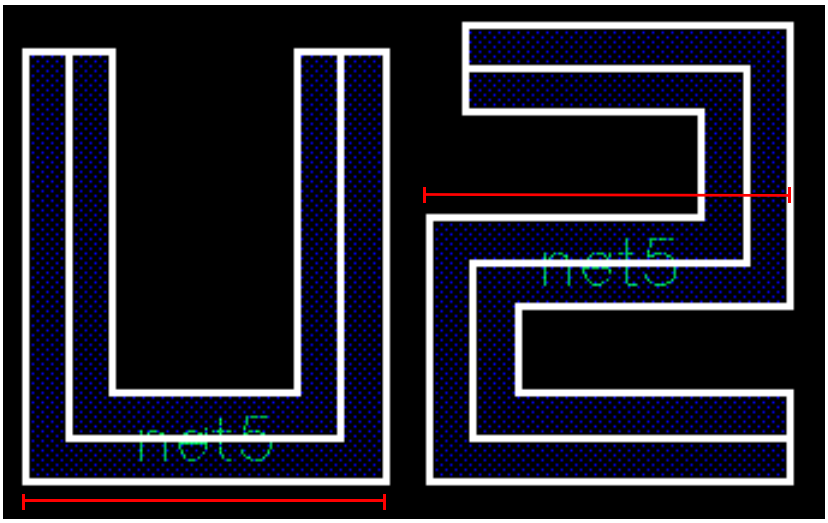
- For pathsegs, the label is placed at the midpoint of the centerline.



Label is placed at the midpoint of each pathseg.

© 2023 Cadence Design Systems, Inc.

- For paths, the label is placed at the midpoint of the middle segment's centerline.



Middle segment of the path

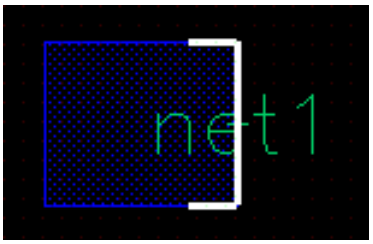
Middle Segment of the Path

Partial Selection

To enable partial selection, you can select the *Partial Select* icon  on the *Options* toolbar or press `F4`.

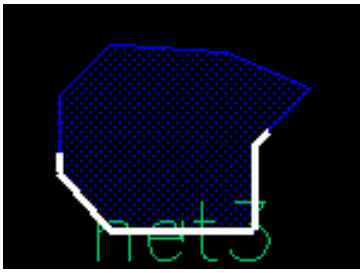
Partial selection or post-selection of a shape in auto mode labels the shapes in the following way:

- The label is placed at the midpoint of a selected edge.

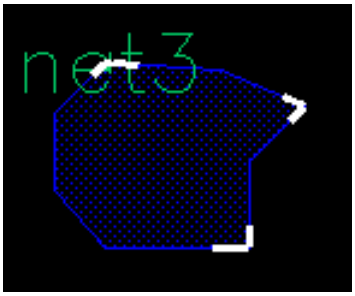


© 2023 Cadence Design Systems, Inc.

- With multiple selected edges, the label is placed at the midpoint of the longest selected edge. If two or more selected edges are of the same length, label is placed on the midpoint of a randomly selected same-length edge.

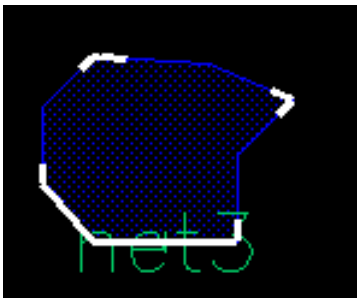


- If multiple non-consecutive vertices are selected (no edge selected), the label is placed at the vertex closer to the midpoint of the shape.



Selected Vertices

- If a combination of vertices and edges are selected, the label is placed at the midpoint of the longest selected edge.



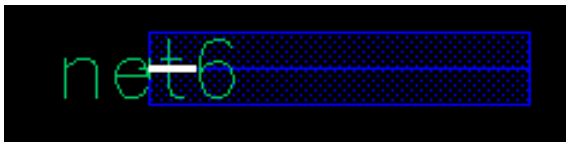
- Partial selection of a path or pathSeg places the label origin at the end of the selection.

© 2023 Cadence Design Systems, Inc.

- If the begin or end style is *truncate*, the label is placed on the centerline where it meets the edge.

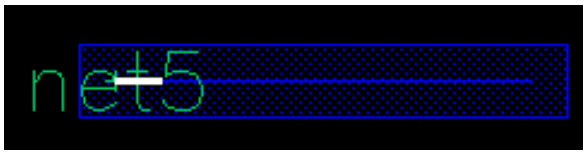


Path with end style Truncate

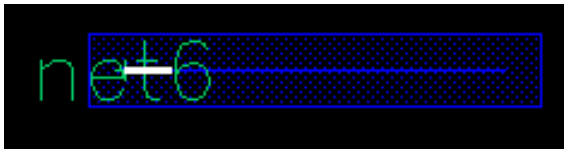


Pathseg with begin and end style Truncate

- If the begin or end style is *extend*, the label is placed at the end of the centerline.



Path with end style Extend



PathSeg with begin and end style Extend

Related Topics

Create Label Form

Label Creation

Configuring Label Creation Options

Generating Labels Using the Scan Line

Generating Labels Using Auto Step in Manual Mode

Label Generation for Overlapping Shapes

Label Generation for Overlapping Shapes

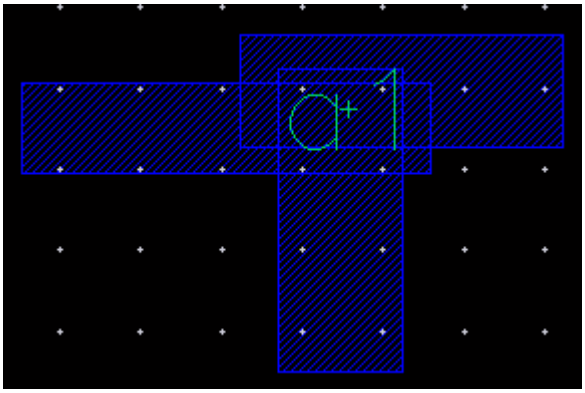
The way labels are generated for overlapping shapes vary in manual and auto modes selected in the Create Label form.

© 2023 Cadence Design Systems, Inc.

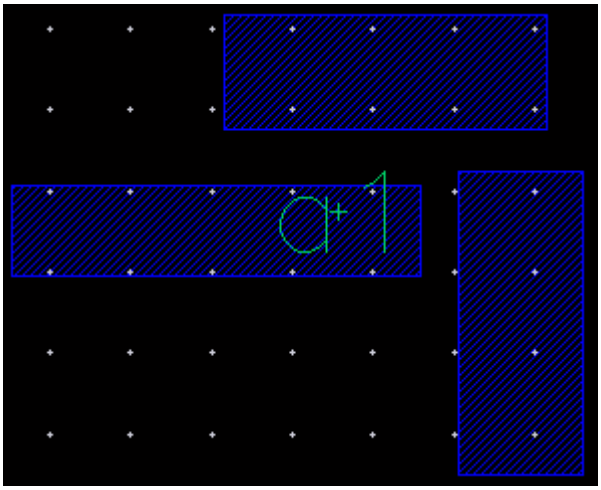
- Manual Mode
- Auto Mode

Manual Mode

- When the *Label Layer/Purpose* field is set to *Use current entry layer* or *Select layer*: If the click point or intersection point of scan line lies on overlapping shapes, a single label is created at the click or intersection point. The label is created on the layer-purpose pair that is the current entry layer in the Layers panel or on the one that you select from the *Select layer* list. Single label string is consumed from the *Label (Pattern)* field.



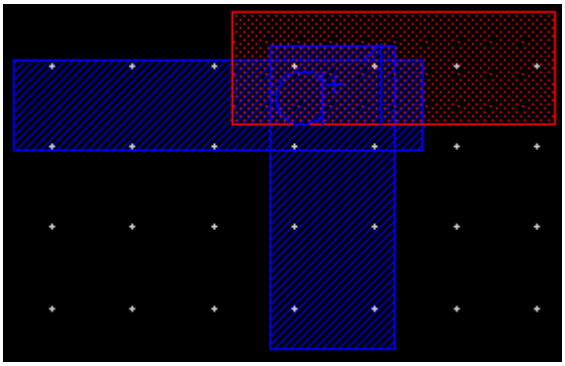
You can move the shapes to confirm that only one label is created.



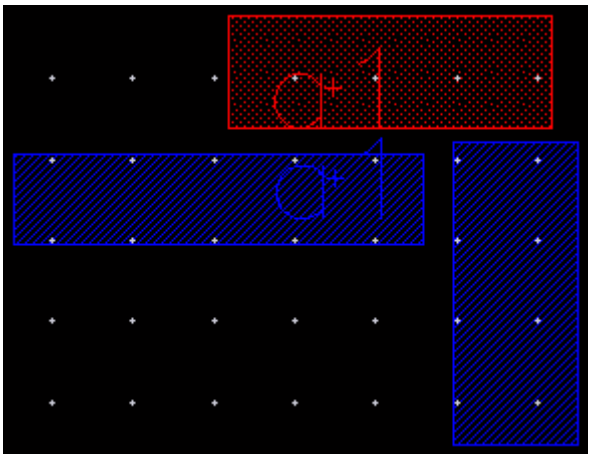
- When the *Label Layer/Purpose* field is set to *Use same layer as shape, select purpose* or *Auto (Using stampLabelLayers section from technology file)*: If the click point or intersection point of scan line lies on overlapping shapes, multiple labels are created at the click or intersection point. One unique label is created on each layer-

© 2023 Cadence Design Systems, Inc.

purpose pair. The following figure illustrates this when the *Label Layer/Purpose* is set to *Use same layer as shape, select purpose*. The two shapes in blue are on the `metal1` layer and the red shape is on the `metal2` layer.



If you move the shapes, you will notice that the labels are created on the same layers as shapes. However, since there are two shapes on `metal1`, only one label is generated for the two shapes.

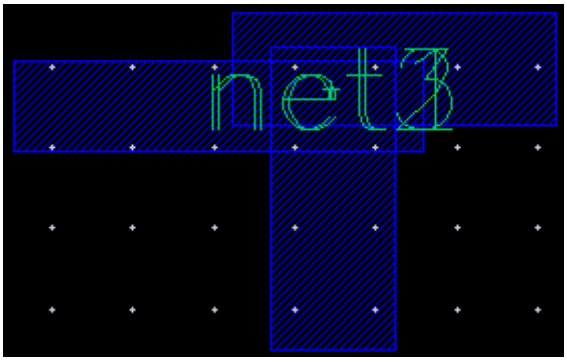


Auto Mode

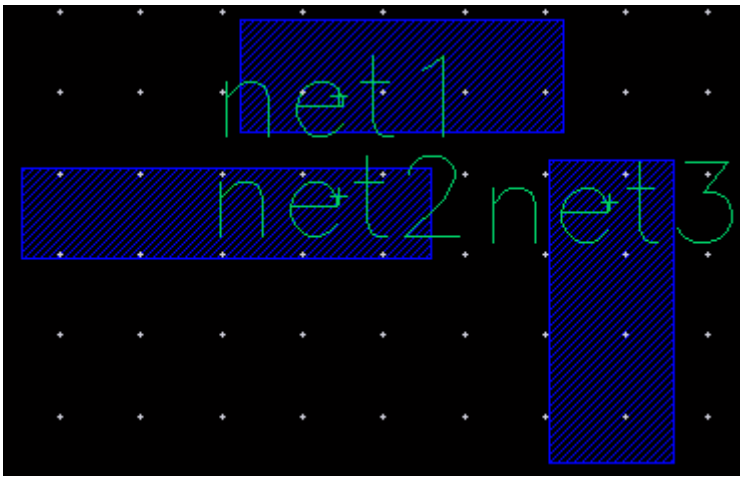
- When the *Label Layer/Purpose* field is set to *Use current entry layer* or *Select layer*: Separate labels are generated for overlapping shapes. The label name for each corresponds to the net name associated with the respective overlapping shape. Multiple labels with different label names overlap at the click location. The overlapping labels attach to the respective overlapping shapes by default. The labels are created on the

© 2023 Cadence Design Systems, Inc.

layer-purpose pair that is the current entry layer in the Layers panel or on the one that you select from the *Select layer* list.



Move the shapes to see that each shape has been labeled with its net name and all labels are created on the same layer-purpose pair.

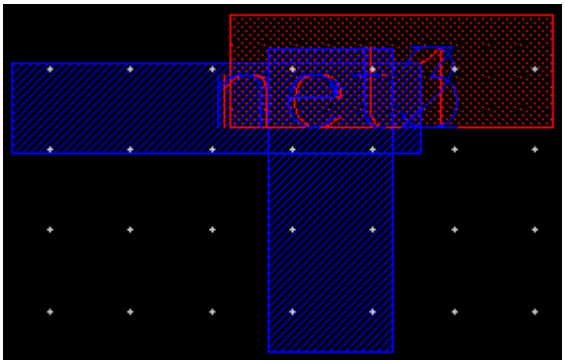


If the shape you click or the shape that intersects the scan line does not have connectivity information, no label is assigned to the shape and a warning message is generated in the CIW.

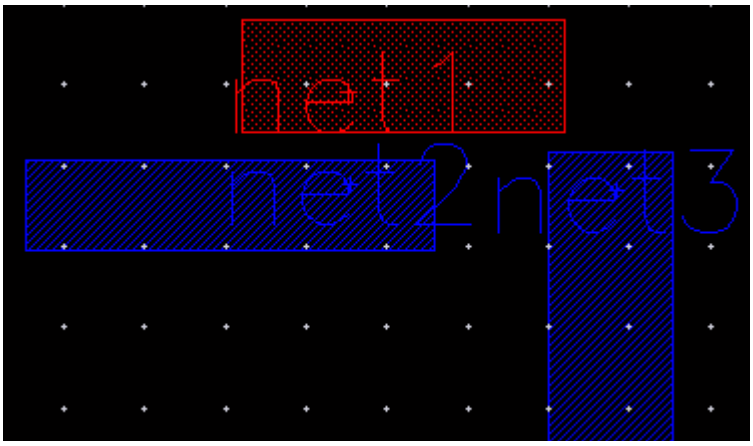
- When the *Label Layer/Purpose* field is set to *Use same layer as shape, select purpose* or *Auto (Using stampLabelLayers section from technology file)*: Separate labels are generated for overlapping shapes with the label names corresponding to the net name associated with the respective overlapping shapes. The

© 2023 Cadence Design Systems, Inc.

labels are created on the same layer as the shapes if you select *Use same layer as shape, select purpose*, as shown in the following figure.



Move the shapes to see that each shape has been labeled with its net name and all labels are created on the same layer-purpose pair.



Note: Small labels, less than 4 pixels, are not displayed in the design. To view small labels, you can enable the drawSmallLabelOrigin environment variable.

You can create and attach a label to an object inside a figure group if you are in Edit In Place (EIP) mode or the transparent group option is ON. You can create and attach a label to an object inside a figure group in the following scenarios:

- If you create a label on an object inside the figure group and attach it to the target object
- If you click an empty area outside the figure group and attach the label to a target object inside the figure group

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

Label Creation

Configuring Label Creation Options

Generating Labels Using the Scan Line

Generating Labels Using Auto Step in Manual Mode

Via Creation

To connect objects on different layers within a cellview, you can create vias by using the *Create Via* command. The various types of vias are OA standard, cds, syContact, Pcell, and fixed vias.

Vias are defined in the `viaDefs` section of the technology file. Via definitions (viaDefs) are of the following types:

- Standard via definition (`standardViaDefs`)
- Custom via definition (`customViaDefs`)
- Standard via variant (`standardViaVariants`)
- Custom via variant (`customViaVariants`)

Vias must exist in the technology file in order for them to appear in the *Via Definition* field in the Create Via form. In both the Create Via and the Edit Via Properties forms, viaDefs are listed in the order as they are defined in the `validVias` list in the technology file.



Important

Always use the Create Via form to place vias in a cellview or add them by using the Select Via Form while creating wires. Do not use the Create Instance form to create vias because these will not be interpreted as vias by the OpenAccess.

Parameter Caching

When you first open the Create Via form, the parameter values for the respective viaDefs are seeded in the form from the technology database values, based on the values specified in the Create Via section in the Layout Editor Options form.

While creating a via or via stack, you can change the parameter values for a viaDef. If you successfully create a constraints-valid via or via stack using the updated parameters, the parameter set is cached for that viaDef. The next time you access the form in the same

© 2023 Cadence Design Systems, Inc.

Virtuoso session with the same viaDef and layer/width values, the cached parameter set is used. Each time you create a via with a different viaDef or different parameter values, the cache is created or updated, respectively.

The cached parameter values are validated against existing constraints before you can use the parameter set to create a via or via stack. If the parameters are not valid, they are recalculated based on enclosure calculation settings.

Similarly, if while creating a via, you manually update the enclosure values in the Create Via form, the values are cached if they comply with the minimum rules. The manually specified enclosure values are not re-calculated when you update other viaDef parameters, as long as the enclosure values do not violate the minimum rules.

You cannot create a custom or standard via that creates an invalid geometry.

Related Topics

Technology File Via Definitions and Via Specifications

Via Definitions and Via Specifications Functions

Create Via Form

Layout Editor Options Form

Installing a cdsVia Device in the Technology File

To install a `cdsVia` device in the technology file by using the Install Device form:

1. In the CIW, choose *Tools – Technology File Manager*.

   The Technology File Manager window opens.

2. Click *Install Device* in the *Utilities* section.

   The Install Device form appears.

3. From the *Technology Library* list, select the technology library in which you want to create the `cdsVia` device.

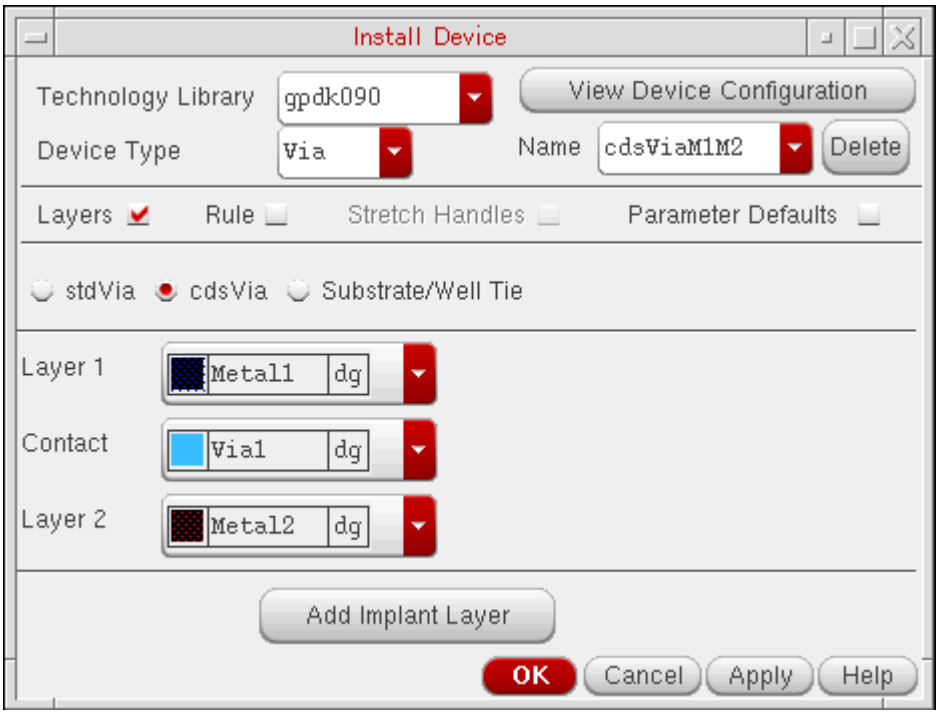
4. Select *Via* from the *Device Type* list.

5. Select the *cdsVia* option.

6. In the *Name* field at the top of the form, specify the name of a new via device or select an existing via device from the list.

© 2023 Cadence Design Systems, Inc.

The following image shows the sample name as `cdsViaM1M2`. The form gets updated accordingly.



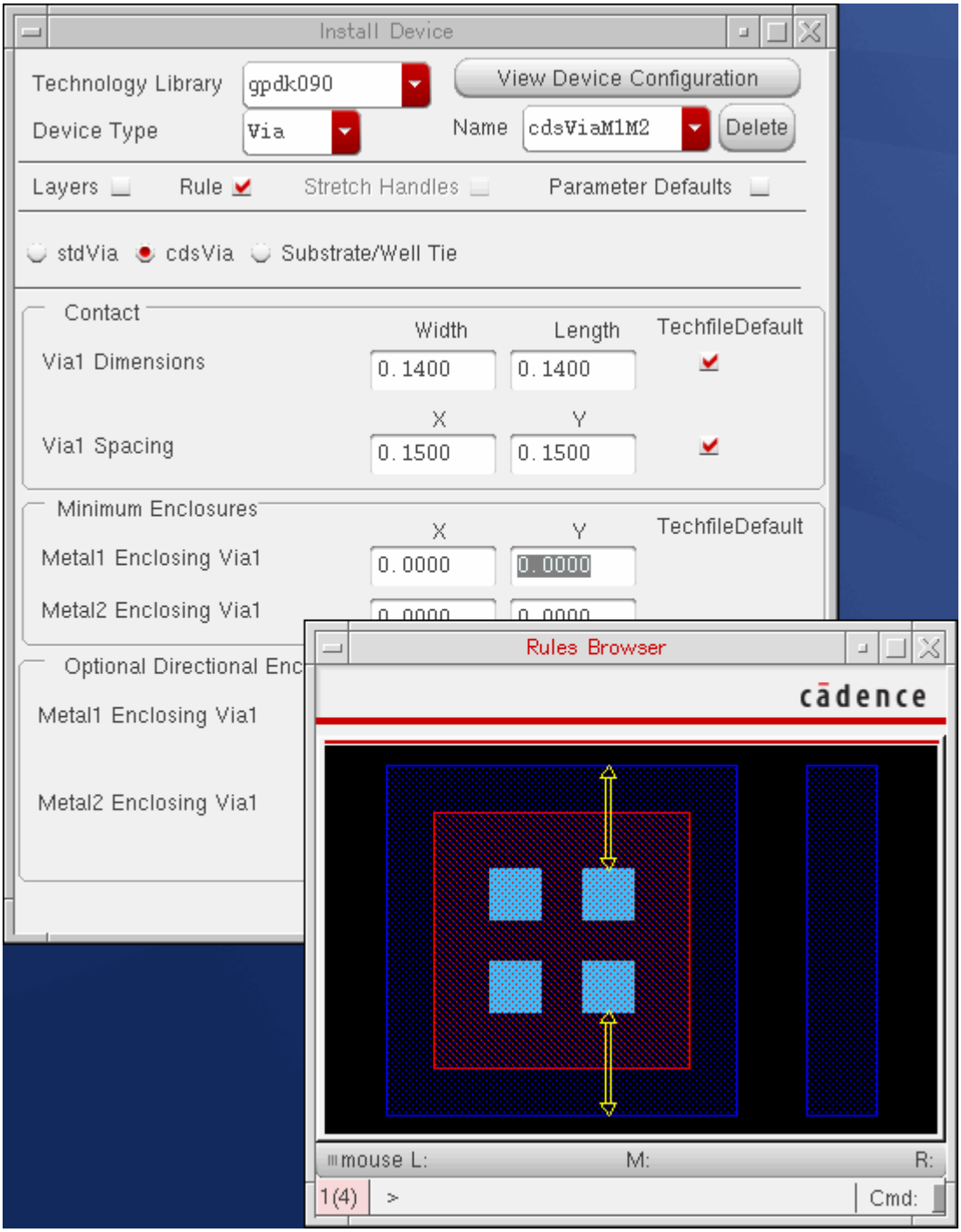
To delete an existing via device, select the via name from the *Name* list and click the *Delete* button.

7. With the *Layers* check box is selected, select the required layers from the *Layer 1*, *Contact*, and *Layer 2* lists. For example, you can set the following:
   - *Layer 1* as *Metal1 dg*
   - *Contact* as *Via1 dg*
   - *Layer 2* as *Metal2 dg*
8. Select the *Rule* check box.

   Ensure that all fields are populated with values.

© 2023 Cadence Design Systems, Inc.

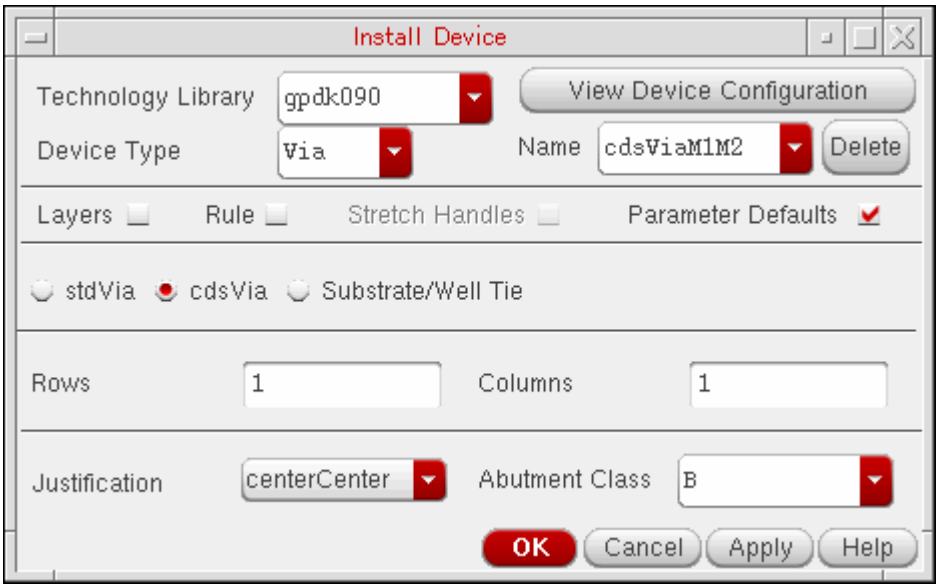
Notice that when you select the *Rule* check box, the Rules Browser window opens. If you place the pointer in any field, an arrow in the Rules Browser window indicates the physical area where the rule applies, as shown in the image.



9. Select the *Parameter Defaults* check box.

© 2023 Cadence Design Systems, Inc.

Ensure that all fields are populated with values. The *Abutment Class* field is optional for a `cdsVia` device.



10. Click *OK* or *Apply*.

If you click *Apply*, a message appears in the CIW to let you know if there are any errors or whether the `cdsVia` device has been created. If the `cdsVia` device already exists in the technology file, you are prompted to confirm if you want to overwrite the existing `cdsVia` device.

If you click *OK*, in addition to the message that appears in the CIW, you are prompted to confirm if you want to save the new rules to the technology file and then save the technology library to the disk. If you click *Yes*, the *Install Device* form closes.

You can check the `DEVICES` section in the technology file to confirm that the `cdsVia` device has been defined as per the values specified in the Install Device form.

Related Topics

Adding the cdsVia Device to the validVias Constraint

Defining and Installing Devices

Adding the cdsVia Device to the validVias Constraint

To make an installed cdsVia device available in the Create Via form, you need to add it to the `validVias` constraint of the required constraint group in the technology file. You can do this

© 2023 Cadence Design Systems, Inc.

by manually editing the technology file and then reloading it or by using the Process Rule Editor to edit the technology file.

You can add the cdsVia device to the constraint group that contains the purpose-aware rules. The following image shows the example of the user-defined `LPPtestCG` constraint group, containing various purpose-aware `minWidth` and `minSpacing` rules.

```
 ;( group       [override] )
 ;( -----       ---------- )
  ( "LPPtestCG" nil
    interconnect(
     ( validLayers  ( ( Metal1  highVoltage) ( Metal2  highVoltage)
( Metal3  highVoltage) ( Metal1  lowVoltage) ( Metal2  lowVoltage)
( Metal3  lowVoltage) Via1  Via2  ) )
    ) ;interconnect
    spacings(
     ( minWidth                  "Metal1"      0.3 )
     ( minWidth                  ("Metal1" "lowVoltage")    1.0 )
     ( minWidth                  ("Metal1" "highVoltage")   2.0 )
     ( minWidth                  "Metal2"      0.4 )
     ( minWidth                  "Metal3"      0.5 )
     ( minWidth                  "Via1"        0.3 )
     ( minWidth                  ("Via1" "lowVoltage")      0.4 )
     ( minWidth                  ("Via1" "highVoltage")     0.6 )
     ( minWidth                  "Via2"        0.3 )
     ( minSpacing                "Metal1"      0.3 )
     ( minSpacing                "Metal2"      0.4 )
     ( minSpacing                "Metal3"      0.5 )
     ( minSpacing                "Via1"        0.2 )
     ( minSpacing                ("Via1" "lowVoltage")       0.4 )
     ( minSpacing                ("Via1" "highVoltage")      0.6 )
     ( minSpacing                "Via2"        0.3 )
     ( minSpacing                "Via3"        0.4 )
     ( minSpacing                "Via4"        0.5 )
     ( minSpacing                "Via5"        0.6 )
      .....
      .....
    ) ;spacings
  ) ;LPPtestCG
```

To edit the technology file by using the Process Rule Editor:

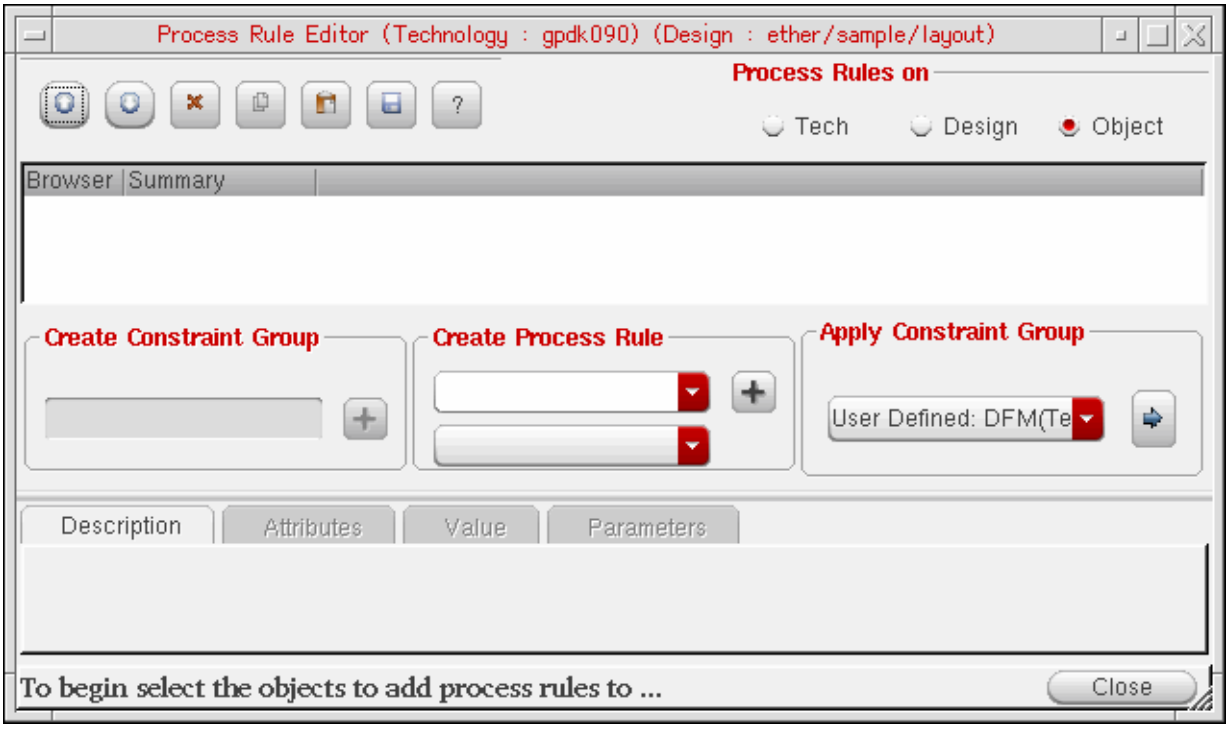
© 2023 Cadence Design Systems, Inc.

1. In the layout window, choose *Window – Assistants – Constraint Manager*.

   The *Constraint Manager* assistant opens.

2. Click the *Process Rules Editor* icon .

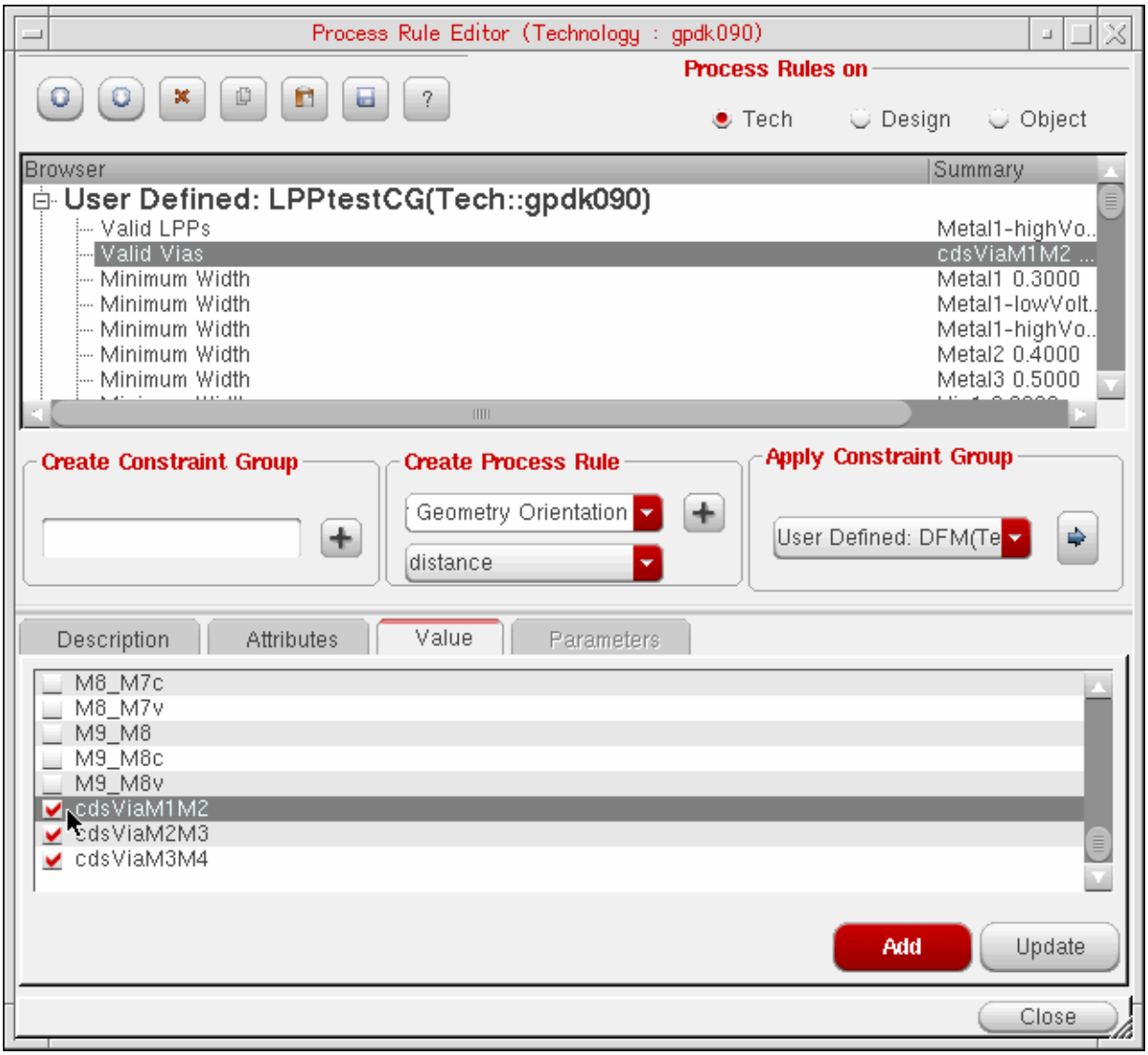
   The Process Rules Editor window opens.



3. Select the *Tech* option in the *Process Rules on* section.
4. In the *Browser* section, expand the constraint group to which you want to add the cdsVia.
5. Click the *Value* tab in the lower section of the window.
6. Scroll to select the installed cdsVia device.

© 2023 Cadence Design Systems, Inc.

For example, the image shows the `cdsViaM1M2`, `cdsViaM2M3`, and `cdsViaM3M4` cdsVias selected.



7. Click *Update*.
8. Click *Close*.

© 2023 Cadence Design Systems, Inc.

The following image shows the `cdsViaM1M2`, `cdsViaM2M3`, and `cdsViaM3M4` cdsVias added to the `validVias` constraint in the `LPPtestCG` constraint group.

```
;( group       [override] )
;( -----       ---------- )
  ( "LPPtestCG" nil
    interconnect(
     ( validLayers  ( ( Metal1  highVoltage) ( Metal2  highVoltage) ( Metal3
highVoltage) ( Metal1  lowVoltage) ( Metal2  lowVoltage) ( Metal3  lowVoltage) V
Via2  ) )
     ( validVias     (cdsViaM1M2  cdsViaM2M3 cdsViaM3M4) )
    ) ;interconnect
  ) ;LPPtestCG
```

To instantiate the installed `cdsVia` device in your design, you can create a cdsVia.

Related Topics

Setting Up Via Creation Options

Creating a cdsVia

Installing a cdsVia Device in the Technology File

Via Creation

Setting Up Via Creation Options

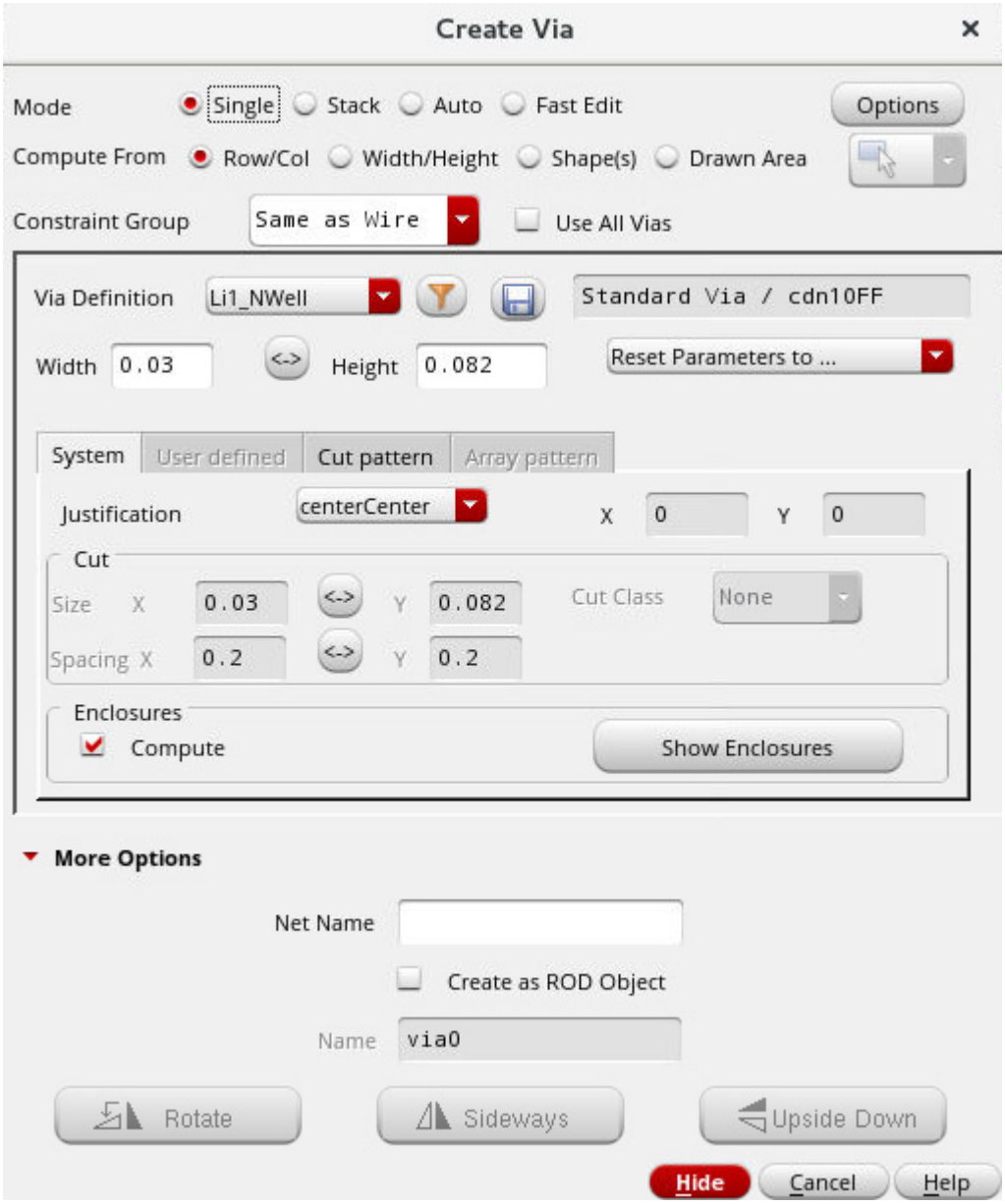
Before creating a single, stack, or auto via, you can set up the viaDefs to be available for via creation.

To set up via creation options:

1. In the layout window, choose *Create – Via* or click the *Create Via* icon  on the *Create* toolbar.

© 2023 Cadence Design Systems, Inc.

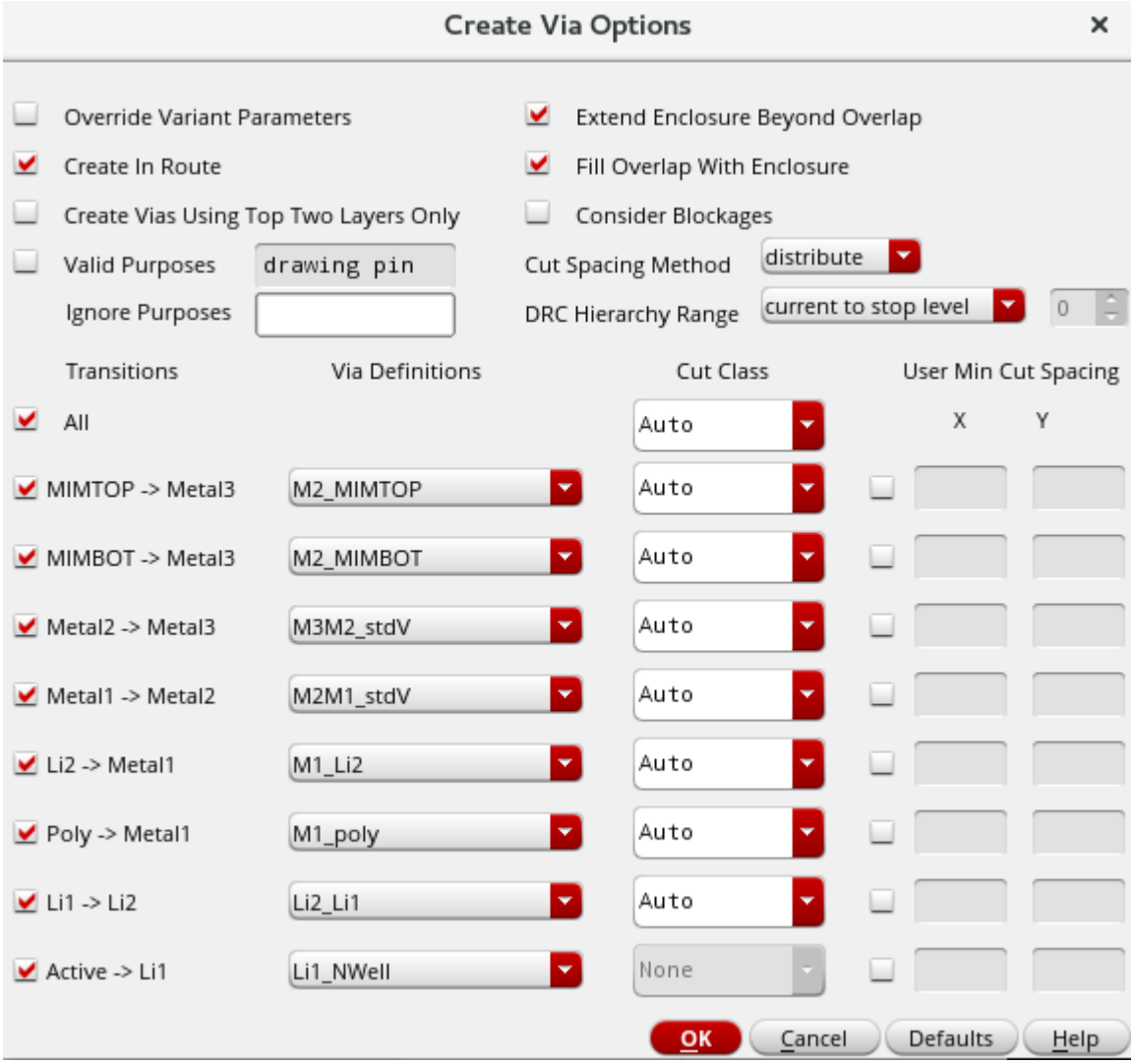
The Create Via form opens.



2. Click the *Options* button. This button is available in all the via modes.

© 2023 Cadence Design Systems, Inc.

The Create Via Options form opens.



3. Select the *Override Variant Parameters* check box to allow the update of via variant parameter values.
4. Select the *Extend Enclosure Beyond Overlap* check box if you want the via enclosures to extend beyond the overlap region of the intersecting shapes.

   In the *Single* and *Stack* mode, use this check box along with selecting the *Compute From – Shape(s)* check box on the Create Via form.
5. Select the *Fill Overlap With Enclosures* if you want the via enclosure to extend and cover the complete overlap region between two intersecting shapes in *Auto* mode.

© 2023 Cadence Design Systems, Inc.

6. Select other via options such as *Create Vias Using Top Two Layers Only* and *Cut Spacing Method* as required.
7. From the *Cut Class* list, select the cut class you want to be available by default for the respective viaDef/via variant in *Single* mode. The selected cut class is also used by the *Stack* and *Auto* automatic via creation engines.
8. From the *Via Definitions* list, select a viaDef/via variant for each layer pair. The selected viaDef/via variant is also used by the *Stack* and *Auto* automatic via creation engines.

Related Topics

Create Via Form

Create Via Options Form

Creating a Via with Arrayed Via Cuts

Creating a Via Automatically in Single and Stack Modes

Creating a Single Via

Use *Single* mode option in the Create Via form to place a via in a cellview. By default, in *Single* mode, the via width, length, row and column spacing, and the enclosure values are derived based on the calculation mode set in the *Via Parameters Calculation Mode* list on the Layout Editor Options form. For standard and custom vias, you can update values for origins, enclosures, and via arrays by using the *System* tab on the Create Via form. Via parameter values are calculated such that via metal shapes are minimized.

Creating a Via with Arrayed Via Cuts

An array of vias is a group of vias with a single origin. You can create such an array to connect two wide paths or to create part of a large transistor.

To place a standard or custom via with arrayed via cuts:

1. In the layout window, choose *Create – Via* or click the *Create Via* icon  on the *Create* toolbar.

   Alternatively, right-click any free space on the design canvas to display the *Layout* shortcut menu and select *Create – Via.*

   The Create Via form opens.

© 2023 Cadence Design Systems, Inc.

2. Select *Single* as the *Mode* for creating vias.
3. Optionally, use the *Options* button to set up the default viaDefs to be available during the via creation and to enable overriding of the via variant parameters in the Create Via Options form.
4. Select a constraint group from the *Constraint Group* list on the Create Via form. This controls the vias listed in the *Via Definition* list. When the *Use All Vias* check box is enabled, all vias are listed in the *Via Definition* list irrespective of the constraint group selected.
5. Select a viaDef or a via variant from either the *Via Definition* list or the Via Definition Selection widget, which displays when you click the *Filter* icon.
   a. To use the selection widget, click the *Filter* icon to open the Via Definition Selection widget.
   b. Specify a search expression in the *Filter* field using wild card characters.
   c. Select a *Sort By* mechanism.
   d. Select the layers in the *From* and *To* lists for specifying the layers between which the viaDef or via variant should be available.
   e. Select one of the filtered viaDefs or via variants from the list box. The via variants are listed as child nodes of the parent standard or custom viaDefs. Press `Esc` if you do not want to select a viaDef from the list.

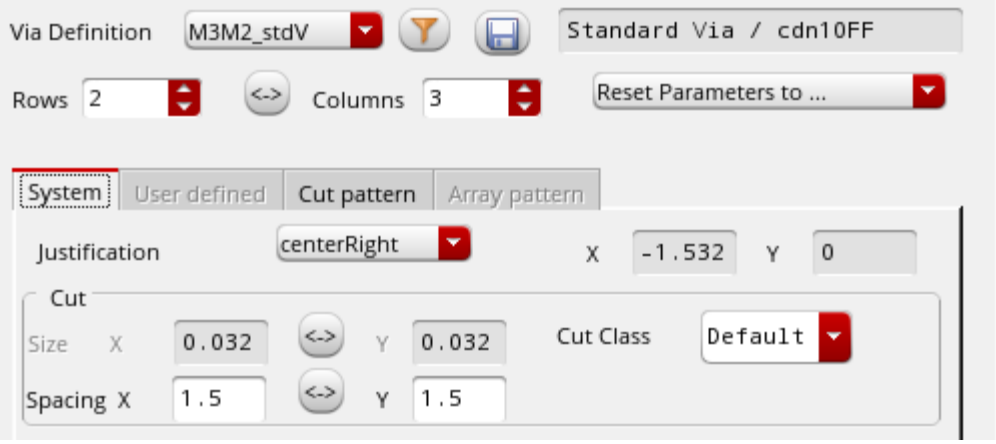
   The Via Definition Selection widget closes and the selected viaDef or via variant displays in the *Via Definition* list in the Create Via form. The *Via Type / Technology Library* read-only field updates to show whether the selected viaDef is a standard or custom viaDef or is a standard or custom via variant and the name of the technology library in which the viaDef is defined.
6. Click *Reset Parameters to* if you want to revert to the default values. The reset value you select depends on whether or not *Compute Enclosures* is enabled in the *System* tab.
7. On the *System* tab, choose a *Justification* setting to change the via origin.

   When set to `offset`, the *X* and *Y* fields are editable and display the offset in the X and Y direction, respectively.
8. If the `cutClass` parameter is specified in the `viaSpacing` rule, select a cut class or a cut size from the *Cut Class* list.

   If a cut class for the viaDef/via variant in the *Via Definitions* list is set up in the Create Via Options form, the value is selected by default in the *Cut Class* list in the *Create Via* form.

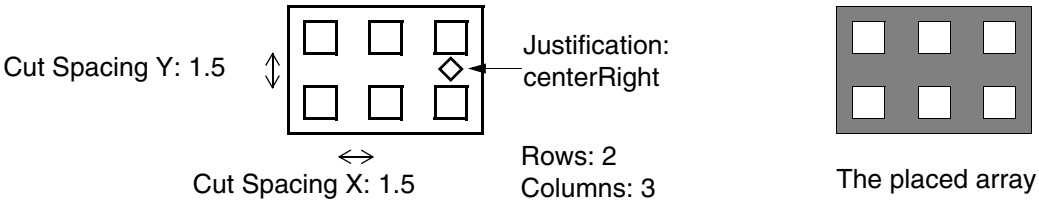
© 2023 Cadence Design Systems, Inc.

9. Update the number of *Rows* and *Columns*, if required.
10. Specify new values for *Spacing X* and *Y* if you want to change the spacing between the center of each via cut.



11. To associate the via to a particular net, type the name of the net in the *Net Name* field under *More Options*.
12. To create the via as a ROD object, select *Create as ROD Object*. Specify a name for the via ROD object in the *Name* field.
13. Click on the canvas where you want to place the via array.

    If the size of a via cut is not an even number of the manufacturing grid unit, the via cannot be centered around the origin. In such a case, the via origin is changed to ensure that via cut shapes are on the grid. The via cuts are shifted up for vertical adjustment and to the right for horizontal adjustment by an offset of half of the manufacturing grid.



14. Place as many vias as required.

    If a via has more than 100 000 cuts, only a box is displayed on the canvas. Else, all the cuts of the via are displayed on the canvas.
15. Press *Esc* or click *Cancel* in the Create Via form to finish creating vias.

© 2023 Cadence Design Systems, Inc.

Related Topics

Create Via Form

Create Via Options Form

Setting Up Via Creation Options

Creating or Updating a Via Variant

Via Creation

Layout Editor Options Form

Creating a cdsVia

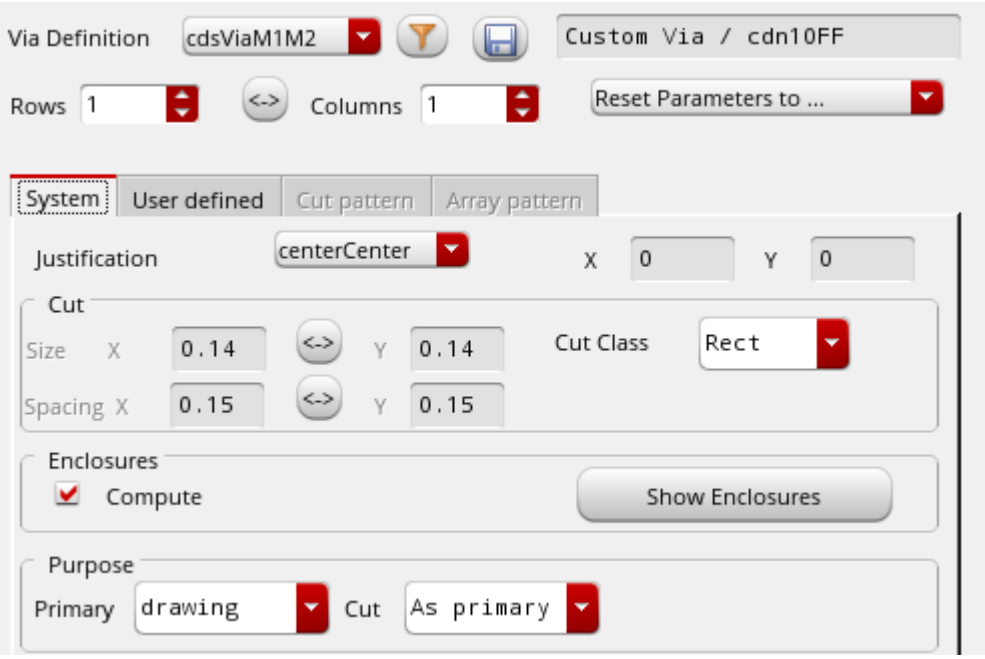
You can create a cdsVia in your design if you have installed the `cdsVia` device in the technology file.

To add a cdsVia to a design:

1. In the layout window, open the Layout Editor Options form and ensure that the required constraint group is selected in the *Via* list in the *Create Via* section. You can access the form by using *Options – Editor*.
2. Choose *Create – Via* to open the Create Via form.
3. Select *Single* as the *Mode* for creating the cdsVia.
4. Select a cdsVia from the *Via Definition* list.

© 2023 Cadence Design Systems, Inc.

The *Purpose* section is added to the Create Via form if you select a cdsVia from the *Via Definition* list.



5. In the *Purpose* section, select another purpose from the *Cut* list.

   The cut values are automatically re-calculated. These values are picked from the entries in the constraint group.

6. Click the *Show Enclosures* button

   The enclosure values update automatically depending on the purpose you select from the *Cut* list in the *Purpose* section and based on the constraints defined in the constraint group.

   Note that implant layer enclosures cannot be specified for custom vias.

   Enclosure parameters are supported on Pcell vias.

7. Click on the canvas where you want to place the cdsVia.
8. Place as many vias as required.
9. Press *Esc* or click *Cancel* in the Create Via form to finish creating cdsVias.

Related Topics

Installing a cdsVia Device in the Technology File

© 2023 Cadence Design Systems, Inc.

Creating a Single Via

Creating a Via with Arrayed Via Cuts

Layout Editor Options Form

Create Via Form

Via Creation

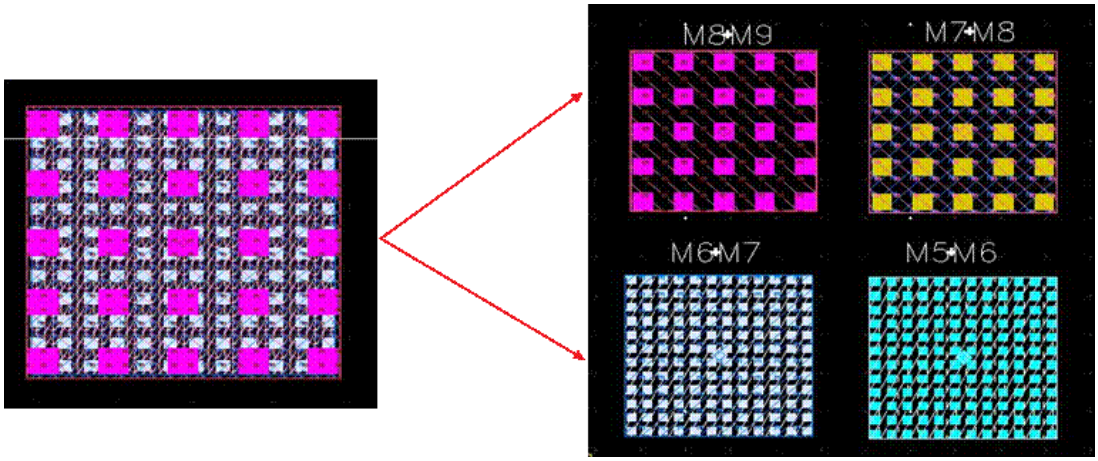
Creating a Stack Via

You can create a via stack by physically stacking vias belonging to adjacent layers on top of each other. The footprint of each via in the via stack matches the footprint of the top via layer. This ensures that the metal shapes across all vias in the via stack coincide. Vias in a via stack have the same origin. If a via stack has connectivity, all vias in that via stack have the same net name.

By default, in *Stack* mode, the via width, length, row and column spacing, and the enclosure values are derived based on the calculation mode set in the *Via Parameters Calculation Mode* list in the Layout Editor Options form.

You can use via variants for creating a via stack. Such via stacks can be edited by using both the *Edit Via Properties* form and the *Property Editor* assistant.

In the following example, the figure on the left is an M5M9 via stack created using the *Stack* option in the Create Via form. The figure on the right shows the constituent vias `M5M6`, `M6M7`, `M7M8`, `M8M9` of the via stack.



To create a via stack:

© 2023 Cadence Design Systems, Inc.

1. In the layout window, choose *Create – Via* or click the *Create Via* icon  on the *Create* toolbar.

   Alternatively, right-click any free space in the design canvas to display the *Layout* shortcut menu and select *Create – Via.*

   The Create Via form opens.

2. Select the *Stack* option.

   The lower portion of the form updates to display the *Start Layer* and *End Layer* lists.

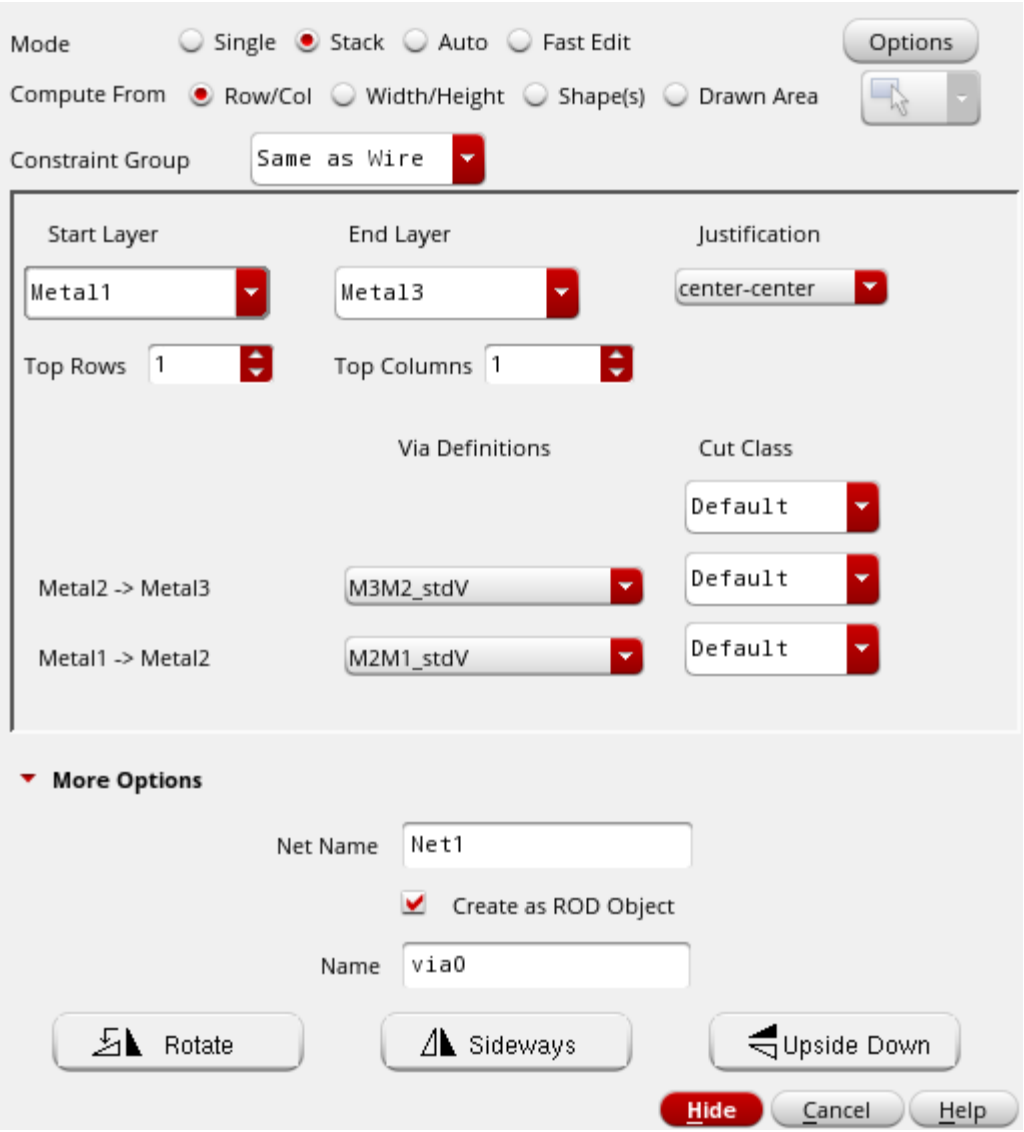
3. Use the *Options* button if you want to set up the default viaDefs to be available during the via creation and to enable overriding of the via variant parameters.

4. Specify the *Net Name* you want to associate with the via stack, in the *More Options* section of the form. If the specified net does not exist, a new net is created. If you leave this field blank, no net is associated with the via stack.

5. Optionally, select *Create as ROD Object* to create the via stack as a ROD object. You can specify a name in the *Name* field.

© 2023 Cadence Design Systems, Inc.

6. In the *Start Layer* and *End Layer* lists, select the layers between which you want to place the via stack. After you specify both *Start Layer* and *End Layer* values, the form updates to display all the vias available between the two selected layers.



7. Specify the number of *Top Rows* and *Top Columns* you want in the via stack.
8. Select a viaDef for each layer pair from the *Via Definitions* list.

© 2023 Cadence Design Systems, Inc.



If you update the technology file while the Create Via form is still active with the Mode set to Stack, your changes to the technology file will not reflect in the form. To ensure the form gets updated based on your changes in the technology file, close and reopen the form.

9. Click on the canvas where you want to place the via stack.
10. To rotate or mirror the via stack as you place it, click *Rotate*, *Sideways*, or *Upside Down* and then place the via stack. After placing the via stack, each time you click any one of these buttons, the via stack is further rotated or mirrored.

    Alternatively, you can click the middle mouse button to replicate *Rotate,* press `Ctrl` and click the middle mouse button to replicate *Sideways*, or press `Shift` and click the middle mouse button replicate *Upside Down.*

Using the Property Editor assistant, you can edit the layers of the via that is part of the via stack from the via stack. You cannot edit the layers from the via object.

In addition to *Stack* mode in the Create Via form, you can use any of the following ways to create a via stack:

- *Auto* mode on the Create Via form on two intersecting and non-adjacent routing layers
- *Single* mode on the Create Via form and stacking the vias manually
- *Create – Wire* command while switching from one routing layer to a non-adjacent routing layer

Related Topics

Layout Editor Options Form

Create Via Form

Attributes in the Edit Via Stack Properties Form

Creating or Updating a Via Variant

Setting Up Via Creation Options

Creating a Via with Arrayed Via Cuts

Via Creation

© 2023 Cadence Design Systems, Inc.

Creating or Updating a Via Variant

You can create via variants and update them as required.

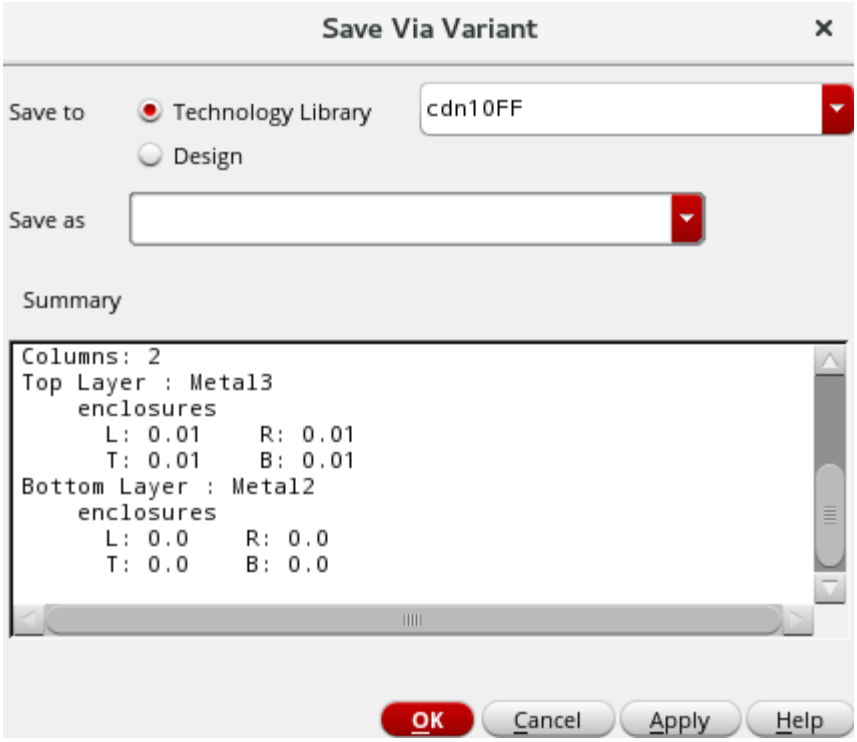
- Creating a Via Variant
- Updating a Via Variant

Creating a Via Variant

To create a via variant:

1. In the layout window, choose *Create – Via* to open the Create Via Form.
2. Set *Mode* as *Single*.
3. Select a standard or custom viaDef from the *Via Definition* list.
4. On the *System* tab, change one or more via parameters, such as number of rows and columns, cut spacing, offset, or justification, as required.
5. Click the *Save Via Variant* icon.

   The Save Via Variant form opens.



© 2023 Cadence Design Systems, Inc.

6. Choose where to save the via variant, either in the *Technology Library* or in the *Design*. If you want to save the via variant in the technology library, select one from the list.
7. Specify a name for the via variant in the *Save as* field.
8. Review the updated parameters in the *Summary* read-only list box.
9. Click *OK*.

   If you specify an existing viaDef name, a message is displayed in the CIW that a viaDef with the specified name already exists.

   The created via variant appears in the *Via Definition* list as a child node of the selected parent viaDef.
10. In the *Via Definition* list, expand the node and select the saved via variant.

Updating a Via Variant

To update an existing via variant:

1. In the Create Via Form, select *Mode* as *Single*.
2. Click the *Options* button.

   The Create Via Options form opens.
3. Ensure that the *Override Variant Parameters* check box is selected.

   You cannot update the via parameters of an existing via variant if the *Override Variant Parameters* check box is not selected. When this check box is not selected, the fields for specifying the cut class, cut rows and columns, and enclosure values are grayed out.
4. Click *OK*.
5. Select a standard or custom via variant from the *Via Definition* list in the *Create Via* form.
6. On the *System* tab, change one or more via parameters, such as number of rows and columns, cut spacing, offset, or justification, as required.
7. Click the *Save Via Variant* button.

   The Save Via Variant form opens.
8. Choose where to save the via variant, either in the *Technology Library* or in the *Design*. If you want to save the via variant in the technology library, select one from the list.
9. Specify a name for the via variant or select one from the *Save as* list.

© 2023 Cadence Design Systems, Inc.

10. Review the updated parameters in the *Summary* read-only list box.
11. Click *OK*.
    - If you specify the same name as that of an existing via variant name, a message box appears to confirm whether you want to overwrite the existing via variant. You can click *Yes* or *No* on the message box to specify your choice. The via variant is not updated if you click *No*.
    - If you create a via variant that has the same viaDef and parameters as another existing via variant with a different name, a message box appears to confirm whether you want to rename the existing via variant. You can click *Yes* or *No* on the message box to specify your choice. The via variant is not renamed if you click *No*.

    The updated via variant appears in the *Via Definition* cyclic list as a child node of the selected parent viaDef.
12. In the *Via Definition* cyclic list, expand the node and select the saved via variant.

Related Topics

Create Via Form

Create Via Options Form

Save Via Variant Form

Via Creation

Creating a Via Automatically in Single and Stack Modes

In the Create Via form, use the *Compute – From Shape(s)* check box to generate vias automatically in *Single* or *Stack* modes when you click the overlap region of intersecting shapes.

1. In the layout window, choose *Create – Via* or click the *Create Via* icon  on the *Create* toolbar.

   Alternatively, right-click any free space in the design canvas to display the *Layout* shortcut menu and select *Create – Via.*

   The Create Via form opens.
2. Use the *Options* button if you want to set the via creation options using the Create Via Options form. For example, you can enable *Extend Enclosure Beyond Overlap* to extend via enclosures beyond the overlap region between two intersecting shapes.

© 2023 Cadence Design Systems, Inc.

3. Select *Mode* for creating vias in the Create Via form.

   a. If you select *Single* mode, do the following:

      i) Select the *Compute From Shape(s)* check box.

         Notice that the fields for specifying the *Cut Class*, cut *Rows* and *Columns*, and enclosure values are grayed out.

      ii) Select a viaDef from the *Via Definition* field.

         Ensure that the selected viaDef matches at least one of the two intersecting shapes for generating the automatic via.

      iii) In the *Cut* section, specify the *Width*, *Length*, *Spacing X* and *Y* values.

   b. If you select *Stack* mode, do the following:

      i) Select the *Compute – From Shape(s)* check box.

         Notice that the fields for specifying *Top Via Rows* and *Top Via Columns* are grayed out.

      ii) Select the start and end layers for creating the via stack.

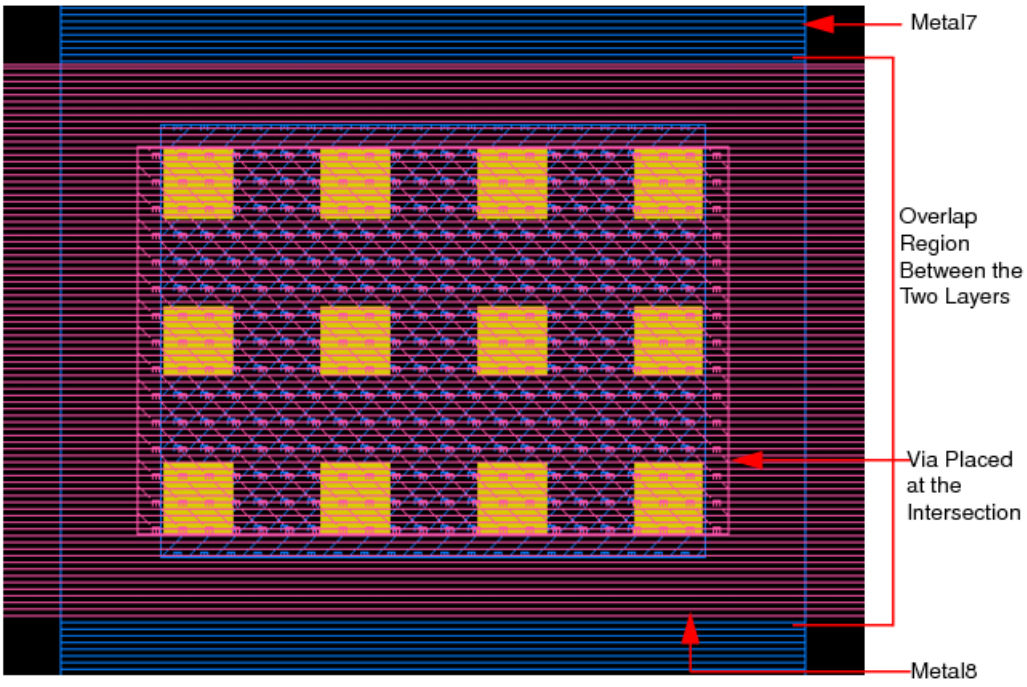
      iii) Select a viaDef for each layer pair from the *Via Definition* list.

4. Click in the overlapping area of the intersecting shapes on the canvas to create a via or stacked via.

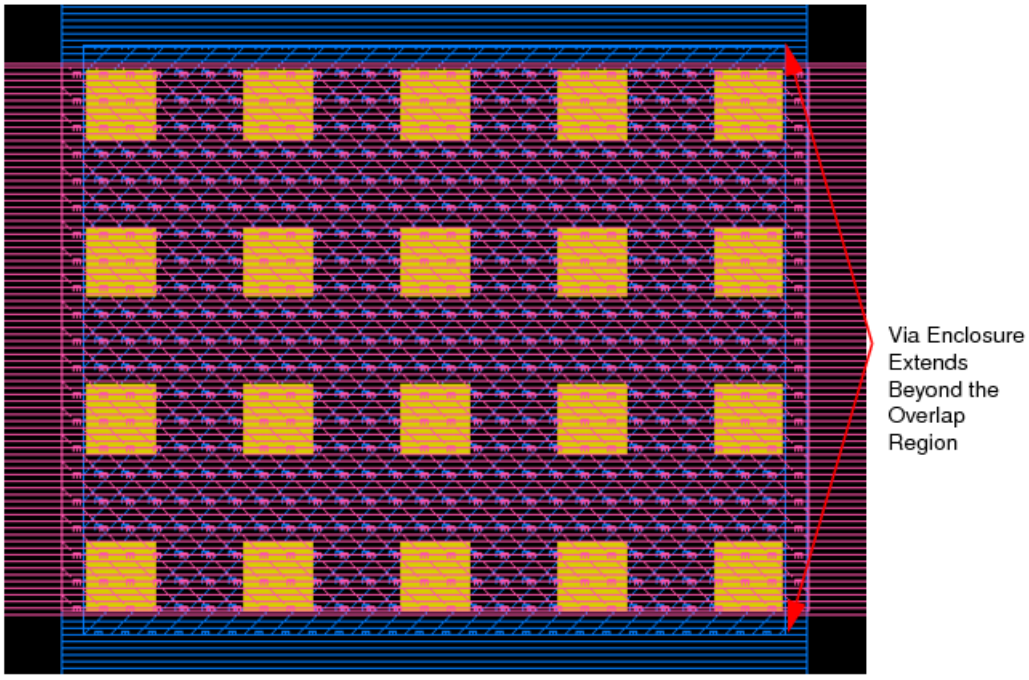
   The following images show an example of a *Single* mode via created at the intersection of two shapes when the *Compute – From Shape(s)* check box is selected. The vias in the first image are generated when the *Extend Enclosure Beyond Overlap* check box is not selected in the Create Via Options form. The second image illustrates the

© 2023 Cadence Design Systems, Inc.

enclosure extension when the *Extend Enclosure Beyond Overlap* check box is selected.



Extend Enclosure Beyond Overlap is off.



Extend Enclosure Beyond Overlap is on.

© 2023 Cadence Design Systems, Inc.

5. Continue to place vias at the required locations.
6. Press `Esc` or click *Cancel* on the form to finish creating vias.

Note: Auto via mode supports shapes like circles, ellipses, and donuts.

A statistics report is generated for auto via in the form of a table in the CIW. This report is best viewed when the default CIW font is mono.

```
(VIA-1010): Running the auto via command at point (1.5, 3.5).
*****************************************************************************
*                          Auto Via Statistics                              *
* Point : (1.5, 3.5)                                                        *
* Run Time = 0.01 sec                                                       *
* Total number of vias created: 6                                           *
*****************************************************************************
#
# Summary
=============================================================================================================================
| Stack |  Metal Overlaps   |      Via      |    Initial Overlaps     |        Final Results        |  Percent   |
| Depth |  Bottom - Top     |               |   Total   | Unconnected | Newly Connected | Connected |            |
-----------------------------------------------------------------------------------------------------------------------------
|   2   |  Metal1 - Metal3  |  Via1 - Via2  |         1 |           1 |               1 |         1 |   0 -> 100 |
-----------------------------------------------------------------------------------------------------------------------------
|                Totals                     |         1 |           1 |               1 |         1 |   0 -> 100 |
=============================================================================================================================
t
```

Auto via considers stopped layers as defined by the connectivity extractor in the `virtuosoDefaultExtractorSetup` constraint group. For more information, see Specifying Layer-Purpose Pairs in the validLayers Constraint.

Related Topics

Create Via Form

Setting Up Via Creation Options

Create Via Options Form

Via Creation

Auto Mode Via Creation on Intersecting Shapes

You can automatically place a single via or a stacked via in the overlap area of intersecting polygons, rectangles, paths or path segments. The via is created if there are available valid vias in the current constraint group in the technology database that connect the two layers on which the shapes exist. If there is a taper type constraint in the constraint group in the technology database, the vias are added to the valid vias list. When you place vias with *Auto*

© 2023 Cadence Design Systems, Inc.

option selected in the Create Via form, the via origin snaps to the center of the overlap, irrespective of the *Snap Spacing* settings in the Display Options form.

- When a via is created in *Auto* mode or when the *Compute – From Shape(s)* check box is selected in the Create Via form, by default the *Minimum Rules* calculation mode is applied. However, you can change the calculation mode by enabling the createViaAlwaysUsesDefinedCalcMode environment variable.
- Auto via mode supports polygonal overlaps. If a non-metal via is created in a polygonal overlap, such a via is non-editable. In case the overlap area is a path or path segment, the via is created in every path or path segment subpart.
- When shapes abut, auto via mode creates a via only when the shapes are path or path segment and when they are connected by their begin or end center line point.

In case there are multiple shapes on different layers overlapping at the same place, the auto via mode considers the adjacent metals. For example, if there is an `M1`, `M2`, and `M4` shape at the same location, auto via mode considers two different overlaps, `M1`-`M2` and `M2`-`M4`. It does not consider `M1`-`M4` as an overlap, unless `M2` is not visible in the Palette assistant.

By default, in auto via mode, the via width, length, row and column spacing, and the enclosure values are derived based on the calculation mode set in the *Via Parameters Calculation Mode* list in Layout Editor Options form.

By using *Auto* mode in the Create Via form, you can do the following:

- Automatically place a via with a single click
- Automatically place a via based on the connectivity information of the intersecting shapes

Auto vias cannot be created at the click point within an intersection area if no valid via or constraint group is defined in the technology file for connecting the intersecting layers. You can either edit the technology file to include the definition for vias connecting the intersecting layers or change the layers of the intersection objects to match a via defined in your technology file. An appropriate warning message is issued in the CIW for each case.

Related Topics

Layout Editor Options Form

Placing a Via Automatically with a Single Click

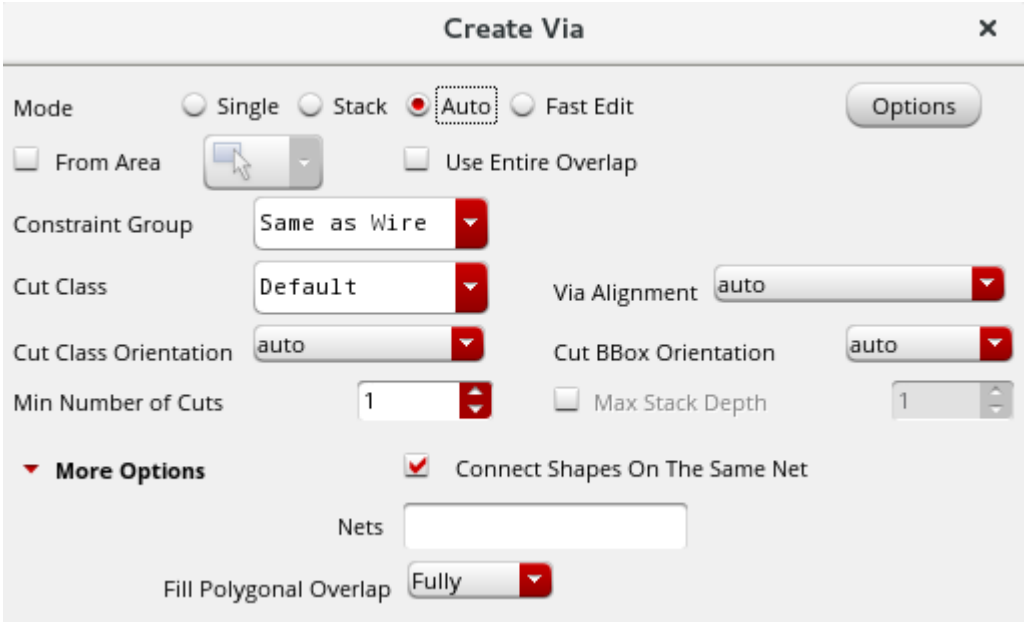
Placing a Via Automatically Based on Connectivity Information of Intersecting Shapes

© 2023 Cadence Design Systems, Inc.

Placing a Via Automatically with a Single Click

To create a via using *Auto* mode and automatically place the via with a single click:

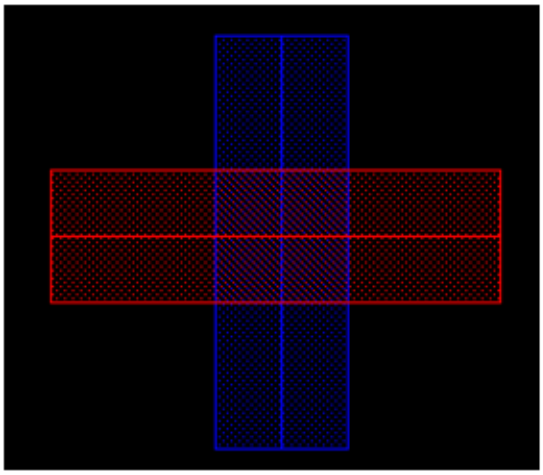
1. In the layout window, choose *Create – Via* to open the Create Via form.
2. Select *Auto* option for *Mode*.



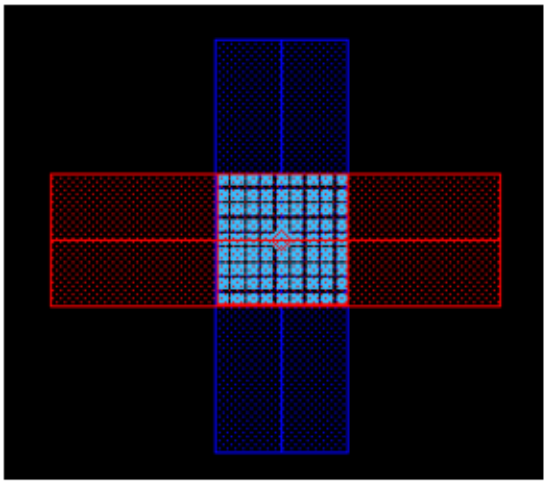
3. Use the *Options* button if you want to open the Create Via Options form to set the default viaDefs to be available during via creation and to enable overriding of the via variant parameters.
4. Type the net name in the *Net Name* field of the Create Via form to associate the via to a particular net.

© 2023 Cadence Design Systems, Inc.

5. On the canvas, click on the overlapping area of the intersecting shapes.



Intersecting Paths

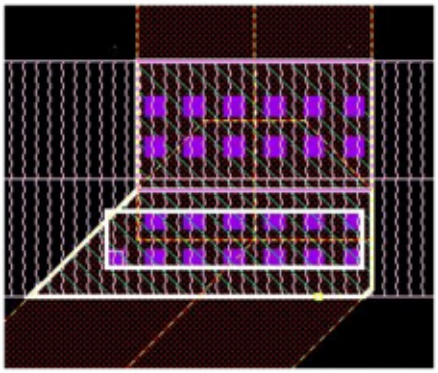


Via Created at the Path Intersection By Using the Auto Mode

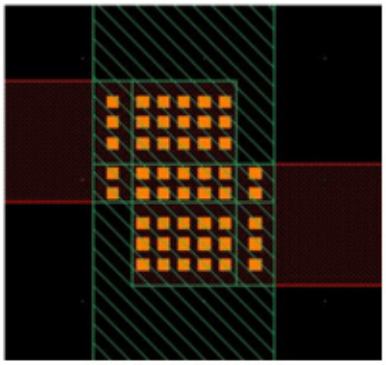
5. Continue to place vias at the required locations, if needed.
6. Press `Esc` or click *Cancel* in the form to complete creating vias.

© 2023 Cadence Design Systems, Inc.

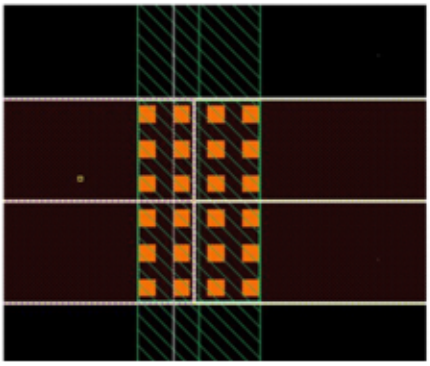
The following figures illustrate a few cases of auto via generation.



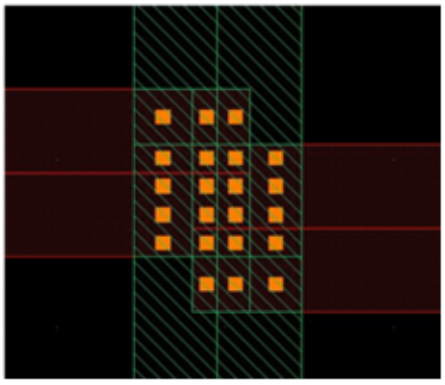
Non-orthogonal pathSegs; Non-rectangular metal intersection area



Non-rectangular intersection area; Overlapping shapes for one metal layer

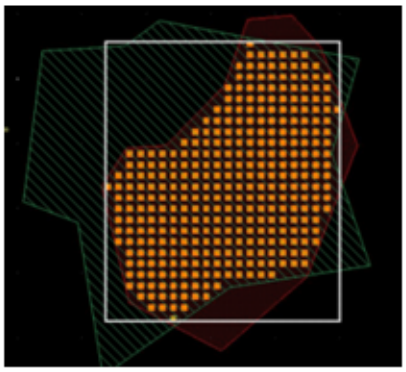


Butting edges of pathSegs at metal intersection, rectangular metal intersection area

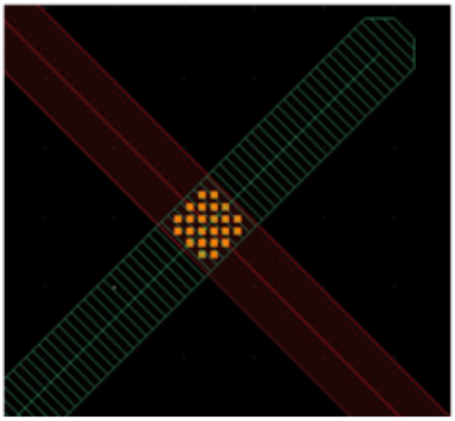


Non-rectangular metal intersection area; overlapping pathSegs for one metal layer

© 2023 Cadence Design Systems, Inc.



Intersection of complex polygon shapes

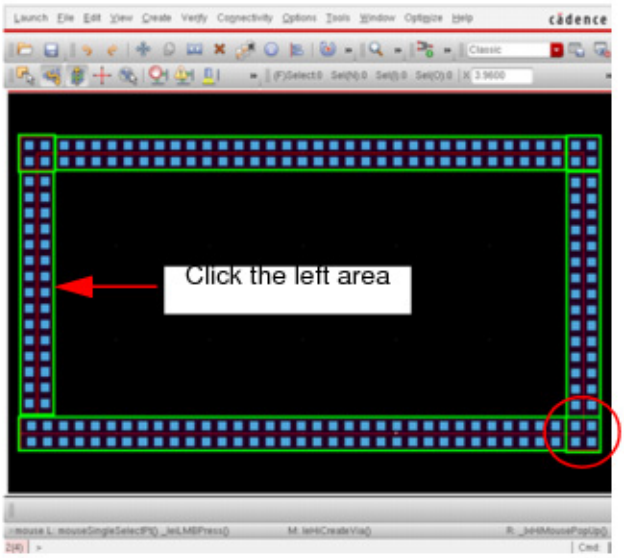


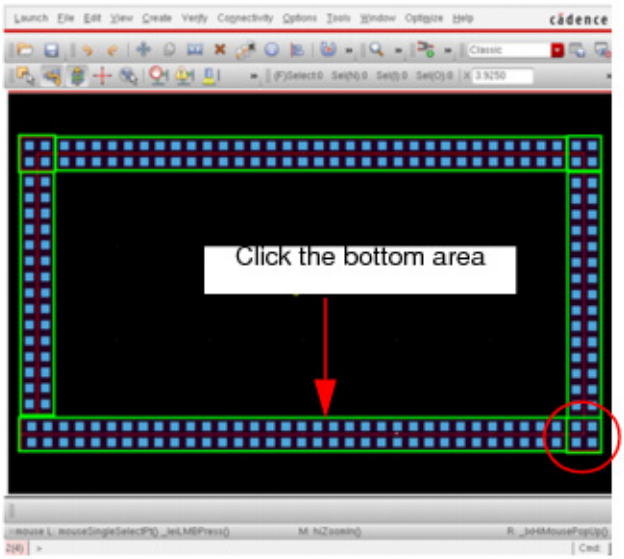
Intersection of 45 degree pathSegs

When the auto via mode creates polygonal vias, the metal shape is reduced to a single-cut-sized shape under the lower-most left cut. The via cuts must be covered by the existing metal shapes in the design. For stacked vias, extra metal shapes for the intermediate layers are created, which are then associated with the vias as child shapes. Editing polygonal via parameters by using property editor or properly assistant or modifying the via geometries with Chop or Stretch commands is not supported.

© 2023 Cadence Design Systems, Inc.

For polygons or pathsegs, the coverage and cuts between vias are always same, irrespective of where you click on the shape.





If there are multiple overlapping shapes, you can center-click and drag to select an area that encloses the required overlapping shapes. Vias are generated at all the intersecting shapes.

In auto via mode, a via is created in the intersection between the overlapping shapes and the area selected by center-click and dragging the pointer.

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

Setting Up Via Creation Options

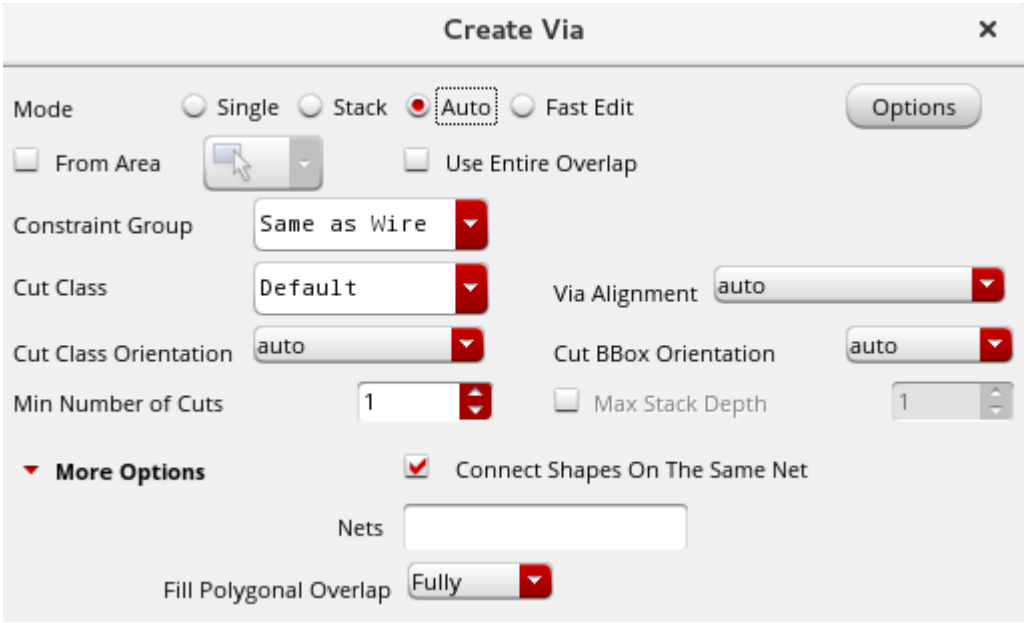
Create Via Options Form

Via Creation

Placing a Via Automatically Based on Connectivity Information of Intersecting Shapes

To create a via using *Auto* mode and automatically place the via based on the connectivity information of intersecting shapes:

1. In the layout window, choose *Create – Via* to open the Create Via form.
2. Select the *Auto* option for *Mode*.
3. Use the *Options* button to open the Create Via Options form to set the default viaDefs to be available during via creation and to enable overriding of the via variant parameters.
4. Select the *Connect Shape On The Same Net* check box in the Create Via form.



5. Optionally, to place the vias on only those intersecting shapes that belong to a specific net, enter the particular net name in *Net(s) Name*.
6. On the canvas, click the overlapping area of the intersecting shapes. Alternatively, click and drag the middle mouse button to select the intersecting shapes.

© 2023 Cadence Design Systems, Inc.

A via is placed in the intersecting area based on the connectivity information of the intersecting shapes.

5. Continue to place vias at the required locations.
6. Press `Esc` or click *Cancel* in the form to complete creating vias.

Related Topics

Create Via Form

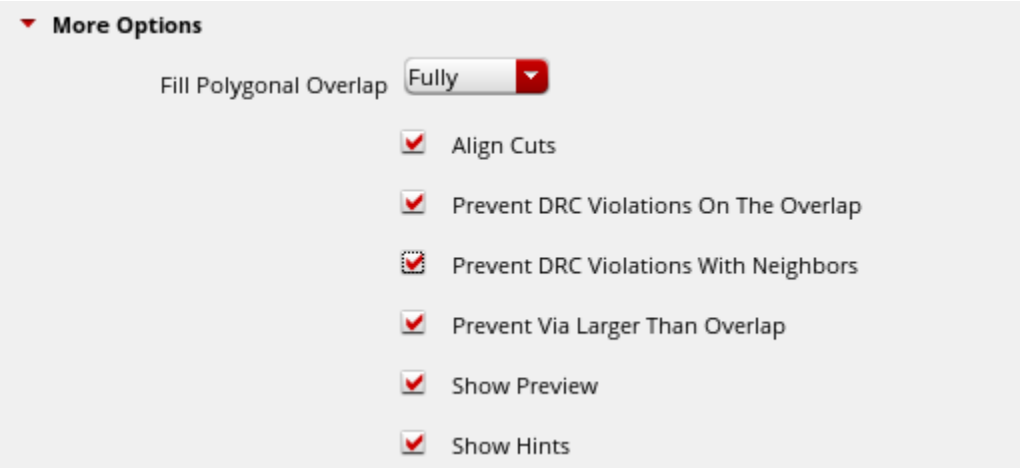
Setting Up Via Creation Options

Create Via Options Form

Via Creation

Auto and Fast Edit Via Mode Options

Using the *Auto* and *Fast Edit* via creation modes in the Create Via form, you can specify settings to prevent DRC violations, prevent creation of a via larger than overlap, show preview, and display hints during via creation.



- Prevent DRC Violations On The Overlap
- Prevent DRC Violations With Neighbors
- Prevent Via Larger Than Overlaps
- Show Preview

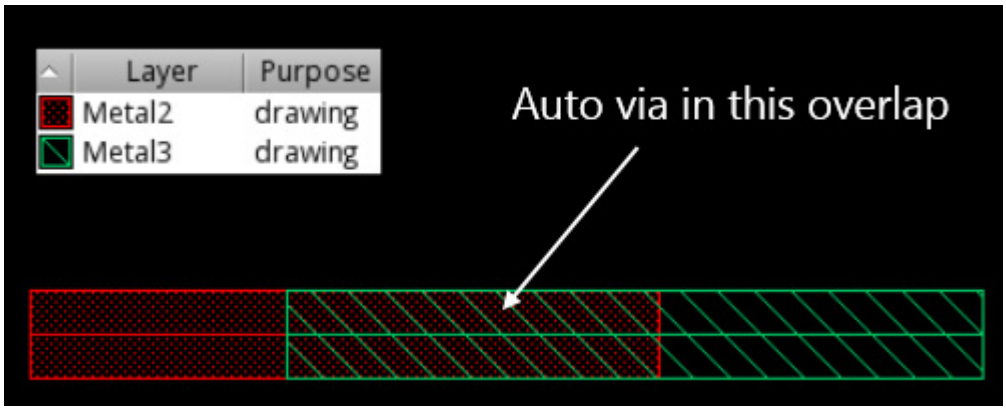
© 2023 Cadence Design Systems, Inc.

- Show Hints

Prevent DRC Violations On The Overlap

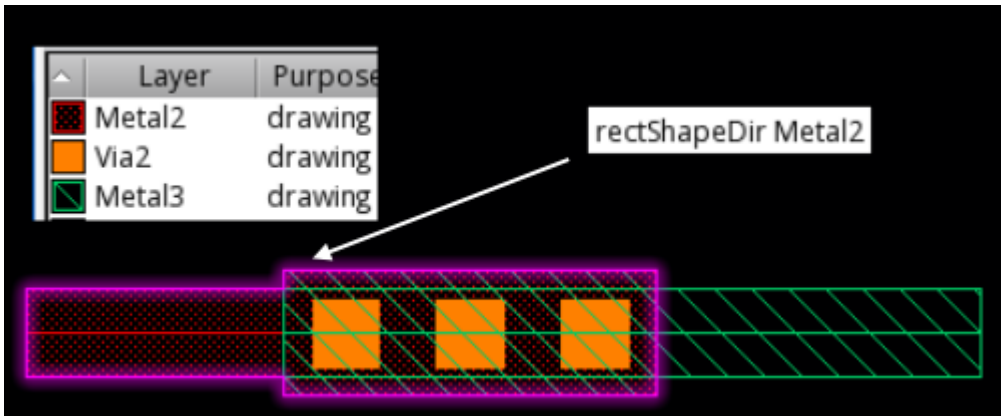
You can use the *Prevent DRC Violations On The Overlap* option to prevent the creation of a via if it introduces DRC violations on the overlap.

The following example depicts auto via creation using the *Prevent DRC Violations On The Overlap* option.



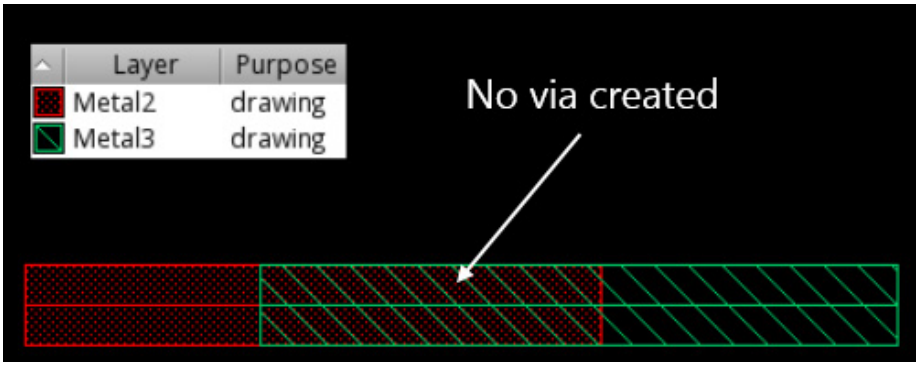
Auto via on Metal2 - Metal3 overlap creates a rectShapeDir violation on Metal2 (polygonal shape on Metal2 forbidden in this process node)

When the *Prevent DRC Violations On The Overlap* option is not selected, the auto via is created even though it introduces a DRC violation on the overlap, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

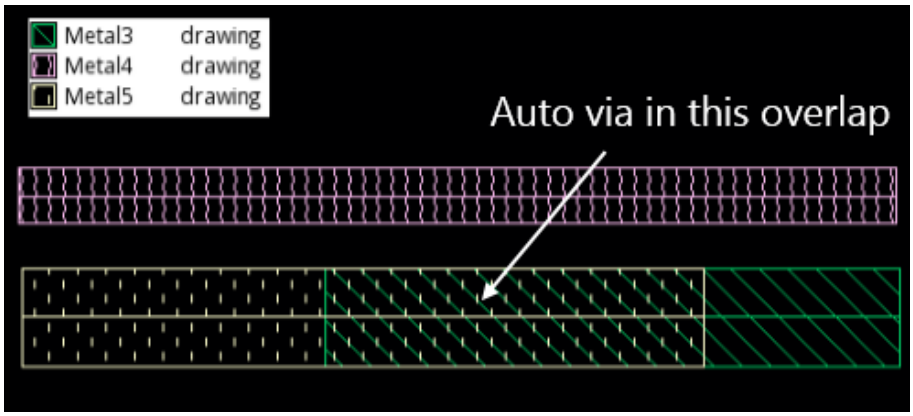
When the *Prevent DRC Violations On The Overlap* option is selected, the auto via is not created as it introduces a DRC violation. A warning message is displayed in the CIW.



Prevent DRC Violations With Neighbors

You can use the *Prevent DRC Violations With Neighbors* option to prevent the creation of a via if it introduces DRC violations with the neighboring shapes.

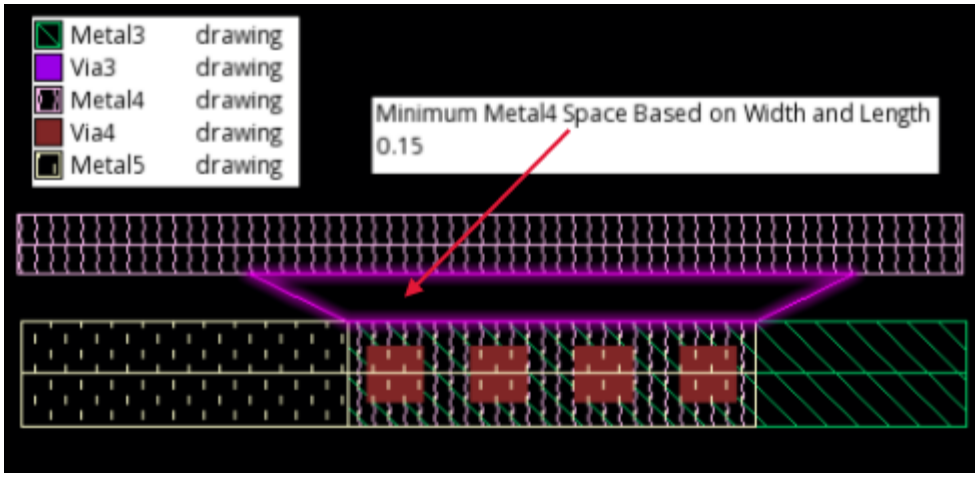
The following example depicts auto via creation using the *Prevent DRC Violations With Neighbors* option.



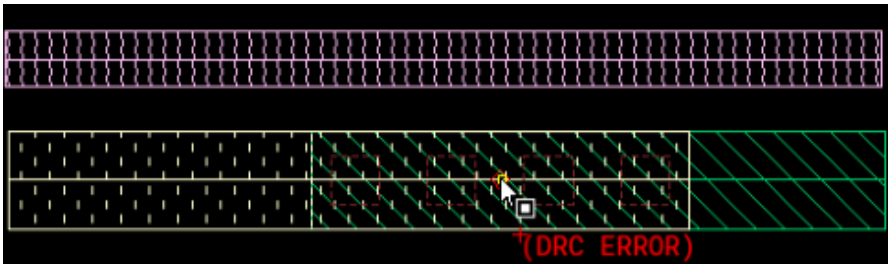
Auto via on Metal3- Metal5 overlap creates a minSpacing violation with neighboring shape in Metal4

© 2023 Cadence Design Systems, Inc.

When the *Prevent DRC Violations With Neighbors* option is not selected, the auto via is created even though it introduces a DRC violation with the neighboring shapes, as shown in the following figure.



When the *Prevent DRC Violations With Neighbors* option is selected, the auto via is not created as it introduces a DRC violation. A shaded via and the *(DRC ERROR)* label are displayed on the canvas.



© 2023 Cadence Design Systems, Inc.

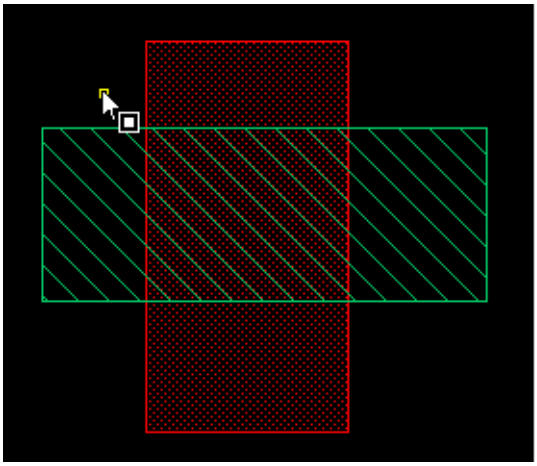
Prevent Via Larger Than Overlaps

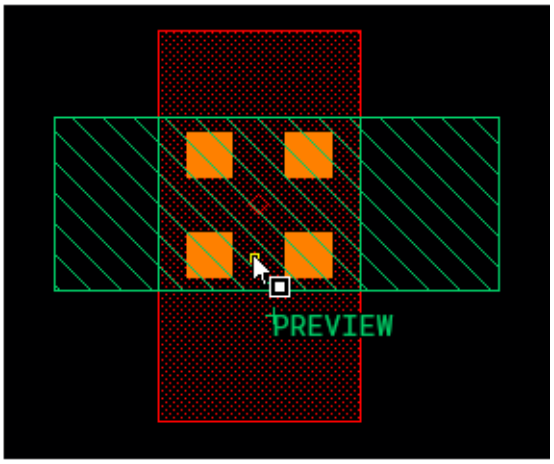
You can use the *Prevent Via Larger Than Overlap* option to prevent the creation of a via if it is larger than the overlap.



Show Preview

In the auto and fast edit via creation modes, you can also use the *Show Preview* option to preview the via before it is created, as shown in the following figure.



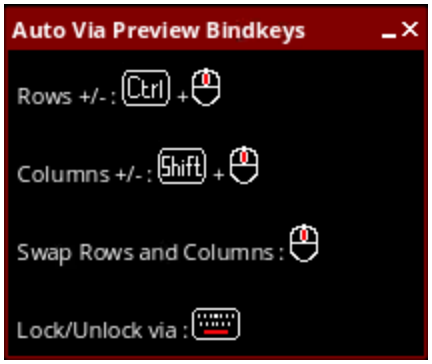


When you move the pointer over the overlap, the preview of the via is displayed

© 2023 Cadence Design Systems, Inc.

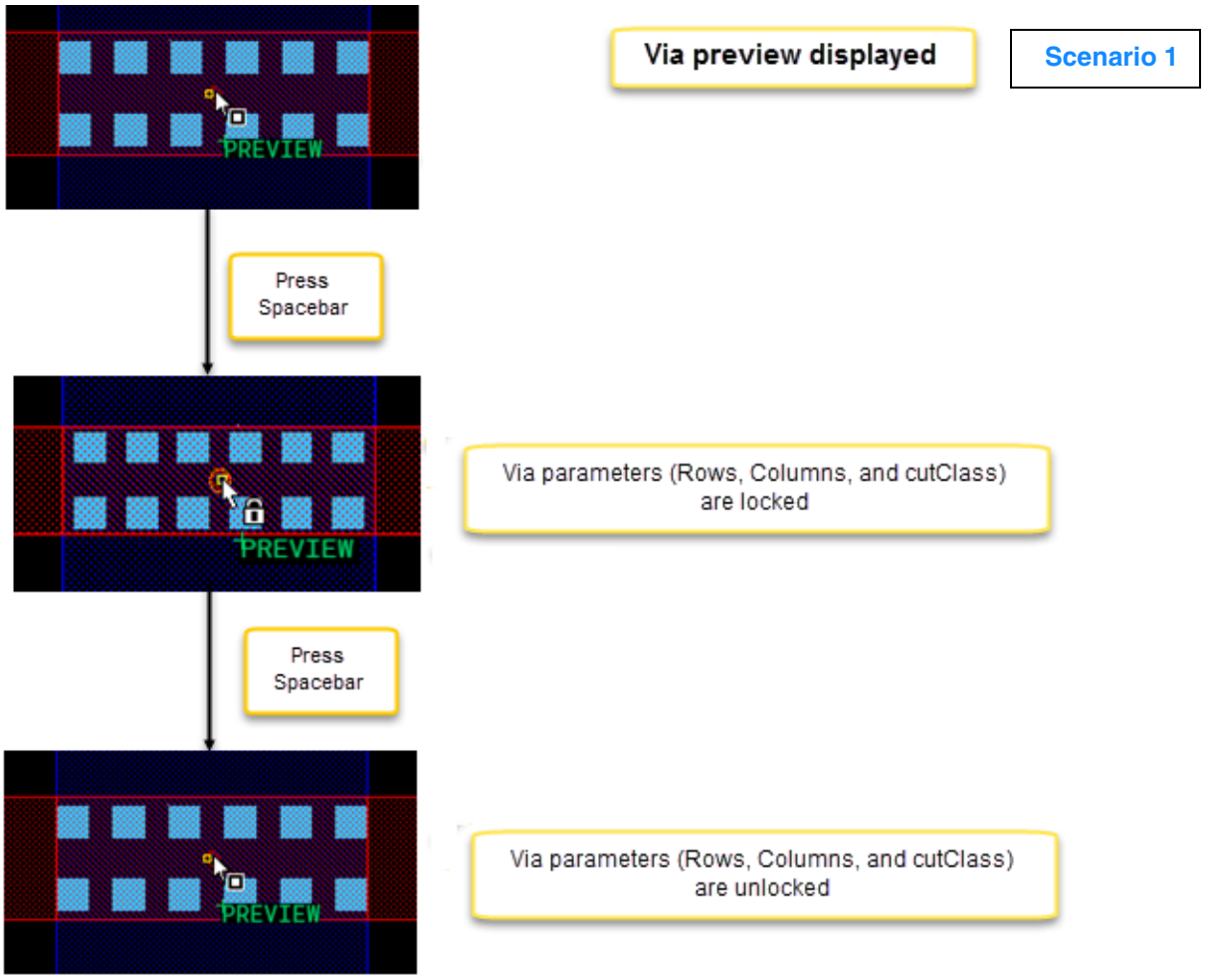
Show Hints

You can use the *Show Hints* option to display the *Auto Via Preview Bindkeys* hint box on the canvas.



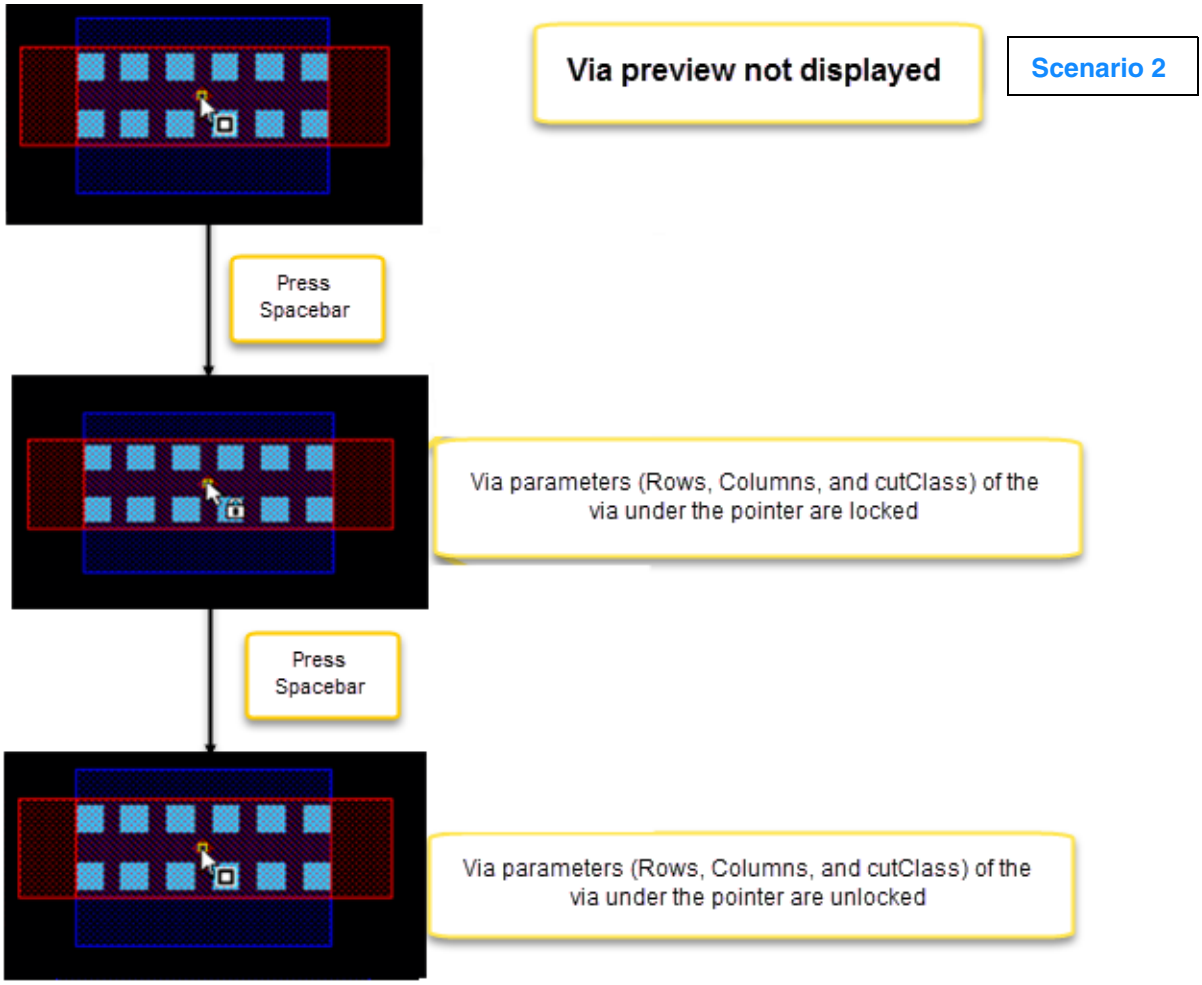
© 2023 Cadence Design Systems, Inc.

Locking a via captures the number of rows/columns along with the `cutClass`.



© 2023 Cadence Design Systems, Inc.

The following figure shows the lock/unlock via option when the via preview is not displayed.

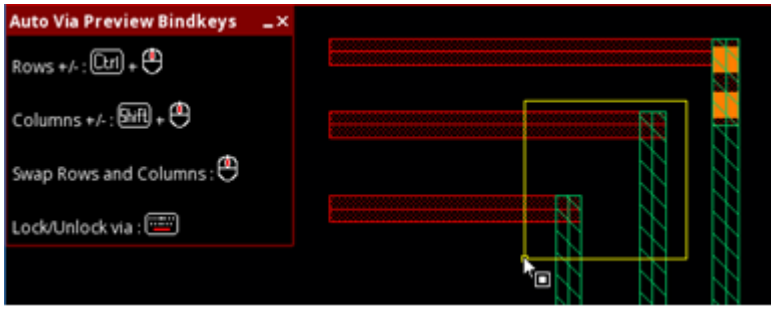


In the following example, the *Create Via* command is invoked in auto mode. If you press the spacebar when the pointer is over an existing via, via parameters of the existing via are locked, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

You can then draw an area using the middle mouse button to drop the vias on the other overlaps, as shown in the following figure.



The vias that are dropped have the same parameters as the via that has been locked.



Related Topics

Create Via Form

Via Creation

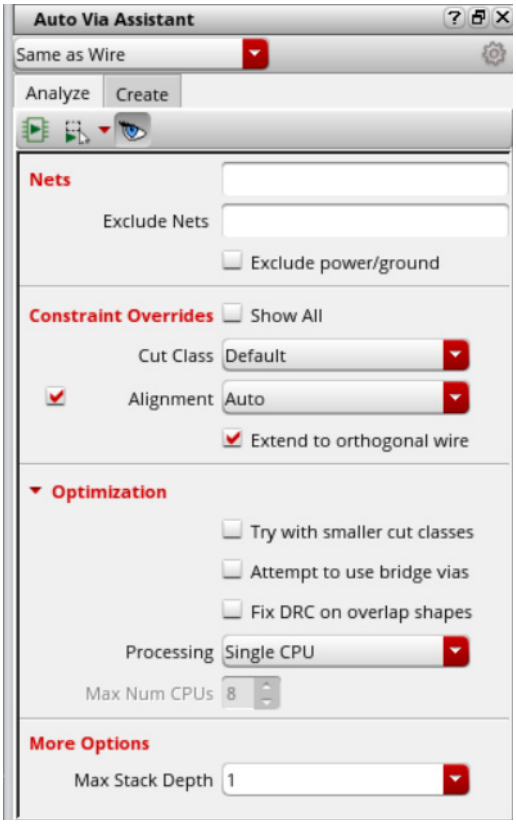
Analyzing a via using Auto Via Assistant

In Virtuoso Layout Suite EXL and higher tiers use the *Analyze* tab of the Auto Via Assistant to launch auto vias automatically and preview and analyze these vias.

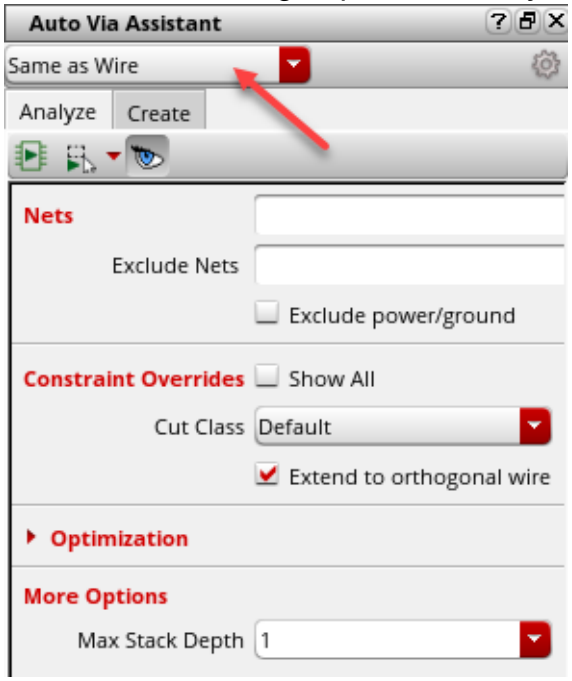
1. In the layout window, choose *Window – Assistants – Auto Via Assistant* to open *Auto Via Assistant.*

   Alternatively, right-click any empty space on the Layout toolbar and choose to display the Auto Via Assistant.

© 2023 Cadence Design Systems, Inc.

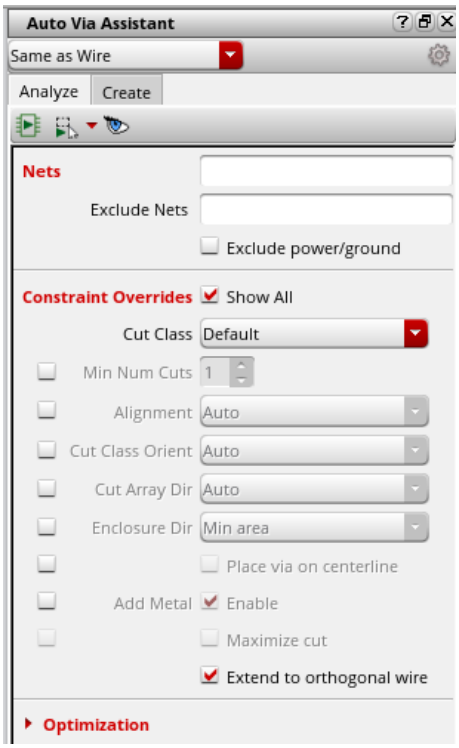


2. Select a constraint group for the analysis of the vias.



© 2023 Cadence Design Systems, Inc.

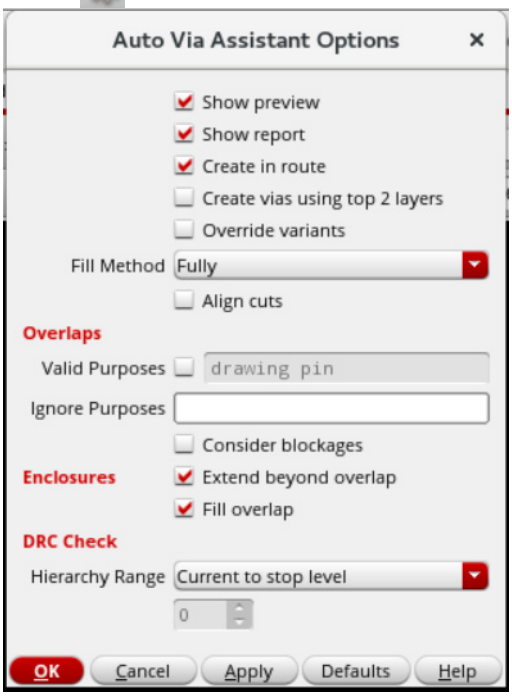
3. Specify net names in the *Nets* field to check the possibility of creating vias for the specified nets.
4. Specify net names in the *Exclude Nets* field to exclude the specified nets from analysis.
5. Select *Exclude power/ground* option to exclude power and ground nets from the analysis.
6. Select *Show All* option in *Constraint Overrides* field to display constraint override options.
7. Specify constraint override options in fields: *Cut Class*, *Min Num Cuts*, *Alignment*, *Cut Class Orient*, *Cut Array Dir*, *Enclosure Dir*, *Place via on centerline*, *Add Metal*, *Maximize cut*, and *Extend to orthogonal wire*.



8. Specify the analysis optimization options in the fields: *Try with smaller cut classes*, *Attempt to use bridge vias*, *Fix DRC on overlap shapes*, *Processing*, and *Max Num CPUs*.
9. Specify *Max Stack Depth* in *More Options* field.

© 2023 Cadence Design Systems, Inc.

10. Click  button to set the via creation options using the Auto Via Assistant Options form.



For example, select *Show preview* to see a preview of the vias from analysis, select *Show report* to see an analysis report generated in the CIW, and *Create in route* to create every new via in a new route. Also, you can select *Extend Enclosure Beyond Overlap* to extend via enclosures beyond the overlap region between two intersecting shapes.

Specify the options in the available fields for analyzing auto vias.

11. Click  from the toolbar to run the analysis in the entire cellview.
12. Click  from the toolbar to run the analysis in a rectangle, polygon, or on visible area. Select the *Use entire overlap* option to use the entire overlap for analysis.
13. Click  from the toolbar to view the vias that you can create.

© 2023 Cadence Design Systems, Inc.

An auto via analysis report is generated in the CIW, as shown below.

```
(VIA-1010): Running the analyze command in area (-3.325:6.46 -3.325:73.51 59.725:73.51 59.725:6.46).
* Overlaps found             :      2,604

* Begin generating vias...

* End generating vias        :      2,605 vias generated.

* Begin checking DRC violations...

* End checking DRC violations:          1 DRC violations found.

* Begin fixing DRC violations...

* End fixing DRC violations  :      2,603 vias remaining.

*****************************************************************************
*                          Analyzer Statistics                              *
* Area : (-3.325, 6.46) (-3.325, 73.51) (59.725, 73.51) (59.725, 6.46)      *
* Run Time =  2 s [cpu:  2 s]                                               *
* Total number of candidate DRC clean vias:2603                             *
*****************************************************************************
#
# Net: A
=============================================================================================================
| Stack |  Metal Overlaps   |      Via      |    Initial Overlaps     |       Analyze Results       |  Percent   |
| Depth |  Bottom - Top     |               |   Total   | Unconnected | Newly Connected | Connected |            |
-------------------------------------------------------------------------------------------------------------
|   2   |  Metal1 - Metal3  |  Via1 - Via2  |         1 |           1 |               0 |         0 |   0 ->   0 |
|   1   |  Metal1 - Metal2  |     Via1      |     2,603 |       2,603 |           2,603 |     2,603 |   0 -> 100 |
-------------------------------------------------------------------------------------------------------------
|                Totals                     |     2,604 |       2,604 |           2,603 |     2,603 |   0 ->  99 |
-------------------------------------------------------------------------------------------------------------
```

Related Topics

Auto Via Assistant

Auto Via Assistant Options Form

Creating a via using Auto Via Assistant

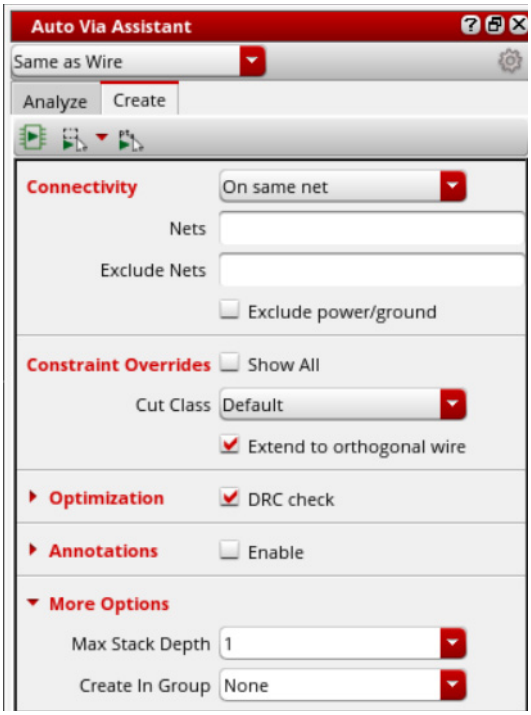
Creating a via using Auto Via Assistant

In Virtuoso Layout Suite EXL and higher tiers use the *Create* tab in the Auto Via Assistant to create vias automatically at the suggested locations from analysis.

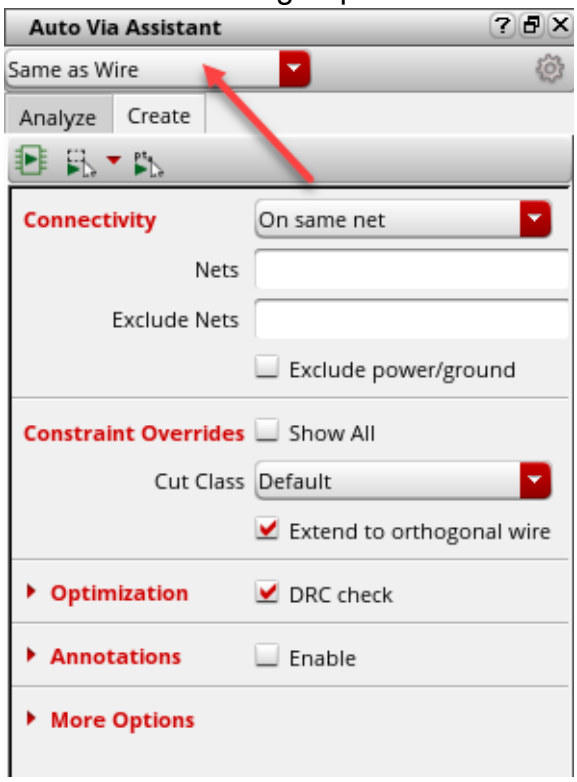
1. In the layout window, choose *Window – Assistants – Auto Via Assistant* to open *Auto Via Assistant.*

   Alternatively, right-click any empty space on the Layout toolbar and choose to display the Auto Via Assistant.

© 2023 Cadence Design Systems, Inc.



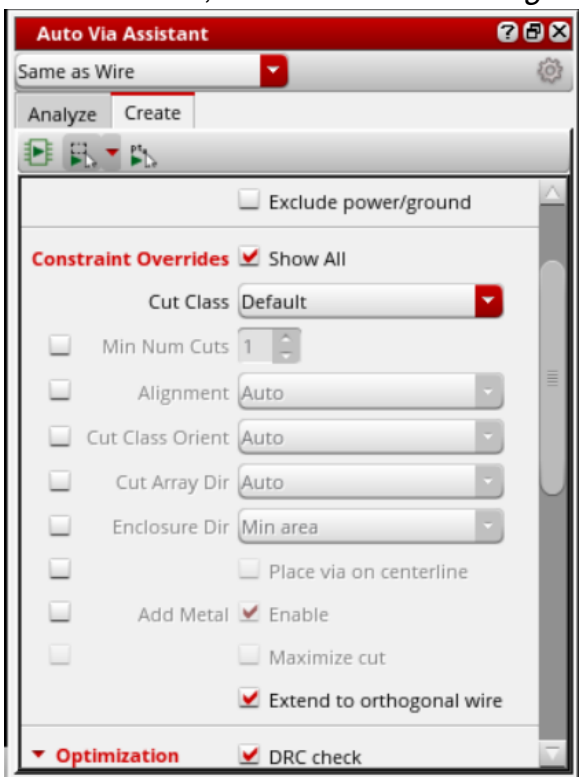
2. Select a constraint group for the creation of vias.



3. Specify *Connectivity*. The options are: *On same net*, and *Ignored*.

© 2023 Cadence Design Systems, Inc.

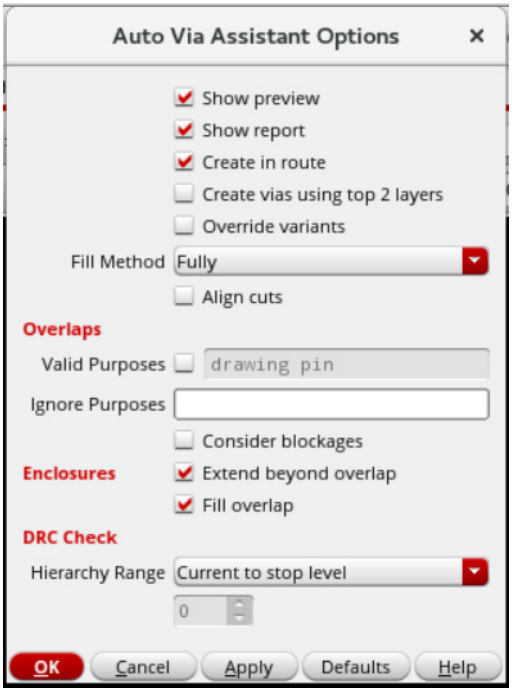
4. Specify net names in the *Nets* field to create vias for the specified nets.
5. Specify net names in the *Exclude Nets* field to exclude the specified nets from creating vias.
6. Select *Exclude power/ground* option to exclude power and ground nets from creating vias.
7. Select *Show All option* in *Constraint Overrides* field to see various constraint override options.
8. Specify constraint override options in fields: *Cut Class*, *Min Num Cuts*, *Alignment*, *Cut Class Orient*, *Cut Array Dir*, *Enclosure Dir*, *Place via on centerline*, *Add Metal*, *Maximize cut*, and *Extend to orthogonal wire*.



9. Specify the optimization options for creating vias in the fields: *DRC check, Try with smaller cut classes*, *Attempt to use bridge vias*, *Fix DRC on overlap shapes*, *Processing*, and *Max Num CPUs*.
10. Select *Enable* option in *Annotations* field to specify different annotation options for creating vias. Specify the annotation options in the fields: *Vias adding metal*, *Vias not adding metal*, *Vias not retained*, and *Overlaps with no solution*.
11. Specify *Max Stack Depth,* and *Create In Group* options in *More Options* field.

© 2023 Cadence Design Systems, Inc.

12. Click button to set the via creation options using the Auto Via Assistant Options form. The Auto Via Assistant Options form opens.



For example, select *Show preview* to see a preview of the vias from analysis, *select Show report* to see an analysis report generated in the CIW, and *Create in route* to create every new via in a new route. Also, you can select *Extend Enclosure Beyond Overlap* to extend via enclosures beyond the overlap region between two intersecting shapes.

Specify the options in the available fields for creating auto vias.

13. Click from the toolbar to create vias in the entire cellview.
14. Click from the toolbar to create vias in a rectangular area, polygon area, or the entire visible area. Select *Use entire overlap* to use the entire overlapped area for creating vias.
15. Click from the toolbar to create vias at the suggested locations.

Related Topics

Auto Via Assistant

Auto Via Assistant Options Form

Analyzing a via using Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

Setting the Technology to Create Vias

The following are the aspects related to the technology setup for creating vias.

- Calculation mode for via parameters
- `viaEngineContext` user type
- Supported constraints

Related Topics

Calculation Modes for Via Parameters

viaEngineContext User Type

Supported Constraints

Via Creation

Calculation Modes for Via Parameters

You can set the calculation mode for via parameters by using the *Via Parameters Calculation Mode* option in the Layout Editor Options form. You can set the calculation mode to any of the following:

- *Min Rules & ViaDef*: For all the via parameter fields, such as the number of cut rows and columns, and cut sizes, the minimum rule or the viaDef default values, whichever is greater, are used.
- *Minimum Rules*: For all the via parameter fields, only the minimum rules, such as `minWidth` and `minSpacing`, are used.
- *ViaDef Defaults*: For all the via parameter fields, viaDef defaults are used.

Supported Via Parameters

The following table lists the parameters supported by the via engine for customVia. The `cutPattern` parameter is supported for custom Pcell vias.

`row`	`column`
`w`	`l`
`xCutSpacing`	`yCutSpacing`

© 2023 Cadence Design Systems, Inc.

xPitch	yPitch
xBias	yBias
layer1XDefOverride	layer1YDefOverride
layer2XDefOverride	layer2YDefOverride
layer1XEnclosure	layer1YEnclosure
layer2XEnclosure	layer2YEnclosure
layer1Direction	layer2Direction

The following table lists the parameters supported by the via engine for cdsVia.

row	column
cutLayerW	cutLayerL
xCutSpacing	yCutSpacing
layer1XDefOverride	layer1YDefOverride
layer2XDefOverride	layer2YDefOverride
layer1XEnclosure	layer1YEnclosure
layer2XEnclosure	layer2YEnclosure
layer1Direction	layer2Direction
origin	

Related Topics

Setting the Technology to Create Vias

viaEngineContext User Type

Supported Constraints

Via Creation

viaEngineContext User Type

When you call pre or post viaEngine callback function, the `viaEngineContext` user type is specified.

© 2023 Cadence Design Systems, Inc.

The following example depicts the sample usage of the user type information.

```
procedure(myPreViaEngineCB(veCxt)
printf("call pre viaEngine procedure with context '%s' and viaDef '%s'.\n" veCxt->operationType veCxt->viaModel->viaDef->name)
printf("overlapInfo is: ")
println(veCxt->overlapInfo)
t
)
viaRegisterPreViaEngineCallback("myPreViaEngineCB")
```

When creating a via in single via mode, the following is the output of the above sample callback function:

```
\o call pre viaEngine procedure with context 'check' and viaDef 'M3_M2'.
\o overlapInfo is: nil
```

When creating a via in auto via mode, the following is the output of the sample callback function:

```
\o call pre viaEngine procedure with context 'generate' and viaDef 'M1_PO'.
\o overlapInfo is: ((((0.25 -4.0) (0.25 1.0) (-0.25 1.0) (-0.25 -4.0)) "Metal1" "drawing" "bottom to top" 0.5) (((1.0 -1.0) (1.0 2.0) (-1.0 2.0) (-1.0 -1.0)) "Poly" "drawing" "vertical" 2.0) ((-0.25 -1.0) (0.25 1.0)))
\r t
```

In this example, the value, `0.5`, means that the first involved shape on `Metal1 drawing` has a width of `0.5`. The value `2.0` means that the second involved shape on `Poly drawing` has a width of `2.0`. And the intersection of the two shapes gives the following overlap box:

```
((-0.25 -1.0) (0.25 1.0))
```

Related Topics

Appendix 5, "Setting the Technology to Create Vias"

Calculation Modes for Via Parameters

Supported Constraints

Via Creation

viaRegisterPreViaEngineCallback

Supported Constraints

- Supported Constraints for Via Creation and Editing

© 2023 Cadence Design Systems, Inc.

- Supported Constraint Parameters
- Additional Supported Constraints for Stack and Auto Modes
- Supported Constraints for Local Interconnect Vias

Supported Constraints for Via Creation and Editing

The *Create Via* for a single via mode, *Edit Via Properties* for a single mode via, and *Property Editor* assistant for vias use technology library constraints and constraint groups to adjust via creation and editing to adhere to the following constraints:

- `minCutClassSpacing (One layer)`
- `viaSpacing`
- `forbiddenCutClassSpacingRange`
- `minEndOfLineEdgeExtension`
- `minCenterlineExtension`
- `minNumCut`
- `minEdgeAdjacentLength`
- `minExtensionDistance`
- `minOppExtension`
- `minWidth`
- `minLargeViaArraySpacing`
- `minLargeViaArrayCutSpacing`
- `minLargeViaArrayWidth`
- `largeRectViaArrayAllowed`
- `allowedCutClass`
- `minSpacing (One layer)`
- `minNeighborViaSpacing`
- `minExtensionToCorner`
- `allowedWidthRanges`
- `minAreaEdgeLength`

© 2023 Cadence Design Systems, Inc.

- maxWidth
- viaKeepoutZone
- rectShapeDir

Supported Constraint Parameters

The via editing and creation commands also comply with the following parameters:

- The 'coincidentAllowed parameter of the minExtensionDistance and minOppExtension constraints. If both minExtensionDistance and minOppExtension are used, minOppExtension takes precedence.

  For example, in the following sample, with the 'coincidentAllowed parameter in the minExtensionDistance rule, an enclosure of 0.0 is allowed. The allowed enclosure can be either 0.0 or > = 0.06.

  ```
  orderedSpacings(
      ( minExtensionDistance   "Metal1"   "Via1"   0.06   'coincidentAllowed)
      ( minExtensionDistance   "Metal2"   "Via1"   0.06   'coincidentAllowed)
  ) ;orderedSpacings
  ```

- The 'centerToCenter parameter of the viaSpacing and minViaSpacing (One layer) constraints and the constraint minCenterToCenterSpacing. If 'centerToCenter is specified, both distance and spacing are measured from cut center to cut center instead of from cut edge to cut edge.

  For example, the following sample sets the minimum via cut spacing on Via1 to 1.0 when there are three or more cuts that are 1.5 or fewer user units apart.

  ```
  Spacings(
    (viaSpacing  "Via1"  'centerToCenter  'numCuts 3  'distance  1.5 1.0)
  ); spacings
  ```

- The 'cutClass parameter of the viaSpacing and minOppExtension constraints.

  For example, cut class can be specified by cut sizes.

  ```
  ( viaSpacing  "Via2" 'cutClass 0.14 'numCuts 3 'distance 0.21 0.3  )
  ( viaSpacing  "Via2" 'cutClass (0.14 0.28) 'numCuts 3 'distance 0.21 0.4  )
  ( viaSpacing  "Via2" 'cutClass 0.28 'numCuts 3 'distance 0.21 0.5  )
  ```

  In the first rule, width is 0.14. Since length is not been specified, it is the same as width. Similarly, in the third rule, both width and length are 0.28. In the second rule, width is 0.14 and length is 0.28.

  Cut class can also be specified by the cut class name.

© 2023 Cadence Design Systems, Inc.

```
( viaSpacing  "Via2" 'cutClass "Small" 'numCuts 3 'distance 0.21 0.3  )
( viaSpacing  "Via2" 'cutClass "Bar" 'numCuts 3 'distance 0.21 0.4  )
( viaSpacing  "Via2" 'cutClass "Large" 'numCuts 3 'distance 0.21 0.5  )
```

- The `'noPrl` parameter of the `viaSpacing` constraint.
- The `'extraCut`, `'area`, `'distanceWithin`, and `'nearbyLayer` parameters of the `minOppExtension` constraint.
- The `'distanceWithin` and `'nearbyLayer` parameters of the `minOppExtension` constraint.
- The `'overLayer` and `'exceptViaConnectivityType` parameters of the `minViaSpacing (One layer)` constraint.
- The `'maxNumCuts` and `'applyPerDirection` parameters of the `minLargeViaArrayCutSpacing` constraint.
- The `'above`, `'below`, and `'bothBelowAndAbove` parameters of the `minNumCut` constraint.

Additional Supported Constraints for Stack and Auto Modes

The *Stack* and *Auto* modes of the *Create Via* command adhere to the following additional constraints:

- `minArea` (for the intermediate layers in a via stack)
- `minOutsideCornerEdgeLength`
- `minQuadrupleExtension`
- `minSize`
- `minStepEdgeLength`
- `viaStackingLimits`
- `orthogonalWSPGrid` (Auto via only)

If you change parameters on the *Create Via* or *Edit Via Properties* forms that cause a violation of these constraints, the values are adjusted to comply with the constraints defined in the technology library. The rule checking excludes Origin X and Y.

Before you create or edit a via, set the constraint group in the *Wire Editing* section of the Layout Editor Options Form. The selected constraint group determines the viaDefs that are available for creating and editing vias.

© 2023 Cadence Design Systems, Inc.

The via engine uses both shapes and layer blockages to define regions for area-based constraints. For more information, see *Area Constraints*.

Supported Constraints for Auto Mode and Fast Mode Vias

The *Auto* and *Fast Edit* modes of the Create Via command support area-based constraints. These constraints are not supported for single and stack vias.

In Layout XL and higher tiers, the `'edgeExtension`, `'layer`, and `'extensionDirection` parameters are supported for the `viaSpacing` constraint when the *Prevent DRC Violations With Neighbors* option is selected.

Auto via also supports the `minVoltageExtension` constraint.

Supported Constraints for Local Interconnect Vias

Local interconnect vias are vias between local interconnect (LI) layers. LI layers do not need a physical via or contact shape in order to be connected. However, the automatic router model requires a via for every transition from one layer to another in order for a route to be considered valid. This is why LI vias are required to establish connections between LI layers, even though there is no contact layer between these layers.

At advanced process nodes, many foundries add support for LI layers to the process stack. LI layers form connections through overlaps rather than using separate cut layers. Typically, only rectangles are allowed on LI layers and these shapes often have a fixed direction.

LI vias adhere to the following constraints:

- `allowedLengthRanges`
- `minExtensionOnLongSide`
- `minSideExtension`
- `minWireOverlap`
- `minDirectionalOverlap`
- `rectShapeDir`

Related Topics

Setting the Technology to Create Vias

© 2023 Cadence Design Systems, Inc.

viaEngineContext User Type

Calculation Modes for Via Parameters

Support for Voltage-Dependent Rules

Via Creation

Layout Editor Options Form

Support for Voltage-Dependent Rules

Voltage-dependent rules are critical for design processes at 45 nm and less. The support for these rules is based on the ability to use user-defined layer purposes that represent voltage ranges and on the ability to define design rule constraint groups within these voltage ranges.

To support these rules in Virtuoso, you can define non-drawing purposes, purpose-aware constraints, such as those for width and spacing, in the technology file. Additionally, you can define cdsVia, which is a via device that can accept non-drawing purposes.

Standard vias accept only the `drawing` purpose.

You can define a cdsVia either directly in the technology file or by using the *Install Device* form.

To be able to create and edit cdsVias on non-drawing, voltage-aware purposes, the following tasks need to be performed in the following order:

1. Install a cdsVia device in the technology file
2. Add the cdsVia device to the `validVias` constraint
3. Create a cdsVia

Related Topics

Installing a cdsVia Device in the Technology File

Adding the cdsVia Device to the validVias Constraint

Creating a cdsVia

Setting the Technology to Create Vias

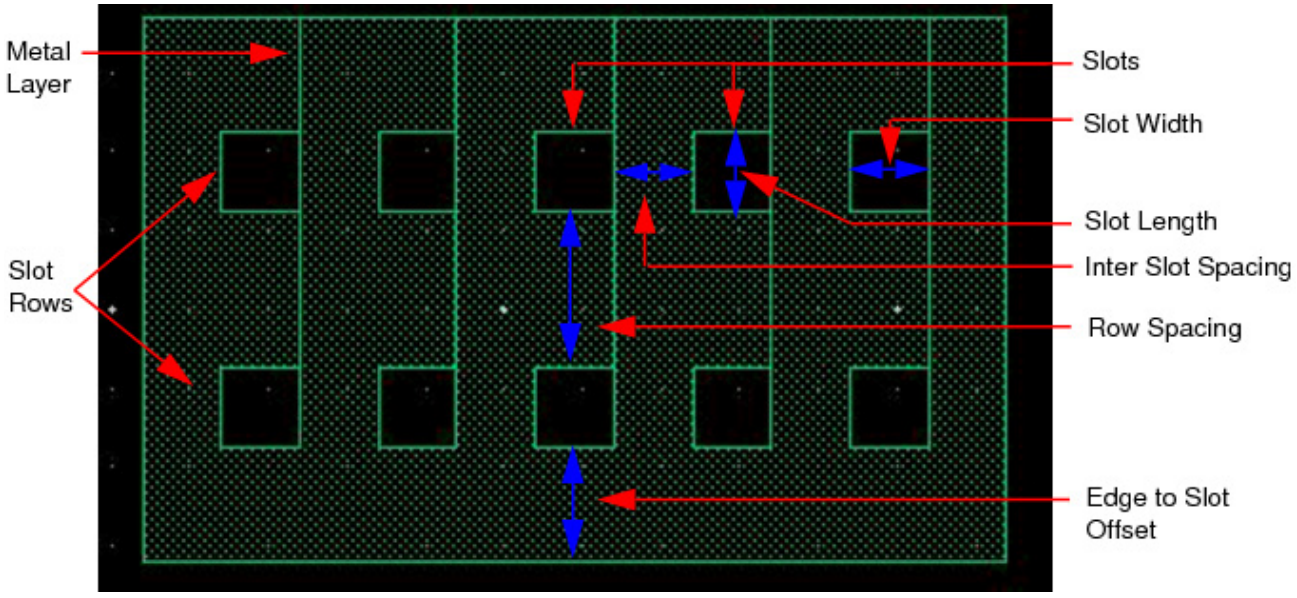
Via Creation

© 2023 Cadence Design Systems, Inc.

Creating Slots

Metal slots are the holes that you create in the metal layer to relieve the stress that results from heat generation, especially in large metal shapes. Creating slots in large shapes improves the structural integrity of shapes, making them less likely to fracture because of constant expansion (heating) and contraction (cooling).

The following figure shows a rectangle with slots and illustrates the various slot parameters.



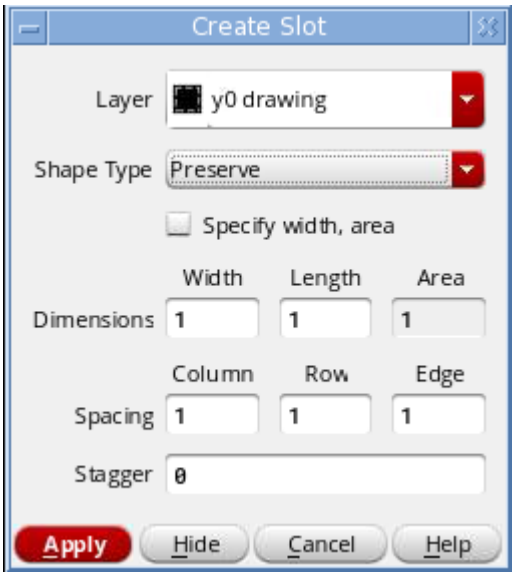
You use the *Create Slot* command to create slots in existing rectangle and path shapes. You can also create slots while creating rectangle and path shapes. The *Create Slot* command is available only for rectangles and paths.

To create slots in an existing rectangle or path that does not have slots:

1. In the layout window, choose *Create – Slot*.

© 2023 Cadence Design Systems, Inc.

2. Press `F3` to open the Create Slot form.



3. Select a layer-purpose pair from the *Layer* list.

   The layer-purpose pair information is required if the resulting shape with slots is a ROD object.

4. Set *Shape Type* as *Preserve*, *Convert to Polygon*, or *Convert to ROD* based on the type of shape you want to create after slots are added to it.

5. Specify *Width/Length* or *Width/Area*.

   If you select the *Specify width, area* check box, the *Length* and *Area* fields are available, else the *Width* and *Length* fields are available.

6. Specify slot dimensions in the *Width* and *Length* or *Width* and *Area* fields.

7. Specify the values as required in the *Spacing Column, Row, Edge*, and *Stagger* fields.

8. On the canvas, click an existing rectangle or path that does not have slots. You can also select multiple shapes to create slots in them. The selected shapes can be a mix of ROD objects and polygons.

   If the rectangle or path that you select already contains slots, a message is displayed in the CIW, informing you to select only the rectangles and paths that do not have slots.

   Slots are created in the selected shape. You can continue to select more rectangles or paths to create slots in them.

9. Press *Esc* or click *Cancel* in the form to finish creating slots.

© 2023 Cadence Design Systems, Inc.

You can use the `lxShapeSlotting` function to create slots on metal shapes.

Related Topics

Create Slot Form

Editing Slots

Creating a Rectangle

Creating a Path

Wide Wires Slotting Flow

You can use the density-aware `lxShapeSlotting` SKILL function to automatically create slots on metal shapes during wide wire routing. This saves you from having to manually draw complex wires and other shapes that are time-consuming to create and that do not always take full account of the required electrical and design rule constraints.

The following is the recommended flow to create slots on metal shapes during wide wire routing:

1. Pre-route the design.

   In this step, the wide wires match the expected resistance (R) value as well as the electromigration criteria. Some margin is considered in terms of wire width to anticipate further degradation that slotting might introduce.

2. Use the Virtuoso Electrically-Aware Design (EAD) flow as early as possible in order to verify whether the electrical constraints are satisfied.

3. Run the density aware slotting SKILL function `lxShapeSlotting` so that slots are added to the shapes to fix `maxDensity` and `maxWidth` rules, wherever required.

4. Use EAD to verify that the electrical constraints are satisfied.

   If the electrical constraints are still not satisfied, you might need to increase the width of some of the shapes that are concerned by violations or remove some of the slot shapes to ease out the current flow. Then re-run the `lxShapeSlotting` SKILL function.

5. Run the DRD checker to verify the `maxDensity` and `maxWidth` rules.

© 2023 Cadence Design Systems, Inc.

Related Topics

Setting up the Interactive Routing Environment

Introduction to Virtuoso Electrically Aware Design Flow

Design Rule Driven Editing

Creating a Group

You can group shapes and/or instances together to manipulate the objects together, rather than individually. As an owner, a group has an area boundary object that conforms to the bounding box of the entire group, which is the composite of all the bounding boxes of the members of the group. This area boundary is automatically created, edited, and deleted by the *Group* command as objects are added to or deleted from the group.

A group has the following characteristics:

- Is a named database figure, which you can query in the database.
- Can contain any number of figures or instances, including other groups. It may have an applied constraint group.
- Is selectable and has an origin and an orientation.
- Does not have a layer/purpose or connectivity, though its members may.

To create a group:

1. In the layout window, choose *Create – Group*.

   The Create Group form opens.

2. Optionally, name the group by typing the name in the *Name* field.

   The group is automatically named; the first group is named `Group0` and as groups are added, the number increments by one.

3. Optionally, choose the orientation from the *Orientation* list.
4. On the canvas, click the objects that you want to add to the group.
5. Click *Apply* in the form.

   The group is created. When you place the mouse pointer over a group, a dotted line surrounds the objects in the group.

© 2023 Cadence Design Systems, Inc.

Use the *Edit – Group* command to add an object to a group, remove an object from a group, or ungroup a group of objects. To edit the properties of objects in a group, use the *Edit – Properties* command.

Related Topics

Create Group Form

Editing Groups

Placement and Routing Objects

Placement and routing (PR) objects are objects that help you to place instances in a design and make connections and routes between instances. PR objects include the following:

- Blockage: Specifies an area in a design where certain objects cannot be placed or are to some degree restricted in their placement.
- Row: Represents a location for the placement of standard cells, macros, or devices and is a repetition of the site element in a given direction, horizontal or vertical.
- Custom placement area: Used for placement of standard cells and custom cells.
- Track pattern: Provides guidelines for laying interconnect routes. Track patterns are uniformly-spaced orthogonal patterns extending across the base array.
- PR boundary: Represents the boundary of a block for place-and-route applications.
- Snap boundary: Used for placing instances of a design in rows, thereby aligning the snap boundary origin with the site origin. A snap boundary is used for designs that implement any rectilinear macros or standard cells.
- Area boundary: Indicates a region of a block for various purposes defined by a design methodology. An area boundary is a generic boundary in a physical block.
- Cluster boundary: Defines an area in which the contents of a specific cluster are included or excluded, depending upon their definition.

Related Topics

Blockages

Rows

© 2023 Cadence Design Systems, Inc.

Creating a Custom Placement Area

Track Patterns

Creating a PR Boundary

Creating a Snap Boundary

Creating an Area Boundary

Creating a Cluster Boundary

Blockages

A blockage object specifies an area in a design where certain objects cannot be placed or are to some degree restricted in their placement. The type of objects to exclude from the blockage area depends on the blockage type.

Blockages are defined by points. You can use the command to seed the *Create Blockage* form. The command captures the layer information of the blockage, if applicable. It also captures the blockage type and updates the environment variable blockageType. In addition, if the blockage is of type `fill` or `screen`, the command captures the value of maximum density per cent and updates the environment variable blockageMaxDensity.

The following are the different types of blockages:

- **Placement blockage**: Prohibits the placement instances in the blockage area.
- **Routing blockage**: Prohibits creating or placing interconnect shapes such as path segments, paths, multipart paths, single-level shapes such as rectangles and polygons, or vias in the blockage area on the defined layer.
- **Pin blockage**: Prohibits creating or placing shape, instance, or via pins in the blockage area on the defined material and layer.
- **Feedthru blockage**: Prohibits creating feedthrus, which is routing through multiple gates through the terminal on the same layer, in the blockage area.
- **Screen blockage**: Allows you to create routing in the blockage area until the maximum density specified for the blockage area is reached on the defined material and layer.

When Design Rule Driven Editing (DRD) is enabled, blockages participate in DRD rule checking so that error markers/messages are created when you violate certain design rules.

© 2023 Cadence Design Systems, Inc.

Effective Width for a Blockage

The effective width for a blockage is an attribute that you can assign to any blockage types defined by a layer or layers such as routing, pin, or feedthru. The following considerations apply when using effective width:

- The effective width if set, rather than the size of the block object, is used for the entire blockage when a `minSpacing` value in the table spacing rules is defined in the technology file. These values determine which spacing rules apply. DRD flags the error in Notify mode or prevents placement in Enforce mode if the object is closer than the defined width for the layer.
- If a single `minSpacing` value is defined in the technology file, it is applied and the effective width is not used.
- It is not necessary to have the effective width defined for all layer blockages.
- When the blockage type is placement, the *Effective Width* field is grayed out, since it applies to instances and not layers.

Blockage Owner

You can optionally use the *Attach to Owner* option to assign an object as the blockage owner, depending upon the type of blockage. In the case of a placement halo blockage, the owner should be an instance or PR boundary. An owner can have any number of blockages. Cyclical relationships are not allowed. For example, if object A is a parent to object B, then object B cannot be a parent for object A.

Blockage Summary

You can use the *File - Summary* command to view information about blockage objects. The summary contains the total number of blockage objects and the blockage count of each type of blockage in the cellview. The information is sorted by layer.

Related Topics

Creating a Blockage

Placement and Routing Objects

© 2023 Cadence Design Systems, Inc.

Creating a Blockage

To create a blockage:

1. In the layout window, choose *Create – P&R Objects – Blockage.*

   The Create Blockage form opens.

2. Select the type from *Blockage Type*.
3. If the blockage is a layer blockage:
   a. Select the appropriate material type from *Material*.
   b. Select the appropriate layer from the *Layer* list.
4. Specify how you want to compute blockage spacing. You can choose from *Blockage geometry or Effective width*, or use *Specify* to enter a spacing value.
5. Optionally, specify blockage settings such as the color, owner, maximum density, and ROD object name.
6. If you want to use *Manual* as the *Create Method* method:
   a. Select the appropriate snap mode in the *Create Method* section.
   b. On the canvas, click the first point of the blockage. Then, click the next point of the blockage. Just as you would for a polygon, continue to enter points until the shape is the final shape you want for the blockage.
   c. Press `Enter` or double-click on the canvas to complete the blockage object.
7. If you want to use *Automatic* as the *Create Method* method:
   a. Select the appropriate method from *Derive From*.

      Depending on the method you select, you are prompted to select objects or instances to derive the blockage.

   b. Click the instances or objects on the canvas that you want to include in the blockage area.
   c. Click the *Apply* button in the form or press `Enter` to complete the blockage object.

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

Blockages

Blockage Halos

A blockage halo defines an area around an instance that cannot overlap the placement blockage.

- **Placement halo:** A placement halo is a type of blockage associated with an instance or a master, representing an area around the master's PR boundary. The shape of the halo is determined from the shape of the PR boundary of the master and the given offsets, and it is automatically updated to reflect changes in the master's PR boundary.

  The placement halo is most commonly used to prevent standard cells from being placed within a certain range of an instance. This is usually done to make sure there is enough space to properly route to the presumably large number of pins on the side of a block.

- **Layer halo:** A layer halo is a type of blockage associated with an instance or a master, representing an area on a given layer around the master's PR boundary. The shape of the halo is determined from the shape of the PR boundary of the master and the given offsets, and it is automatically updated to reflect changes in the master's PR boundary.

  The `oaLayerHalo` applies around the perimeter of an instance. It prohibits routing on the given layer parallel to the edge of the instance, but it allows routing perpendicular to the instance edge. This prevents long lines being routed parallel to the edge of the instance, which causes cross-coupling. However, it allows routing straight into the pins. Since power lines do not switch, they can be routed anywhere within the halo area.

You can create a blockage halo by using the Halo Editor form. You can access this form by clicking the *Define Halo* button in the *Create Instance* form or the *Create P&R Boundary* form.

Related Topics

Halo Editor Form

Create Instance Form

Create P&R Boundary Form

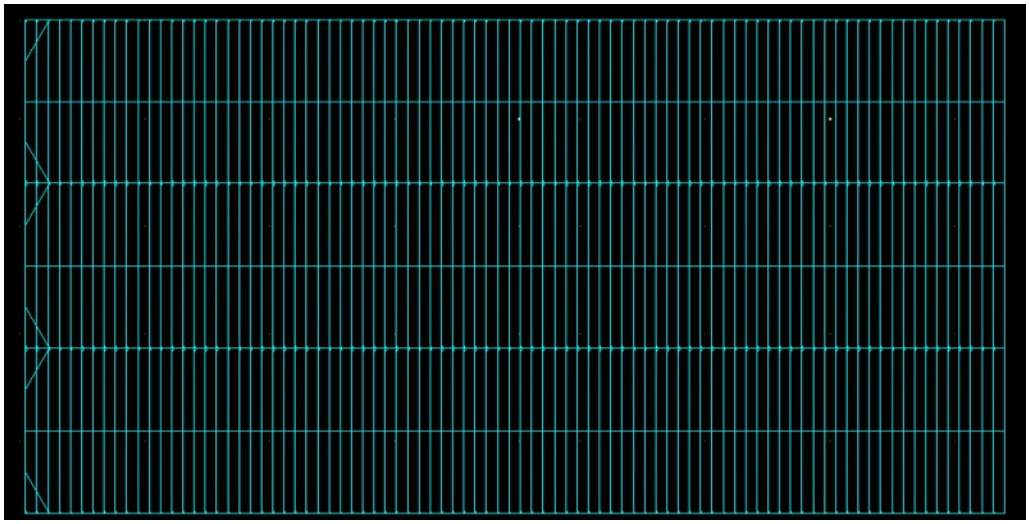
Creating a Blockage

Blockages

© 2023 Cadence Design Systems, Inc.

Rows

A row represents a location for the placement of standard cells, macros, or devices and is a repetition of the site element in a given direction, horizontal or vertical. A row is basically a one-dimensional array of a given site. All sites in one row must be the same size. Rows are drawn using the display attributes for layer-purpose pair row boundary, a Cadence reserved layer which must already be in the Palette assistant. Rows are drawn as a small rectangle with a diagonal line in the corner of rows to show the row orientation. Sites are drawn as a large rectangle with a diagonal line in the corner of sites to show the site orientation. The display of row name and row site can be turned on or off in the Display Options form by using the *Row Name* and *Row Site* check boxes. You can control the snapping of instances to rows. Adding appropriate site pattern to the instance masters results in successful snapping of instances to rows.



Site Definitions

Site definitions `siteDefs` are technology library objects that are the basic elements from which placement rows are constructed. Each row is associated with `siteDefs`. A site defines a type, width, height, and allowable symmetry of each placement location row. A new site is defined for each cell height. Cell widths must be a multiple of the site width. The height of a row equals the height of the site and the width of a row equals the multiple of a site width.

The following are the types of site definitions:

- Scalar site definitions `scalarSiteDefs` define sites where you can place a standard row.

© 2023 Cadence Design Systems, Inc.

- Array site definitions `arraySiteDefs` define a pattern of `scalarSiteDefs`. The `arraySiteDefs` define a pre-determined set of sites that you use to place a set of cells in part of a standard row.

  The `arraySiteDefs` site definitions are primarily used in gate-array designs. Each scalar site definition in the array, along with its offset from the origin (the lower-left corner) of the array and its orientation are defined. Each pattern is a named reference to a scalar site definition.

Before you can create rows, you must first define site definitions in the `siteDefs` section of the technology file as `scalarSiteDef`. The following is an example of the `siteDefs` section.

```
siteDefs(
scalarSiteDefs(
;( siteDefName          type width  height  symInX symInY symInR90)
;( -----------          ---- -----  ------  ------ ------ -------)
 ( gpdk090site          core 0.29    2.61     nil    t      nil)
) ;scalarSiteDefs
```

You can add multiple site names and definitions in the `siteDefs` section. Then, in the Create Row form, from the *Site Definition Name* list, you can select the one you want to use.

Templates

After defining site definitions, you may define the rails and/or components as a SKILL program entered into the input line of the CIW. This information appears in the *Site Information* field of the *Create Row* form.

The following is an example of the site information:

```
procedure( createDefs()
let(((cv geGetEditCellView()))
    railDef1 = dbCreateRailDef(cv "rail1" "vdd!"
            list("metal1" "drawing") "bottomAlignment" "bottomAlignment" 2.0
0.0 0.0 0.0)
    railDef2 = dbCreateRailDef(cv "rail2" "gnd!"
            list("metal1" "drawing") "topAlignment" "topAlignment" 2.0 0.0 0.0
0.0)
    compDef1 = dbCreateCompTypeSetDef(cv "compDef1" "bottomAlignment"
            "bottomAlignment" list("R0" "MX") list("MyComp1" "MyComp2") 0 0)
    compDef2 = dbCreateCompTypeSetDef(cv "compDef2" "topAlignment"
            "topAlignment" list("R0" "MX") list("MyComp1" "MyComp2") 0 0)
    t
)
)
```

The rails and components are displayed in the *Pre-Defined* list in the *Templates* section of the *Create Row* form. Use the arrow buttons to move the rail/component definitions from the *Pre-Defined* pane to the *Selected* pane or vice-versa. Only the *Selected* definitions are used for creating the row or cluster boundary.

© 2023 Cadence Design Systems, Inc.

Related Topics

Creating Rows

Creating Rows

To create rows:

1. Ensure the required site definitions are present in the `siteDefs` section of the technology file.
2. In the layout window, choose *Create – P&R Objects – Row.*

   The Create Row form opens.

   Opening the *Create Row* form turns on *Row* visibility in the Objects panel, even if you have previously turned it off. If site definitions are not present in the technology file, a warning message displays in the CIW, the *Create Row* form does not appear, and the command ends.
3. Specify row details in the *Definition* section as required.
4. Type the row name in the *Base Name* field.

   When you enter the base name, an underscore is added followed by a number, which is greater than or equal to 1. As you enter new sites, each site is incremented by 1. For example, if your base name is ROW_1, the name assigned to the first site is ROW_1_1, the next is ROW_1_2, and so on. If you do not enter a custom row name, the software assigns the name row_number, which increments by 1 each time you add a site and rows.
5. Select the orientation of the row as *Horizontal* or *Vertical*.
6. Select the *Starting Site Orientation*.

   All subsequent rows follow the orientation that you set in this field. The valid values are *R0*, *R90*, *R180*, *R270*, *MX*, *MY*, *MXR90*, and *MYR90*.
7. Specify the flip and abut settings for rows in *Options*. For example, you may set every other row to flip or set only the first row to flip. You may specify the number of rows to abut or set to abut the first and second rows. You can also specify the space between the rows to be created.
8. Select the site definitions as core, pad, or both, based on which you want to filter the row created.

© 2023 Cadence Design Systems, Inc.

9. Select the site from the *Site Definition Name* box whose elements are to be placed in the row. Selecting each site definition name one-by-one displays the dimension information about the site in the *Site Information* box.

10. Templates allow you to place rails and/or components automatically into the rows. Choose pre-existing rails or components or both to create the row. If pre-defined rails or components exist, they will display in the *Pre-Defined* pane. Select the ones you want to use for row creation one-by-one and move them to the *Selected* pane by using the left and right arrows.

    Template definitions are associated only with the cellview in which they are defined. Therefore, the rail and component templates available to one cellview are not available to another cellview.

11. Specify how the row should be created by choosing options in the *Create Method* section.

    a. In the *Assisted* method, specify the *Number of Rows* to be created and the *Number of Sites* to be created for each row.

    When you move the pointer to the canvas, the wireframe displays the number of rows specified in the form. Click the cellview at the position where you want to place the rows. Each row has the number of sites you specified in the form. When you open the form again, the last values entered in the fields are retained.

    b. In the *Area Fill* method, select the type of area you want for placing the rows, *Rectangular*, *Rectilinear*, or *Partition*. Then, drag and select the area on the canvas where you want to place the rows. The software calculates and creates the maximum number of rows in the selected area and creates the maximum number of sites for each row.

    - *Rectangular*: Draw a rectangle. The software creates rows in the selected area until there is no more room in the rectangle for an additional row. Maximum number of sites are created for each row. If you set offsets, they are obeyed within the rectangular shape.

    - *Rectilinear*: It works in the same way as for a rectangle, but the rows are of different lengths and widths depending on the shape of the polygons. If you set offsets, they are obeyed within the rectilinear shape. The rectilinear polygon must not be self intersecting.

    - *Partition*: Create an area consisting of pre-defined areas by choosing either a PR boundary or snap boundary.

    In the *Area Fill* method, you can specify *Offset* values to place asymmetrical rows in the boundary areas. The offset must be a floating number — one or greater than

© 2023 Cadence Design Systems, Inc.

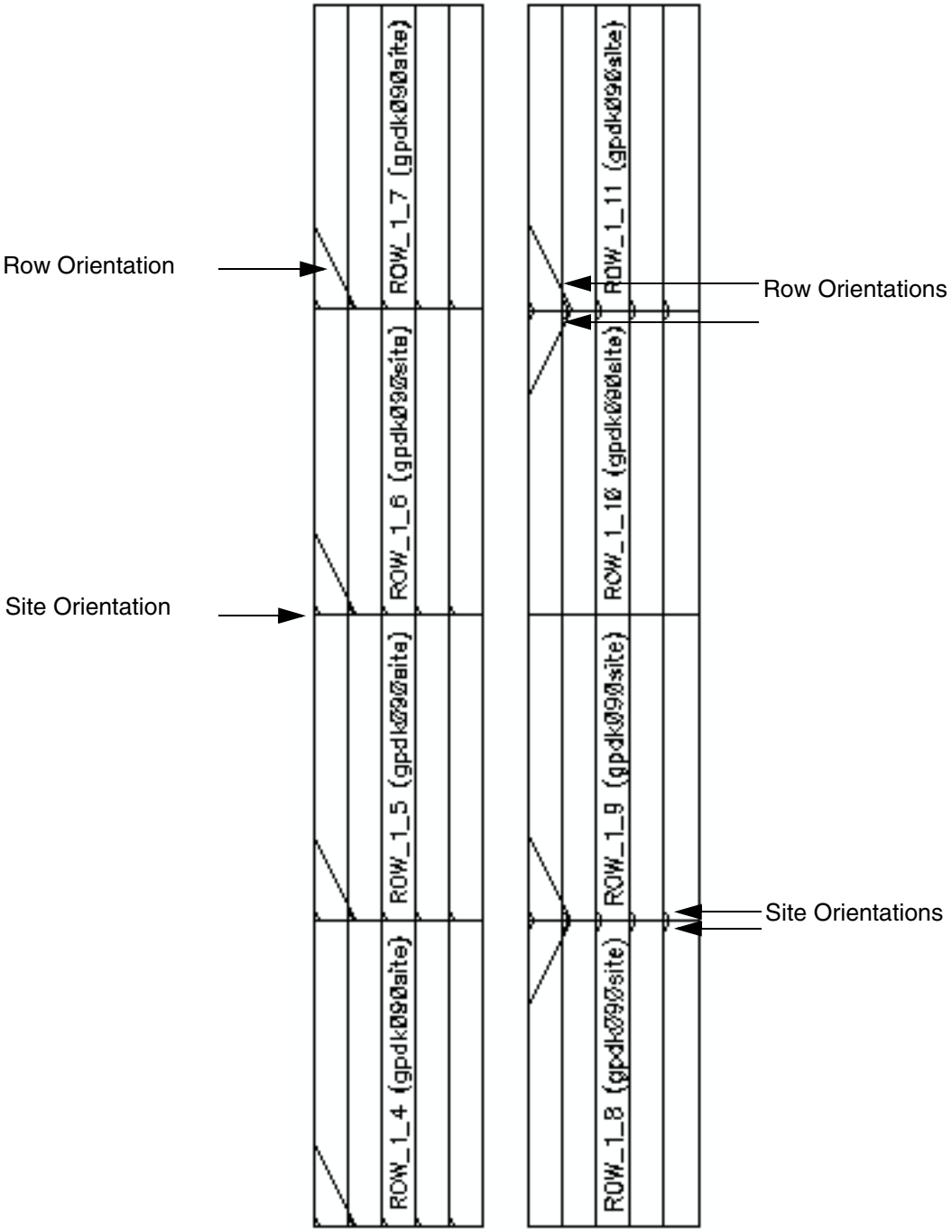
one. Set the distance from *Top*, *Bottom*, *Left*, and *Right* of the boundary in which the rows should not enter.

In the following example, the left image is *Row Orientation* as *Horizontal*, *Starting Site Orientation* is *R0*, *Create Method* is *Assisted*, *Number of Rows* is 4, *Number of Sites* is 5.

The right image also shows rows with the same settings, with additional settings *Flip Every Other Row* and *Flip The First Row* enabled and *Abut* set at 4 rows. This is useful to make

© 2023 Cadence Design Systems, Inc.

rows more compact and if you have rails in the rows, place source at the top of the rows and ground at the bottom. This avoids shorts in the rows.



© 2023 Cadence Design Systems, Inc.

Related Topics

Create Row Form

Rows

Technology File Site Definitions

Creating a Custom Placement Area

You use a custom placement area for placement of standard cells and custom cells. You use the Create Custom Placement Area form to create a custom placement area. The custom placement area is automatically snapped to an appropriate grid depending on the state of the snapCpaToPlacementGrid environment variable.

To create a custom area for placement:

1. In the layout window, choose *Create – P&R Objects – Custom Placement Area.*

   The Create Custom Placement Area form opens.

2. Select the method for creating the custom placement area as *Manual*, *Assisted*, or *Auto*.
3. Specify the attributes in the *Definition* section.
   - Type the name of the custom placement area in the *Base Name* field.

     When creating the custom placement area using *Manual* mode, you do not need to specify attributes such as the height, width, and orientation of the area.
   - Specify the *Height* and *Width* of the placement area, if using *Assisted* or *Auto* creation mode.
   - Specify *Number of Placement Areas* you want to create, if using *Auto* creation mode.
   - Specify the width between the placement areas in the *Channel Width* field.
   - Select the orientation of the placement area as *Horizontal* or *Vertical*.
   - Specify the flip and abut settings for the placement area. For example, you can set every other area to flip or set only the first area to flip, and specify the number of areas to abut or set to abut the first area.
4. Specify the template you want to use, in the *Templates* section.

© 2023 Cadence Design Systems, Inc.

Templates allow you to place rails and/or components automatically into the custom placement areas. Choose pre-existing rails, components, or both to create the custom placement area. If pre-defined rails or components exist, they are listed in the *Pre-Defined* box in the form. Select the ones you want to use for custom placement area creation one-by-one and move them to the *Selected* box by using the arrow icons.

Template definitions are associated only with the cellview in which they are defined. Therefore, the rail and component templates available to one cellview are not available to another cellview.

5. Draw the custom placement area on the canvas if you selected the *Manual* or *Assisted* create method or click on the canvas if you selected *Auto*.

Related Topics

Create Custom Placement Area Form

Templates

Placement and Routing Objects

Creating Clusters

Clusters provide a grouping mechanism for placement control interpreted and used by floor planning and placement tools. They are identified by a unique name. Clusters are not graphical objects.

Clusters are used to associate instances and cluster boundaries. They can own area and layer blockages. Each instance can belong to only one cluster, but a cluster can have any number of cluster boundaries. Cluster boundaries are used to indicate the collective area into which you can place the contents of a specific cluster of instances.

Clusters can be hierarchical, so that you can associate clusters with other clusters. Clusters can also own blockages. You define each cluster as one of the following types:

- Inclusive: Specifies that instances other than instances in this cluster may be placed in its associated cluster boundaries.
- Exclusive: Specifies that only instances in this cluster may be placed in its associated cluster boundaries.
- Suggested: Specifies that instances in this cluster may be placed outside of its associated cluster boundaries.

© 2023 Cadence Design Systems, Inc.

To create a cluster:

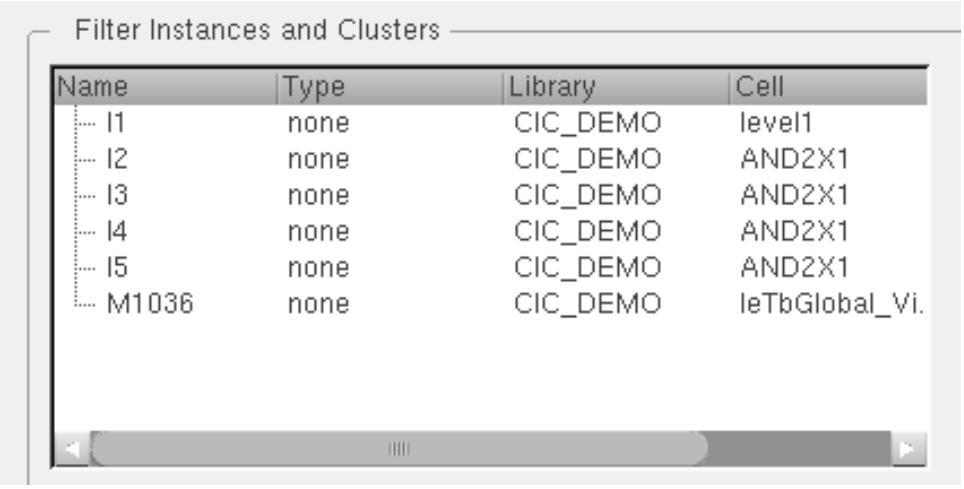
1. In the layout window, choose *Create – P&R Objects – Clusters*.

   The Create and Edit Clusters form appears.

2. In the *Filter Instances and Clusters* section, set the filtering criteria and click *Filter By.*

   The filtered objects are listed in the section.

3. Select the instances, clusters, and sub clusters to add new or existing clusters in the current cluster editing area of the form.

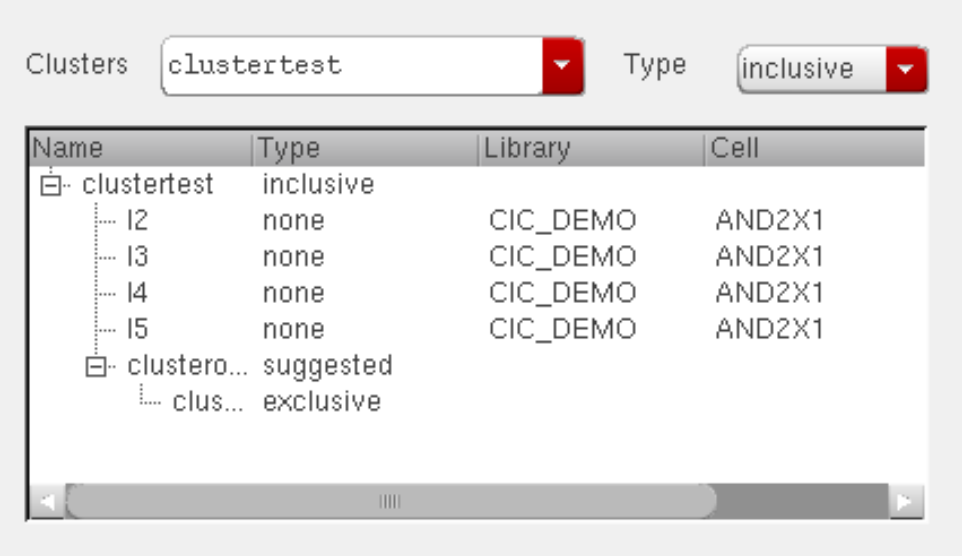


4. Click *Add to Cluster* to move the instances or clusters you have selected in *Filter Instances and Clusters* to the current cluster editing area of the form.

   The selected instances and clusters are added to the trunk. Or, if you select the branch in the current cluster editing area, the instances and clusters are added to the correct branch. The following figure shows the current cluster editing area listing the clusters

© 2023 Cadence Design Systems, Inc.

defined in the cellview in the tree table format. Use the scroll bar to display the view names.



5. Create new and edit existing clusters and sub clusters using the *Create*, *Create Sub Cluster*, *Flatten* and other options in the form. Because you cannot select clusters, you can use the options to edit the properties of clusters and to delete them.

   The newly created cluster is also added to the *Clusters* list, and the current cluster editing area displays the newly created cluster.

Related Topics

Create and Edit Clusters Form

Placement and Routing Objects

Track Patterns

Tracks patterns are uniformly-spaced orthogonal patterns extending across the base array.

The tracks that make up the track pattern provide guidelines for laying interconnect routes.

Gridded routers place the center-line of each route collinear with an unused segment of a routing track.

© 2023 Cadence Design Systems, Inc.

Related Topics

Technology File Requirements for Track Patterns

Displaying Track Patterns

Creating a Track Pattern

Placement and Routing Objects

Technology File Requirements for Track Patterns

You can create track patterns on `Metal` and `Poly` layers. The following shows the technology file requirement for creating track patterns on a metal layer.

```
layerDefinitions(
techPurposes(
(track 229 trk)
...
)
techDisplays(
(metal1 track packetName t t t t t)
...
)
techLayerPurposePriorites(
(metal1 track)
...
)
layerRules(
functions(
(metal1 "metal" 3)
...
)
```

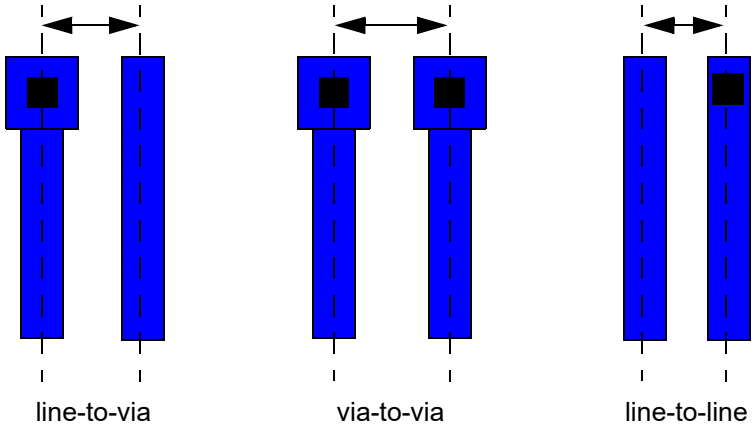
To efficiently utilize the routing area, tracks should be created considering the following design aspects:

- Minimum Routing Pitch
- Via Stacking
- Preferred and Non-Preferred Routing Directions
- Diagonally Adjacent Vias
- Track Spacing
- Number of Tracks
- Track Patterns Assigned to Routing Layer “None”

© 2023 Cadence Design Systems, Inc.

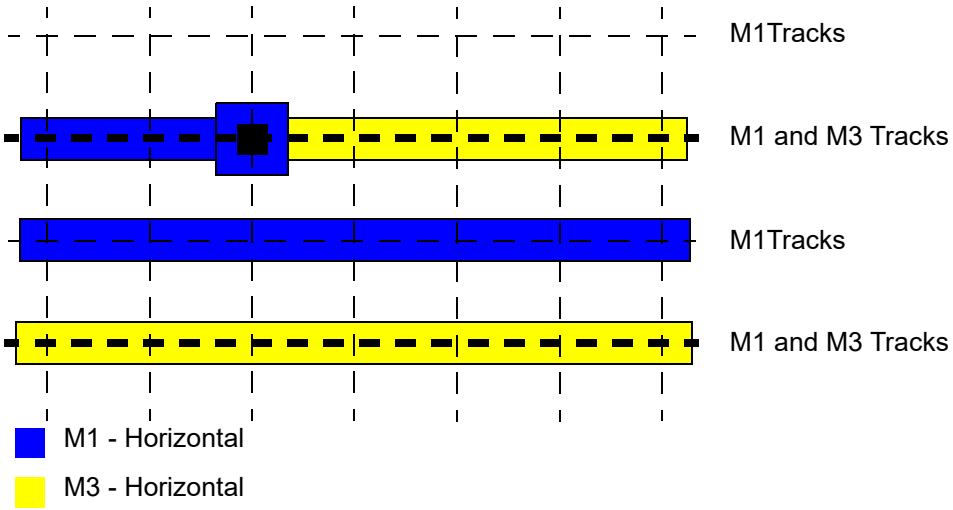
Minimum Routing Pitch

Track spacing is determined from the routing pitch. The most common routing pitch is line-to-via because it allows the minimum pitch between routes in which a via can be placed next to a route. A larger pitch does not efficiently utilize the routing area, where as a smaller pitch may result in low utilization of the routing area because the router will not be able to place vias next to a route.



Via Stacking

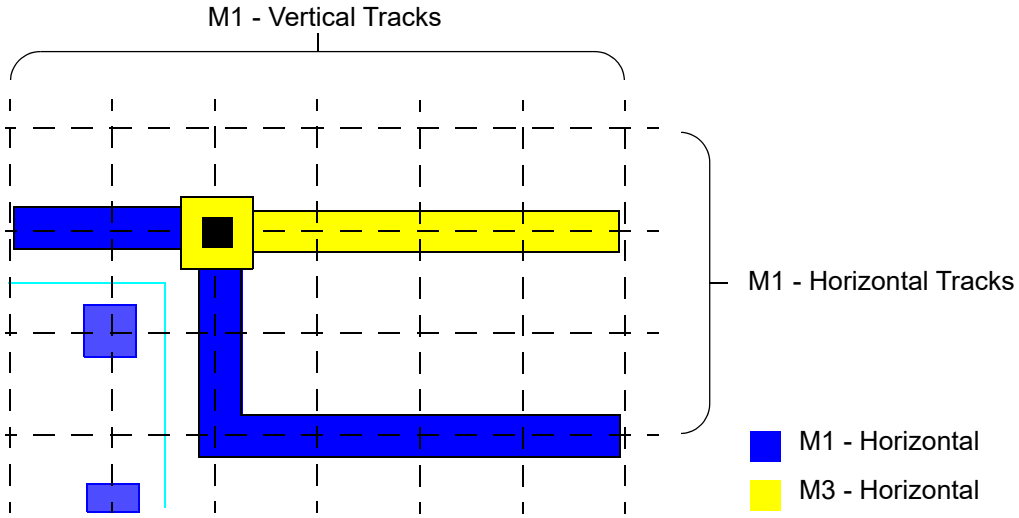
Routing layers using the same preferred direction can be placed on coincident tracks to allow for via stacking. If tracks cannot be coincident, place the tracks at a ratio, for example 1:2 or 2:3.



© 2023 Cadence Design Systems, Inc.

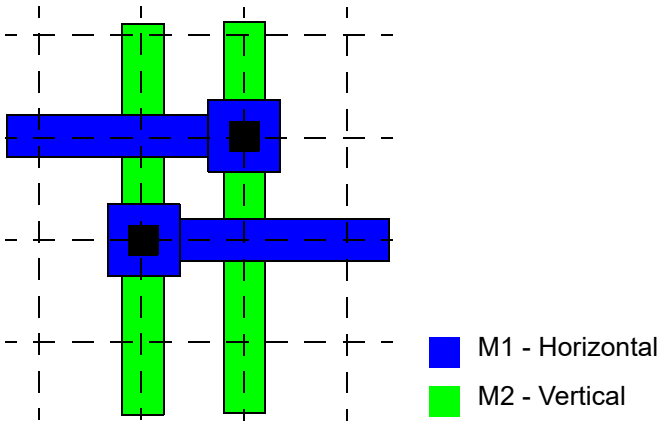
Preferred and Non-Preferred Routing Directions

Tracks are created in both the preferred and non-preferred routing direction to allow for X Y locations where routes can be jogged and vias and pins can be placed and connected. Points where a horizontal track on one layer crosses over a vertical track on another layer are via locations where a vias can be placed. Most routing is done on the tracks aligned in the preferred direction. Small-distance jogs use tracks in the non-preferred direction.



Diagonally Adjacent Vias

Increasing the routing pitch may be necessary to allow for diagonally placed adjacent vias. This may increase the orthogonal spacing but can reduce the overall die size by readily allowing connections to via rather than longer routes around congested areas.

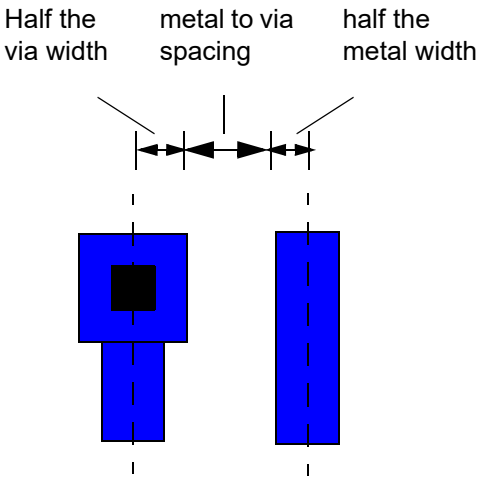


© 2023 Cadence Design Systems, Inc.

Track Spacing

You can determine the line-to-via track spacing (routing pitch) for each routing layer by adding the following:

Half the via width + metal to via spacing + half the width of the metal

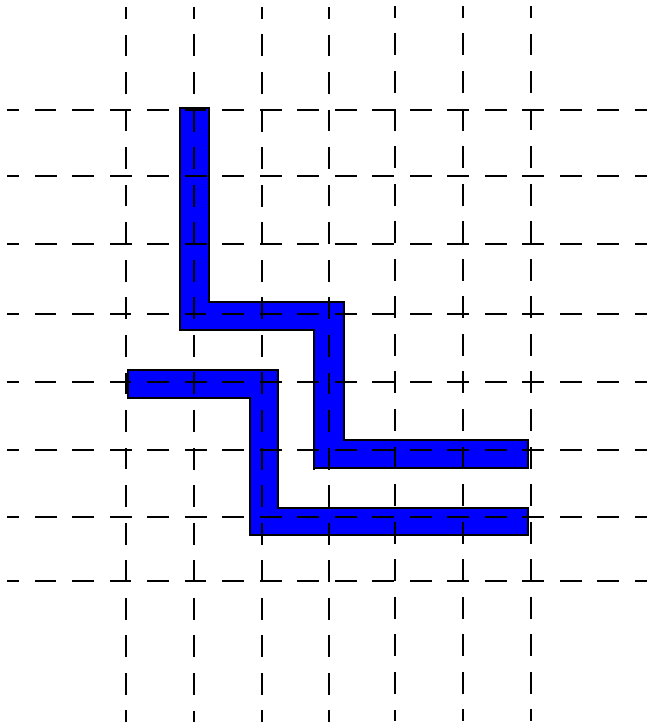


Number of Tracks

You must determine the number of tracks needed for the width and the height of a cellview. Calculate the number of horizontal tracks by dividing the height of the design area by the routing pitch of the metal. Determine the number of tracks needed in vertical direction using the same method. Create horizontal and vertical track patterns for each routing layer. When

© 2023 Cadence Design Systems, Inc.

Snap to Track is on in the Layout Editor Options form, you are only allowed to route in areas where there are intersecting horizontal and vertical tracks.



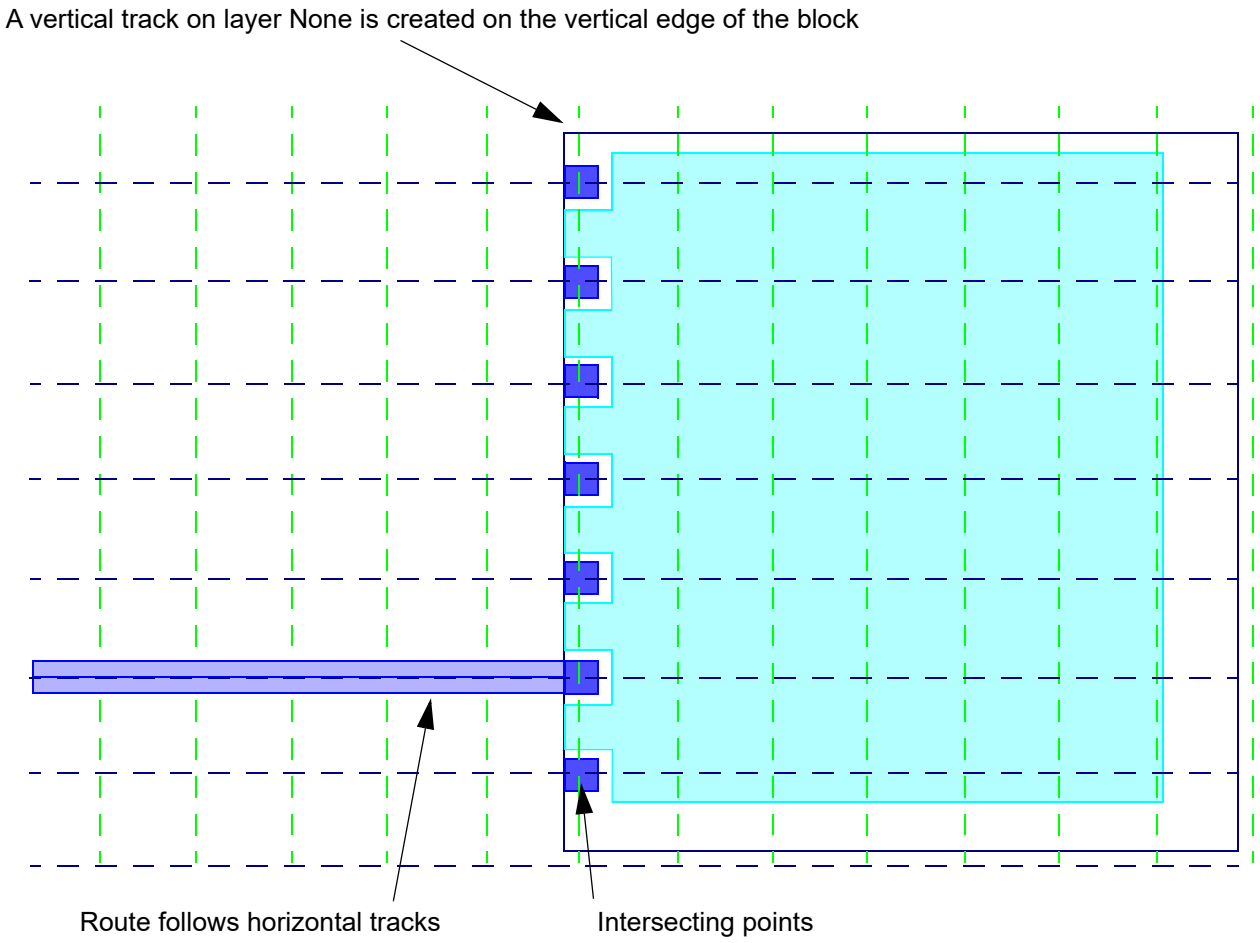
Track Patterns Assigned to Routing Layer “None”

Track patterns can be created without an assigned routing layer. This can be done to save a routing track for use with other applications. Some tools, such as a global router, section the chip by looking at the number of routing tracks. Creating additional tracks without a routing layer can trick the tool into seeing more sections.

This can also be used to defined points for placing pins without allowing routing in one direction. For example, on the vertical edge of a block, you may want to place pins for a horizontal route but you may not want to route on the vertical edge because the route would be half outside the block. To resolve this, you can create a vertical track directly on the vertical edge. The intersection points of the vertical track and the horizontal tracks define valid points

© 2023 Cadence Design Systems, Inc.

to place pins. The router can route to the pin inside the block using the horizontal tracks, but cannot route to the pin, inside the block, vertically.



Related Topics

Displaying Track Patterns

Creating a Track Pattern

Displaying Track Patterns

To display all track patterns:

1. Ensure track layers, which are layers with the purpose *track,* are valid and visible.

   If necessary, update the technology file.

© 2023 Cadence Design Systems, Inc.

2. In the Grids panel in the Palette assistant, enable the *Visibility* check box for *Tracks*.
3. To display individual track patterns, turn the visibility on or off, as required for specific track layers in the Layers panel.
4. Use the *View – Redraw* command to update the view each time display changes are made to track patterns.

   The track pattern spacing and how closely the design is viewed using *Zoom* determines if track patterns can be seen.

Related Topics

Palette Panels

Creating a Track Pattern

Technology File Requirements for Track Patterns

Creating a Track Pattern

Create tracks in both preferred and non-preferred layer directions.

To create a track pattern:

1. In the layout window, choose *Create – P&R Objects – Track Patterns*.

   The Track Pattern Editor form opens.
2. Filter the routing layers of the track patterns that you want to display in the form by selecting from the *Filter by Routing Layer* list.

   This allows you to limit the form to only show the track patterns on the routing layer you want to edit. This does not affect the routing layers displayed in the design window.
3. Select the routing layer for the track pattern from the *Layer* field.
4. Select track pattern *Direction* as *Horizontal* or *Vertical*.
5. Specify the track spacing routing pitch in the *Spacing* field.
6. Optionally, type a value to offset the wire from the track, in the *Origin Offset* field.
7. Type the number of required tracks in the *Number of Tracks*.
8. Set other options such as the color pattern for track patterns, width associated with the track, and the constraint group, as required.

© 2023 Cadence Design Systems, Inc.

9. Click *Update*.
10. Redraw the design window by using *View – Redraw* to view the track pattern you have created.

Related Topics

Track Pattern Editor Form

Displaying Track Patterns

Technology File Requirements for Track Patterns

Creating a PR Boundary

A placement and routing (PR) boundary is the boundary of a block for place-and-route applications. In a top-down design flow, it is commonly passed down to constrain the area in which the block's contents are placed. Any shape or instance created for the block should be enclosed within the PR boundary. In a bottom-up design flow, a PR boundary is used to indicate the primary extent of the block's contents. For cell placement that is directed by sites in a row object, the PR boundary determines the aspect of the cell that must align with the site. Only one PR boundary is allowed per cellview.

When you create or edit a PR boundary, to prevent instances from being placed off the grid, the snap period is taken into consideration when the following conditions are met:

- The snapPatternIndexSnappingLayers environment variable is set to `t`.
- The snapGridPeriod snap period constraint is defined for a global grid.
- The leSnapPatternSnappingHorizontalGridDetail or leSnapPatternSnappingVerticalGridDetail property is defined for instances.

To create a PR boundary:

1. In the layout window, choose *Create – P&R Objects – P&R Boundary*.
2. Press `F3` to open the Create P&R Boundary form.

   If the visibility setting of *P&R Boundary* in the Objects panel in the Palette assistant is turned off, a dialog box appears with the option to set the PR boundary to visible and proceed with the boundary creation. When you click *Yes*, *P&R Boundary* is set to visible in the Objects panel.

© 2023 Cadence Design Systems, Inc.

3. Select *Create as ROD Object* if you want to create the PR boundary as a ROD object. The *Name* field becomes editable, allowing you to change the name from the default.
4. Use the *Define Halo* button if you want to create a halo blockage for the PR boundary.
5. From *Create Method* choose a method and create the PR boundary:
   - *From Selected Set*: Select objects on the canvas to add to the selected set. Once you have selected the objects, click the *Apply* button to create the PR Boundary.
   - *Rectangle*: Specify a rectangle area on the canvas for the boundary. The *Apply* button and *Snap Mode* list are grayed out when using this method.
   - *Polygon*: Select a snap mode and specify a polygon area on the canvas for the boundary. The *Apply* button is grayed out when using this method.

Related Topics

Create P&R Boundary Form

Blockage Halos

Creating a Snap Boundary

A snap boundary is used for designs that implement any rectilinear macros or standard cells. A snap boundary must be a rectangle. Placement tools use the snap boundary to place instances of the design in rows, thereby aligning the snap boundary origin with the site origin. If the PR boundary is a rectangle, no snap boundary is needed. You may need a snap boundary for cells with rectilinear (but non-rectangular) PR boundaries. Only one snap boundary is allowed per cellview.

To create a snap boundary:

1. In the layout window, choose *Create – P&R Objects – Snap Boundary.*
2. Press `F3` to open the Create Snap Boundary form.
3. Select *Create as ROD Object* if you want to create the snap boundary as a ROD object. The *Name* field becomes editable, allowing you to change the name from the default.
4. From *Create Method* choose a method and create the snap boundary.
   - From *Selected Set*

© 2023 Cadence Design Systems, Inc.

Select *From Selected Set.* Then, select objects on the canvas to add to the selected set. Once you have selected the objects, click the *Apply* button in the form to create the snap boundary.

- *Rectangle*

  Select *Rectangle*. Then, specify a rectangle area on the canvas for the snap boundary. The *Apply* button field is grayed out when using this method.

Related Topics

Create Snap Boundary Form

Creating an Area Boundary

Area Boundary is a generic boundary in a physical block. It is used to indicate a region of a block for various purposes defined by a design methodology, thus each area boundary has a unique name. You can tie an area boundary to a constraint group that can direct advanced placement and routing solutions.

To create an area boundary:

1. In the layout window, choose *Create – P&R Objects – Area Boundary.*
2. Press `F3` to open the Create Area Boundary form.
3. Assign a name to the area boundary in the *Name* field. The name must be unique to the cellview.

   The software automatically assigns names beginning with `AB` (for area boundary), followed by an underscore and a number, to make the name unique to the cellview.
4. Select *Create as ROD Object* if you want to create the area boundary as a ROD object. The *Name* field becomes editable, allowing you to change the name from the default.
5. Select *Soft Fence* if you want to create the soft fence.

   Objects enclosed by the area boundary with a soft fence are compacted relative to each other as well as compacted as an entity relative to objects outside the boundary.
6. From *Create Method* choose a method and then create the area boundary on the canvas:

   - *From Selected Set*

© 2023 Cadence Design Systems, Inc.

Select *From Selected Set.* Then, select objects on the canvas to add to the selected set. Once you have selected the objects, click the *Apply* button in the form to create the area boundary.

- *Rectangle*

  Select *Rectangle*. Then, specify a rectangle area on the canvas for the boundary. The *Apply* button field is grayed out when using this method.

- *Polygon*

  Select *Polygon* and select a snap mode in the form. Then, specify a polygon area on the canvas for the boundary. The *Apply* button is grayed out when using this method.

Related Topics

Create Area Boundary Form

Creating a Cluster Boundary

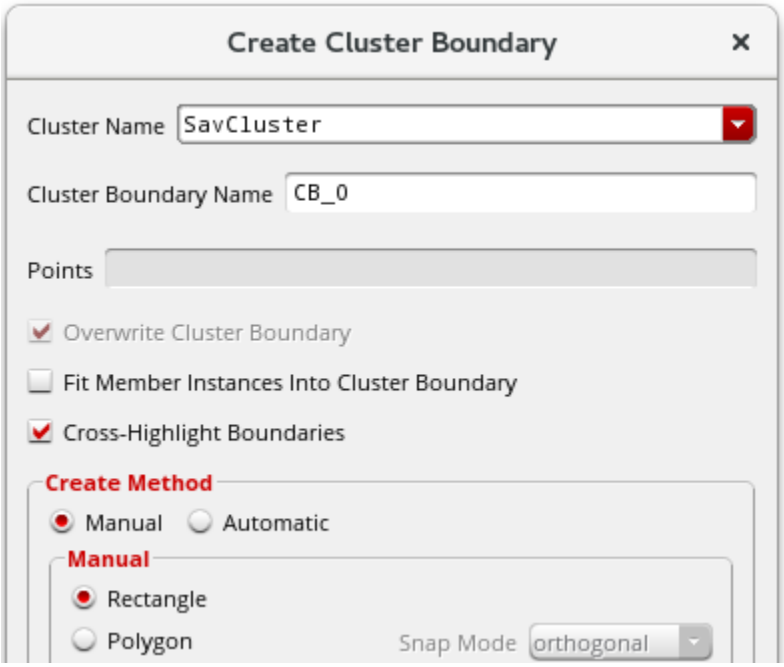
Cluster Boundary defines an area in which the contents of a specific cluster are included or excluded, depending upon their definition. The cluster boundary is attached to a specific cluster object. Each cluster boundary has a unique name with respect to all other cluster boundaries in a given cellview.

To create a cluster boundary:

1. In the layout window, choose *Create – P&R Objects – Cluster Boundary.*
2. Press `F3` to open the Create Cluster Boundary form.

© 2023 Cadence Design Systems, Inc.

3. Select an existing cluster for the boundary from the *Cluster Name* list.



4. Optionally, overwrite the automatically assigned name by typing a name for the boundary in the *Cluster Boundary Name* field. The name must be unique to the cellview.

   The software automatically assigns names beginning with CB (for cluster boundary), followed by an underscore, then a number unique to the cellview.

5. Turn on *Cross-Highlight Boundaries* when generating new cluster boundaries; use this option to help you associate existing clusters with their boundaries.

6. From *Create Method*, choose either *Manual* or *Automatic*.

   - When *Manual* is selected, choose either *Rectangle* or *Polygon*
     - *Rectangle*

       Specify a rectangle area on the canvas for the boundary.
     - *Polygon*

       Select a snap mode and specify a polygon area on the canvas for the boundary.
   - When *Automatic* is selected, the boundary is created using the aspect ratio, origin, and target area. *Aspect Ratio W/H* is the width-to-length ratio of the cluster boundary. For example, a value of 1 specifies a square boundary, 2 specifies a boundary twice as long as it is wide, and .5 specifies a boundary half as wide as it is long. The default is 1.

© 2023 Cadence Design Systems, Inc.

If the target area is greater than 0 and you create a rectangular cluster boundary whose area is less than the target area, a warning is issued in the CIW stating that the utilization value is greater than 100%.

Related Topics

Create Cluster Boundary Form

Create Commands Supporting Snap Pattern Snapping

Snap pattern snapping is supported in the *Create Rectangle*, *Create Pin*, *Create Instance*, *Create Via*, and *Create Wire* object creation commands.

In the *Create Rectangle* command, both the points are snapped to the grid by applying negative enclosure for the lower edge and positive enclosure for the upper edge of the rectangle. While drawing rectangles on layer-purpose pairs (LPPs) with the `singleTrackCenter` snapping mode, the points are snapped with respect to the same track, restricting the dimension to twice the enclosure value. You can disable snapping while drawing rectangles using a special mode enabled by using the `Shift` key.

Related Topics

Rectangles Created on a Layer with singleTrackCenter Snapping Mode

Pin Snapping to Width Spacing Patterns

Dynamic Display During Pin Creation and Editing

© 2023 Cadence Design Systems, Inc.

6

Object Editing in Layout XL

Using the Virtuoso® Layout Suite layout editor, you can edit objects in your designs. For example, you can move, copy, and stretch the objects. You can also flip, rotate, and align objects according to your design requirements.

- Object Moving
- Object Copying
- Object Stretching
- Object Deletion
- Object Flipping
- Object Rotation
- Object Chopping
- Object Merging
- Object Yanking and Pasting
- Object Reshaping
- Object Splitting
- Modifying Corners of a Polygon
- Modifying Object Size
- Parent-Child Relationship Between Objects
- Object Conversion
- Object Alignment
- Editing Slots
- Editing Groups
- Edit Commands Supporting Snap Pattern Snapping

© 2023 Cadence Design Systems, Inc.

■ Pin-Path and Pin-PathSeg Connections Preserved

Object Moving

You can use the *Move* command to move an object to another location in the current or another cellview. The following table describes the support available for moving objects.

Object	Support for the Move Operation
Multipart paths, Multipart rectangles	You can move multipart paths and multipart rectangles to a new cellview. All data remains intact. If the multipart path is chopped, that data also moves with it. Multipart paths and multipart rectangles are renamed if there are objects with similar names in the new cellview. To retain alignments while moving to another cellview, each of the aligned objects must be part of the selected set to be moved.
ROD objects	You can move a ROD object within the same cellview or to another cellview. Within the same cellview, moving a ROD object that has other objects aligned to it causes the aligned objects to move as well. ROD objects are renamed if there are objects with similar names in the new cellview. When you move a ROD object between cellviews and the ROD object is aligned to other ROD objects, the system preserves the alignment only when the aligned ROD objects are also in the selected set; otherwise the alignment is broken. You should avoid rotating aligned ROD objects during a move operation because the aligned handle names are not updated after the move, so the results might not be what you want.
Rulers	You cannot move a ruler from one cellview to another.
Blockages	When blockages are in the selected set, the *Change To Layer* functionality remains enabled. The new layer information applies to all shapes in the selected set. If an owner is moved, the blockage follows the owner.
Boundaries	All instances associated with the cluster, part of the cluster, and inside the bounding box are moved as a group. You can use the *Rotate*, *Sideways*, and *Upside Down* buttons on the Move form for boundary objects.

© 2023 Cadence Design Systems, Inc.

(Virtuoso RF Option) When you move a curved path, with the gravity feature enabled, the end points of the curved path snap to the end points or centerline of another curved path or to the center of a pin or a via. When you move a pin or a via, its center is snapped to the end points of a curved path or to the center of another pin or via.

Related Topics

Moving an Object by Direct Manipulation

Moving Objects Manually

Moving an Object by Using the Move Command

Moving or Copying an Object to Another Layer

Moving an Object to Another Cellview

Moving a Multipart Rectangle

Moving Objects Using the Move Form

Moving an Object by Direct Manipulation

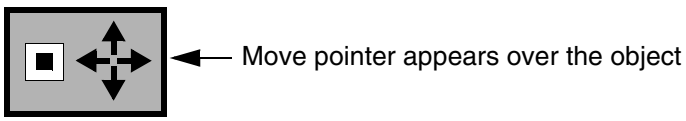
You can move objects by selecting and dragging them on the canvas.

To move an object:

1. On the layout canvas, select an object.

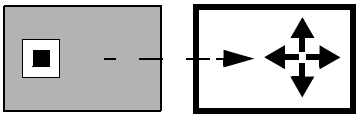
   Alternatively, you can select an object by specifying the points at which the object is placed in the design area by using the Enter Points form.

2. Move the pointer over the object to see the move pointer.

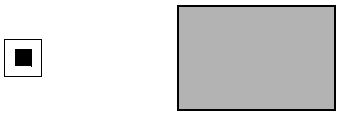


© 2023 Cadence Design Systems, Inc.

3. Press the left mouse button at the reference point, which is the point from which the move operation starts, and drag the pointer to the destination point. Alternatively, you can specify the reference and destination points using the Enter Points form.



4. Release the left mouse button.



The object is moved

Related Topics

Enter Points Form

Moving Objects Manually

To move an object manually on the layout canvas:

1. From the layout window menu bar, choose *Edit – Move*.

   The layout window prompts you to select an object to move.

2. Click an object in either the layout or the schematic.

   The object is highlighted in both the schematic and layout windows. An outline of the object appears in the layout canvas and follows the cursor until you click in the layout canvas to place the object.

   Draglines show the connections from pins of the selected object to pins of the nearest instance or the nearest I/O pin.

   The layout window prompts you to click the new location for the object.

3. Click the location where you want to place the object.

   The object is moved to the new location.

© 2023 Cadence Design Systems, Inc.

If you select a component in the layout or the schematic before you choose the *Move* command, Layout XL highlights the corresponding component in the schematic or layout.

4. Click another object to move or press `Esc` to cancel the command.

Using the *Move* command cancels any existing commands in the current or in other windows.

Related Topics

Changing the Appearance of Draglines

Moving an Object by Using the Move Command

To move an object using the *Move* command:

1. In the layout window, choose *Edit – Move* or click the *Move* icon  on the *Edit* toolbar.

   Alternatively, right-click at any free space on the design canvas to display the *Layout* shortcut menu and select *Move.*

2. Press `F3` to open the Move form.
3. Select one or more objects. If you click a single object, the click point is taken as the reference point, which is the point from where the move starts. If you create a selection box or press `Shift` while selecting multiple objects, the *Move* command prompts you for a reference point.
4. Click the reference point for the move operation, if prompted. Alternatively, you can specify the reference point for the move operation by using the Enter Points form.
5. Click on the canvas where you want to move the objects. Alternatively, you can specify the destination point by using the Enter Points form.

Related Topics

Move Form

Enter Points Form

Moving or Copying an Object to Another Layer

To move or copy an object to another layer:

© 2023 Cadence Design Systems, Inc.

1. In the layout window, choose *Edit – Move* or *Edit – Copy*.
2. Press `F3` to open the form.

   The Move or Copy form opens.
3. From *Change To Layer*, select the layer to which you want to move or copy the object.
4. On the canvas, click the object whose layer you want to change.
5. Click where you want to move or copy the object.

   The object is moved or copied and is now on the new layer.

Related Topics

Move Form

Copy Form

Moving an Object to Another Cellview

To move an object to another open cellview:

1. On the layout canvas, select the object you want to move.
2. Choose *Edit – Move* to invoke the *Move* command.
3. Place the pointer in the other open cellview.
4. Click on the canvas to place the object.

Related Topics

Move Form

Moving a Multipart Rectangle

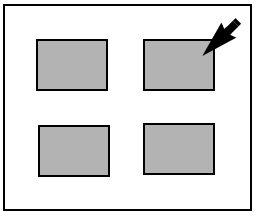
When you move a multipart rectangle within a cellview, the master rectangle and all of its associated arrays of subrectangles move together. You cannot move a master rectangle separately from its associated subrectangles, nor can you move subrectangles separately from their master rectangle.

© 2023 Cadence Design Systems, Inc.

When you move a multipart rectangle that has objects aligned to it, the objects move to stay in alignment with the multipart rectangle.

To move a multipart rectangle:

1. In the layout window, choose *Edit – Move*.
2. In full selection mode, choose the multipart rectangle by clicking on the master rectangle or any subpart.



Click any part of the multipart rectangle

3. Click on the canvas to place the multipart rectangle. The aligned objects follow.

The multipart rectangle and all objects aligned to it appear in the new location.

Related Topics

Move Form

Moving Objects Using the Move Form

You can use the Move form to change the following:

- Layer of the object you select
- Angles at which you can move the object
- Presence of draglines on the object as you move it
- Orientation of the object you select

To move an object using the Move options form:

1. From the layout window menu bar, choose *Edit – Move*.
2. Press `F3` to display the Move form.
3. To change the direction in which you can move an object, choose one of the options from the *Snap Mode* cyclic field: *anyAngle*, *diagonal*, *orthogonal*, *L90XFirst*, or *L90YFirst*.

© 2023 Cadence Design Systems, Inc.

4. To change the layer on which a shape is drawn, choose the new layer from the *Change To Layer* cyclic field. You can search for a specific layer-purpose pair using the search filter in the *Change To Layer* list box.
5. Choose *Treat as absolute point* to consider the points specified in *Delta X,Y* fields as absolute points.
6. To specify a distance to move the selected component, use the *Delta X* and *Y* fields.
   a. Type the distance to move in the horizontal direction into the *X* field.
   b. Type the distance to move in the vertical direction into the *Y* field.
   c. Click *Apply* to complete the move.

   To show draglines for an object you move, turn on *Display draglines*.
7. To keep chained devices together as you move them, expand the *Connections* section and select a *Chain Mode*. Choose from *All*, *Selected*, *Selected Plus Left*, or *Selected Plus Right*.
8. To snap the moved object to the nearest grid, expand the Snap section and select *Pins and boundary to grid.* The snapping behavior depends on the block type. For digital blocks, the place and route boundary is snapped to the placement grid and pins with placement status placed are snapped to the routing grid. For analog blocks, both place and route boundary and placed pins are snapped to the manufacturing grid. Pins with placement status unplaced, fixed, locked, or unknown are not snapped, even if they were touching the boundary before the move.
   a. Select *Pins to boundary* to automatically snap pins to the place and route boundary.
   b. Select *Allow pin resizing* to specify that the pin needs to be resized during pin snapping.

      Environment Variable: addIxStickyNetPropOnDummy

      The *Allow Pin Resizing* setting that you select in the *Move* form is applicable only for the current instance. For all other instances, the *Allow Pin Resizing* setting in the *Floorplan Global Options* form is considered as the default setting. Changes that you make to the *Allow Pin Resizing* setting in the *Move* form cannot overwrite the global setting.
9. To change the orientation of the object, click the button describing the transformation you want: *Rotate*, *Sideways*, or *Upside Down*.

© 2023 Cadence Design Systems, Inc.

Related Topics

Move Form

Changing the Appearance of Draglines

Snapping Soft Block Pins to Grid

Object Copying

You can use the *Copy* command to place the copy of an object in a cellview. You can copy an object to the same cellview or to another cellview.

Important

When you copy an object to another cellview, any constraints associated with the object are not copied to the resultant cellview.

The following table describes the support available for copying various objects.

Object	Support for the Copy Operation
ROD objects	■ You can copy a ROD object within the same cellview or to another cellview. The system automatically assigns unique names to the copies. ■ Alignment between ROD objects in a selected set determines the alignment between the corresponding copied objects. ■ Alignment with objects that are not in the selected set is ignored. ■ You can copy a whole multipart path (MPP) or a whole multipart rectangle (MPR), but not an individual subpart of an MPP or MPR.

© 2023 Cadence Design Systems, Inc.

Object	Support for the Copy Operation
Curved Shapes	The design must be associated with the technology file that has non-IC fabric. **(Virtuoso RF Option)** When you copy a curved path, with the gravity feature enabled, the end points of the curved path snap to the end points or centerline of another curved path or to the center of a pin or a via. When you copy a pin or a via, its center is snapped to the end points of a curved path or to the center of another pin or via.
Rulers	You cannot copy a ruler from one cellview to another.
Blockages	■ If only the blockage object is selected, the resultant copied blockage does not have any owner attached to it. ■ If both the blockage object and its owner are selected, the resultant copied owner and blockage objects retain the ownership information. ■ If only the owner is selected, all blockages owned by that owner are copied to the resultant owner object and the blockage ownership information is retained. This behavior also applies when you copy blockages from one cellview to another.
Boundaries	■ There can only be one PR boundary and one snap boundary in a cellview. Therefore, the *Copy* command works only when you copy a boundary from one cellview to another. If a PR boundary or snap boundary already exists in the destination cellview, a copy is not created. ■ If you copy an area boundary object with an associated constraint group between cellviews, the constraint group is copied with the area boundary.

You can use the Copy form to create group arrays. For information, see Group Arrays.

Related Topics

Copying an Object

Array of Copied Objects

© 2023 Cadence Design Systems, Inc.

Copying an Object to Another Cellview

Cellviews Copied Across Libraries

Copying an Object in Repeat Mode

Copying an Object

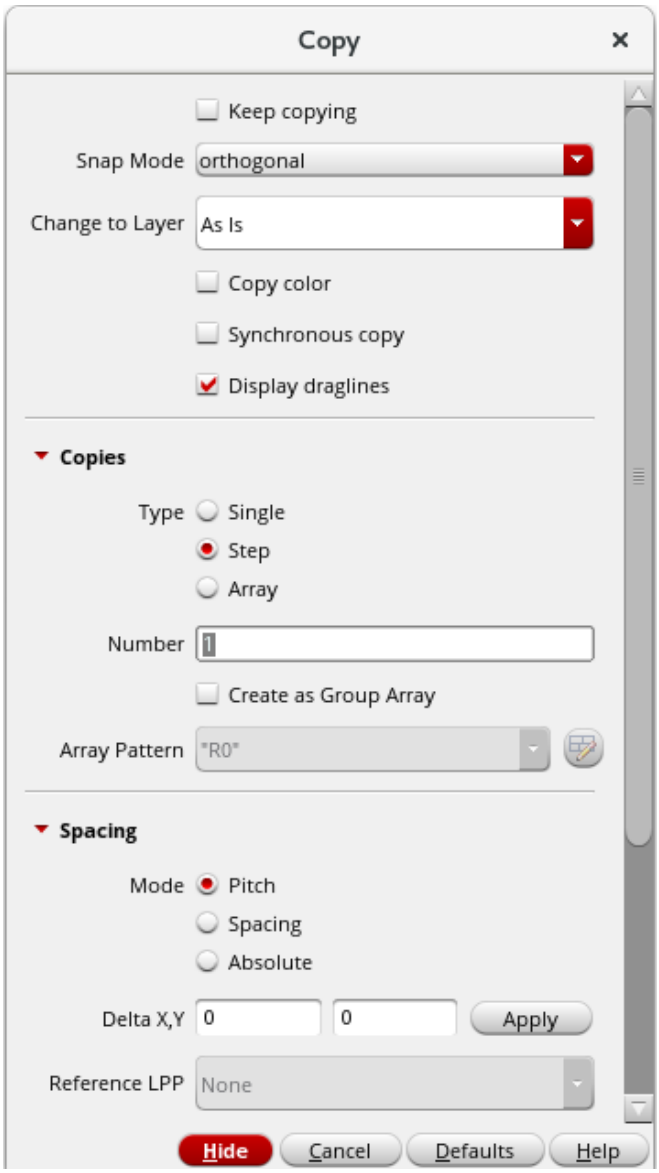
To copy an object:

1. In the layout window, choose *Edit – Copy* or click the *Copy* icon  on the *Edit* toolbar.

   Alternatively, right-click an empty space on the design canvas and choose *Copy* from the *Layout* shortcut menu.

© 2023 Cadence Design Systems, Inc.

2. Press `F3` to open the Copy form.



3. Select the *Keep copying* check box if you want to create multiple copies of an object.
4. Select an option from the *Snap Mode* list based on the direction in which you want to move the copied object.
5. Select a layer-purpose pair from the *Change to Layer* list if you want to copy an object to a different layer.
6. Select the *Type* of *Copies* you want to create. For example, select *Array* and specify the number of rows and columns if you want to create an array of copied objects.

© 2023 Cadence Design Systems, Inc.

For information about creating a group array, see Creating a Group Array by Using the Copy Form.

7. Optionally, select *Copy connectivity* if you want to copy the connectivity information on a path, pathSeg, fluid guard ring, or via to the new object.
8. In the *Chain Mode* list, select the appropriate option if you want to copy a chain of instances.
9. On the design canvas, select one or more objects by using either of the following methods:
   - To select a single object, click it. Alternatively, you can select an object by specifying the points at which the object is placed on the design canvas by using the Enter Points form. To select multiple objects, hold down the `Shift` key and click the required objects, one at a time.

     A copy of the selected object appears on the design canvas, attached to the pointer. This copy stays attached to the pointer and moves when you move the pointer. The *Copy* command prompts you to click at the location where you want to place the copy.
   - Drag the pointer over the objects you want to select. You can add more objects to the selected set by holding down the `Shift` key and then clicking the required objects or by dragging the pointer over the required objects.

     The *Copy* command prompts you to click on the canvas to specify a reference point.
10. Click on the design canvas to place the copy. Alternatively, you can specify the points at which you want to place the copy using the Enter Points form.

    If you copy a parent object, copies for all its child objects are created, whether or not you select the child objects individually. In pre-selection mode, the copies of the parent and all the child objects remain selected after you place them on the design canvas.

Important

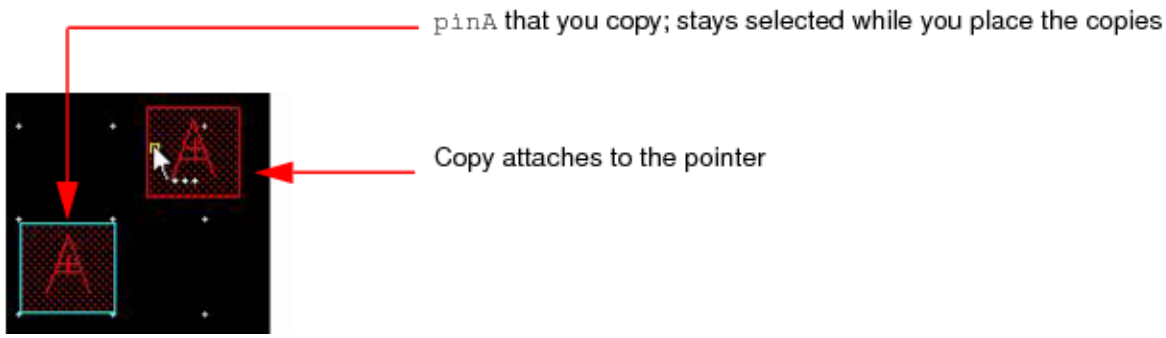
The reference point is set to the lower-left corner of the selection set on clicking the *Apply* button in the Spacing section of the Copy form.

11. Press `Esc` or click *Cancel* in the form to finish copying.

The following example illustrates the creation of multiple copies of `pinA` on `Metal1`, where the copies are placed on different layers, `Metal3` and `Metal4`. In the *Copy* form, you select the *Keep copying* check box, set *Snap Mode* to *anyAngle,* and set *Change To Layer* to

© 2023 Cadence Design Systems, Inc.

`Metal3`. On the canvas, when you select `pinA`, a copy of the pin attaches to the pointer and `pinA` remains selected.

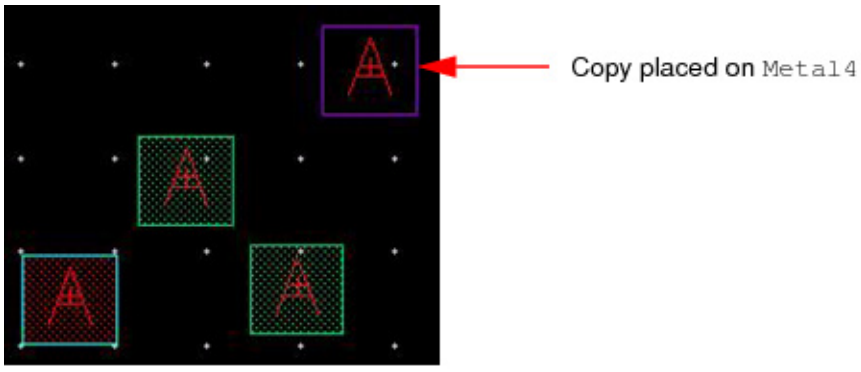


You need to click on the design canvas to place the first copy. Notice the change in the layer-purpose pair of the copy you place. Copy remains attached to the pointer.



You need to click on the design canvas to place the second copy, also on `Metal3`.

Next, you press `F3` to display the Copy form and select `Metal4` from the *Change To Layer* list. As a result, the next copy is placed on `Metal4` when you click on the design canvas to place the copy. Press `Esc` to finish creating the copies.



© 2023 Cadence Design Systems, Inc.

If the enableCopyAndPaste environment variable is enabled, then when using the *Copy* command, the selected objects and the reference point are automatically stored in the buffer after the reference point for the copy command has been specified. You can use the *Paste* command to paste the objects stored in the buffer. The *Copy* and *Yank* commands use the same buffer to store objects.

 Video

To view how to use the Copy Connectivity option in the Copy command, watch the video, *How to Use Copy Connectivity Option in Copy Command*.

Related Topics

Copy Form

Enter Points Form

Array of Copied Objects

Array of Copied Objects

You can create an array of copied objects. The array of copied objects is flat, which means that all objects in the array can be selected individually. The array can be placed only in the original cellview; it cannot be copied to another cellview.

You can create an array of copied objects by using the *Copy* command interactively. This involves specifying the distance between the copies by using the mouse pointer on the canvas. You can also create an array by specifying the distance between the copies in the Copy form by using the following spacing mode options:

- *Pitch*: Lets you specify the delta that applies between the objects in an array the distance specified in the *Delta X, Y* field. The distance is measured between the same-edge corners of the objects in the array.
- *Spacing*: Applies the distance between the objects in an array that is specified in the *Spacing X, Y* fields. The distance is measured between the adjacent-edges of the objects in the array.
- *Absolute*: Enables you to copy an object at the location specified in the *X, Y* fields.

In pre-selection mode, all copies of the array are selected in the end. The original object is not selected. In post-selection mode, no object remains selected after the copies are placed. The object you copy is part of the resultant array only when you specify the corner-to-corner (*Delta*) or edge-to-edge (*Spacing*) distance.

© 2023 Cadence Design Systems, Inc.

You can use the Copy form to create group arrays. Group arrays cannot be created by using the *Copy* command interactively. For information, see Group Arrays.

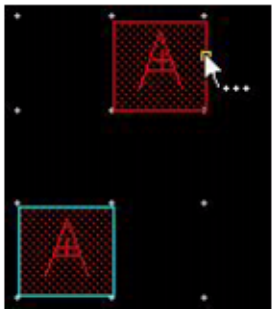
Related Topics

Copy Form

Creating an Array by Using the Copy Command Interactively

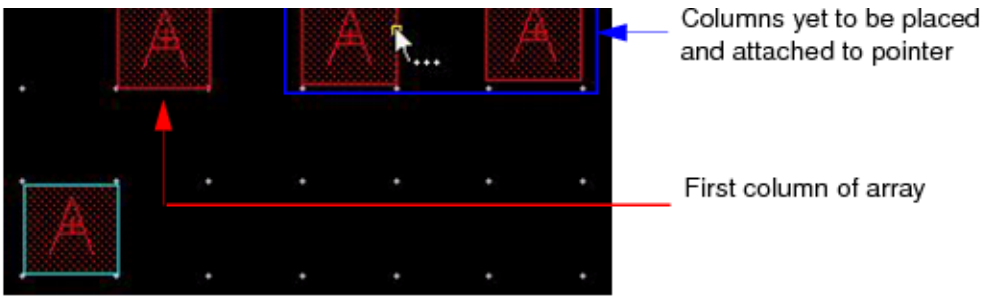
To create an array without entering the distance between the copies in the Copy form:

1. On the layout canvas, select the object to copy.
2. Choose *Edit – Copy* and press `F3` to open the Copy form.
3. Select the *Array* radio button in the *Copies* section.
4. In the *Rows* and *Columns* fields, type the number of rows and columns you want in the array, say `3` rows and `3` columns.
5. Click on the canvas to place the first object of the array.



6. Click on the canvas to place the second column of the array.

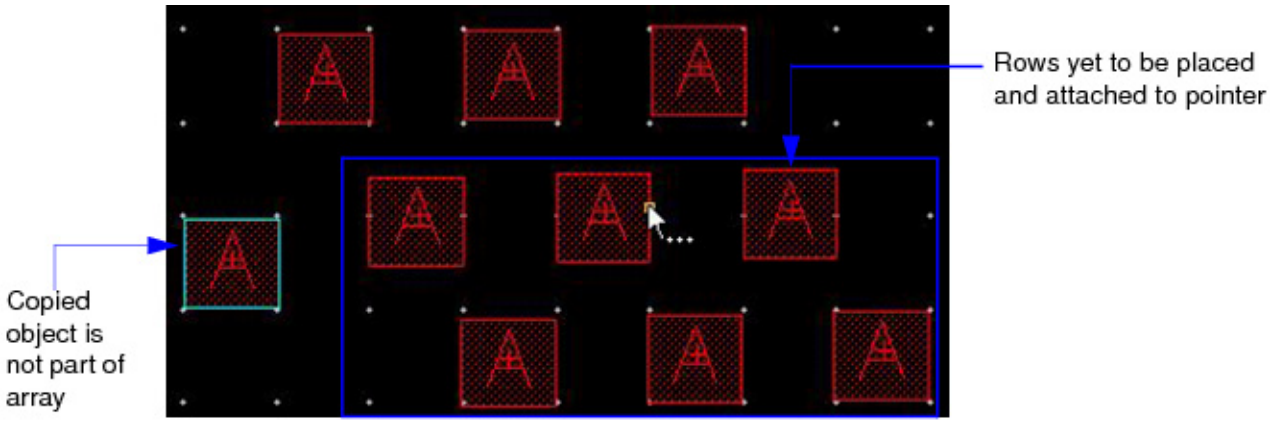
   The rest of the columns are automatically placed. The distance between the first and second copies controls the spacing and orientation of the remaining columns.



© 2023 Cadence Design Systems, Inc.

7. Similarly, click on the canvas to place the rows of the array and complete the array.

   The distance between the first and second rows controls the spacing and orientation between the remaining rows.



Group arrays cannot be created by using the *Copy* command interactively.

Related Topics

Copy Form

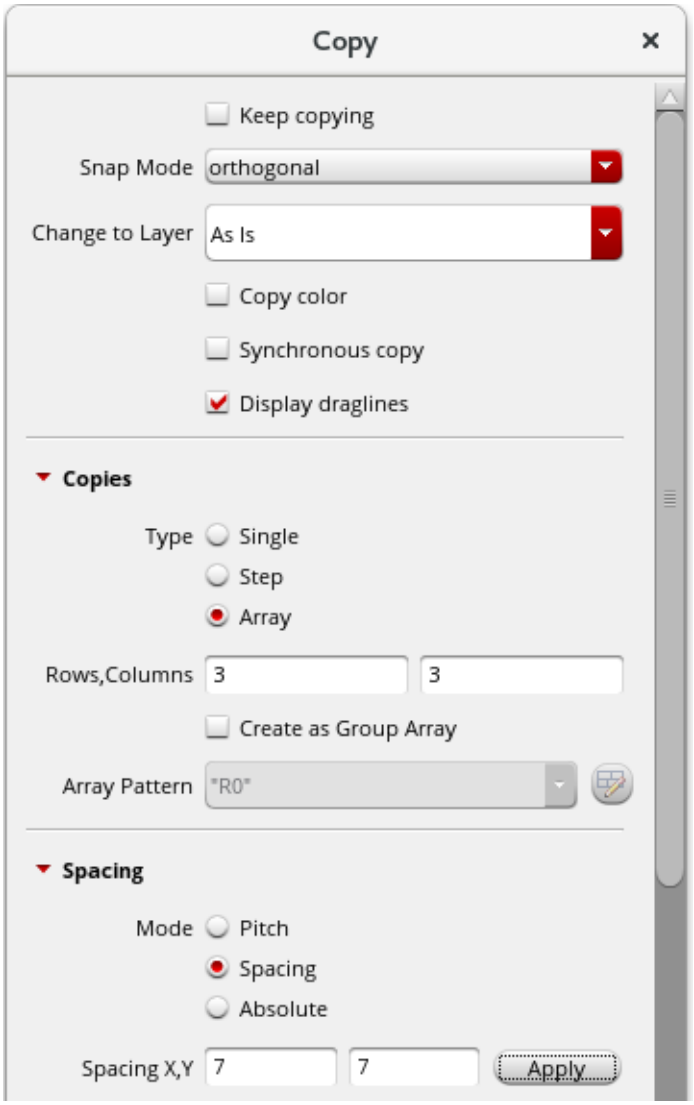
Creating an Array by Specifying Spacing Mode and Values

To create an array of copied objects by specifying spacing mode and spacing values in the Copy form:

1. On the layout canvas, select the object to copy.
2. Choose *Edit – Copy* and press `F3` to open the Copy form.
3. With the *Array* check box selected as the copy type, in the *Rows* and *Columns* fields, type the number of rows and columns you want in the array.
4. Select one of the following modes in the *Spacing* section and specify the spacing value:
   a. *Pitch*: Select the *Pitch* radio button and specify the spacing in the *Delta X,Y* fields.
   b. *Spacing*: Select the *Spacing* radio button and specify the spacing in the *Spacing X,Y* fields.

© 2023 Cadence Design Systems, Inc.

c. *Absolute*: Select the *Absolute* radio button and specify the spacing in the *X, Y* fields.

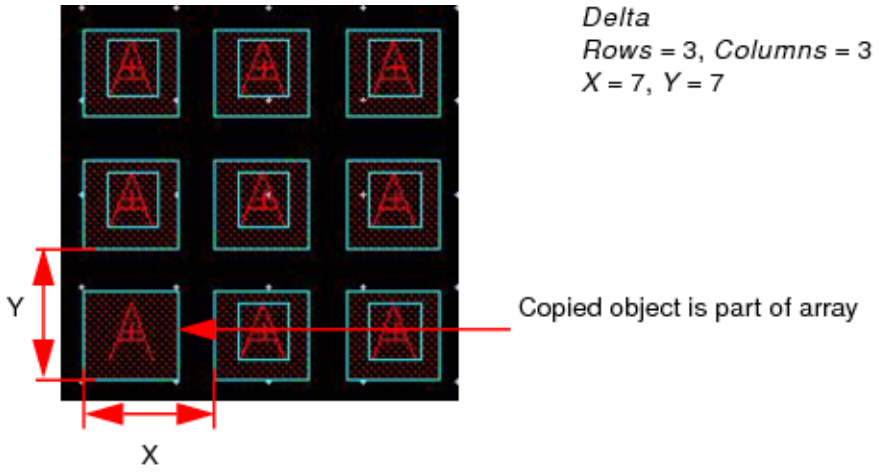


5. Click *Apply* in the *Spacing* section.

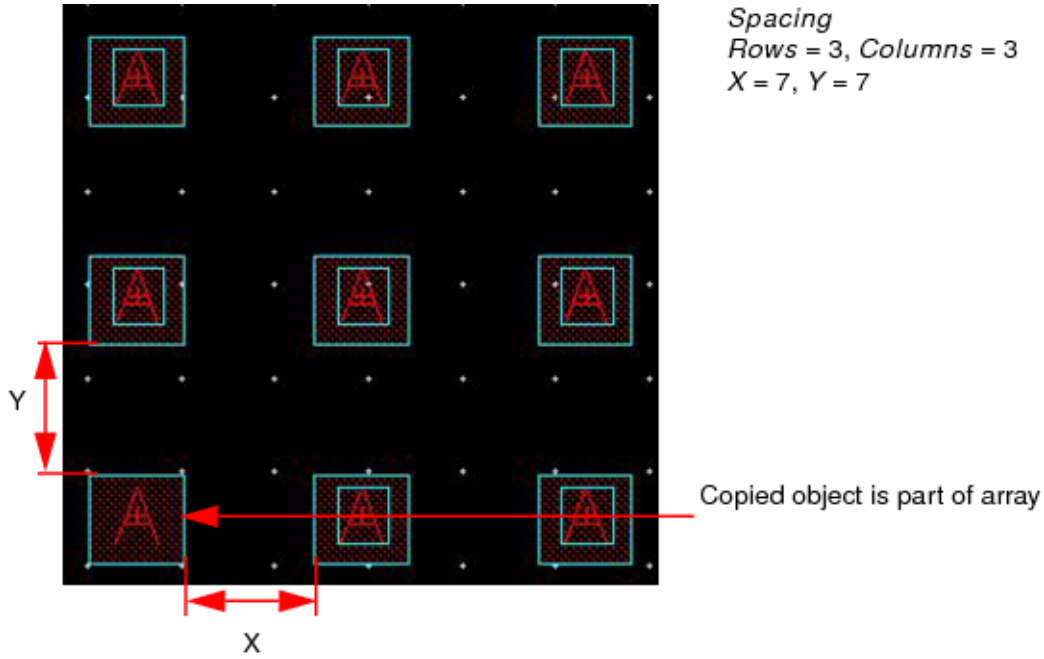
An array with the specified settings is created. The object you copied is part of the array.

© 2023 Cadence Design Systems, Inc.

The following figure shows an array of copied objects created using the *Pitch* option.



The following figure shows an array of copied objects created using the *Spacing* option.



You can use the Copy form to create group arrays. For information, see Creating a Group Array by Using the Copy Form.

Related Topics

Copy Form

© 2023 Cadence Design Systems, Inc.

Group Arrays

Copying an Object to Another Cellview

You can copy objects from one cellview to another interactively or by specifying spacing mode and values in post- or pre-selection modes.

- Copying an Object Interactively
- Copying an Object Using Spacing Values in Post-Selection Mode
- Copying an Object Using Spacing Values in Pre-Selection Mode

Copying an Object Interactively

To copy objects interactively to another open cellview:

1. In the source cellview, select the objects you want to copy.
2. Choose *Edit – Copy.*
3. Move the pointer into the other open target cellview.
4. Click on the canvas to place the copies.

You cannot use the *Array* fields in the Copy form to create and copy an array of copied objects from one cellview to another. Only the original objects you selected are copied to the next cellview even if you have specified values in the *Rows* and *Columns* fields.

Copying an Object Using Spacing Values in Post-Selection Mode

To copy objects to another open cellview in post-selection mode by specifying spacing values:

1. In the source cellview, choose *Edit – Copy* to invoke the *Copy* command.
2. Select the object on the canvas that you want to copy.
3. Move the pointer into the target cellview.
4. Press `F3` to open the Copy form.
5. Select one of the following modes in the *Spacing* section and specify the spacing values:
   - *Pitch*: Select the *Pitch* radio button and specify the spacing in the *Delta X,Y* fields.

© 2023 Cadence Design Systems, Inc.

- *Spacing*: Select the *Spacing* radio button and specify the spacing in the *Spacing X,Y* fields.
- *Absolute*: Select the *Absolute* radio button and specify the spacing in the *X,Y* fields.



6. Click *Apply* in the *Spacing* section.

   The object is placed at the specified location in the target cellview.

Copying an Object Using Spacing Values in Pre-Selection Mode

To copy objects to another open cellview in pre-selection mode by specifying spacing values:

1. In the source cellview, select the object on the canvas that you want to copy.
2. Choose *Edit – Copy* to invoke the *Copy* command.
3. Click on the canvas to specify the reference point for the copy operation.
4. Move the pointer into the target cellview.
5. Press `F3` to open the Copy form.
6. Specify the spacing mode and values in the *Spacing* section.
7. Click *Apply* in the *Spacing* section.

The object is placed at the specified location in the target cellview.

Related Topics

Copy Form

© 2023 Cadence Design Systems, Inc.

Cellviews Copied Across Libraries

You can use one of the following ways to copy a layout cellview from a source to a target library:

- Library Manager
- dbCopyCellView SKILL API
- *Copy* command to interactively copy portions of a source layout cellview to a target layout cellview

During the copy operation, the following components of the source and target technology libraries are compared:

- dbuPerUU and userUnits
- Layer names and layer numbers
- Purpose names and numbers
- Name, types, layer names and numbers of all viaDefs
- Name, types, width, and height of all siteDefs
- Names of array siteDefs

Related Topics

Copy Form

Object Copying

Copying an Object in Repeat Mode

The *Repeat Copy* command enables you to repeatedly place copies of the selected object. The command supports both pre-selection and post-selection modes.

To repeatedly place copies of an object:

1. In the layout window, select *Edit – Repeat Copy* or click the *Repeat Copy* icon  on the *Edit* toolbar.

© 2023 Cadence Design Systems, Inc.

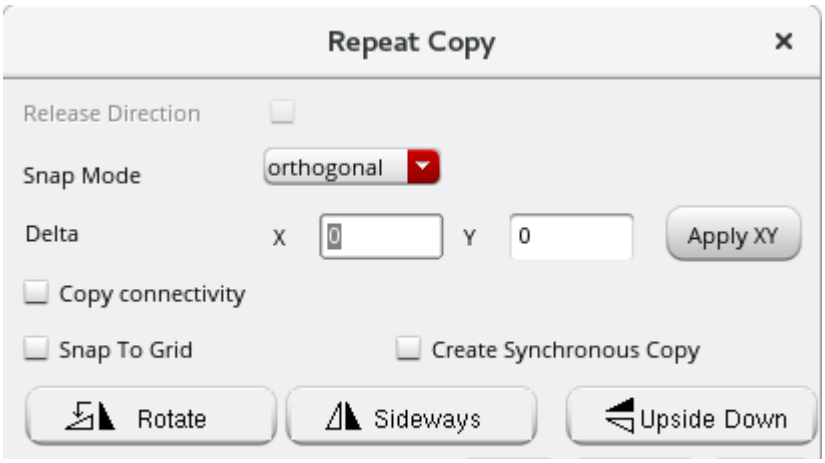
2. Click the object to copy.



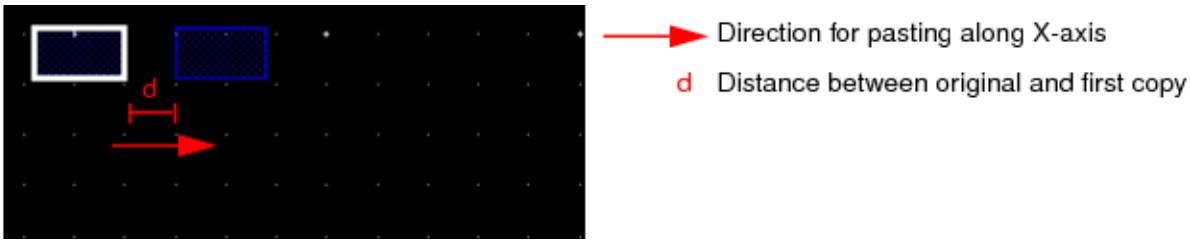
A copy of the object attaches to the pointer.

You cannot change the selection set after you have placed the first copy.

3. Press `F3` to open the Repeat Copy form.



4. From the *Snap Mode* list, select a direction in which to place the copy.
5. Click on the canvas to place the first copy.



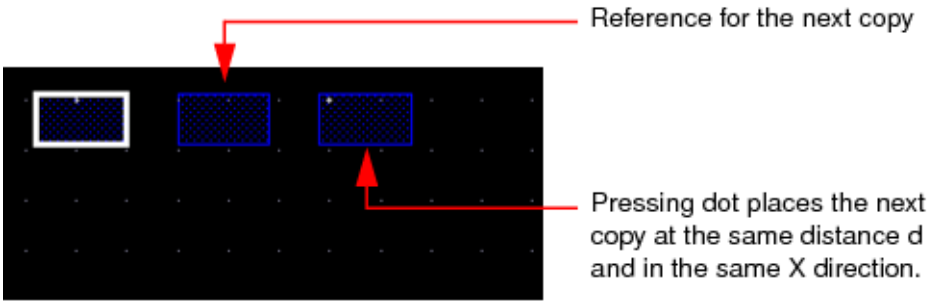
The distance `d` between the object and its first copy and the direction in which the copy is placed determine the placement distance and direction of the subsequent placed copies. After you place the first copy, the distance and direction are fixed.

© 2023 Cadence Design Systems, Inc.

Alternatively, you can specify the distance at which to place the copy in the *Delta X* and *Y* fields in the *Repeat Copy* form and click the *Apply XY* button. The copy is placed at the specified distance from the object.

6. Press dot (.) to repeat placing the copy on the canvas.

   The previous placed object is taken as the reference. A copy is placed at the same *d* distance from the previous copy and in the same direction in which the previous copy was placed.



   If you specified the *Delta* distance, it will be used to place the remaining copies.

   You can continue to press dot (.) to place the copy on the canvas as many times as required.

7. To start placing the copy from a different location, while retaining the distance:

   a. Press F3 to open the *Repeat Copy* form.

   b. Select the *Release Direction* check box.

      The option is grayed out after you select the check box.

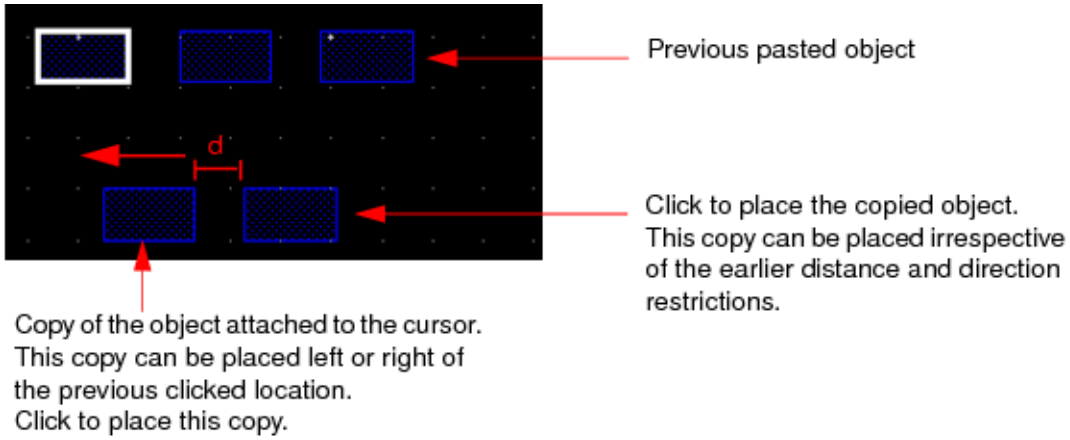
   c. Click *Hide*.

      You can move the pointer freely in any direction on the canvas. You can place the copy in any direction and at any distance from the previous placed copy.

   d. Click on the canvas to place the copy of the object.

© 2023 Cadence Design Systems, Inc.

The copy is placed at the clicked location and a copy of the object attaches to the pointer.



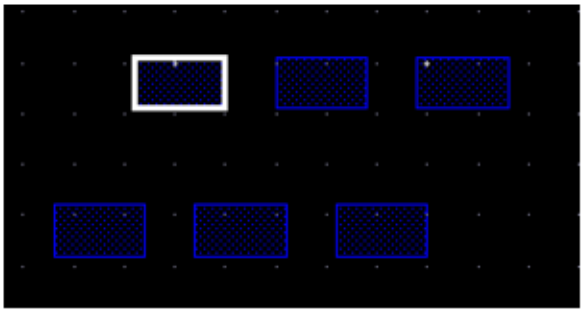
e. Move the pointer and click at the next location to place the copy.

To place this copy, the pointer movement is limited to only either horizontal or vertical direction, depending on the earlier direction in which the copy was placed. For example, if the earlier direction of placing the copy was along the X axis, you will be able to move the pointer only to the left or right of the placed copy. If the earlier direction of placing the copy was along the Y axis, you will be able to move the pointer only to the top or bottom of the placed copy.

The same distance `d` will be used to place the copy. *Delta* will be used if you specified it in the *Repeat Copy* form.

f. Press dot (.) to place the copy in the fixed direction at the fixed distance.

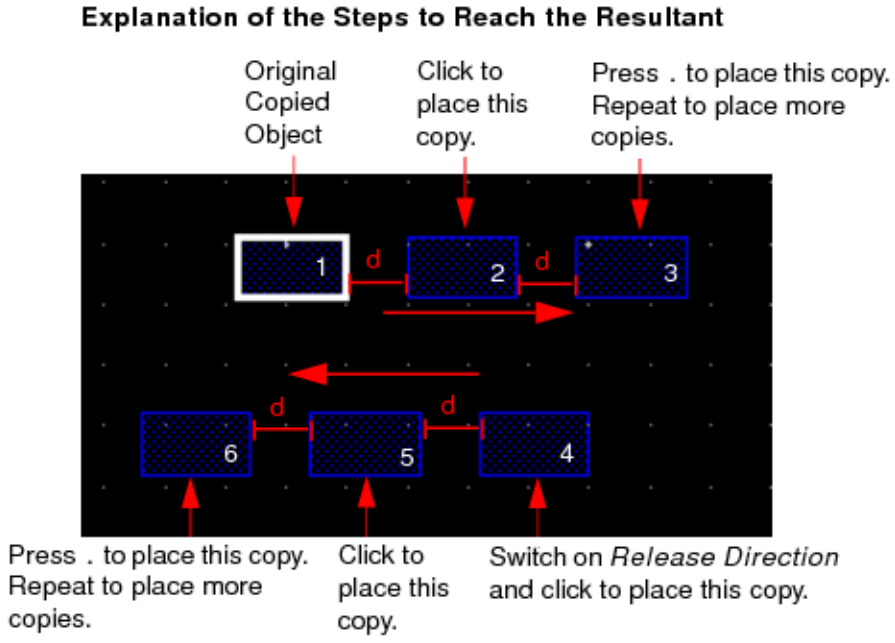
Final Result



8. Press `Esc` or click *Cancel* in the form to finish placing the copies.

© 2023 Cadence Design Systems, Inc.

The following image explains the steps involved in reaching the resultant copies.



The original copied object remains selected throughout.

You can use the `leRepeatCopyMoveStretch` SKILL function to copy, move, or stretch the selected objects and apply the same transform as done in the last copy, move, or stretch command and select the newly created object.

Related Topics

Object Copying

Repeat Copy Form

Object Stretching

You can use the *Stretch* command to stretch an edge or corner of an object. The following table describes the support available for stretching various objects.

Object Type	Support for the Stretch Operation
ROD objects	The *Stretch* command fully supports ROD objects.

© 2023 Cadence Design Systems, Inc.

Object Type	Support for the Stretch Operation
Blockages	The *Stretch* command stretches blockages, except for placement halos, in the same way it stretches a polygon by using points. You can stretch both fully and partially selected blockages. When you stretch fully selected blockages, the *Stretch* command works the same way as the *Move* command. When you stretch a fully selected owner, its blockages also move.
Boundaries	The *Stretch* command stretches boundaries in the same way as it stretches a polygon by using points. ■ You can stretch both fully and partially selected boundaries. ■ When you stretch fully selected boundaries, the *Stretch* command works the same way as the *Move* command. ■ The *Stretch* command supports IOBox and CoreBoxSpec during PR boundary stretch. Stretching retains the relative distance of IOBox and CoreBoxSpec from the PR boundary.
Mosaics	You can use the *Stretch* command to expand or shrink mosaics. The number of rows or columns increase or decrease according to the stretching direction and the target location. ■ Mosaics can be stretched in different directions depending on the edge selected and the current *Edit Snap* mode. ■ Multiple mosaics can be selected and stretched at the same time. ■ When the *Mosaic Partial Selection* option is selected on the Selection Options form, you can partially select and stretch mosaics. In this mode, you can select edges and corners. You can also select edges from multiple mosaics in partial selection mode.
Vias	You can use the *Stretch* command to expand or shrink vias in X or Y direction when the *Via Partial Selection* option is selected on the Selection Options form. The number of rows or columns increase or decrease according to the stretching direction and the target location.

(Virtuoso RF Option) When you stretch a curved path, with the gravity feature enabled, the end points of the curved path snap to the end points or centerline of another curved path or to the center of a pin or a via. When you stretch a pin or a via, its center is snapped to the end points of a curved path or to the center of another pin or via.

© 2023 Cadence Design Systems, Inc.

Related Topics

Stretching an Object by Direct Manipulation

Stretching a Circle

Changing the Angle of a Corner by Stretching

Stretching and Moving Objects

Stretching Paths

Stretching Multipart Rectangles

Selection Options Form

Stretching an Object by Direct Manipulation

You can stretch objects by selecting and dragging their edges or corners.

To stretch an object:

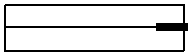
1. Choose *Edit – Stretch* or click the *Stretch* icon on the *Edit* toolbar.

   Alternatively, right-click any free space on the design canvas to display the *Layout* shortcut menu and select *Stretch.*

2. Select one or more edges or corners in partial selection mode as follows:



Press `F4` and click an edge or corner of a polygon.



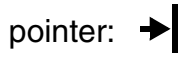
Press `F4` and click an endpoint on a path centerline.



Press `F4` and click a point on a circle or donut.

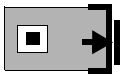
   In partial selection mode, if you press `Shift` and click an edge or corner, that edge or corner is the reference point for the stretch.

   If you create a selection box around the edges, *Stretch* prompts you for a reference point. The reference point does not have to be on the selected shapes.

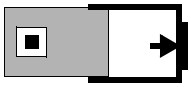
3. Move the mouse pointer over the edge until you see the stretch pointer: 

© 2023 Cadence Design Systems, Inc.

4. Click and drag the mouse pointer to stretch the object and then release the mouse button.



Click on a reference point.



Drag the pointer.



Release the mouse button. The object is stretched.

You cannot use *Stretch* to delete objects. If you try to stretch an object to a zero area, a warning appears in the CIW and a dialog box opens stating that the shape could not be modified because the operation is invalid.

Related Topics

Stretch Form

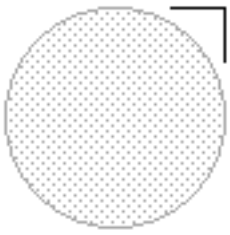
Stretching a Circle

Circles are special cases of ellipses, so if you stretch a circle it will not remain a circle, unless you use one of the correct methods. There are two ways to stretch a circle so that it will remain a circle.

- *Edit* option is set to *diagonal* in the *Snap Mode* section of the Display Options form.
- Selection mode is set to Partial.

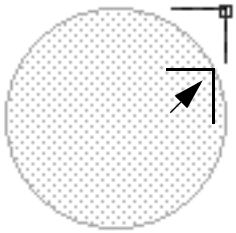
To stretch a circle:

1. Set *Edit* to *diagonal* in the *Snap Mode* section of the Display Options form.
2. Set the selection mode to Partial by using the `F4` key.
3. On the canvas, select one corner (the vertex) of the circle, as shown below:



© 2023 Cadence Design Systems, Inc.

4. Move the mouse pointer to the vertex edge until you see the stretch pointer, as shown below:



You can enlarge or shrink the circle by moving the stretch pointer, and the circle remains a circle if you stretch at a 45-degree angle, however, the origin point of the circle will not remain the same.

Caution

You must move the pointer carefully to ensure it remains at a 45-degree angle. Even though the snap mode is set to diagonal, it is still possible to move the pointer orthogonally. If you move the pointer orthogonally, the circle will become an ellipse.

Related Topics

Object Stretching

Stretching a Via

You can stretch partially selected vias in X or Y direction. The stretch functionality is not supported for a cdsFixedVia via.

To stretch a via:

1. In the layout window, choose *Edit – Stretch* or click the *Stretch* icon on the *Edit* toolbar.

   Alternatively, right-click any free space on the design canvas to display the *Layout* shortcut menu and select *Stretch.*

2. Press `F3` to open the Stretch form.
3. Select *Via Partial Selection*.

   Alternatively, you can select this option in the Selection Options form.

© 2023 Cadence Design Systems, Inc.

4. On the canvas, select the corner of the via that you want to stretch.
5. Move the pointer over the edge until you see the stretch pointer.
6. Click and drag the pointer to stretch the via. You can stretch the via in X or Y direction.
7. Release the mouse button.

Related Topics

Stretch Form

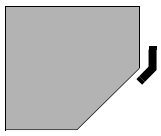
Selection Options Form

Changing the Angle of a Corner by Stretching

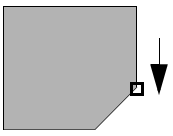
To change the angle of a corner by stretching:

1. In the layout window, choose *Edit – Stretch* or click the *Stretch* icon on the *Edit* toolbar.

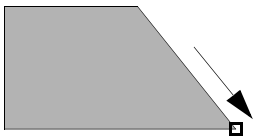
   Alternatively, right-click at free space on the design canvas to display the *Layout* shortcut menu and select *Stretch.*
2. Press `F3` to open the Stretch form.
3. Deselect the *Lock Angles* check box.
4. On the canvas, select the corner that you want to stretch, drag the pointer to the desired location and then click on the canvas to complete the command.



The selected corner



With *Lock Angles* on, you cannot change the angle of the corner.



With *Lock Angles* off, you can stretch the corner in any direction.

If you create a non-orthogonal segment while stretching the corner of a multipart path, subrectangles cannot regenerate in that particular segment. Subpaths can and will regenerate.

© 2023 Cadence Design Systems, Inc.

Related Topics

Stretch Form

Multipart Paths

Stretching and Moving Objects

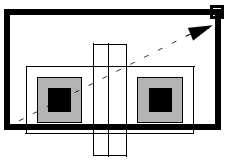
You can simultaneously stretch and move objects. This is often called a window stretch.

To stretch and move objects:

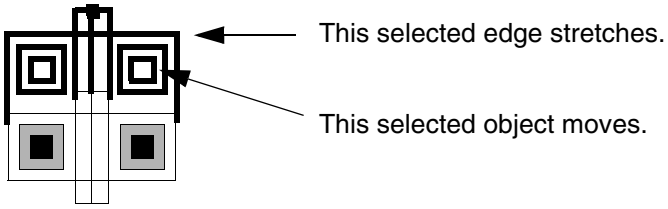
1. In the layout window, choose *Edit – Stretch* or click the *Stretch* icon on the *Edit* toolbar.

   Alternatively, right-click any free space on the design canvas to display the *Layout* shortcut menu and select *Stretch.*

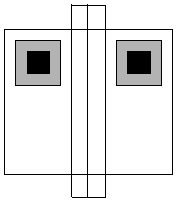
2. Press `Shift` and create a selection box around the objects that you want to move and the edges you want to stretch.



3. Select a reference point for the stretch.
4. Move the pointer to stretch the edges where you want.



5. Click on the canvas.



The stretched and moved objects

© 2023 Cadence Design Systems, Inc.

Related Topics

Stretch Form

Stretching Paths

You can stretch the ends, segments, and/or corners of a path. You stretch ends and corners in the same way as you stretch any other object. To stretch segments, you need to select the path centerline in the segments you want to stretch.

The resultant of stretching a path should not be malformed or self-intersecting, with overlapping edges. A resultant path with coincident edges (abutted edges) can be manufactured and is allowed.

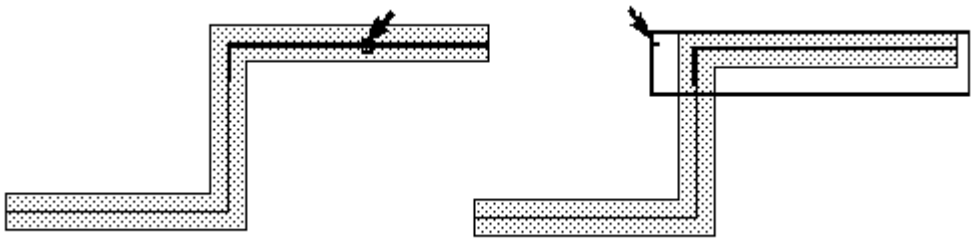
To stretch path segments:

1. In the layout window, choose *Edit – Stretch*.

   Stretch automatically changes to partial selection mode and prompts you to select the object you want to stretch.

2. Select one or more segments by doing one of the following:

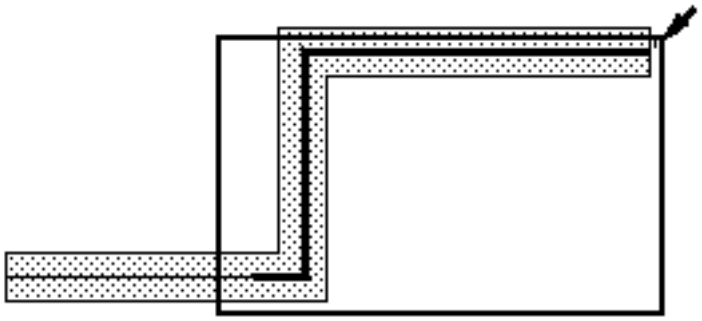
   - For a single segment, click the middle of the centerline of the segment or create a selection box around the centerline of the segment, making sure to include the ends of the segment centerline.



   Highlighting shows the path segment is selected.

© 2023 Cadence Design Systems, Inc.

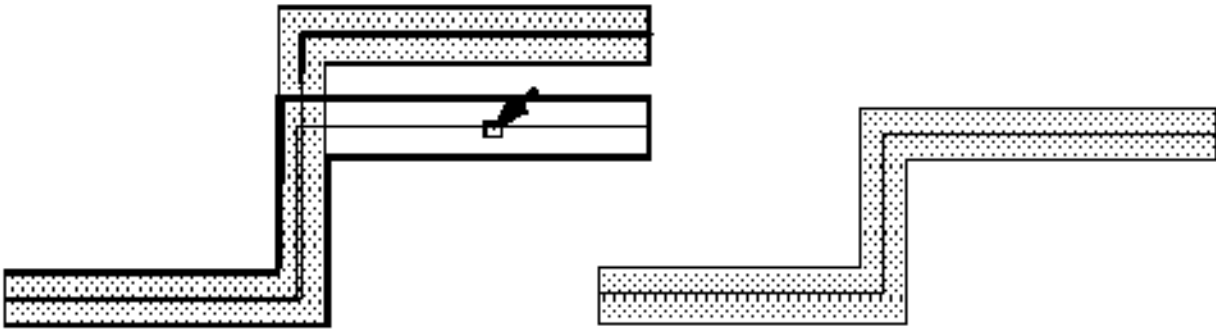
- For multiple segments, to add another segment, press `Shift` and click or create a selection box around the centerline of the segments, making sure to include the ends of the segment centerlines.



Highlighting shows the path segments are selected.

When you use a selection box, *Stretch* prompts you for a reference point. The reference point does not have to be on the selected shapes.

3. If prompted for a reference point, click on the canvas where you want the stretch to begin.
4. Point to the new location for the stretch and click on the canvas.



5. Press `Esc` to end the *Stretch* command.

Related Topics

Object Stretching

Stretching Multipart Rectangles

You can stretch the edges and/or corners of a master rectangle of a multipart rectangle in the same way as you stretch regular rectangles: click an edge or a corner and then click at a new location. The system regenerates the arrays of subrectangles associated with the stretched

© 2023 Cadence Design Systems, Inc.

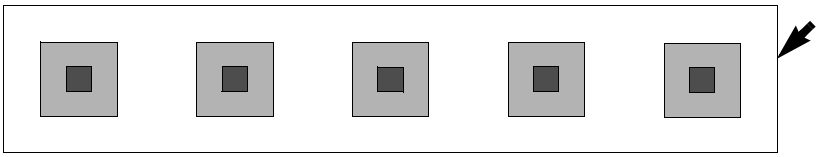
master rectangle. How the subrectangles regenerate depends on the definition of the multipart rectangle. You cannot stretch a master rectangle separately from its associated arrays of subrectangles, nor can you stretch an array of subrectangles separately from its master rectangle.

To stretch an edge of a multipart rectangle:

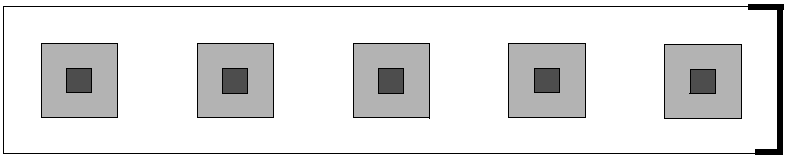
1. In the layout window, choose *Edit – Stretch*.

   Stretch automatically changes to partial selection mode and prompts you to select the object you want to stretch.

2. Select the edge you want to stretch by clicking it:

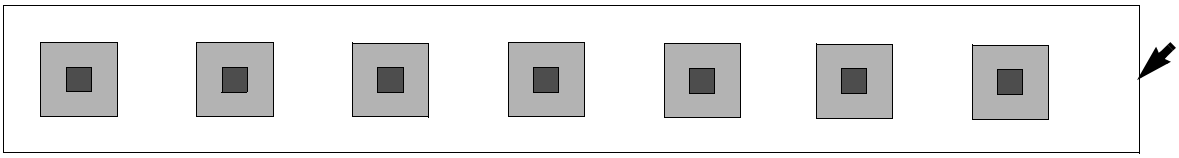


   The selected edge is highlighted, as shown below.



   Stretch prompts you to point to the new location. In this example, the master rectangle is stretched to the right.

3. Point to a new location for the stretch and click on the canvas.



Related Topics

Multipart Rectangles

rodCreateRect

© 2023 Cadence Design Systems, Inc.

Object Deletion

You can use the *Delete* command to delete objects. The following table describes the support available for deleting objects.

Object	Support for the Delete Operation
ROD objects	The *Delete* command fully supports ROD objects. ■ When you delete using the *Net Interconnect* option, all ROD objects on the selected net are deleted, except for pins. For multipart paths (MPP) and multipart rectangles (MPR), when the master path or master rectangle is on the selected net, the MPP or MPR and all subparts are deleted; otherwise, no part of the MPP or MPR is deleted. ■ When you delete a segment of an MPP with the *Path Segment* option, the remaining segments of the MPP become two separate multipart paths, each with its own name.
Boundaries	■ You can delete pre- or post- selected boundaries. ■ You cannot delete partially selected edges of boundary objects.
Blockages	■ You can delete pre- or post- selected blockages. ■ You cannot delete partially selected edges of the blockage object. ■ When you delete the owner of a blockage, the blockage object is also deleted.

Related Topics

Deleting Objects

Deleting Objects

To delete objects:

1. On the layout window canvas, choose the objects, edges, or corners you want to delete.
2. Choose *Edit – Delete* or press `Delete` or click the *Delete* icon  on the *Edit* toolbar. Alternatively, right-click any free space on the design canvas to display the *Layout* shortcut menu and select *Delete.*

© 2023 Cadence Design Systems, Inc.

All selected objects are deleted. If you selected an edge or corner, the edge or corner is deleted and the object is redrawn.

You cannot delete a point if it changes the shape in a way that the shape becomes invalid. For example, a path must have at least two points; if you try to delete one point of a two-point path, a warning dialog box is displayed. You must close the dialog box and choose a different point to delete.

3. Press `F3` to open the Delete form if you want to specify the type of objects to be deleted.
4. Select the type of objects that you want to delete. You can choose to delete net interconnect and path segments and specify the device chains.

   If you select the *Net Interconnect* option, all ROD objects and shapes on the selected net, except pins, are deleted.
5. Select the objects on the design canvas that need to be deleted.
6. Click *Cancel* in the Delete form.

Canceling Delete

If you discover that you deleted the wrong object, choose *Edit – Undo* to restore the object.

Delete automatically repeats if you select the *Delete* command first and then choose objects. If delete automatically repeats, press `Esc` to cancel the command.

You can cancel an ongoing delete operation by pressing `Ctrl` + `C`.

Related Topics

Delete Form

Object Flipping

You can flip an object horizontally or vertically. When flipping multiple objects as a group, you can choose to flip them around their own respective centers or around the center of their combined boundary box. The *Flip around combined bbox* option in the Layout Editor Options form and in the Flip, Flip Vertical, and Flip Horizontal forms helps you specify whether the multiple selected objects need to be flipped around their combined bounding box. If the *Flip around combined bbox* option is disabled, by default, the selected objects are flipped around their own centers.

© 2023 Cadence Design Systems, Inc.

You can use the following Flip commands to flip objects in pre-selection and post-selection modes. You can access the commands from the *Edit – Flip* menu option. You can access the *Flip Vertical* and *Flip Horizontal* commands from the *Rotate* option on the shortcut menu displayed when you right-click on the canvas.

- *Flip*: Flips an object or multiple selected objects as a group as per the specified reference point and direction. You can specify a custom or pre-defined reference point and choose whether multiple objects need to be flipped around their combined bounding box.
- *Flip Vertical*: Flips an object vertically. In pre-selection mode, you can use the Flip Vertical form to specify the reference point and whether objects need to be flipped around their combined bounding box in case multiple objects are selected.
- *Flip Horizontal*: Flips an object horizontally. In pre-selection mode, you can use the Flip Horizontal form to specify the reference point and whether objects need to be flipped around their combined bounding box in case multiple objects are selected.

You must set the stop display level to greater than zero before flipping geometries in the via instances. For *Copy* and *Move* commands, the *Flip* icon on the Edit toolbar is disabled.

While editing an instance or a mosaic, you can specify whether the Flip command uses the instance bounding box, place and route (PR) boundary, or snap boundary (if defined) as the instance border for the flip operation. For example, if you have a standard cell with a large label that extends well beyond the PR boundary, the bounding box of that cell will be much larger than its PR boundary. In the Layout Editor Options form, you can set *Instance Editing Mode* to *P&R Boundary* to make the *Flip* command ignore the bounding box and consider only the PR boundary for the flip operation.

Related Topics

Flipping an Object by Specifying a Reference Point on Canvas

Flipping Objects

Flipping Objects

You can choose the point of reference and direction for flipping objects. You can also choose to flip multiple selected objects around a specific reference point of the combined bounding box of objects.

To flip objects by using the Flip form:

1. Open your design in the layout window and choose *Edit – Flip – Flip*.

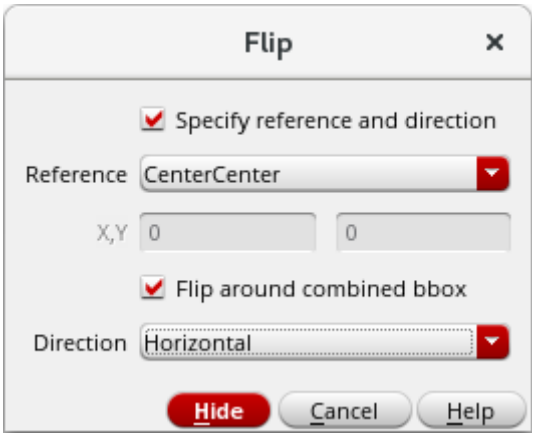
© 2023 Cadence Design Systems, Inc.

2. Press `F3`.

   The Flip form is displayed.

3. Select the *Specify reference and direction* check box.

   The options in the form are enabled.



4. From the *Reference* list, select the point of reference around which you want to flip the objects. The default is *CenterCenter*. You can choose *Custom* to specify the X and Y coordinates around which you want to flip the objects.

5. Ensure the *Flip around combined bbox* check box is selected if you want to flip multiple selected objects around the specified reference point of the combined bounding box of the objects.

6. Set the flip direction to horizontal or vertical in the *Direction* list.

7. Select the objects on the canvas that you want to flip.

   The objects are flipped according to the settings you specified in the form.

Related Topics

Flip Form

Object Flipping

© 2023 Cadence Design Systems, Inc.

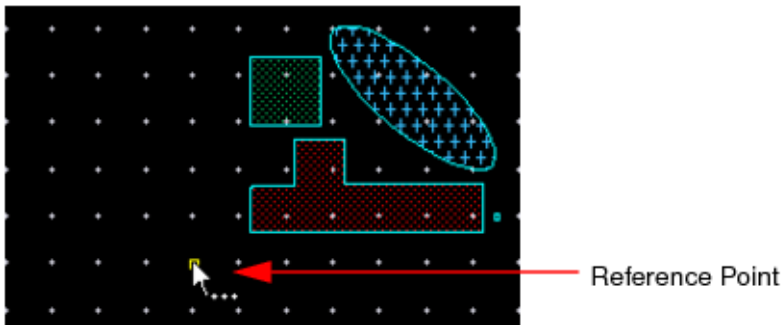
Flipping an Object by Specifying a Reference Point on Canvas

To flip the selected objects by specifying a reference point on the canvas:

1. In the layout window, select the objects on the canvas that you want to flip.
2. Choose *Edit – Flip – Flip*.
3. Press `F3` to open the Flip form.
4. Ensure the *Specify reference and direction* check box is not selected in the form.

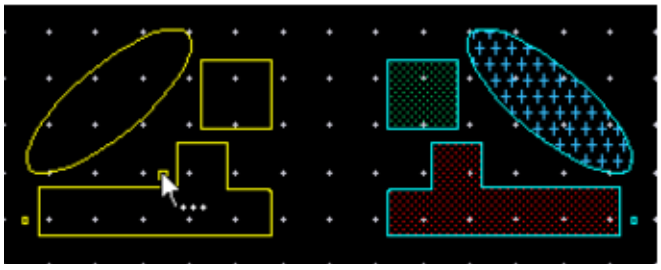
   In this mode, all the options on the form are disabled.
5. Click a reference point on the canvas for flipping the objects.

   The coordinates of the clicked reference point populate in the *X, Y* reference point fields of the Flip form. The ghost image of the resultant flip operation displays. The location of the ghost image moves as you move the pointer on the canvas. The following figure below shows the selected objects and the reference point.



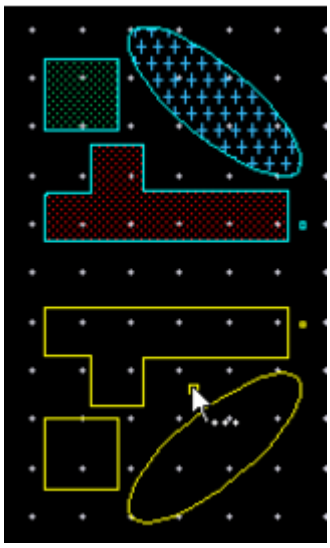
Assuming that the selected objects are in quadrant 2 on the canvas:

- If the pointer is in quadrant 1 relative to the selected objects, the ghost image is also displayed in quadrant 1.

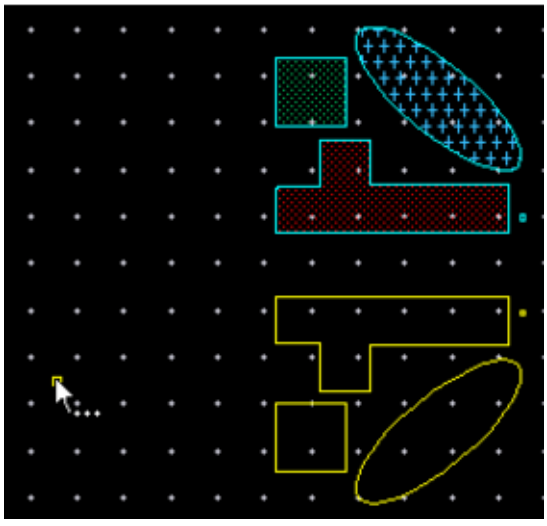


© 2023 Cadence Design Systems, Inc.

- If the pointer is in quadrant 3 relative to the selected objects, the ghost image is also displayed in quadrant 3.



- If the pointer is in quadrant 2, no ghost image is displayed.
- If the pointer is in quadrant 4, nothing changes. If a ghost image is displayed in quadrant 1 or 4, it continues to be displayed.



6. Click on the canvas to select the required ghost image.

   The selected objects are flipped and the command ends.

© 2023 Cadence Design Systems, Inc.

Related Topics

Flip Form

Object Flipping

Object Rotation

Using the Rotate functionality, you can change the orientation of most geometric objects. You can rotate an object at a specific angle and a specific reference point on the canvas.

The following table describes the support available for rotating various objects.

Object Type	Support for Rotation
Donuts and dots	All objects can be rotated except donuts and dots. To rotate donuts and dots, use the *Move*, *Copy*, or *Paste* commands.
Instances and labels	Instances and labels can be rotated only in 90-degree increments.
ROD objects	ROD objects, except multipart rectangles, can be rotated. Avoid rotating aligned ROD objects because aligned handle names are not updated after the rotation, so the result might not be what you want. For multipart rectangles with subrectangles offset from the master rectangle corners, avoid rotation unless the offsets are all equal. The Rotate functionality does not rotate these offsets. Do not rotate a chopped multipart path (MPP) as the chop holes might not rotate correctly. If you want to chop and rotate an MPP, rotate the MPP first, and then chop it.
Blockages	Blockage objects can be rotated. Any angle rotations are allowed when blockage objects are part of a selection set. When you rotate blockage objects without owners, the coordinates are updated. When you rotate owners of blockage objects, the blockage coordinates are also updated.
Boundaries	Boundary objects can be rotated. Any angle rotations are allowed when boundary objects are part of a selection set. The Rotate functionality supports the rotation of IOBox and CoreBoxSpec when the PR boundary is rotated.

© 2023 Cadence Design Systems, Inc.

When rotating multiple objects as a group, you can choose to rotate them around their own respective centers or around the center of their combined boundary box. Objects that are on-grid remain on-grid after rotation. The *Rotate around combined bbox* option in the Layout Editor Options form and in the Rotate, Rotate Left, and Rotate Right forms helps you specify whether multiple selected objects need to be rotated around their combined bounding box.

You can use the following *Rotate* commands to rotate objects in pre-selection and post-selection modes. You can access the commands from the *Edit – Rotate* menu option or the *Rotate* option on the shortcut menu displayed when you right-click on the canvas.

- *Rotate*: Rotates an object or multiple selected objects as a group at the specified reference point and angle. You can use the Rotate form to specify the reference point of rotation and whether multiple objects need to be rotated around their combined bounding box. You can also specify the angle, snap mode, and precision of rotation.
- *Rotate Left*: Rotates an object by 90 degrees in the counterclockwise direction. In pre-selection mode, you can use the Rotate Left form to specify the reference point for rotation and whether objects need to be rotated around their combined bounding box in case multiple objects are selected.
- *Rotate Right*: Rotates an object by 90 degrees in the clockwise direction. In pre-selection mode, you can use the Rotate Right form to specify the reference point for rotation and whether objects need to be rotated around their combined bounding box in case multiple objects are selected.

While editing an instance or a mosaic, you can specify whether the Rotate command uses the bounding box, place and route (PR) boundary, or snap boundary (if defined) as the instance border for rotation. For example, if you have a standard cell with a large label that extends well beyond the PR boundary, the bounding box of that cell will be much larger than its PR boundary. In the Layout Editor Options form, you can set *Instance Editing Mode* to *P&R Boundary* to have the *Rotate* command ignore the bounding box and consider only the PR boundary for the rotate operation.

To rotate any object in 90-degree increments or to mirror an object, you can also use the *Move*, *Copy*, or *Paste* commands. For *Copy* and *Move* commands, the *Rotate* icon on the Edit toolbar is disabled.

Related Topics

Rotate Form

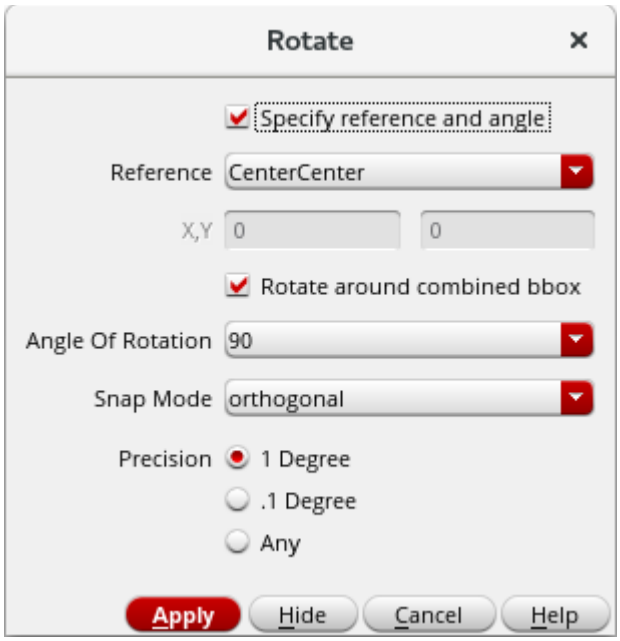
© 2023 Cadence Design Systems, Inc.

Rotating Objects by Using the Rotate Form

To rotate objects by using the Rotate form:

1. In your layout design window, choose *Edit – Rotate – Rotate*.
2. Press `F3` to open the Rotate form.
3. Select the *Specify reference and angle* check box.

   The reference options in the form appear enabled.



4. From the *Reference* list, select the point of reference around which you want to rotate the objects. The default is *CenterCenter*.

   You can choose *Custom* to specify the X and Y coordinates around which you want to rotate the objects.
5. Ensure the *Rotate around combined bbox* check box is selected if you want to rotate multiple selected objects around the specified reference point of the combined bounding box of the objects.
6. In *Angle Of Rotation*, select or enter the angle at which you want to rotate the objects.
7. Select a snap mode from *Snap Mode* for rotating the objects.

   Valid values are *anyAngle*, *diagonal*, *orthogonal*, *horizontal*, or *vertical*.
8. Select the precision of the snap angle from *Precision*.

© 2023 Cadence Design Systems, Inc.

9. Select the objects that you want to rotate on the canvas.

   The objects are rotated according to the settings specified in the form.

Related Topics

Rotate Form

Object Rotation

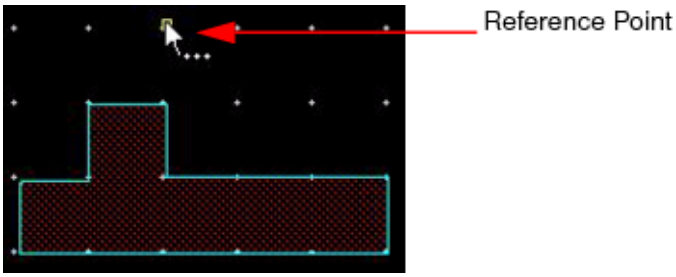
Rotating a Geometric Object by Any Angle

To rotate an object by specifying the rotation reference point on the canvas:

1. Choose *Edit – Rotate – Rotate* or click the *Rotate* icon  on the *Edit* toolbar to invoke the *Rotate* command.

   Alternatively, right-click at any empty space on the canvas and select *Rotate – Rotate* from the shortcut menu.

2. Select the object on the canvas that you want to rotate.
3. Click at a reference point for rotation on the canvas.

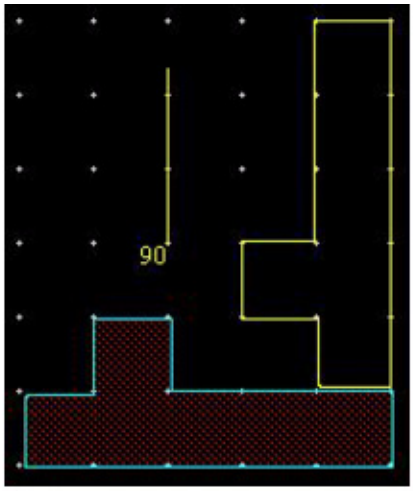


4. Click at a new point on the canvas to indicate the rotation angle as 90, 180, or 270 degrees.

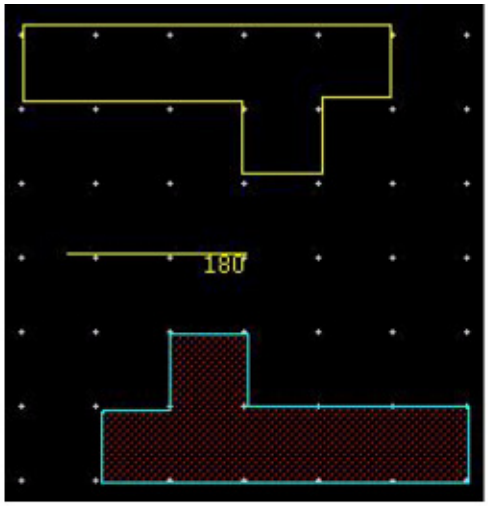
   Before you select the new point, a preview displays the rotation result and the angle of rotation as you move your mouse pointer. When you click on the canvas, the object is rotated and the command is exited.

© 2023 Cadence Design Systems, Inc.

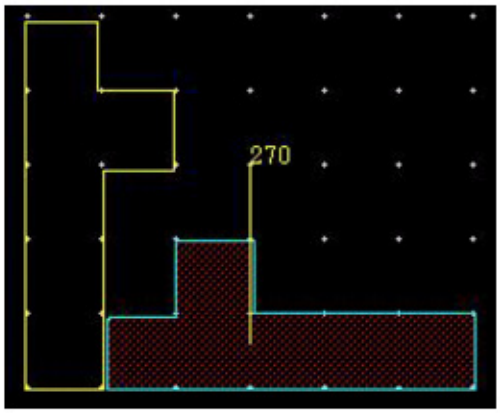
The following figures illustrate how a selected object can be rotated at 90, 180, and 270 degrees.



Rotate 90



Rotate 180



Rotate 270

Related Topics

Rotate Form

Object Rotation

© 2023 Cadence Design Systems, Inc.

Rotating Multiple Objects

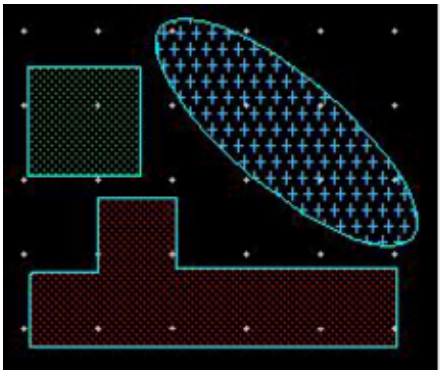
You can rotate multiple selected objects around their respective center points or around the center of their combined bounding box.

- Rotating Objects Around Their Own Center Points
- Rotating Objects Around the Combined Bounding Box
- Rotating Objects as a Group at a Specific Reference Point and Angle

Rotating Objects Around Their Own Center Points

To rotate multiple selected objects around their respective center points by using the *Rotate Left* or *Rotate Right* command:

1. In the Layout Editor Options form, deselect the *Rotate Around Combined BBox* check box and click *OK*.
2. Select the objects on the canvas that you want to rotate.

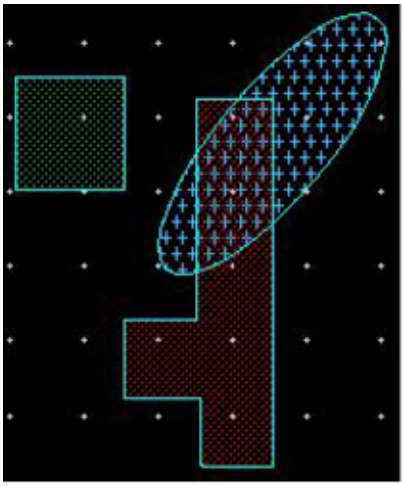


Multiple Selected Objects

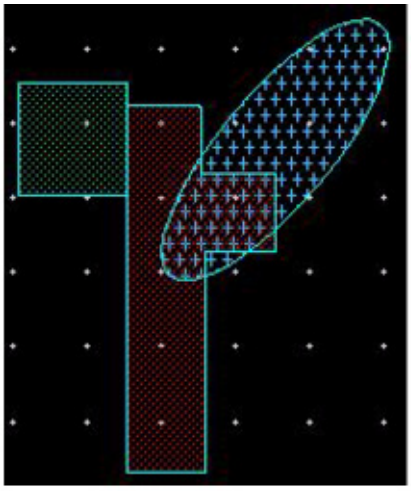
3. Select *Rotate Left* or *Rotate Right*, as required. You can access these commands from the *Edit – Rotate* menu or the *Rotate* shortcut menu on the canvas.

© 2023 Cadence Design Systems, Inc.

The following figures show the result when the *Rotate Left* or *Rotate Right* command is run on the selected set of objects shown in the previous figure.



Rotate Left

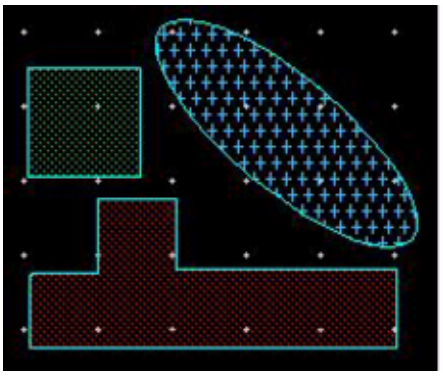


Rotate Right

Rotating Objects Around the Combined Bounding Box

To rotate multiple selected objects around the center of the combined bounding box of the selection set by using the *Rotate Left* or *Rotate Right* command:

1. In the Layout Editor Options form, ensure the *Rotate Around Combined BBox* check box is selected.
2. Select the objects on the canvas you want to rotate.

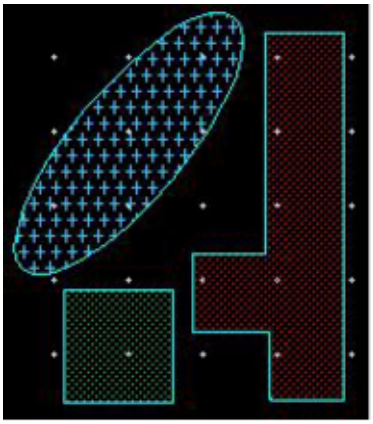


Multiple Selected Objects

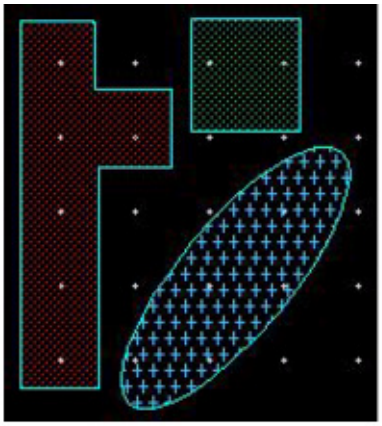
3. Select *Rotate Left* or *Rotate Right*, as required. You can access these commands from the *Edit – Rotate* menu or the *Rotate* shortcut menu on the canvas.

© 2023 Cadence Design Systems, Inc.

The following figures show the result when the *Rotate Left* or *Rotate Right* command is run on the selected set of objects.



Rotate Left

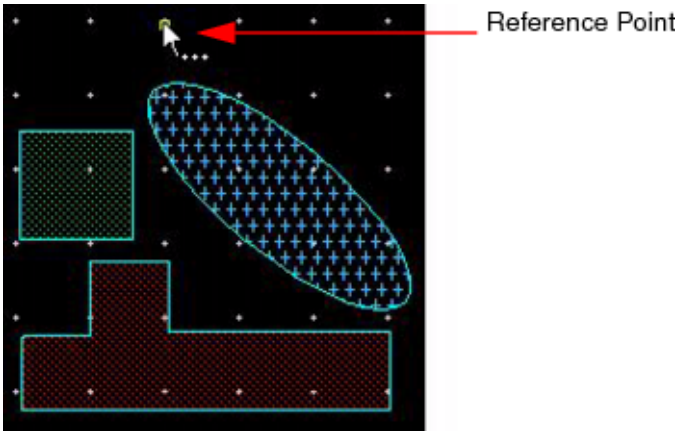


Rotate Right

Rotating Objects as a Group at a Specific Reference Point and Angle

To rotate multiple selected objects as a group and by specifying a rotation reference point:

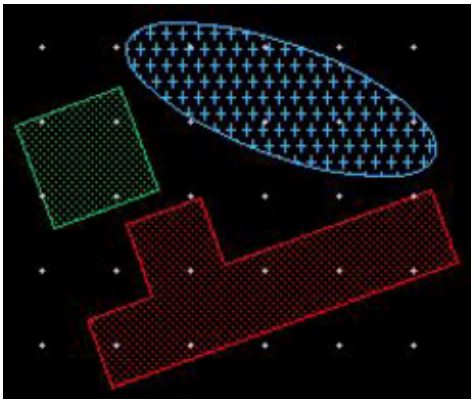
1. Choose *Edit – Rotate – Rotate* to invoke the *Rotate* command.
2. Select the objects on the canvas that you want to rotate.
3. Click on the canvas to specify the reference point.



4. Press `F3` to open the Rotate form.
5. Enter a value in the *Angle Of Rotation* field.
6. Click *Apply*.

© 2023 Cadence Design Systems, Inc.

The objects are rotated at the specified angle. The following figure illustrates the rotation at an angle of 20 degrees.



Related Topics

Rotate Form

Object Rotation

Rotating or Flipping an Object During a Move Operation

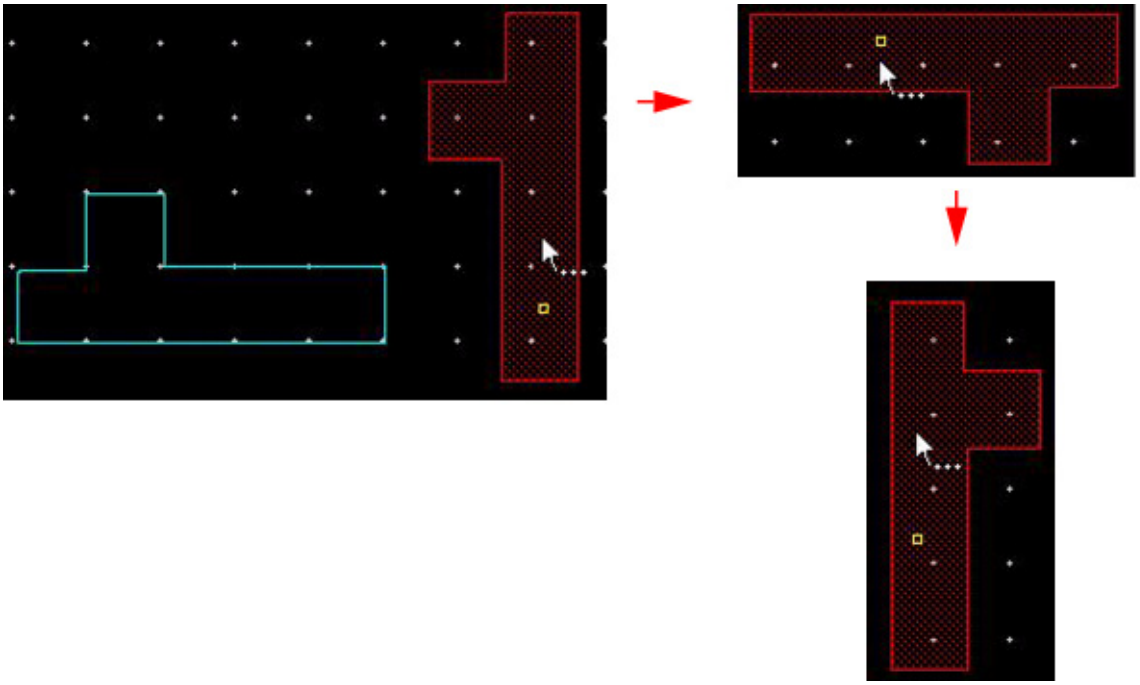
To rotate any object in 90-degree increments or to flip an object, you can use the *Move*, *Copy*, or *Paste* command. To rotate most geometric objects to any angle, use the *Rotate* command.

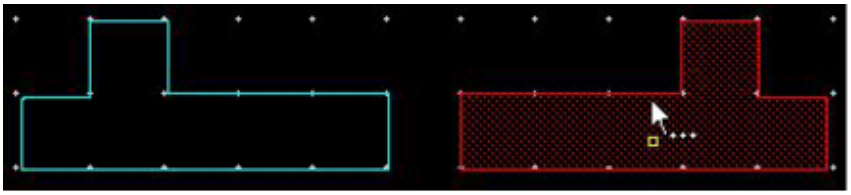
To rotate or flip an object while using the *Move* command:

1. Choose *Edit – Move*.
2. Select the objects on the canvas.
3. Press `F3` to open the Move form.
4. Click on the canvas to specify the reference point for the move operation.
5. Click the *Rotate Left*, *Rotate Right*, *Flip Vertical*, or *Flip Horizontal* button in the form, as required, to rotate or flip the objects during the move operation. Continue clicking the buttons to further rotate or flip the objects.

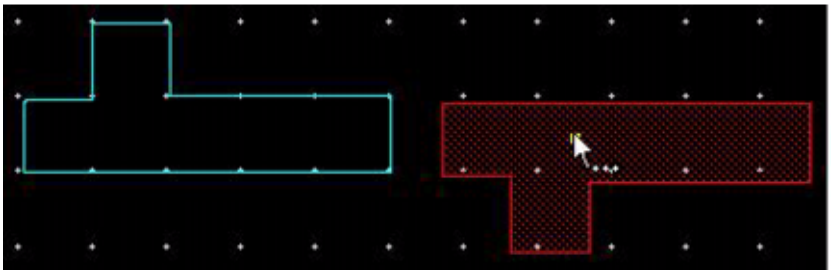
© 2023 Cadence Design Systems, Inc.

Each time you click a *Rotate* button, the object rotates 90 degrees clockwise or counterclockwise.





Rotate 180 mirrors the object along the X axis.



Rotate 270 mirrors the object along the Y axis.

6. Click on the canvas where you want to place the rotated object.
7. Alternatively, use the middle mouse button to rotate or flip the objects.
   - To rotate the objects 90 degrees, click the middle mouse button.
   - To flip the objects, press `Shift` and click the middle mouse button.

© 2023 Cadence Design Systems, Inc.

The objects mirror first along the X axis, then along the Y axis.

To rotate multiple selected objects as a group, use the options on the Rotate form or select the *Rotate Around Combined BBox* check box in the *Layout Editor Options* form.

Rotating Objects: Virtuoso Photonics Option

You need the `Virtuoso_Photonics_Option` license to use the functionality discussed in this section. For information on obtaining the required license, contact your local Cadence representative.

You can rotate an instance by any angle if it has a parameter that can be used for rotation. The value of the `cdsRotateInstParam` parameter on the instance or master is considered as the rotation parameter. If `cdsRotateInstParam` is not specified, the value specified in the `rotateInstParam` environment variable is considered for rotation.

For orthogonal rotation of an instance, the *Rotation* field on the *Attribute* tab of the Edit Instance Properties form is updated. For non-orthogonal rotation of an instance, the *rotation* parameter on the *Parameter* tab of the Edit Instance Properties form is updated. The rotation parameter is defined by the rotateInstParam environment variable.

Related Topics

Rotate Form

Object Rotation

Object Chopping

You can use the *Chop* command to cut away part of an object or cut an object into pieces. If a non-ROD object has a net name and it is cut into pieces, each remaining piece of the object retains the net name. Partial chopping of instances and vias is not supported.

© 2023 Cadence Design Systems, Inc.

The following table describes the support available for chopping various objects.

Object Type	Support for Chop Functionality
ROD objects	Most shapes lose ROD attributes when they are chopped. For multipart paths, the effect of a chop depends on what parts are choppable and how you chop them. For details, see How Chopping Affects Multipart Paths. When you chop the master rectangle of a multipart rectangle, all subrectangles are deleted and the shapes resulting from chopping the master rectangle are regular, unnamed database shapes. When you chop any ROD object, all alignments to that object are deleted.
Blockages	Chopping a blockage object may result in more than one blockage object. If the original blockage object had an owner, then all the resulting blockage objects will have the same owner. One owner will then have multiple associated blockages.

© 2023 Cadence Design Systems, Inc.

Object Type	Support for Chop Functionality
Boundaries	The *Chop* command does not support snap boundary objects. ■ For place and route (PR), area, and cluster boundaries, the command allows rectilinear chops only, therefore only horizontal, vertical, and orthogonal snap modes are allowed. Any other snap mode overrides to orthogonal. ■ For PR boundary, after a chop, the resultant PR boundary snaps to the grid. It is snapped to the grid specified by the boundarySnapXGrids or boundarySnapYGrids environment variables. If the environment variables are not specified, the resultant PR boundary is snapped based on the least common multiple of the global active grids. ■ After a chop, the resultant PR boundary must remain a single object. If you chop the PR boundary into more than one piece, the chop does not take place and the CIW issues the following warning: `*WARNING* Can't chop the PR Boundary into two or more pieces.` ■ The *Chop* command supports editing of IOBox and CoreBoxSpec when PR boundary is chopped. If chopping results in a non-rectangular PR boundary, the rectangle containing all PR boundary points is considered. The relative distance of IOBox and CoreBoxSpec from the PR boundary is maintained in all directions, even until the IOBox and CoreBoxSpec shapes are reduced to zero area. ■ After chopping a cluster boundary, each cluster boundary formed by the chop will remain connected to the original cluster. If you chop a cluster boundary object, each new cluster boundary has an extension added to its name, with a resulting informational message appearing in the CIW.
Pins	The *Chop* command support pins. A single pin is chopped into multiple pins. The new pins is strongly connected, this means that all the chopped pins have new pinFigs but the terminal and pin names are the same. The supported pin shapes are rectangle, polygon, and ellipse. All the properties of the original pin are retained after the chop operation.

© 2023 Cadence Design Systems, Inc.

Object Type	Support for Chop Functionality
Row regions	The *Chop* command supports grid-aware snapping for row regions if grids are specified in the row template manager. Grid-aware snapping is supported for the line, rectangle, and polygon chop shapes. For the line chop shape, snapping is supported for no spacing and the *User Spacing* type; it is not supported for *Min Spacing*.
Arrays	The *Chop* command supports arrays (mosaic or group array) if the vertices of the rectangle and polygon, or end points of a line, are either outside the array or on the boundary of the array.

To cut hierarchical objects, such as parameterized cells, you need to use the *Yank* command.

(Virtuoso RF Option) The *Chop* command is supported for curved shapes in all modes (Rectangle, Polygon, Circle, and Line mode).

Related Topics

Copying and Cutting Through the Hierarchy

Modifying Corners of a Polygon

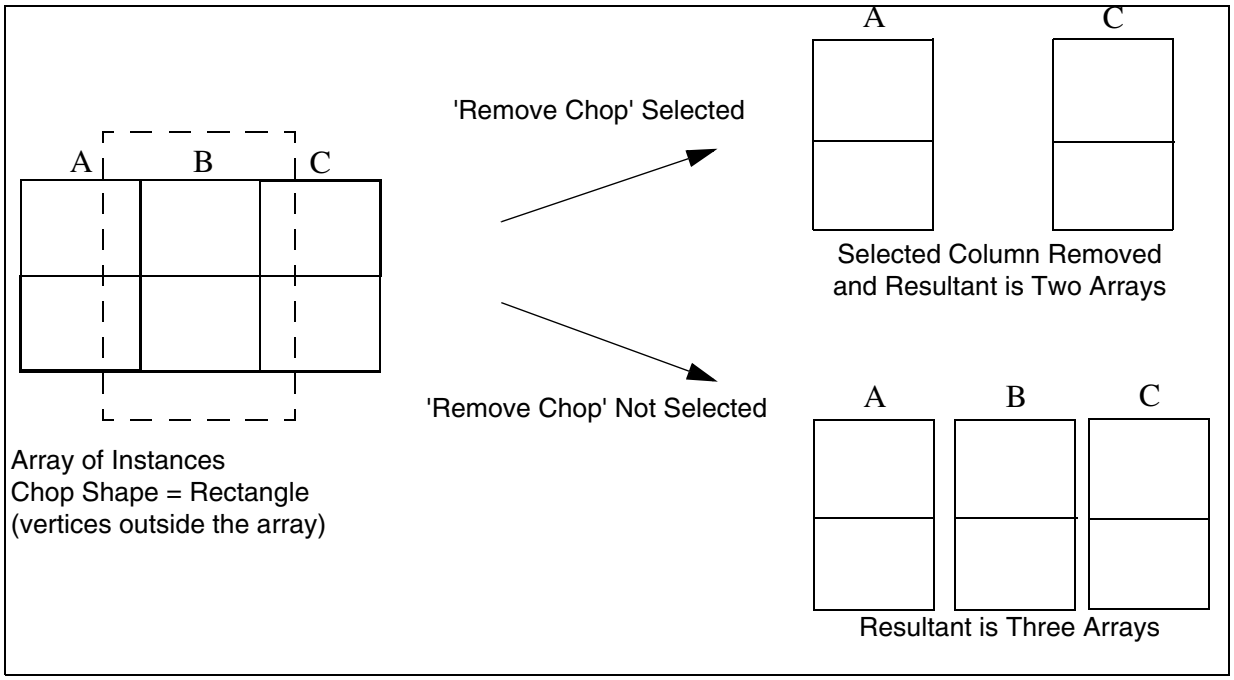
Array Chopping

Array Chopping

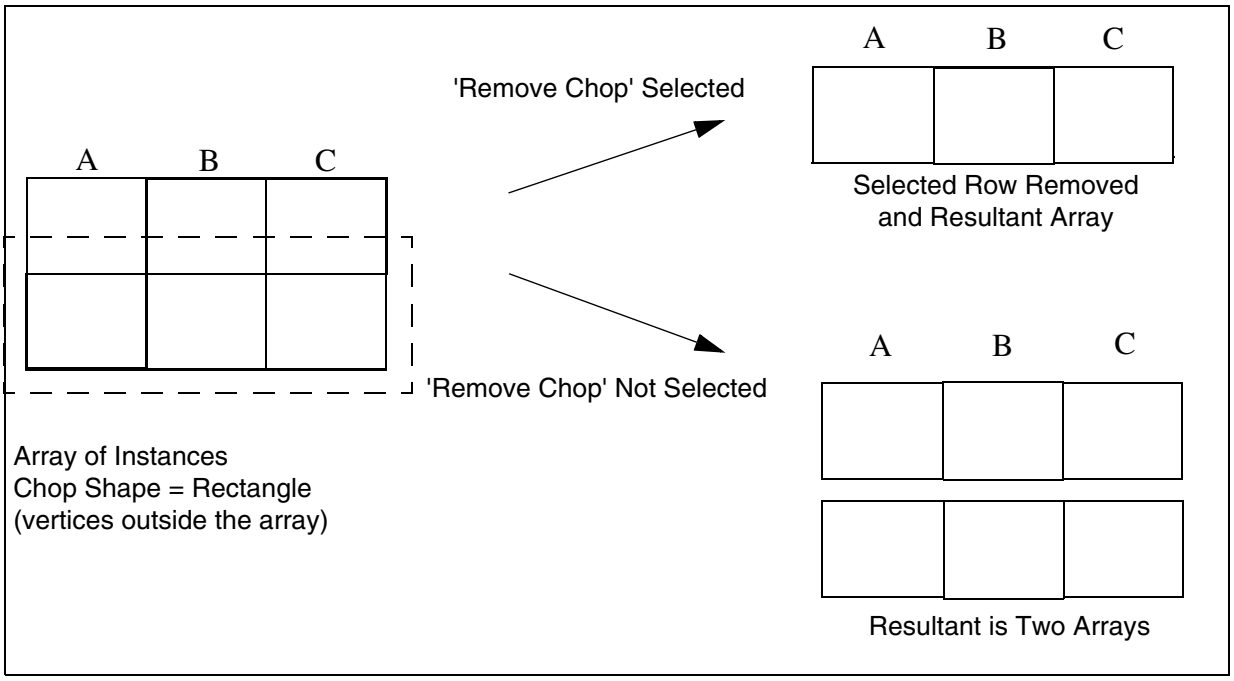
You can chop an array of instances (mosaic) or a group array. When you define the chop area for an array, none of the points of the chop area, which is vertices of the rectangle and polygon and end points of a line, should be contained within the array. The vertices or end points should be either outside the array or on the boundary of the array. This is explained in

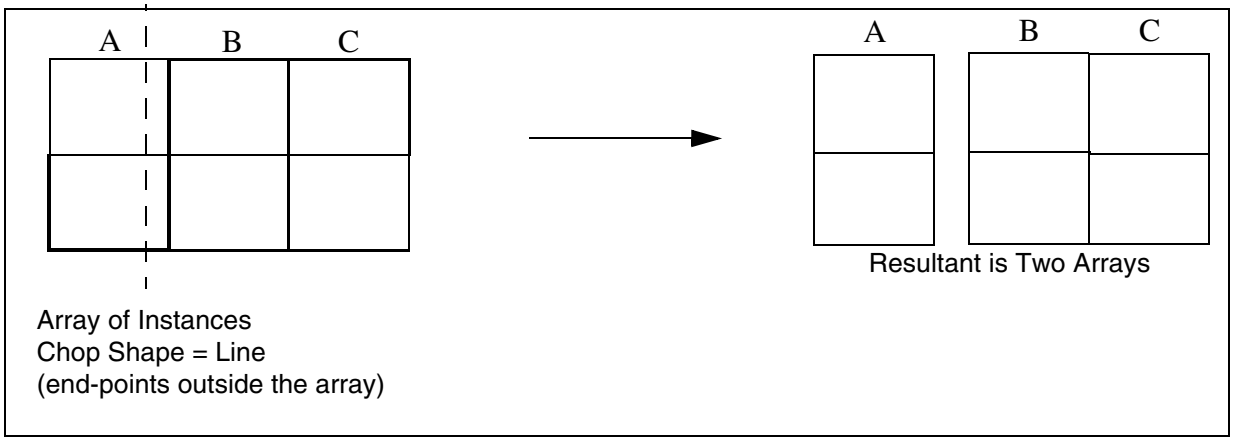
© 2023 Cadence Design Systems, Inc.

the following figures.



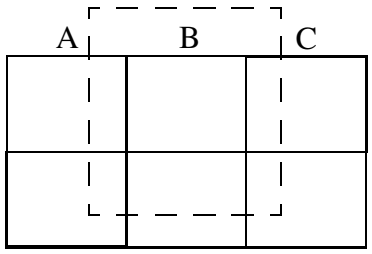
© 2023 Cadence Design Systems, Inc.



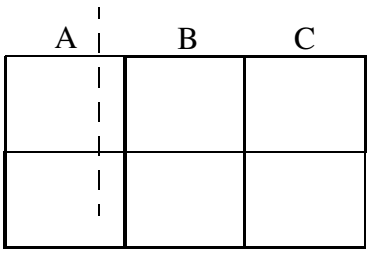


© 2023 Cadence Design Systems, Inc.

In the following chop areas, not all the points are outside the array. Therefore, these arrays of instances will remain unchanged even if you don't select the option to chop arrays.



Two vertices of the rectangle are inside the array.



One end-point of the line is inside the array.

Related Topics

Object Chopping

Chopping an Object

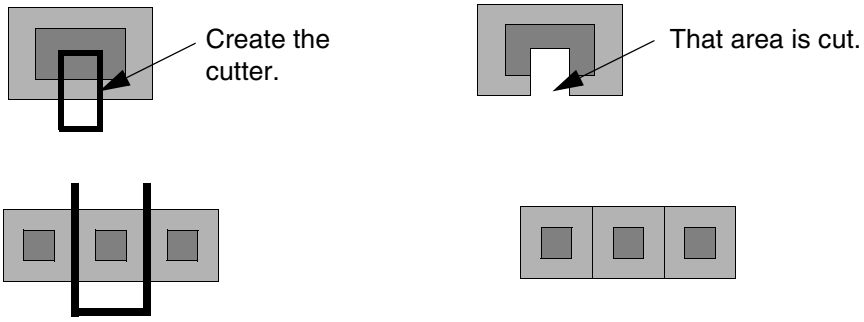
To cut away part of an object:

1. In the layout window, choose *Edit – Basic – Chop*.
2. Press `F3` to open the Chop form.
3. Set *Chop Shape* to *rectangle, line,* or *polygon*.

   If you select *line* or *polygon*, set the *Snap Mode*.
4. You can split an object by deselecting the *Remove chop* check box and using a rectangle or polygon chop shape. This saves both the area you cut and the original objects.
5. Select one or more objects on the canvas.
6. Click to specify the points of the chop.

© 2023 Cadence Design Systems, Inc.

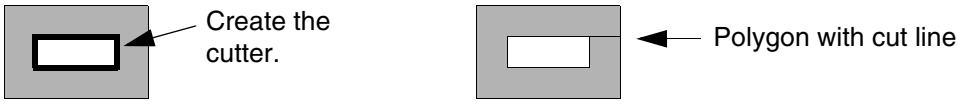
- Depending on the options that are set, the object area is cut away or the object is split.



Create a line that intersects a rectangle.

The rectangle is split into three rectangles.

- If you cut a hole in an object, it is redrawn as a polygon with a cut line.



- Paths convert to polygons when *Chop Shape* is set to *polygon* or *line* and *Snap Mode* is set to *anyAngle* or *diagonal*.
- If you cut a path with a diagonal line, the resulting object is a polygon.



Cutting the path with a diagonal line

After cutting, the path has been converted to a polygon.

The *Chop* command preserves selections even in the post-select mode if the retainSelectionSet environment variable is set to `t`.

Related Topics

Chop Form

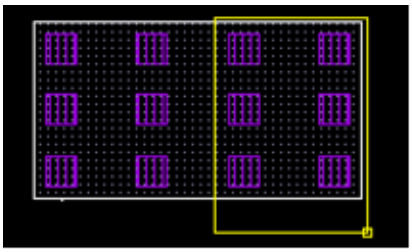
© 2023 Cadence Design Systems, Inc.

Attaching and Detaching Objects

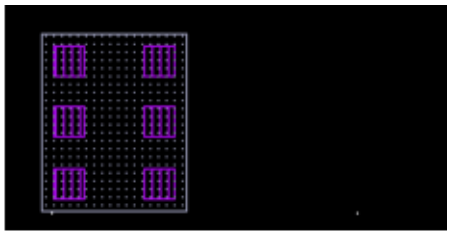
Via Chopping

The *Chop* command also lets you cut away part of a via object or cut a via object into pieces. The following are a few cases of how a via is chopped.

- When a via is chopped with the *Chop Shape* set to *rectangle*, the via is chopped like other shapes, as shown in the figure below.

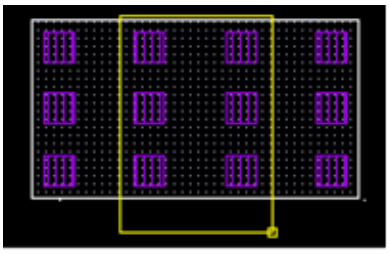


Rectangular Chop

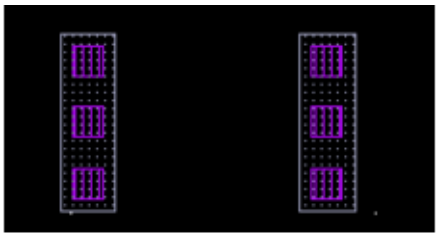


Chopped via

- When a via is chopped with the *Chop Shape* set to *rectangle* and the *Remove chop* check box selected, two distinct vias are created, as shown in the figure below.



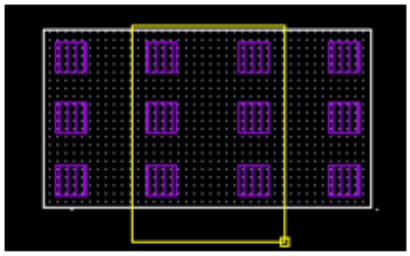
Rectangular chop with *Remove Chop* check box selected



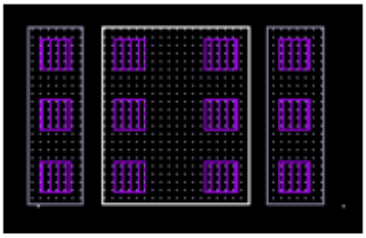
Chopped vias - Two distinct vias created

© 2023 Cadence Design Systems, Inc.

- When a via is chopped with the *Chop Shape* set to *rectangle* and the *Remove chop* check box not selected, three distinct vias are created, as shown in the figure below.



Rectangular chop with *Remove Chop* check box not selected



Chopped vias - Three distinct vias created

If the *Chop Shape* is set to *rectangle or line*, two distinct vias for each original via are created. However, if the *Chop Shape* is set to *polygon*, then *n* distinct vias are created. However, a limitation in case of chopping vias is that if any region of the chop shape is non-rectangular, then a via is not created in the non-rectangular region.

If you are chopping a via stack, each via in the viaStack is chopped independently. The chop functionality is not supported for a cdsFixedVia via.

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

Object Merging

You can merge one or more objects created on the same layer into one object. This is sometimes referred to as performing a logical AND on the objects.

Object Type	Support for the Merge Operation
ROD objects	The result of merging ROD objects is a ROD object. Merging a ROD object with any non-ROD shape produces a non-ROD shape. If the ROD objects being merged are on the same net, the resultant ROD object retains the net. If the ROD objects being merged are on different or no nets, the resultant ROD object will not be on any net. The result of merging two identical multipart paths (MPPs) is an MPP only when the merging of their master paths results in a path and not a polygon. If the result of merging the master paths is a polygon, the resultant is a ROD object. However, the subparts are not ROD objects; only the polygon on the base layer of the MPP is a ROD object.
Blockages	The *Merge* command merges blockage objects in the same way it merges polygons. ■ If one or all selected blockages are attached to the same instance, the resulting merged blockage will remain attached to the instance. ■ If the blockages are attached to different instances, a scroll list displays so that you can choose which instance will own the resulting merged blockage. One of the options is no attachment.
Boundaries	The *Merge* command does not support PR and snap boundary objects. ■ If an area boundary object merge involves more than one constraint group, a scroll list pops up, allowing you to choose the constraint group for the merged area boundary. You must also rename the area boundary. ■ If a cluster boundary object merge involves more than one cluster, a scroll list pops up, allowing you to pick a destination cluster. You must also rename the cluster boundary.

© 2023 Cadence Design Systems, Inc.

(Virtuoso RF Option) The *Merge* command is not supported for curved shapes.

Related Topics

Merging Objects

How Merging Converts Objects

How Merging Converts Paths

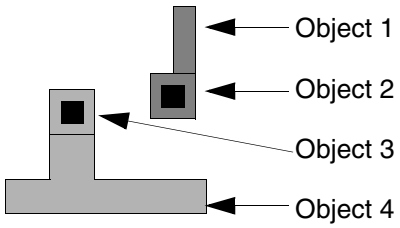
Merging Objects

To merge objects:

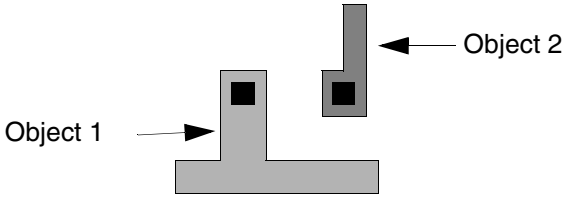
1. In the layout window, choose *Edit – Basic – Merge*.
2. Select one or more objects on the same layer. The objects must touch or overlap each other.

   The objects are merged.

   You can click and drag to create a selection box and merge several objects at once. Only those objects that are on the same layer and that touch are merged.



Select a group of objects while using *Merge*.



Objects on the same layer are merged.

Video

For details on how to use the *Merge* command in pre-select and post-select modes, watch the video *Using the Merge Command in Pre-select and Post-select Modes*.

Related Topics

Object Merging

© 2023 Cadence Design Systems, Inc.

How Merging Converts Objects

How Merging Converts Paths

How Merging Converts Objects

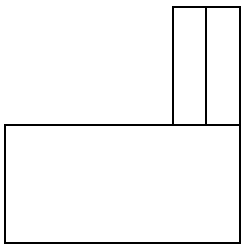
When you merge objects using Merge or Layer Generation, most merged objects are converted to polygons. If you merge a group of polygons and paths, the resulting object is a polygon.

However, when you merge paths of the same width abutting each other, they merge into a single path. When you merge overlapping paths, they merge into a polygon or rectangle.

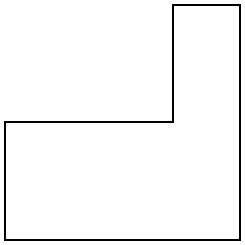
Overlapping or abutting rectangles of the same width adjoining end to end merge into rectangles.

When you merge objects, the resulting polygon should not have more than 4000 points (vertices).

When ROD objects are merged, the resulting shape is not a ROD object.



A polygon and a path before merging.



After merging, the new object is a polygon.

Related Topics

Object Merging

Merging Objects

How Merging Converts Paths

© 2023 Cadence Design Systems, Inc.

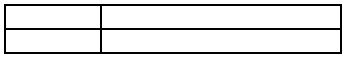
How Merging Converts Paths

When you merge collinear or non-collinear paths, certain conditions must be met for the resulting object to be a path.

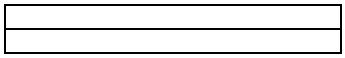
- Merging Collinear Paths
- Merging Non-Collinear Paths

Merging Collinear Paths

When you merge abutted collinear paths, the merged path must have the same width as the original path and should be on the same layer. If the path centers are not aligned, the merged path becomes a polygon. To ensure the path centers are aligned, you can change the X points in the Edit Path Properties form for each vertical path or the Y points for each horizontal path.

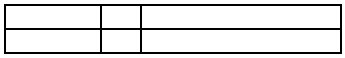


Abutting paths with the same width before merging.

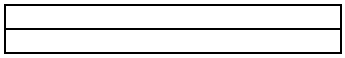


After merging, the new object is a path.

When you merge overlapping collinear paths, the merged path must have the same width and be on the same layer.



Overlapping collinear paths with the same width before merging.



After merging, the new object is a path.

Merging Non-Collinear Paths

When you merge overlapping horizontal and vertical paths, the following conditions should be met for the resulting object to be a path:

- For each end point of the paths at the overlapping ends, the following criterion must be met:

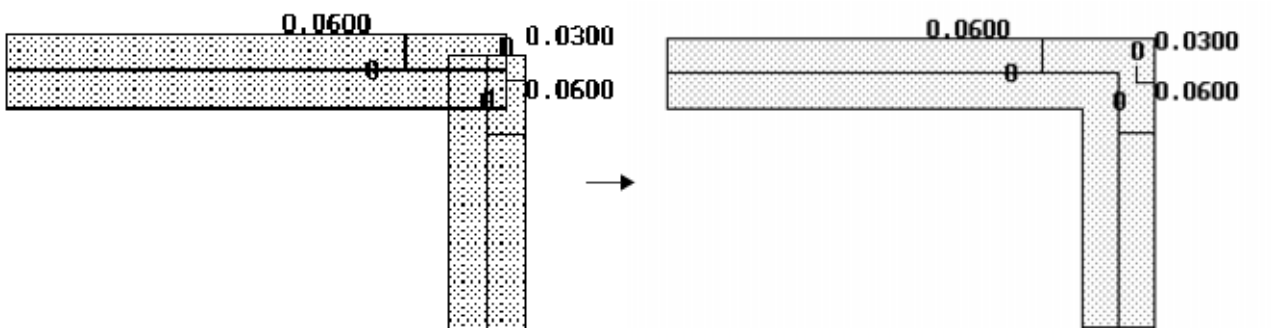
  center line of the other path =< end point of the path =< outer edge of the other path

- For each outer edge of the paths at the overlapping ends, the outer edge must not extend beyond the outer edge of the other path.

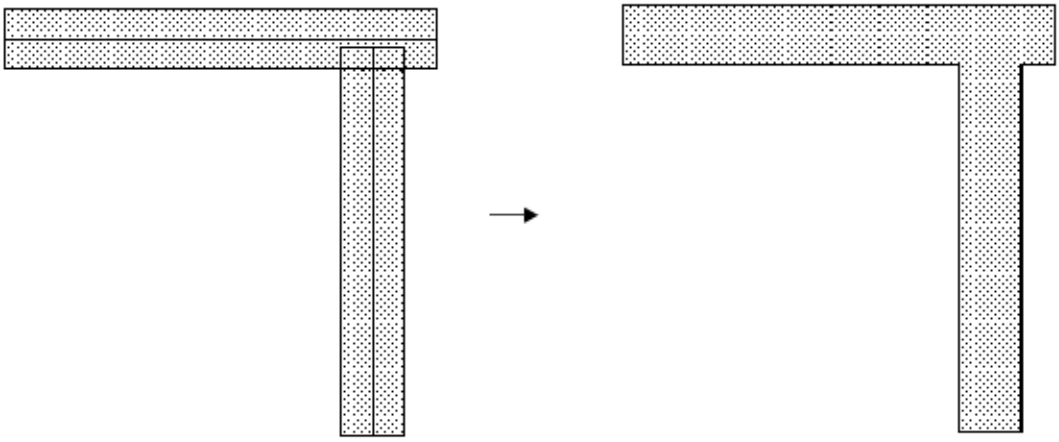
© 2023 Cadence Design Systems, Inc.

If either of these conditions is not met, the resulting object becomes a polygon.

In the following figure, both the above conditions are satisfied by the overlapping paths. Therefore, the resulting merged object is a path. The path ends move to form a right angle to make the path design rule correct.

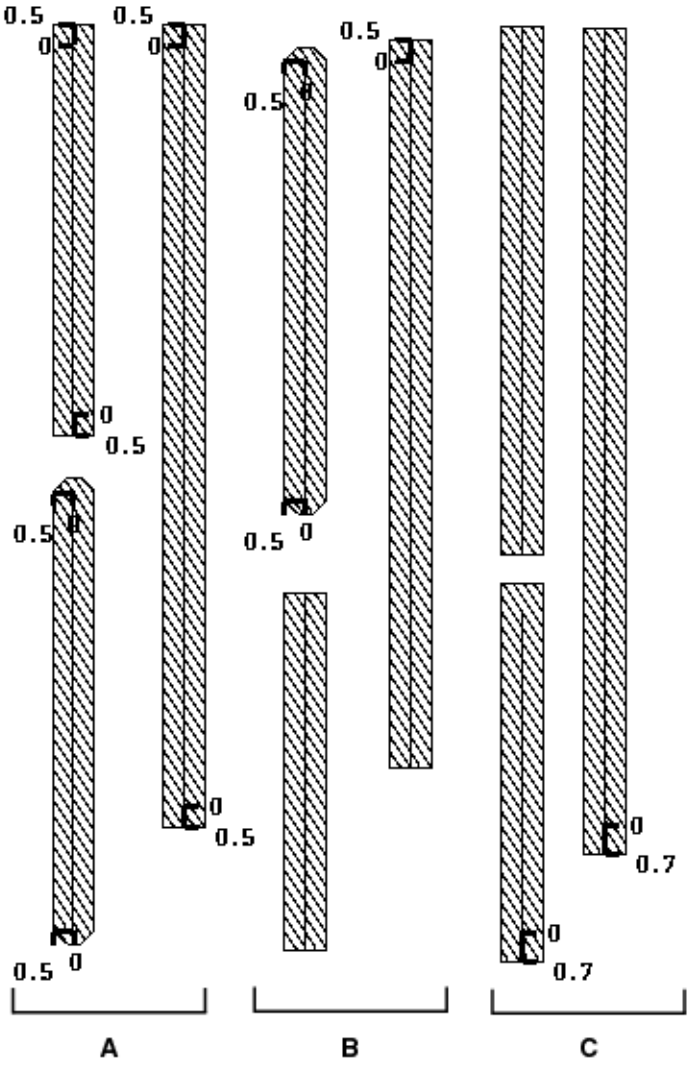


In the following figure, the outer edge of the horizontal path extends beyond the outer edge of the vertical path. Therefore, the resulting object is a polygon.



© 2023 Cadence Design Systems, Inc.

You can merge collinear paths and collinear multipart paths with multiple end types. The supported end types are truncate, extend, round, and variable. Some examples are shown below.



A: The two paths to the left are extend and round paths. The merged path is retained as an extend path with the same offset at the top as the extend path. The 0.5 width of the round path is retained at the end of the new extend path.

B: The two middle paths are round and truncate.The merged path is variable with the truncate end of the original bottom path retained and the 0.5 width of the round end retained at the end of the new variable path.

C: The two paths to the right are truncate and variable. The merged path is a variable path with the truncate end at the top path retained and the 0.7 offset retained of the original bottom path.

All paths being merged must be of the same width.

The following table summarizes the results of merging various path ends, where the resultant object of all merging combinations (columns 1 and 2) is a path (column 3). The table provides examples of begin extension, end extension, or width in the original paths to indicate the parameter values retained in the resultant merged path. In the table:

- **w** represents one half the path width of the round or extend path
- **b** represents begin extension

© 2023 Cadence Design Systems, Inc.

- **e** represents the end extension of any variable path end

A truncate path always has the following: 0, b, and e.

Type of Path 1		Type of Path 2		Type of Resultant Merged Path	
Truncate	(b0:e0)	Truncate	(b0:e0)	Truncate	(b0:e0)
Truncate	(b0:e0)	Extend	(w0.5:w0.5)	Extend	(b0:w0.5)
Truncate	(b0:e0)	Variable	(b2:e2)	Variable	(b0:e2)
Extend	(w0.5:w0.5)	Extend	(w0.5:w0.5)	Extend	(w0.5:w0.5)
Extend	(w0.5:w0.5)	Truncate	(b0:e0)	Truncate	(w0.5:e0)
Extend	(w0.5:w0.5)	Variable	(b2:e2)	Variable	(w0.5:e2)
Round	(w0.5:w0.5)	Round	(w0.5:w0.5)	Round	(w0.5:w0.5)
Variable	(b1:e1)	Variable	(b2:e2)	Variable	(b1:e2)
Variable	(b1:e1)	Truncate	(b0:e0)	Truncate	(b1:e0)
Variable	(b1:e1)	Extend	(w0.5:w0.5)	Extend	(b1:w0.5)

The following table reflects that the merging of any path with a round path results in a polygon.

Type of Path 1	Type of Path 2	Resultant Merged Object
Truncate	Round	Polygon
Extend	Round	Polygon
Round	Truncate	Polygon
Round	Extend	Polygon
Round	Variable	Polygon
Variable	Round	Polygon

Related Topics

Object Merging

Merging Objects

How Merging Converts Objects

© 2023 Cadence Design Systems, Inc.

Object Yanking and Pasting

The *Yank* command performs a cookie-cutter copy. Yank copies objects and parts of objects into a temporary buffer. The *Paste* command places copies of objects from the temporary buffer.

Consider the following while using the *Yank* and *Paste* commands:

- ROD objects are not supported. The *Yank* command copies only the shapes, but not the names of the shapes or the alignments. When you paste, the result is unnamed, unaligned, non-ROD shapes.
- Blockage objects are not supported. The *Yank* command copies only the shapes and ignores the blockage objects.
- Boundaries objects are not supported. The *Yank* command copies only the shapes and ignores the boundary objects.
- Mosaics are not supported.

Yank and *Paste* commands are similar to *Copy* except that objects are chopped by the Yank shape. The *Yank* command is available both in read-only and edit modes.

Related Topics

Yanking and Pasting Objects in Pre-Selection Mode

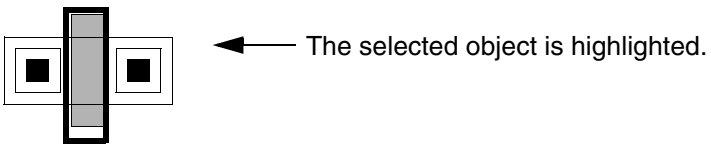
Yanking and Pasting Objects in Post-Selection Mode

Yanking and Pasting Objects in Pre-Selection Mode

You can pre-select objects before you invoke the *Yank* command. In pre-selection mode, only fully selected objects are yanked and partially selected objects are ignored.

To use the *Yank* command to copy an object, group of objects, or an instance:

1. On the layout canvas, select the object that you want to copy.



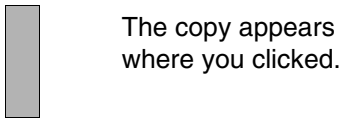
2. Choose *Edit – Basic – Yank*.

© 2023 Cadence Design Systems, Inc.

3. Choose *Edit – Basic – Paste*.

   The outline of the object you yanked appears and follows the pointer.

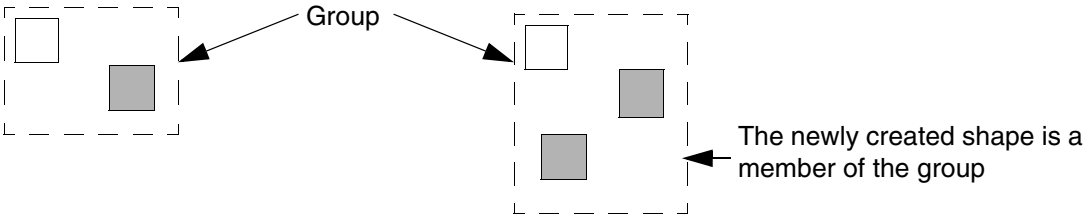
4. Click on the canvas where you want to place the copied object. Continue clicking to place multiple copies.



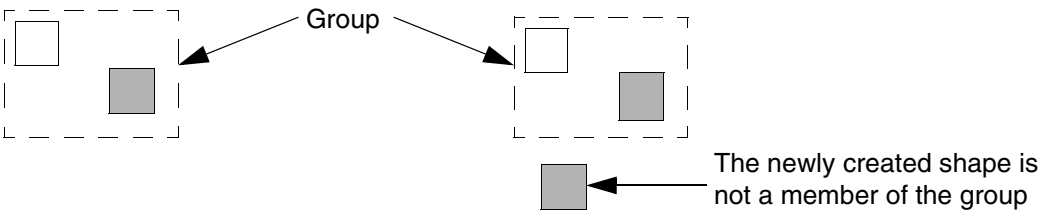
In pre-selection mode, the *Yank* command does not support boundaries (PR, snap, area, and cluster), blockages, guides, and steiners. In case the selected set contains any of these objects, a warning message is displayed in the CIW.

You can copy a group of objects as well using the *Yank* and *Paste* commands.

- When a group of objects is selected, all the members of the group are yanked. When you paste, the newly created objects become members of the newly created group.
- When you want to edit the contents of a group (*Edit – Hierarchy – Edit In Place*), select the shapes that you want to yank. When you paste, the newly created shapes remains a member of the group from which they were copied.

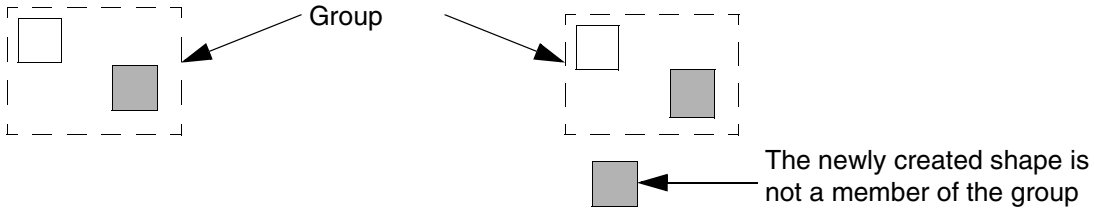


- When you want to edit the contents of a group (*Edit – Hierarchy – Edit In Place*), select the shapes that you want to yank. When you paste after you return (*Edit – Hierarchy – Return to Level*), the newly created shape is not a member of the group from which it was copied.



© 2023 Cadence Design Systems, Inc.

- When the transparent group mode is enabled, select the shapes that you want to yank. When you paste, the newly created shape is not a member of the group from which it was copied.



For information about how parent-child relationship behaves when either the parent or the child is yanked, see Attaching and Detaching Objects.

If you are editing a cell instance in place, you can select the objects to be yanked and then paste these objects. You can also paste in the top cellview even after you return from Edit In Place, using *Edit - Hierarchy - Return*.

Related Topics

Yank Form

Yanking and Pasting Objects in Post-Selection Mode

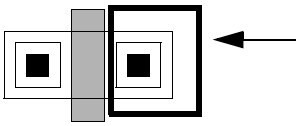
To copy part of an object, group of objects, or an instance:

1. In the layout window, choose *Edit – Basic – Yank*.
2. Press `F3` to open the Yank form.

   In the Yank form, by default, *Yank Shape* is set to create a rectangle. You can change it to a polygon, as required.
3. If you are in full selection mode, press `F4` to switch to partial selection mode.
4. Create the yank shape around the objects you want to copy.

© 2023 Cadence Design Systems, Inc.

All objects and parts of objects inside the box are copied into a special yank buffer. If an instance is inserted by the *Yank* shape, the objects in it are brought to the top level (flattened).



5. Choose *Edit – Basic – Paste*.

   The outline of the objects you yanked appears and follow the pointer.

6. Click on the canvas at the location where you want to place the copied objects.



You can paste an object whose layer is not visible in the *Layers* panel of the Palette assistant. However, you cannot yank an object whose layer is not visible in the *Layers* panel and whose object type is not visible in the *Objects* panel of the Palette assistant.

Related Topics

Yank Form

Object Reshaping

You can change the shape of a selected object by using the *Reshape* command. The following table describes the support available for reshaping objects.

Object Type	Support for the Reshape Operation
ROD objects	Except for multipart paths and multipart rectangles, all ROD objects that can be reshaped remain ROD objects after a reshape operation. When you reshape any ROD object, all alignments to that object are deleted.
Rows	The *Reshape* command does not support rows.
Groups	The *Reshape* command does not support group objects.
Blockages	The *Reshape* command reshapes blockages in the same way it reshapes a polygon.

© 2023 Cadence Design Systems, Inc.

Object Type	Support for the Reshape Operation
Boundaries	The *Reshape* command reshapes PR and cluster boundaries in the same way it reshapes a polygon, using points. The *Reshape* command supports reshaping of IOBox and CoreBoxSpec when PR boundary is reshaped. The relative distance of IOBox and CoreBoxSpec from the PR boundary is maintained. The *Reshape* command does not support snap boundary objects.

Related Topics

Reshape Form

Reshaping a Polygon

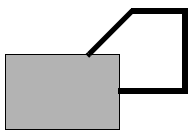
Adding a Rectangle to a Polygon

Reshaping a Path

Reshaping a Polygon

To reshape a polygon:

1. In the layout window, choose *Edit – Advanced – Reshape*.
2. Select a polygon on the canvas.
3. Press `F3` to open the Reshape form.
4. From *Mode*, select *line* as the reshape mode.
5. Create the new section of the polygon on the canvas.

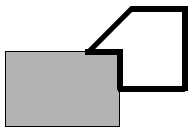


The first and last points must touch the original shape.

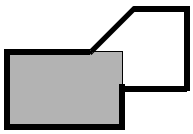
6. Double-click when you are finished entering points.

© 2023 Cadence Design Systems, Inc.

7. Click the middle mouse button to toggle between highlighting the new shape and highlighting both the old and new shape.



Highlight the new shape to replace the original shape.



Highlight both shapes to reshape the original shape.

8. When the shape you want is highlighted, click on the canvas.

Related Topics

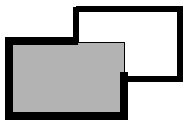
Reshape Form

Reshaping an Object in Level-1 Editing Mode

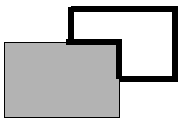
Adding a Rectangle to a Polygon

To add a rectangle to a polygon:

1. In the layout window, choose *Edit – Advanced – Reshape*.
2. Select a polygon on the canvas.
3. Press `F3` to open the Reshape form.
4. From *Mode*, select *rectangle* as the reshape mode.
5. Create a rectangle that intersects the polygon on the canvas.
6. Click the middle mouse button to toggle between highlighting the new shape and highlighting both the old and new shape.



.Highlight both shapes to reshape the original shape



Highlight the new shape to replace the original shape

7. When the shape you want is highlighted, click on the canvas.

Related Topics

Reshape Form

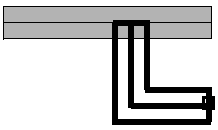
© 2023 Cadence Design Systems, Inc.

Reshaping an Object in Level-1 Editing Mode

Reshaping a Path

To reshape a path:

1. In the layout window, choose *Edit – Advanced – Reshape*.
2. Select a path on the canvas.
3. Press `F3` to open the Reshape form.
4. From *Mode*, select *line* as the reshape mode.
5. Starting from the centerline, create the new section of the path on the canvas.



Start the new segment from the path centerline.

6. Double-click when you are finished entering points.
7. Click the middle mouse button to toggle between highlighting the reshaped path options.



Reshape options

8. When the shape you want is highlighted, click on the canvas.

Related Topics

Reshape Form

Reshaping an Object in Level-1 Editing Mode

Object Splitting

Using the *Split* command, you can split and stretch a portion of an object. You usually use it to add a jog by stretching a section of a path or group of paths (a bus). You can also add a

© 2023 Cadence Design Systems, Inc.

jog by stretching a section of a multipart path. The following table describes the support available for splitting objects.

Object Type	Support for the Split Operation
ROD objects	Except for ellipses, circles, donuts, multipart paths, and multipart rectangles, ROD objects remain ROD objects after a split operation. When you split any ROD object, all alignments to that object are deleted.
Blockages	The *Split* command does not support blockage objects. In pre-selection mode, the command fails, displaying a message in the CIW. In post-selection mode, the command fails to select the blockage object.
Boundaries	The *Split* command does not support PR and snap boundary objects. In pre-selection mode, the command fails, and a message appears in the CIW. In post-selection mode, the command fails to select the boundary object. For area and cluster boundary objects, the *Split* command splits the objects in the same way it splits polygons, using points.

Related Topics

Split Form

Splitting an Object

Changing the Angle While Splitting an Object

Splitting Multiple Objects

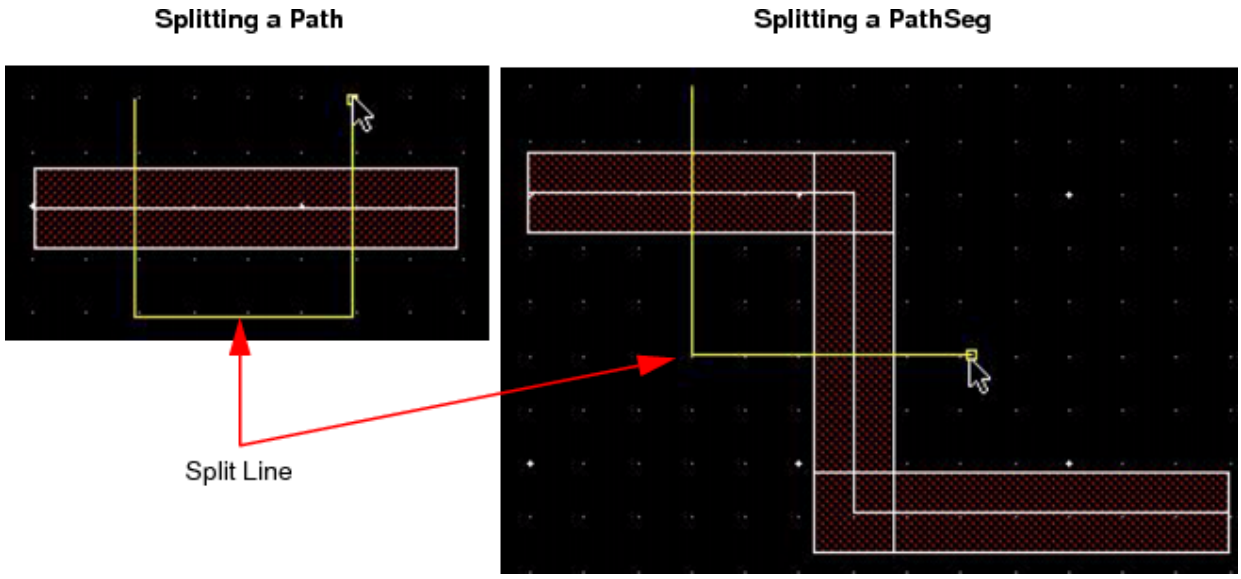
Splitting an Object

To split and stretch an object:

1. On the layout canvas, select the object you want to split.
2. Choose *Edit – Advanced – Split*.

© 2023 Cadence Design Systems, Inc.

3. Click the points of the split line on the canvas.



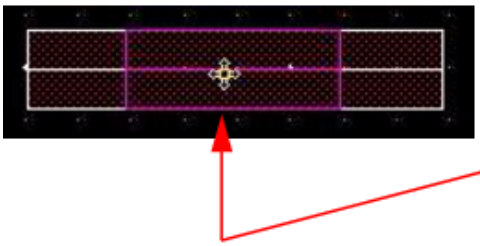
You can use the middle mouse button to toggle between the split segments. In the above example, the split will alternately highlight the path section interior to the split lines, and the two segments of the path exterior to the split lines.

When the split shape is drawn clockwise, the *Split* command selects the parts of the object outside the split lines first. When the split shape is drawn counterclockwise, the command selects the parts of the object interior to the split lines first. In the case of a single split line, toggling with the middle mouse button will alternate between the sides of the object divided by the split line.

© 2023 Cadence Design Systems, Inc.

4. Click the reference point for the stretch.

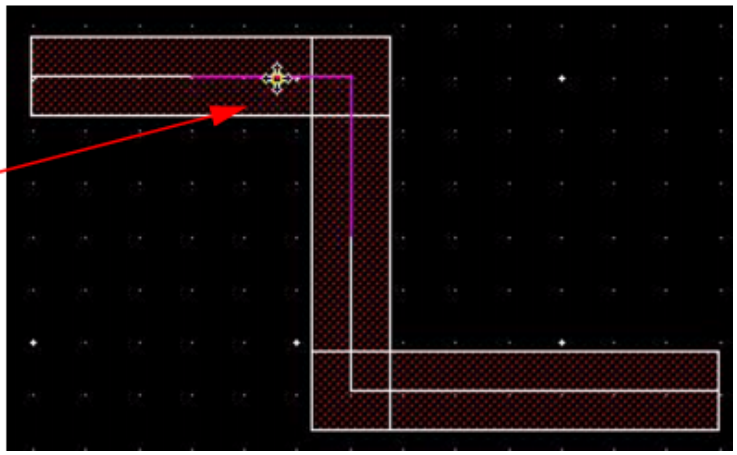
Splitting a Path



Click on the reference point.

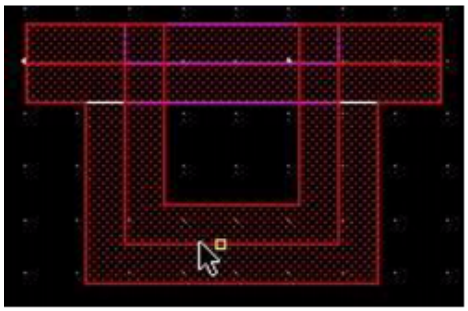
Notice the cursor shape on the reference area to be stretched.

Splitting a PathSeg

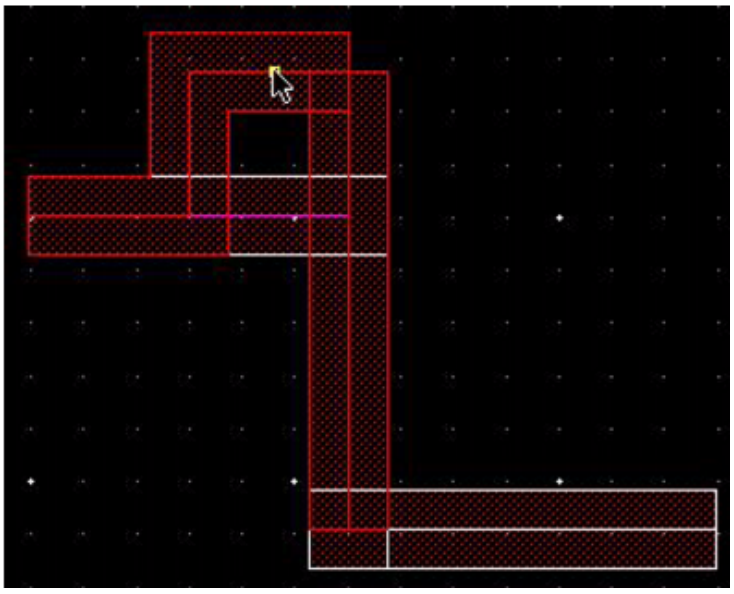


5. Click the new location for the stretch.

Splitting a Path

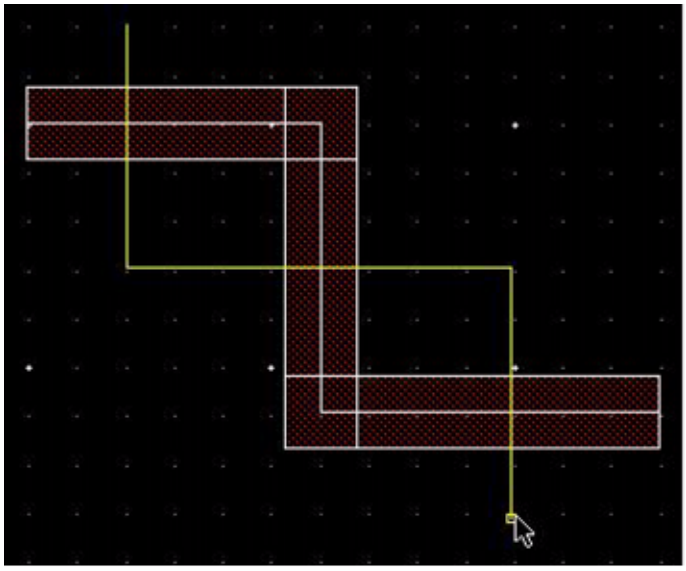


Splitting a PathSeg

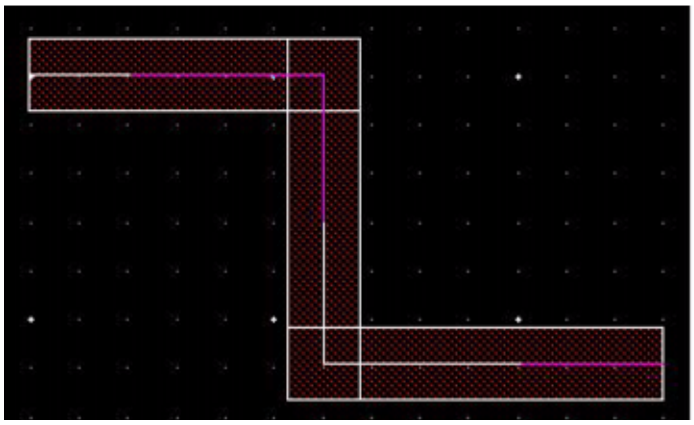


© 2023 Cadence Design Systems, Inc.

The following figures illustrate splitting and stretching of multiple segments of a pathSeg at the same time.

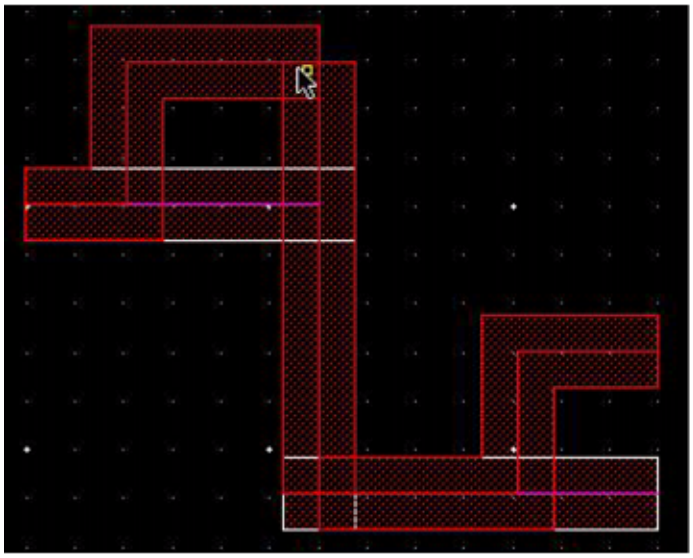


Creating the split line

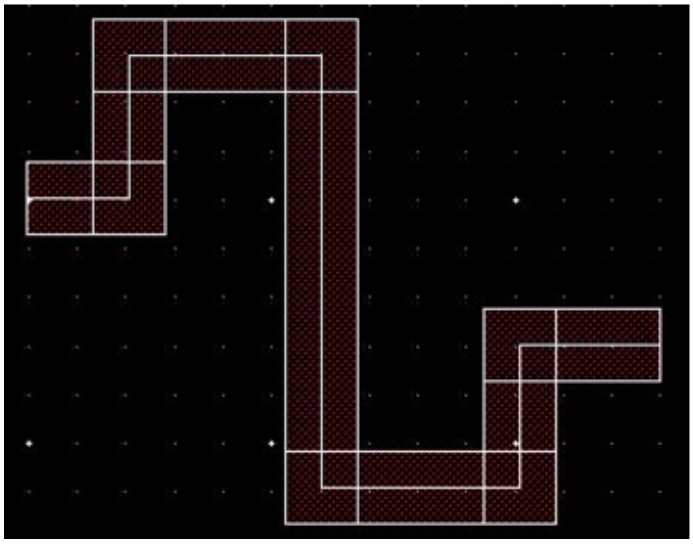


Clicking on the reference stretch area

© 2023 Cadence Design Systems, Inc.



Stretching the split portions



Resultant of the split and stretch

Related Topics

Split Form

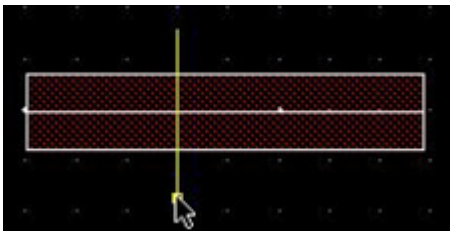
Changing the Angle While Splitting an Object

As you stretch objects, you can change the angle of the segments.

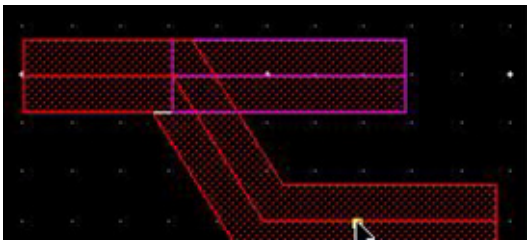
To stretch path segments to any angle:

© 2023 Cadence Design Systems, Inc.

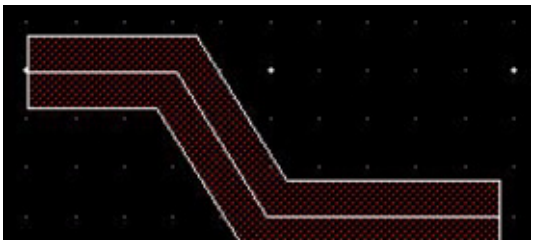
1. In the layout window, choose *Edit – Advanced – Split*.
2. Press `F3` to open the Split form.
3. Deselect the *Lock angles* check box.
4. Ensure the *Snap Mode* is set to *anyAngle* or *diagonal* to change the angle of the segment.
5. Create a cut line through the segment.



6. Stretch the highlighted path segment to the angle you want.



The following figure shows the resultant stretched object.



You can add a jog to several paths (a bus) so that it wraps around an object by splitting a section of the paths and stretching the split section.

Related Topics

Split Form

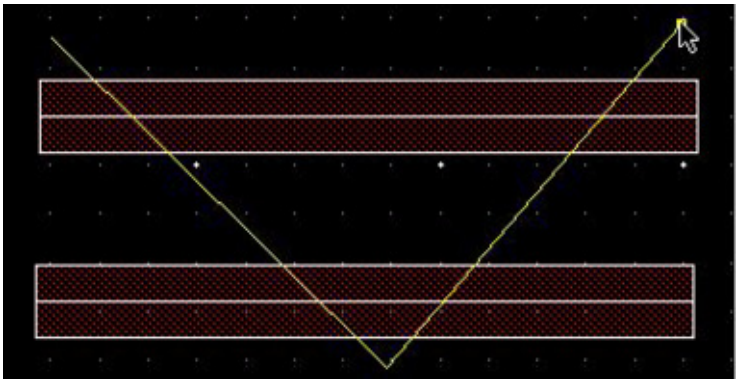
© 2023 Cadence Design Systems, Inc.

Splitting Multiple Objects

To stretch several paths:

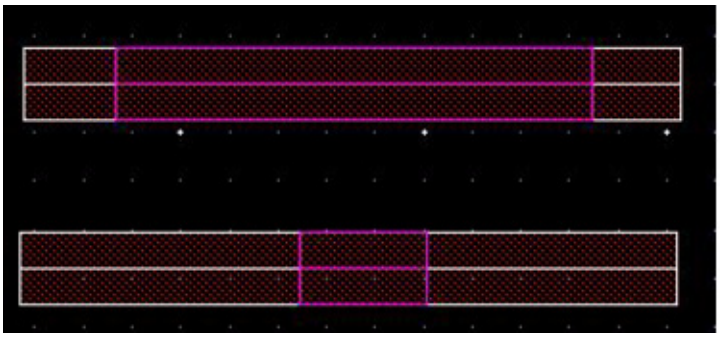
1. On the layout canvas, select all paths and connected objects you want to stretch.
2. Choose *Edit – Advanced – Split*.
3. Press `F3` to open the form.
4. Set *Snap Mode* to *anyAngle*.
5. Click on the canvas to create a split line through the segments.

   To stretch a segment of each path, create a split line that crosses each path twice, once at each end of the segment you want to stretch.

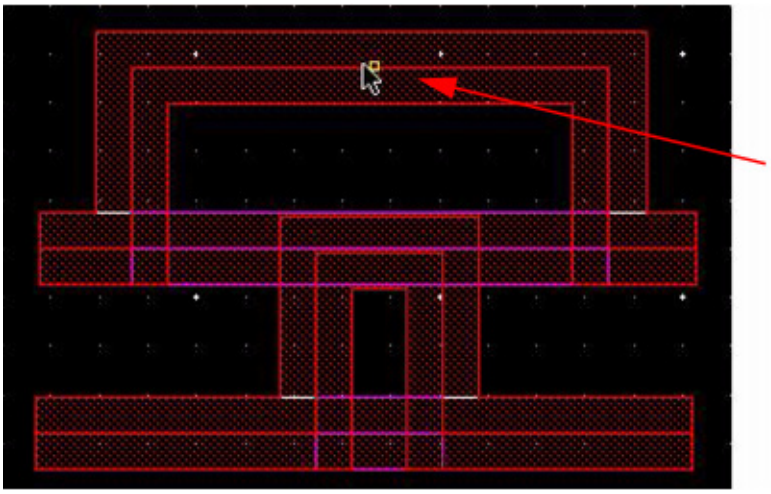
   Create the split line at a 45-degree angle through the segments so that the edges of the segments are offset from each other. This prevents the path segments from overlapping when you stretch them.
6. Double-click at the last point of the split line.

   The reference area to be stretched is highlighted.

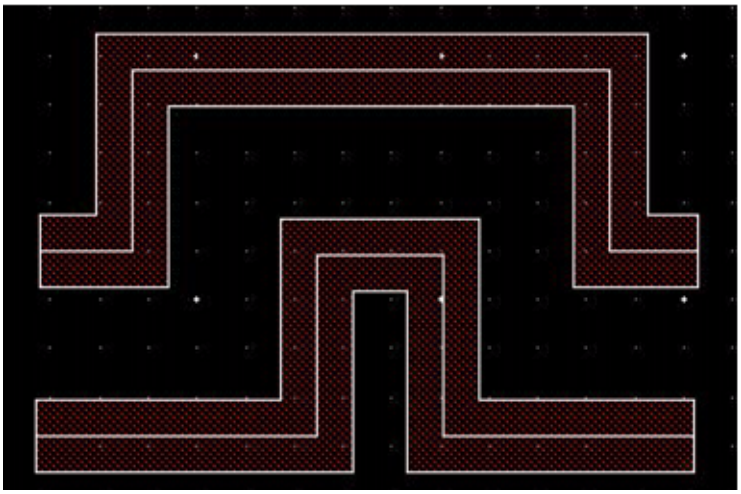
   
7. Click at the starting point for the stretch.

© 2023 Cadence Design Systems, Inc.

8. Click at the new location where you want to stretch the segments.



The following figure shows the resultant stretched objects:



The path segments are stretched. Also, the following other objects move:

- Any objects attached to the segments (such as vias) that have been selected.
- Any objects, such as instances, labels, mosaics, bends, and tapers, not recognized by the *Split* command but attached to the path, if they have been selected.
- Objects aligned to a multipart path you stretch stay in alignment, even if you did not select them.

Subrectangles and subpaths in multipart paths regenerate after the stretch.

© 2023 Cadence Design Systems, Inc.

Related Topics

Split Form

Modifying Corners of a Polygon

Using the *Modify Corner* command, you can reshape the corner of a polygon to make a rounded or chamfered (45-degree) edge. The *Modify Corner* command functions in the following ways for different objects.

- ROD objects: Supports ROD polygons and ROD rectangles, except for master rectangles that are part of a path object
- Rectangles: Supports rectangles, and the resultant modified shape is a polygon.
- Blockages and boundaries: Does not support blockage and boundary objects. In pre-selection mode, the command fail, and a message appears in the CIW. In post-selection mode, the command fails to select the blockage and boundary objects.

To modify a corner:

1. In the layout window, choose *Edit – Advanced – Modify Corner*.

   The Modify Corner form opens.

2. Set the type of corner by doing either of the following:

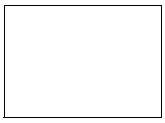
   - Select *radial*. Then set *Radius*, which is the radius of the curve and *Number of Sides*, which is the number of segments to create the curve.
   - Select *chamfer* and set *Distance*.

   The maximum value you can set is half the length of the shortest adjacent line segment. The layout editor applies this value or the maximum allowable value for each corner, whichever is shorter.

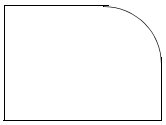
3. On the canvas, click the polygon corner that you want to change. To change more than one corner, press `Shift` and click each additional corner.

© 2023 Cadence Design Systems, Inc.

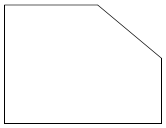
4. Click *OK*.



Original object



Object with a radial corner



Object with a chamfer corner

Related Topics

Modify Corner Form

Modifying Object Size

The *Size* command reduces or enlarges objects by stretching each edge in or out by the given number of user units.

The *Size* command supports ROD objects. However, when you size a multipart path (MPP), the master path becomes a ROD polygon and inherits its name from the original MPP; all subparts become ordinary, unnamed database shapes. When you size any ROD object, all alignments to that object are deleted.

To modify the size of objects:

1. On the layout canvas, select the objects you want to resize.
2. Choose *Edit – Advanced – Size*.

   The Size form opens.

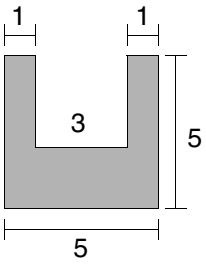


3. Select the *Increase Size by Value* or *Increase Size by Direction* to specify the number of units you want to stretch. If you choose to increase the size by direction, you need to specify the *Left*, *Right*, *Top*, or *Bottom* units. A positive number enlarges the object, a negative number reduces it.

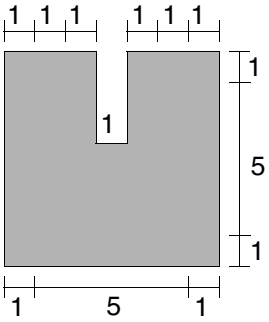
© 2023 Cadence Design Systems, Inc.

Increase Size by Direction works only for orthogonal geometries.

4. Click *OK*.



Object before resizing.



Object after using *Size* with a value of 1 to grow each edge by 1 unit.

(Virtuoso RF Option) The `Size` command does not support curved shapes.

Related Topics

Size Form

Parent-Child Relationship Between Objects

When you attach one object to another, you create a parent-child relationship between the two objects. The attached object is the child of the object to which it is attached. The following table describes how the various commands support the parent-child relationships.

Command	Parent-Child Relationship Support
Move	■ When the parent moves, all child objects move with it. ■ When a child moves, the parent does not move.
Delete	■ When the parent is deleted, all child objects are deleted. ■ When a child is deleted, the parent is not deleted.

© 2023 Cadence Design Systems, Inc.

Command	Parent-Child Relationship Support
Chop	■ When the parent is chopped such that there is only one resultant shape, the parent-child relationship is retained. ■ When the parent is chopped such that there are more than one resultant shapes, one of the chopped shapes becomes the parent of the child. ■ When a child is chopped such that there is only one resultant shape, the parent-child relationship is retained. ■ When a child is chopped such that there are more than one resultant shapes, all the shapes attach as child to the parent.
Yank	■ When the child is selected and the parent is not selected, the child is yanked. After the *Paste* command, the new object is independent. ■ When the parent is selected and the child is not selected, the parent is yanked. After the *Paste* command, the new object does not have any child. ■ When the child and parent are selected, both the child and parent are yanked. After the *Paste* command, the parent-child relationship is retained in the new object.

You use the *Attach/Detach* command to establish parent-child relationships. The command fully support boundary objects. When the parent is deleted, the child is also deleted.

The *Attach/Detach* command provides the following support for blockages:

- Cyclical relationships are not allowed. For example, if object A is a parent to object B, then object B cannot be a parent for object A.
- An instance or cluster cannot be a child to blockage objects.
- If a blockage object is a child object to an instance or cluster, during attach the ownership information is updated, but the explicit parent-child relationship is not retained.
- If a blockage object is attached as a child to any object other than the instance or cluster, the explicit parent-child relationship is retained.
- A blockage object can be a parent to any other objects, such as text labels.

© 2023 Cadence Design Systems, Inc.

- A blockage object cannot be owned by an instance and be a child to any other object. In this case, the owner-blockage relationship takes precedence to the parent-child relationship.

Related Topics

Attaching and Detaching Objects

Attaching and Detaching Objects

You use the *Attach/Detach* command to attach or detach objects.

Attaching Objects

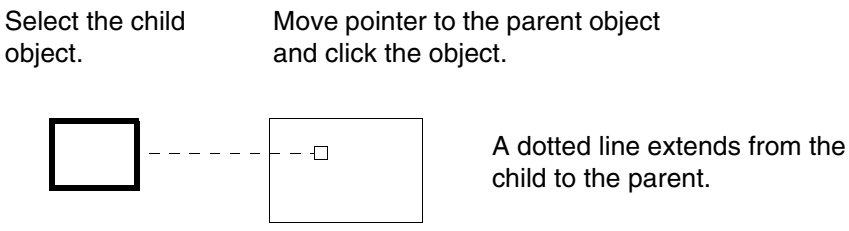
To attach objects:

1. In the layout window, choose *Edit – Advanced – Attach/Detach*.

   The following message appears in the CIW:

   ```
   Select child object to be attached or detached.
   ```

2. Click the object that you want to be the child object. A dotted line extends from the child to the pointer.
3. Move the pointer to the object that you want to be the parent object and click the parent object.



   The child is attached to the parent.

   You can attach all objects to a parent object by pressing `Shift` and clicking the middle mouse button.

Detaching Objects

To detach objects that have been joined using the *Attach/Detach* command:

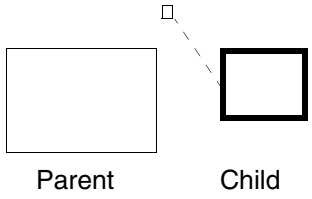
© 2023 Cadence Design Systems, Inc.

1. On the layout canvas, select the child objects that you want to detach.
2. Choose *Edit – Advanced – Attach/Detach*.

   The following message appears in the CIW:

   ```
   To attach child object, click the parent object;
   to detach child object, click in empty area.
   ```

   A dotted line extends from the child to the pointer.
3. Move the pointer to an empty area on the canvas and click the empty area.



   The child object is detached from the parent.
4. If you selected more than one child object, click again in an empty area.

   You can detach all objects from the parent object by clicking the middle mouse button in an empty area.

Related Topics

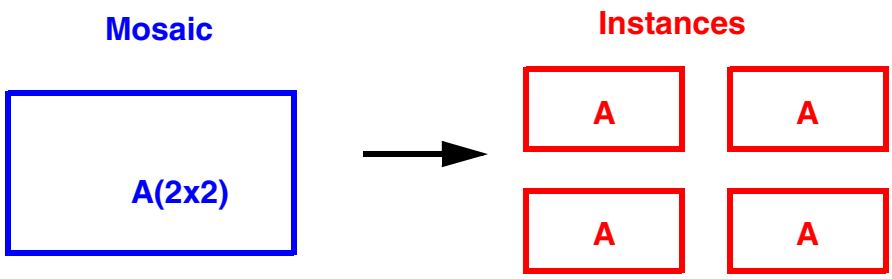
Parent-Child Relationship Between Objects

Object Conversion

The following table describes the commands that you can access from the *Edit* menu to convert objects.

Command	Description
Convert – To Mosaic	Converts the selected instances into mosaics.

© 2023 Cadence Design Systems, Inc.

Command	Description
Convert – To Instance	Converts selected mosaic to instances. Each instance inherits the name, orientation, and origin of the mosaic from which it is derived. 
Convert – To Polygon	Converts circles, donuts, and ellipses to polygons.
Convert – To PathSeg	Converts selected shapes, polygons, and paths to pathsegs. **(Virtuoso RF Option)** The *Convert To PathSeg* command is not supported for curved shapes.
Convert – To Path	Converts polygons and rectangles to paths.
Convert – To Modgen	Converts mosaics to modgens.

Related Topics

Instance to Mosaic Form

Converting Instances to Mosaics

Converting Circles, Donuts, and Ellipses to Polygons

Converting Polygons to Paths

Instance-to-Mosaic Conversion

Using the *Convert – To Mosaic* command, you can convert the selected instances into mosaics. The command works on physical and non-physical instances, fluid guard rings, and Pcell instances.

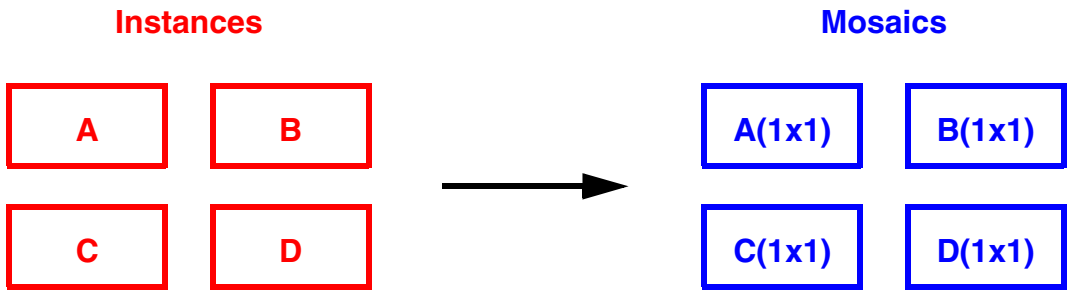
© 2023 Cadence Design Systems, Inc.

You can run the command in two modes: *Convert Instances into Mosaics* or *Convert Instances into Single Mosaic*.



Convert Instances into Mosaics Mode

This is the default mode and converts a number of selected instances into an equivalent number of 1x1 mosaics.



Each mosaic inherits the name, orientation, and origin of the instance from which it was derived. Cluster and group memberships and parent-child relationships are retained, along with any halo blockages and text displays associated with the original instance.

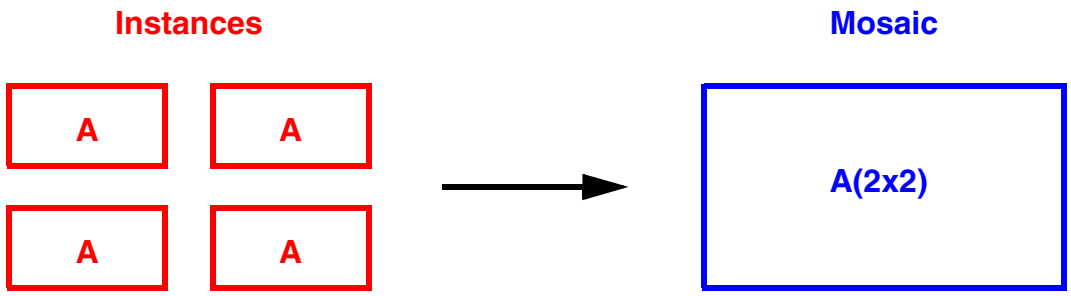
Connectivity information and any process rule overrides on the original instances are lost. Additionally, if the original instance was a member of any constraints, you must check and re-create these manually where necessary. The *Convert to Mosaic* command does not check or correct any constraints.

The *Convert to Mosaic* command is also available on the Edit Instance Properties form. When the *Common* editing mode is on, multiple instances are converted into multiple mosaics.

© 2023 Cadence Design Systems, Inc.

Convert Instances into Single Mosaic Mode

This mode converts a number of selected instances of the same master into a single mosaic. For example, the following figure shows that four instances of the same master, `A`, are selected and converted into a single 2x2 mosaic.

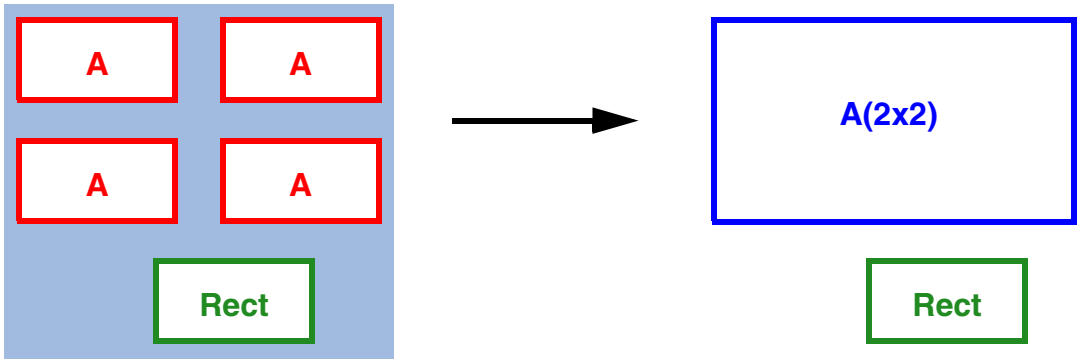


In this mode, all the selected instances must have the same master and must be placed at regular intervals in both X and Y directions. However, the intervals need not be the same in both directions. If the selected instances are Pcells, then all the parameter values must match. The name of the new mosaic is generated automatically and reported in the CIW. It has the same origin as the lower-left instance in the original selected set.

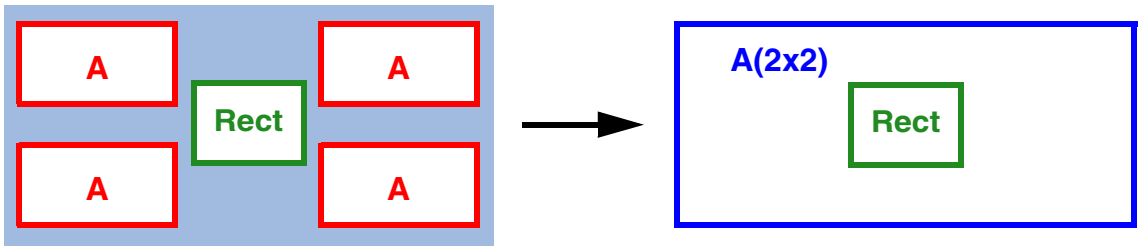
© 2023 Cadence Design Systems, Inc.

Parent-child relationships, properties, and text displays are not maintained in the resultant mosaic. If the selected set includes objects other than instances, these are left untouched when the command is complete.

Example 1



Example 2



Related Topics

Instance to Mosaic Form

Object Conversion

Converting Instances to Mosaics

Converting Circles, Donuts, and Ellipses to Polygons

Converting Polygons to Paths

Edit Properties Form

© 2023 Cadence Design Systems, Inc.

Converting Instances to Mosaics

To convert instances to mosaics:

1. In the layout window, choose *Edit – Convert – To Mosaic*.

   You can choose to select the instances before you start the command.

2. Press `F3` to open the Instance To Mosaic form.

   You can also select *Convert To Mosaic* from the *Instance* shortcut menu to open the form.



3. Choose the *Mode*.

   The default is controlled using the convertEachInstanceIntoMosaic environment variable.

4. On the canvas, select the instances to be converted and double-click on the canvas or press `Enter`.

   The selected instances are converted into one or more mosaics depending on the mode selected.

In the Layout Editor Options form, if the *Instance Editing Mode* is set to *P&R Boundary* or *Snap Boundary*, when converting instances to mosaics, the *Delta X* and *Delta Y* is set from the PR boundary or snap boundary, and not from the bounding box.

Related Topics

Instance to Mosaic Form

Object Conversion

Instance-to-Mosaic Conversion

Converting Circles, Donuts, and Ellipses to Polygons

© 2023 Cadence Design Systems, Inc.

Converting Polygons to Paths

Converting Circles, Donuts, and Ellipses to Polygons

Circles, donuts, and ellipses are stored in the database as true conics with centerpoints and radii. You might need to convert conics to polygons before streaming out your database for chip production.

The *Convert To Polygon* command fully supports ROD objects, except for rectangles and multipart rectangles. The command has no effect on rectangles and multipart rectangles.

When you convert a multipart path, the polygon resulting from the master path is a ROD object and takes the name of the multipart path. Subparts, if any, become unnamed, ordinary database shapes (paths and rectangles). Subparts are not converted to polygons.

When you convert any ROD object (except rectangles and multipart rectangles) to a polygon, all alignments to the converted object are deleted.

The *Convert to Polygon* command does not support blockage and boundary objects. In pre-selection mode, the command fails, and an error message is displayed appear in the CIW. In post-selection mode, the command fails to select the blockage and boundary objects.

To convert a conic to a polygon:

1. In the layout window, choose *Edit – Convert – To Polygon*.
2. Click the conic that you want to convert.

   The conic is redrawn as a polygon, using many straight-line segments to create each curve.
3. Continue clicking on each conic that you want to convert.
4. Press `Esc` to stop converting shapes to polygons.

   No conversion happens if you run the *Convert to Polygon* command on a rectangle or a polygon. A message appears in the CIW if you attempt to convert a rectangle or polygon to a polygon.

Setting Conic Sides

You can set the number of segments used when recreating conics when you convert them to polygons. The minimum number of conic sides is 3, and the maximum number of conic sides is 4000. The default number of segments is 20.

© 2023 Cadence Design Systems, Inc.

To set the number of conic sides:

1. Choose *Options – Editor*.

   The Layout Editor Options form opens.
2. Type a value in the *Conic Sides* field.
3. Click *OK*.

Related Topics

Layout Editor Options Form

Object Conversion

Instance-to-Mosaic Conversion

Converting Instances to Mosaics

Converting Polygons to Paths

Converting Polygons to Paths

With the *Convert – To Path* command, you can convert polygons and rectangles into paths. This command works in both pre-selection and post-selection modes. However, only supported objects, polygons or rectangles, are converted into paths.

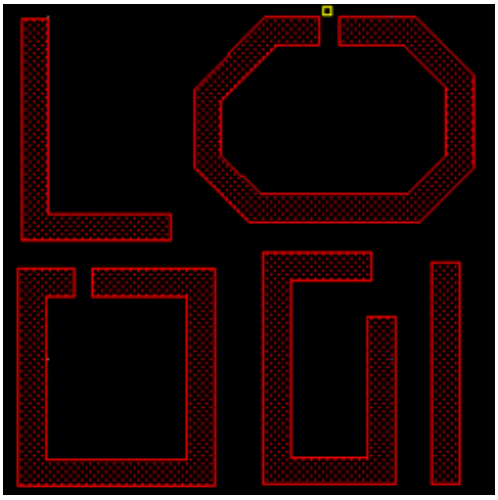
The *Convert – To Path* command can convert the polygons to paths if the following criteria are met:

- Polygons must have a consistent width to be converted to a path.
- The edges of polygons must be orthogonal or 45 degrees.
- The number of vertices of polygons must be even.

Polygons with rounded ends cannot be converted to a path.

© 2023 Cadence Design Systems, Inc.

The following figure shows examples of polygons that can be converted to a path.



The following figure shows examples of polygons that cannot be converted to paths.



The first polygon in the above figure, the Z shape, cannot be converted to a path. This is because it does not have exact matching vertices on the inside and outside of the bends of the polygon. The inside of the bends has one vertex and the outside has two vertices. The second polygon, the I shape, has rounded ends, so this also cannot be converted to a path. The third polygon, mirrored C, has anyAngle segments, so this cannot be converted to a path.

To convert a polygon or rectangle to a path:

1. In the layout window, choose *Edit – Convert – To Path*.
2. On the canvas, click the polygon or rectangle you want to convert.

   The polygon or rectangle is converted to a path.
3. Continue clicking each polygon or rectangle that you want to convert.
4. Press `Esc` to cancel the command.

Alternatively, you can right-click a selected polygon or rectangle on the canvas and select *Convert To Path* from the shortcut menu.

© 2023 Cadence Design Systems, Inc.

You can partially select the point or edge of a polygon, which is then used as the starting point for deducing the path from a polygon or rectangle. This enables you to control the direction of the path.

(Virtuoso RF Option) The *Convert – To Path* command is not supported for curved shapes.

Related Topics

Object Conversion

Instance-to-Mosaic Conversion

Converting Instances to Mosaics

Converting Circles, Donuts, and Ellipses to Polygons

Object Alignment

You can align objects by using the *Edit – Quick Align* command or the *Edit – Advanced – Align* command. The two commands enable you to perform similar tasks but differ in their usage. You can perform the following tasks by using these commands:

- Align objects with a point
- Align objects with an edge
- Align objects with a layer
- Align objects with a target axis

Both the commands work in pre- and post-selection modes.

The *Quick Align* command enables you to align an edge, a centerline, or a point to another edge, centerline, or point, whereas the *Align* command aligns an object bounding box to the bounding box of another object.

The *Quick Align* command enables you to quickly align objects in a more interactive manner than the *Align* command. In comparison to the *Align* command, the *Quick Align* command provides the following additional features:

- Enables smart snapping of edges, centerlines, and points within the specified aperture.
- Enables cyclic highlighting of the reference and target objects within the aperture distance from the pointer by using the spacebar.

© 2023 Cadence Design Systems, Inc.

- Enables alignment with respect to reference or target objects across the hierarchy.
- Supports the spacing rules defined in the technology file.
- Enables alignment by stretching partially selected objects.

(Virtuoso RF Option) The *Quick Align* command supports alignment of curved objects - polygons and paths. You can set references on non-curved edges of a curved object but not on a curved edge or midpoint of a curved edge. You can set target on any aspect of a curved object other than its curved edges.

Related Topics

Object Alignment by Using the Quick Align Command

Object Alignment by Using the Align Command

Object Alignment by Using the Quick Align Command

With the *Quick Align* command, you can use objects, instances, and groups as the reference set and also use them as the target to which the reference set is to be aligned. The reference and target object can be the same object in the *Quick Align* command. By using the command, you can select the edges, centerlines, and points on objects in an instance or a group, which may be present at any level in the hierarchy.

The *Quick Align* command supports design-rule-driven (DRD) *Enforce* mode. You can specify the DRD rules that need to be enforced when the *Quick Align* command is active, by using the *DRD Options* form accessed from the *Options - DRD Edit* menu.

 Video

For details on how to use the smart snapping of ruler in the quick align command, see the video *How to Use Smart Snapping of Ruler in Quick Align Command*.

Related Topics

DRD Options Form

Object Highlighting When Quick Align is Active

Customizing Quick Align Settings

Quick Align Functionality in Full and Partial Selection Modes

© 2023 Cadence Design Systems, Inc.

Aligning Edges and Centerlines with a Target Edge in Full Selection Mode

Aligning Edges and Centerlines with a Target Edge in Partial Selection Mode

Alignment When Reference and Target Are Set on the Same Edge

Aligning Points with a Target Point

Aligning Objects with a Target Axis

Object Highlighting When Quick Align is Active

The Quick Align snapping mechanism uses different object highlighting schemes to set the reference and select the target. You can use the QAHoldTimeMilliSec environment variable to specify the time required to hold the pointer over an object. If the pointer remains stationary over an object for longer than the specified time, the potential points, edges, and centerlines in the object are highlighted.

The following sections demonstrate the results of the *Quick Align* command in various scenarios with pre-selection of objects. Similar results are obtained with post-selection of objects, except that in post-selection, the command remains active and the objects do not remain selected after the alignment.

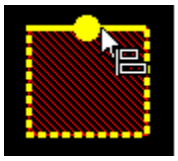
Reference Highlighting

Reference highlighting scheme is dependent on the following factors:

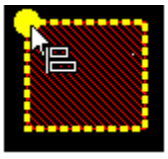
- Selection mode, whether Full Select or Partial Select
- Edge + point mode or edge-only mode

Additionally, highlighting of reference is different after the reference is set.

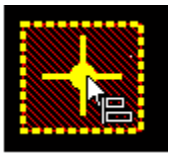
The following figure shows pre-select highlighting when the selection mode is Full Select and edge + point is active.



Edge + Center Point



Vertex

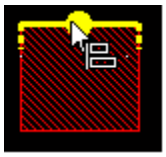


Nexus

© 2023 Cadence Design Systems, Inc.

Nexus is the midpoint of the given shape. In case of a polygon, nexus is the center of its bounding box.

The following figure shows pre-select highlighting when the selection mode is Partial Select and edge + point is active.



Edge + Center Point



Vertex



Nexus

The following figure shows pre-select highlighting in the edge-only mode.

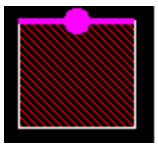


Full Select

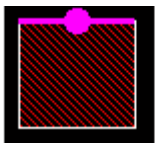


Partial Select

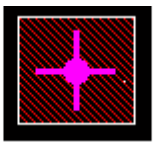
The following figure shows highlighting after the reference is set for both Full and Partial selection modes. Highlighting is same for nexus in both cases.



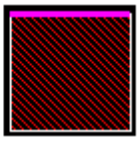
Full Select Edge + Center Point



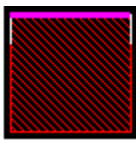
Partial Select Edge + Center Point



Nexus



Full Select Edge Only



Partial Select Edge Only

The reference point is discarded when the `Backspace` key is pressed.

© 2023 Cadence Design Systems, Inc.

Target Object Highlighting

The following figure shows highlighting for the target.



Edge Center *Edge* *Center Point* *Nexus* *Vertex*

If quick align is invoked during edit-in-place (EIP), you can snap the target at a level higher than the current edit level.

Related Topics

Object Alignment by Using the Quick Align Command

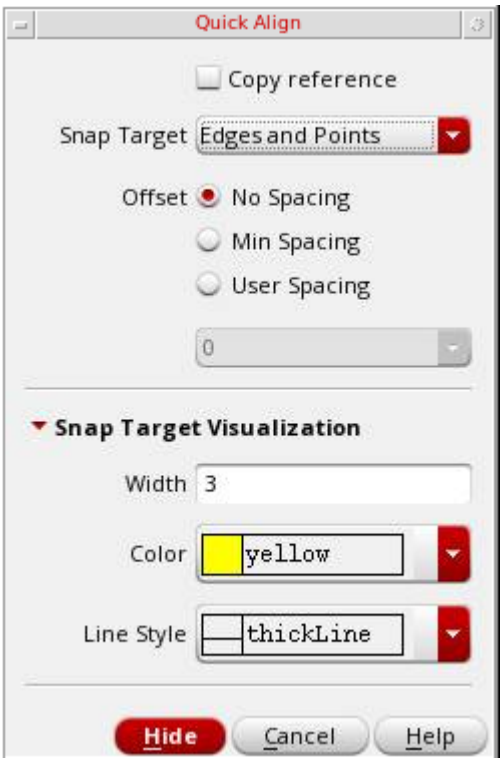
Customizing Quick Align Settings

To customize the *Quick Align* settings:

1. In the layout window, choose *Edit – Quick Align*.
2. Press `F3`.

© 2023 Cadence Design Systems, Inc.

The Quick Align form opens.



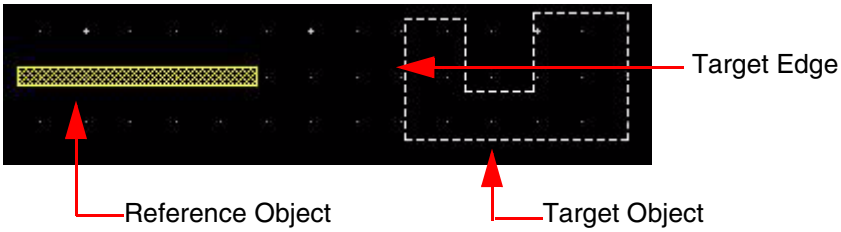
3. Select the *Copy reference* check box to create a copy of the objects during alignment.
4. Set *Snap Target* as *Edges and Points*, *Edges*, or *Points*.
5. Select the *No Spacing*, *Min Spacing*, or *User Spacing* option to set up the spacing to be used between the reference set and the target edge, centerline, or point. You can also access these options from the *Spacing* submenu on the *Quick Align* shortcut menu.

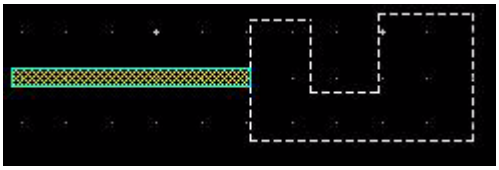
   The spacing of the reference object layer from the target determines the spacing of the other objects in the reference set from the target.

   The following figures show alignment results when the *No Spacing*, *Min Spacing*, and *User Spacing* options are selected while aligning the reference object to the target

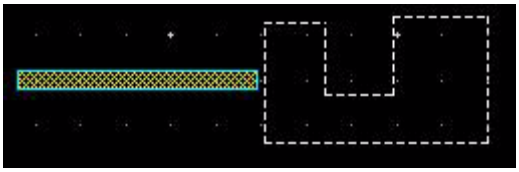
© 2023 Cadence Design Systems, Inc.

edge. The reference object layer is `Poly` and that of the target is `Oxide`. The `minSpacing` value specified in the technology file between the two layers is `0.3`.

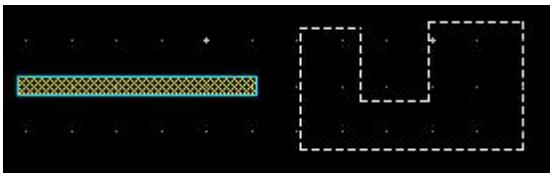




No Spacing



Min Spacing

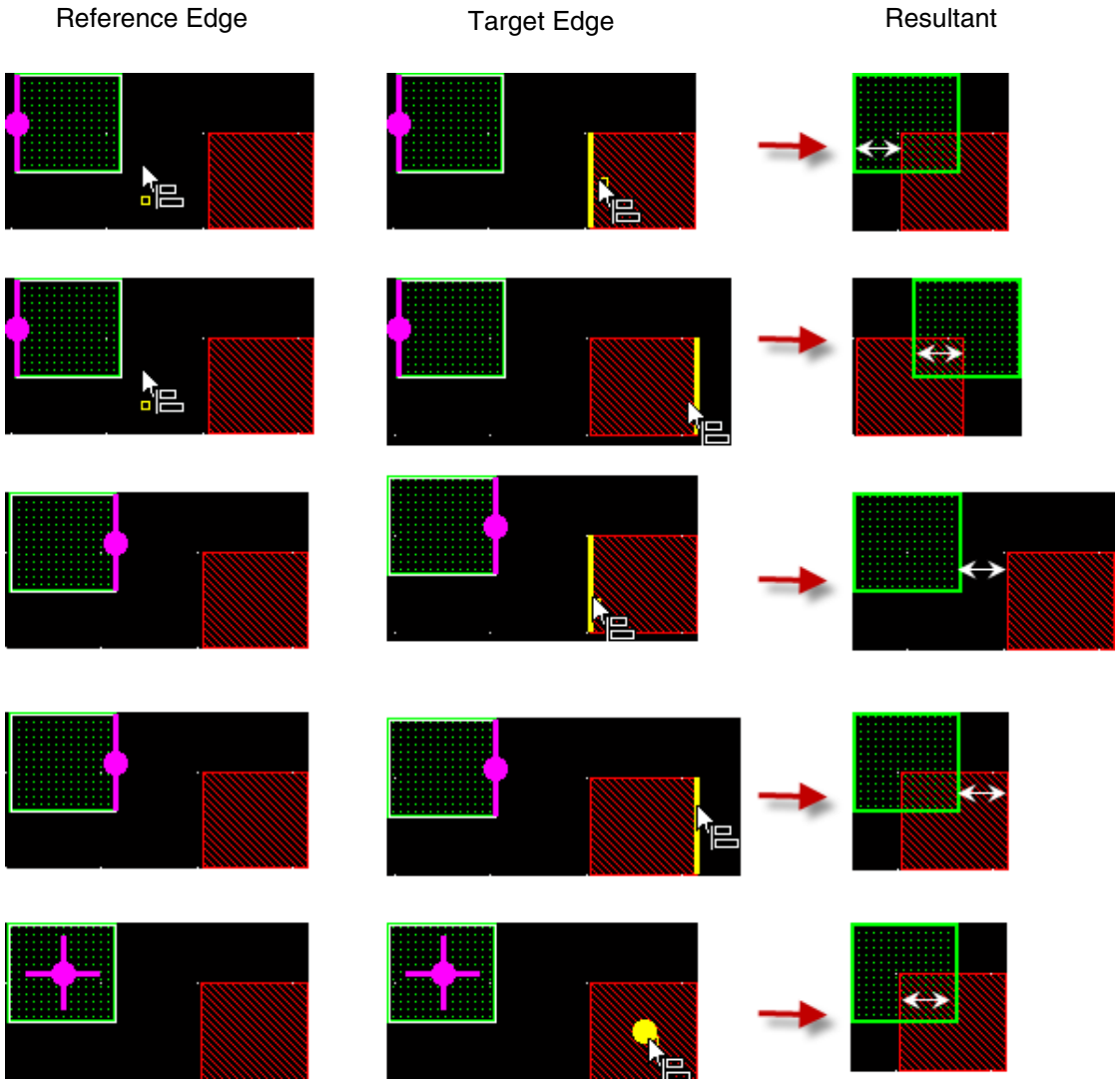


User Spacing = `1.0`

You can specify both positive and negative values in the *User Spacing* field. The following figures depict the examples of resultants with different reference and target edges.

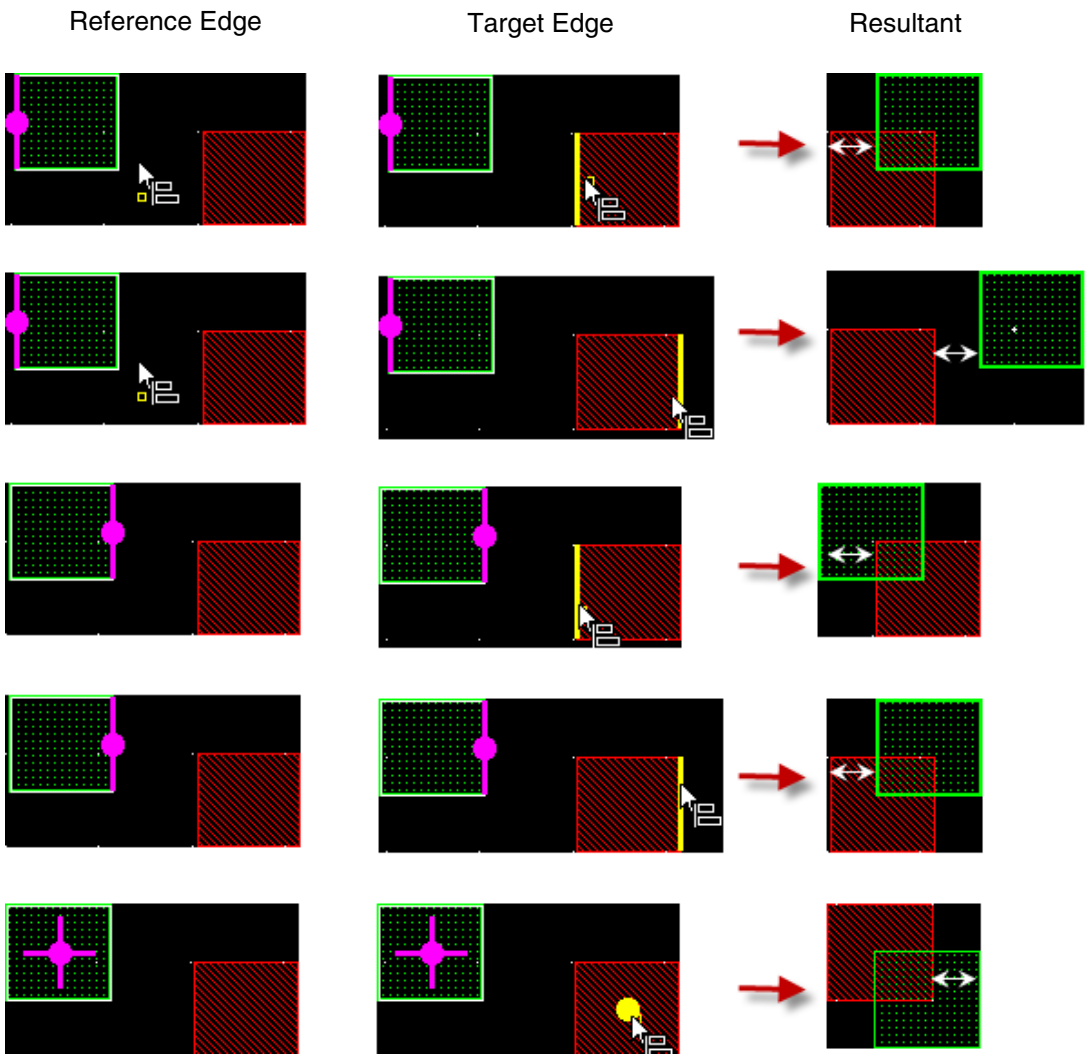
© 2023 Cadence Design Systems, Inc.

- Positive *User Spacing*



© 2023 Cadence Design Systems, Inc.

- Negative *User Spacing*



6. Set the *Width*, *Color*, and *Line Style* of the highlight edge and point in the *Snap Target Visualization* section.

© 2023 Cadence Design Systems, Inc.

Some samples are shown below.



Width = 0, Color = yellow, Line Style = thickline
This is the default setting.



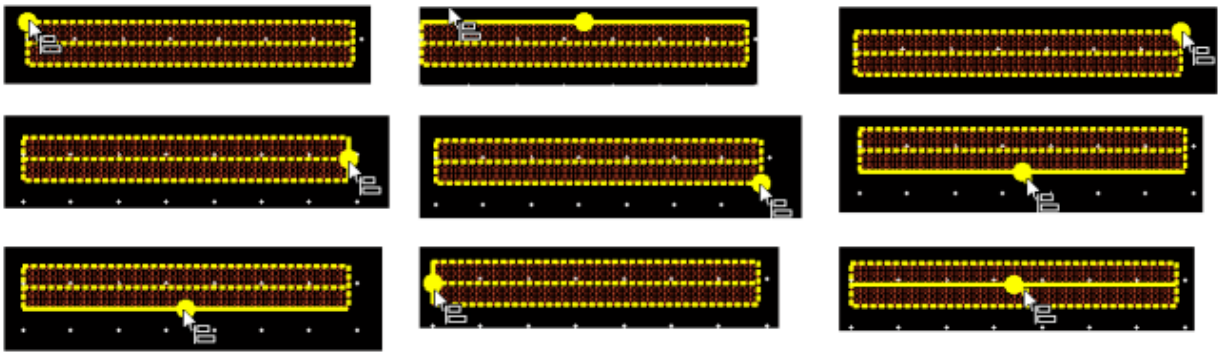
Width = 1, Color = red, Line Style = dashed



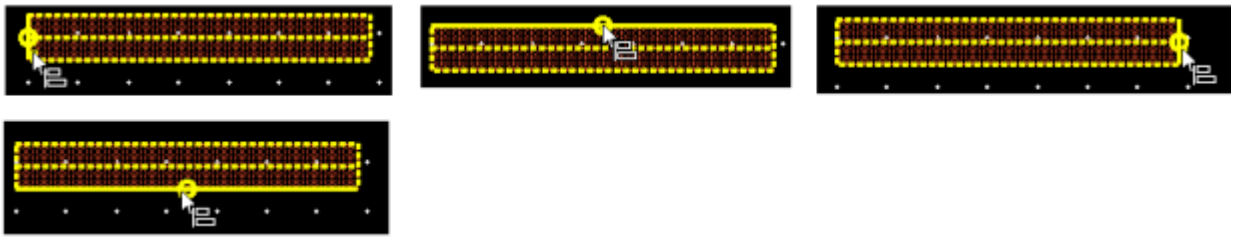
Width = 2, Color = magenta, Line Style = solid

The following figures show some of the points, edges, and centerline highlights in a path.

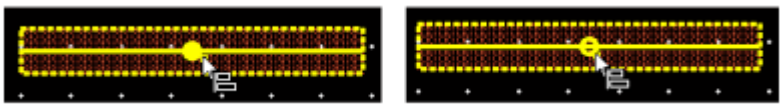
Highlighted Edge and Points



Highlighted Edges Only



Highlighted Centerlines

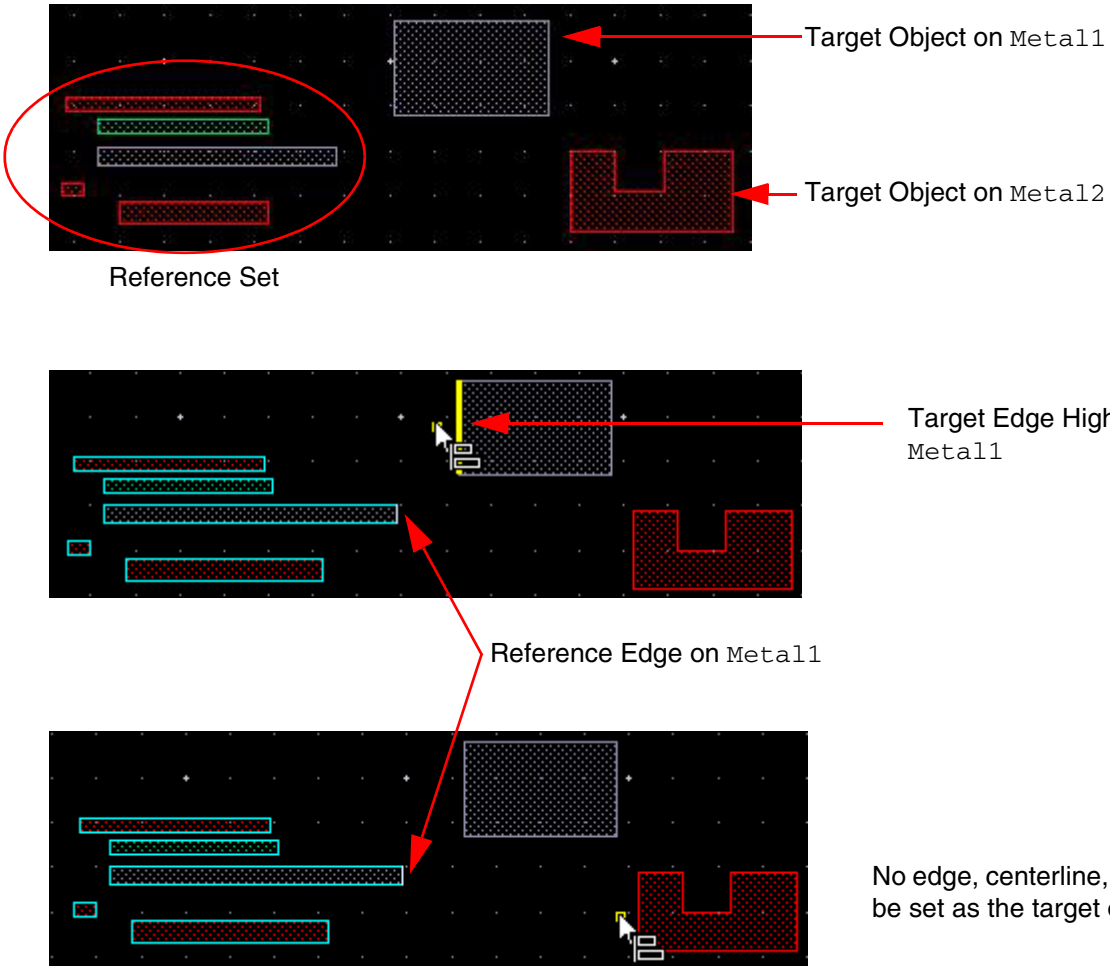


© 2023 Cadence Design Systems, Inc.

Use the highlighted points and edges to select the required reference and target edges, centerlines, or points.

7. Select the required options from the *Quick Align* shortcut menu, which is invoked when you right click on the canvas. Select the *Snap to targets in reference layer* option.

   The following example illustrates a scenario when the *Snap to targets in reference layer* option is selected. The reference set comprises objects on `Metal1`, `Metal2`, and `Metal3` layers. The target objects exist on `Metal1` and `Metal2` layers. If the reference edge is on the `Metal1` object, edges, centerlines, and points on only the `Metal1` target object can be selected for aligning the reference set. Similarly, if the reference edge is on a `Metal2` object, edges, centerlines, and points on only the `Metal2` target object can be selected for aligning the reference set.

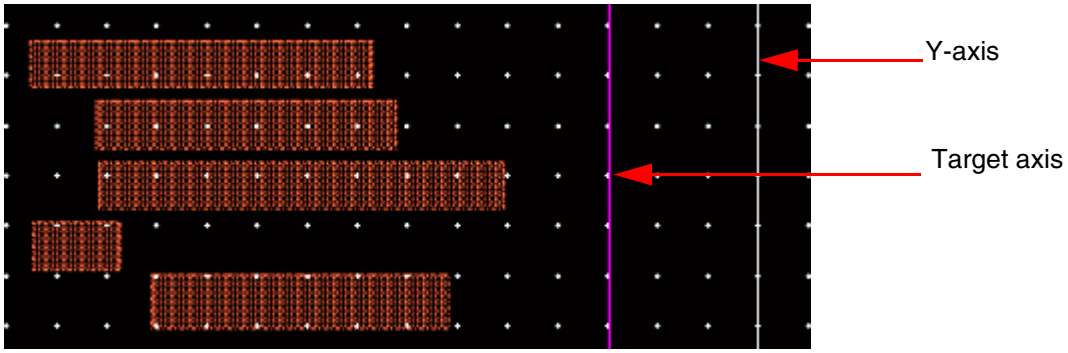


8. Select the *Snap to Edges* option on the *Quick Align* shortcut menu if you want to highlight only the edges and centerlines in the reference set and target object.

© 2023 Cadence Design Systems, Inc.

9. To align the reference objects to a target axis, select *Vertical* or *Horizontal* from the *Target Axis* submenu on the *Quick Align* shortcut menu. Then, click on the design display area.

   The target axis is created at the point of click.



   You can create the target axis, horizontal or vertical, before or after creating a selection set or setting a reference point.

   You can press the `A` + `H` bindkey to display horizontal target axis and the `A` + `V` bindkey to display the vertical target axis.

When snapping labels using the *Quick Align* command, snapping happens only to the origin of the labels.

Related Topics

Quick Align Form

Quick Align Shortcut Menu

Object Alignment by Using the Quick Align Command

Quick Align Functionality in Full and Partial Selection Modes

The following table summarizes the behavior of the *Quick Align* command in full and partial selection modes. The behavior is the same for pre- and post-selected objects.

Action	Result in Full Selection Mode	Result in Partial Selection Mode
Aligning Edges and Centerlines		

© 2023 Cadence Design Systems, Inc.

Action	Result in Full Selection Mode	Result in Partial Selection Mode
Click to select the target edge	All the objects in the reference set *move* and align with the target edge. The relative spacing amongst the reference set objects is maintained.	All the objects in the reference set *stretch* and align with the target edge. The relative spacing amongst the reference set objects is maintained.
Double-click to select the target edge	All the objects in the reference set *move* and align with the target edge. The relative spacing amongst the reference set objects is lost.	All the objects in the reference set *stretch* and align with the target edge. The relative spacing amongst the reference set objects is lost.
Aligning Points		
Click to select the target point	All the objects in the reference set *move*. The reference point coincides with the target point. The relative spacing amongst the reference set objects is maintained.	All the objects in the reference set *stretch* and align with the target point. The reference point coincides with the target point. The relative spacing amongst the reference set objects is maintained.
Double-click to select the target point	All the objects in the reference set *move*. Points on all the objects in the reference set coincide with the target point. The relative spacing amongst the reference set objects is lost.	All the objects in the reference set *stretch* and align with the target point. Points on all the objects in the reference set coincide with the target point. The relative spacing amongst the reference set objects is lost.

Related Topics

Quick Align Form

Object Alignment by Using the Quick Align Command

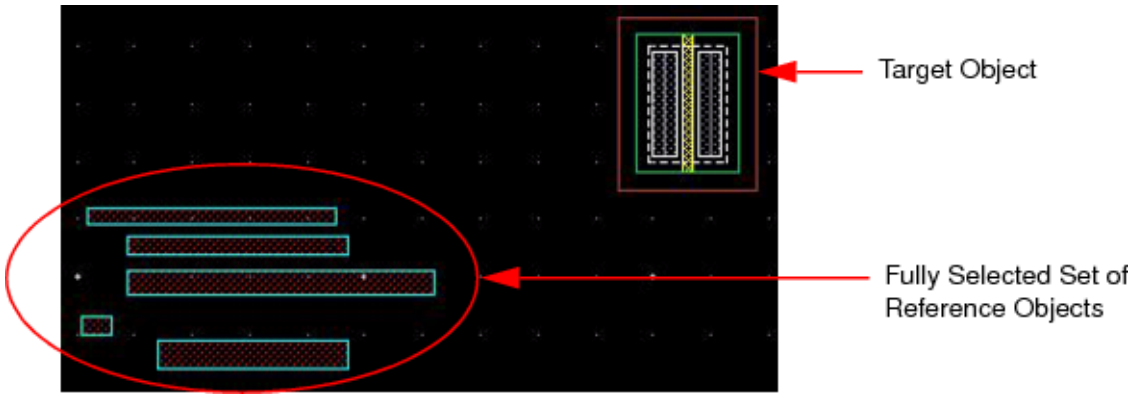
Aligning Edges and Centerlines with a Target Edge in Full Selection Mode

You can align edges and centerlines with a target point using full or partial selection mode.

© 2023 Cadence Design Systems, Inc.

To align edges of objects in a reference set with a target edge or centerline, in full selection mode:

1. On the *Options* toolbar in the layout window, ensure the *Full Select* button is on. If not, press `F4` to enable it.
2. Select the reference set objects.



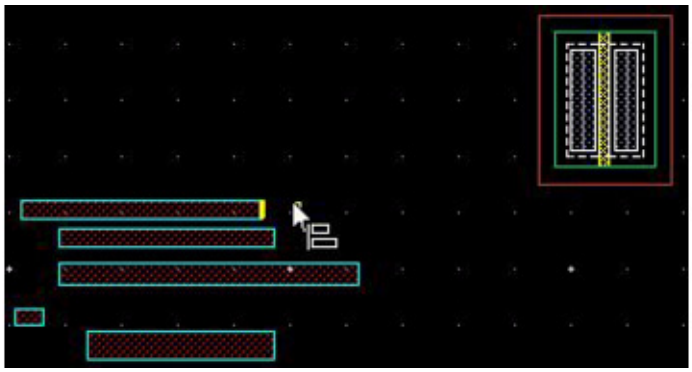
3. Select *Edit – Quick Align* to invoke the *Quick Align* command.
4. Press `F3` or right-click and select *Options* from the Quick Align shortcut menu.

   The Quick Align form appears.
5. Set the options as required.
6. On the canvas, move the pointer around the selected reference set.

   The potential points, edges, and centerlines in the objects in the reference set are highlighted. By default, points, edges, and centerlines highlight. To highlight only the edges and centerlines, right-click and select *Snap to Edges* from the *Quick Align* shortcut menu.
7. Click to set a highlighted edge as the reference edge.

© 2023 Cadence Design Systems, Inc.

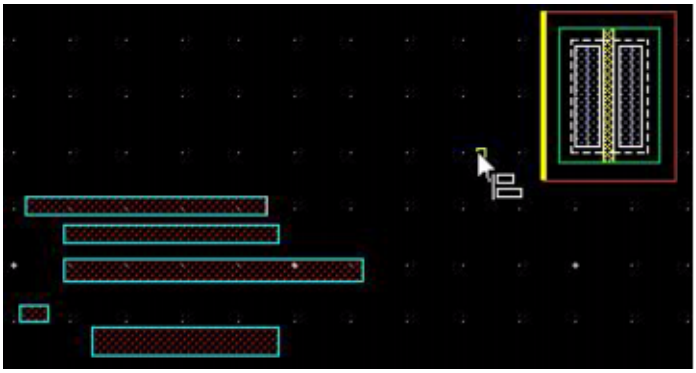
The reference edge must be part of the selected reference set.



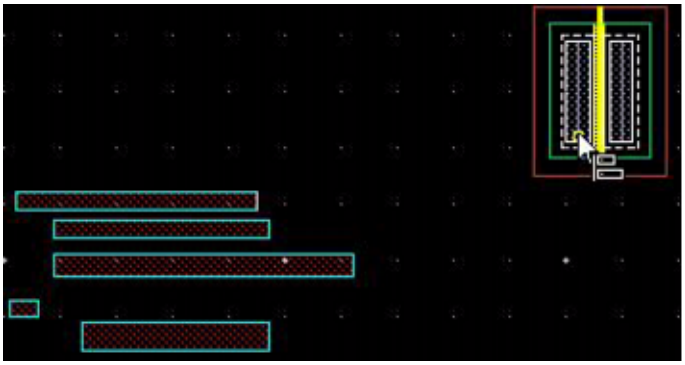
8. Move the pointer close to the target object.

© 2023 Cadence Design Systems, Inc.

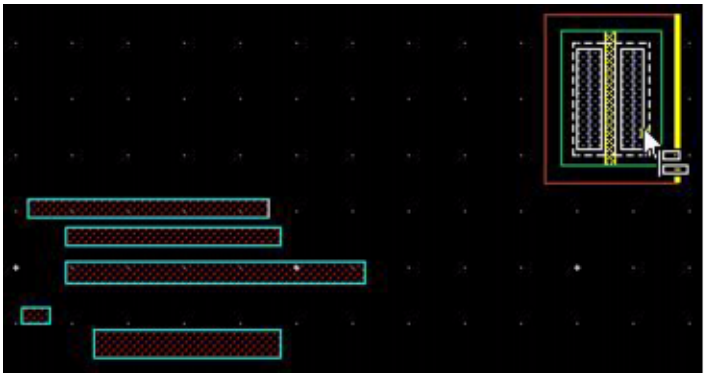
The potential edges and centerlines parallel to the reference edge highlight. The figures below show the potential edges and centerline highlighted when the pointer is moved close to the target object.



Potential Target Edge 1 Highlighted



Potential Centerline Highlighted



Potential Target Edge 2 Highlighted

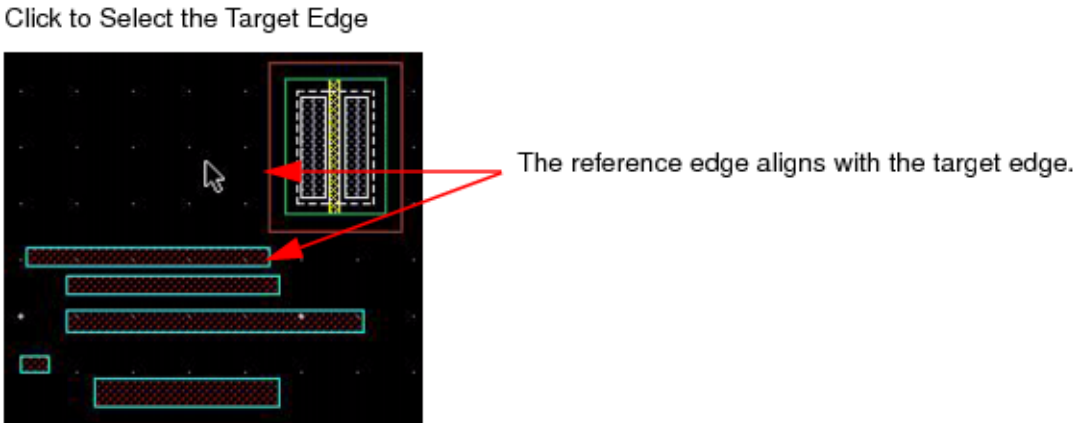
9. Do one of the following to select a highlighted target edge:

   a. Click the highlighted target edge.

   The object with the reference edge moves to align with the target edge. All the objects in the reference set move by the same distance. The relative spacing amongst the reference set objects is maintained.

© 2023 Cadence Design Systems, Inc.

The figure below shows the alignment of the reference set when the potential target edge is set as the target edge.

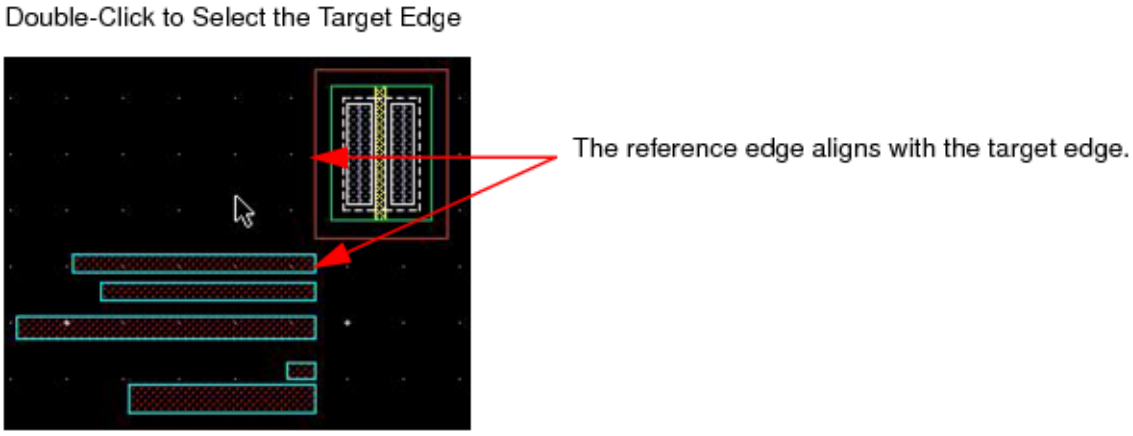


The pre-selected reference set objects remain selected after the alignment.

b. Double-click the highlighted target edge.

The object with the reference edge moves to align with the target edge. All the objects in the reference set move by varying distances to align with the target edge, as the reference edge. The relative spacing amongst the reference set objects is lost.

The figure below shows the alignment of the reference set when the potential target edge is set as the target edge.



The pre-selected reference set objects remain selected.

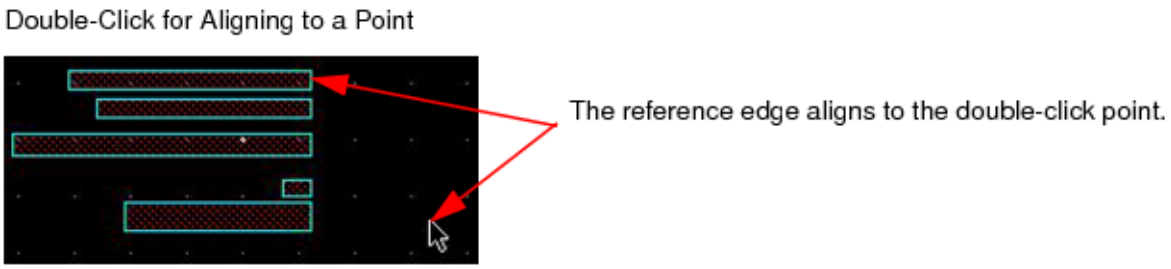
c. Double-click at a point when no target edge is highlighted.

The object with the reference edge moves to align with the point. All the objects in the reference set move by varying distances to align with the target point, as the

© 2023 Cadence Design Systems, Inc.

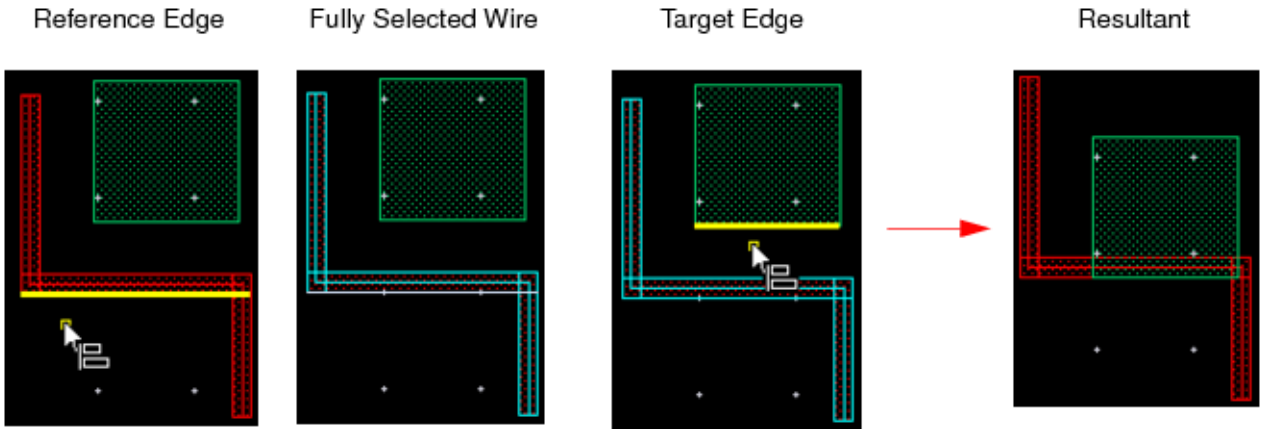
reference edge. The relative spacing amongst the reference set objects is not maintained.

The figure below shows the alignment of the reference set to the point.



The pre-selected reference set objects remain selected.

If you quick align an edge of a wire (pathSeg) in full selection mode to a target edge, the wire moves to align the selected edge to the target edge, as shown below.



Related Topics

Quick Align Form

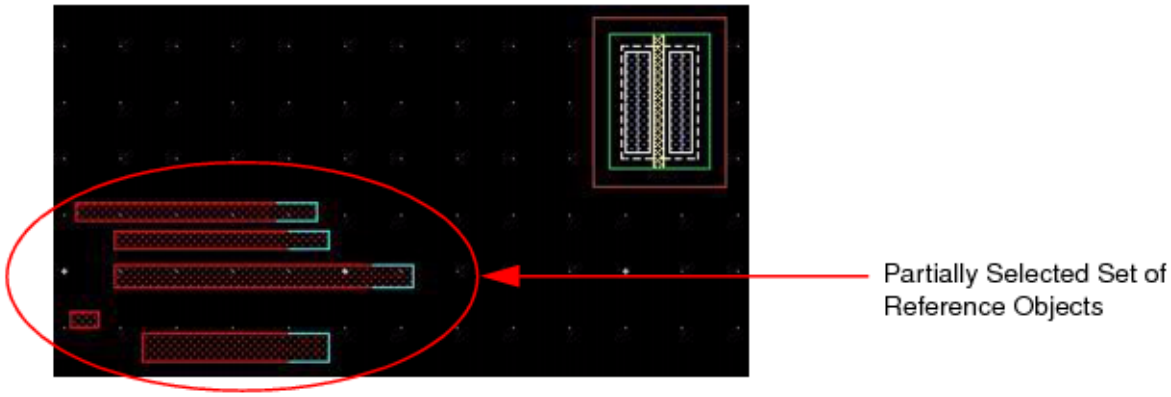
Object Alignment by Using the Quick Align Command

Aligning Edges and Centerlines with a Target Edge in Partial Selection Mode

To align edges of objects in a reference set with a target edge or centerline, in partial selection mode:

© 2023 Cadence Design Systems, Inc.

1. On the *Options* toolbar in the layout window, ensure that the *Partial Select* button is on. If not, press `F4` to enable it.
2. Partially select the set of reference objects. Use `Shift` + click to add to the selection and `Ctrl` + click to remove from the selection.



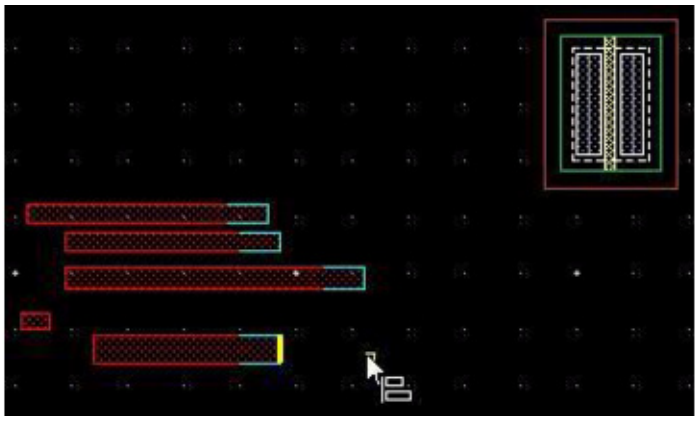
3. Select *Edit – Quick Align* to start the *Quick Align* command.
4. Press `F3` or right-click and select *Options* from the *Quick Align* shortcut menu.

   The Quick Align form appears.
5. Set the options as required.
6. On the canvas, move the pointer around the partially selected reference set.

   The partially selected objects determine the alignment axis. The potential points, edges, and centerlines, along one axis, in the objects in the reference set are highlighted. By default, points, edges, and centerlines are highlighted. To highlight only the edges and centerlines, right-click and select *Snap to Edges* from the *Quick Align* shortcut menu.
7. Click to set a highlighted edge as the reference edge.

© 2023 Cadence Design Systems, Inc.

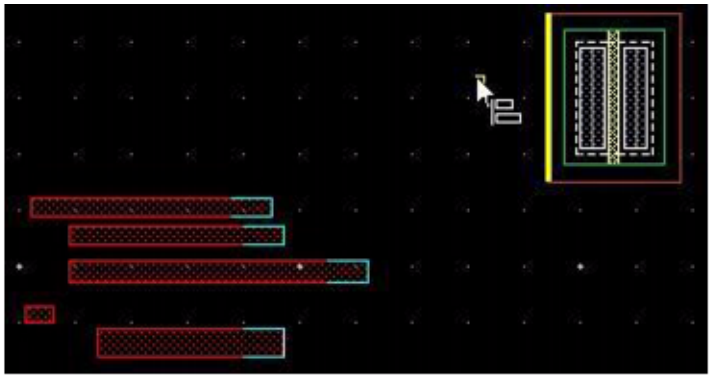
The reference edge must be part of the selected reference set.



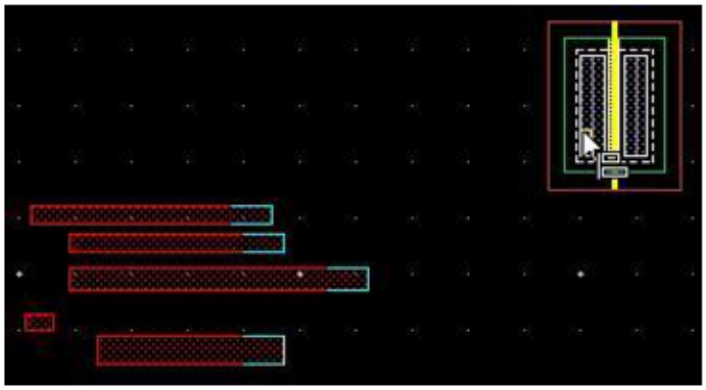
8. Move the pointer close to the target object.

© 2023 Cadence Design Systems, Inc.

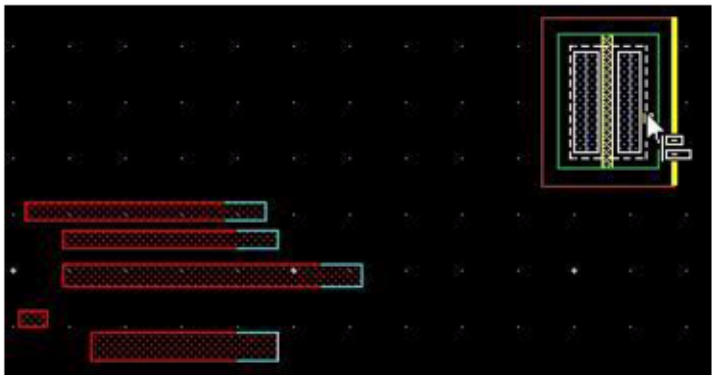
The potential edges and centerlines parallel to the reference edge highlight. The figures below show the potential edges and centerline highlighted when the pointer is moved around the target object.



Potential Target Edge 1 Highlighted



Potential Centerline Highlighted



Potential Target Edge 2 Highlighted

9. Do one of the following to select the highlighted target edge:

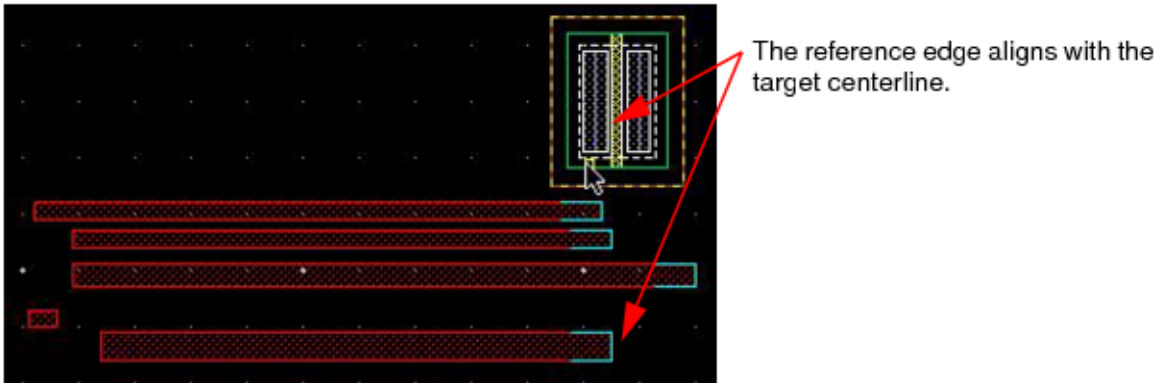
   a. Click the highlighted target edge.

      The object with the reference edge stretches to align with the target edge. All the objects in the reference set stretch by the same amount. The reference edge determines the stretch direction. The relative spacing amongst the reference set objects is maintained.

© 2023 Cadence Design Systems, Inc.

The figure below shows the alignment of the reference set when the potential target centerline is set as the target.

Click to Select the Target Centerline



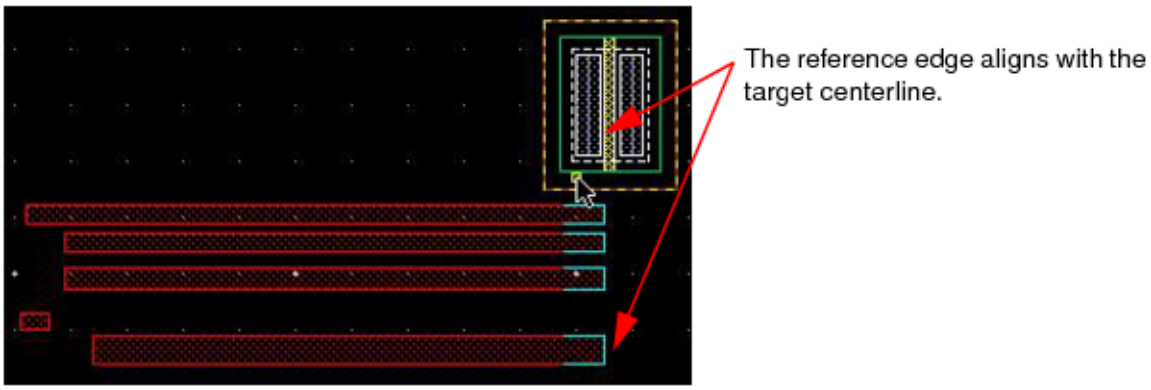
The pre-selected reference set objects remain selected after the alignment.

b. Double-click the highlighted target edge.

The object with the reference edge stretches to align with the target edge. All the objects in the reference set stretch by varying amounts to align with the target edge, as the reference edge. The reference edge determines the stretch direction. The relative spacing amongst the reference set objects is lost.

The figure below shows the alignment of the reference set when the potential target centerline is set as the target.

Double-Click to Select the Target Centerline



The pre-selected reference set objects remain selected after the alignment.

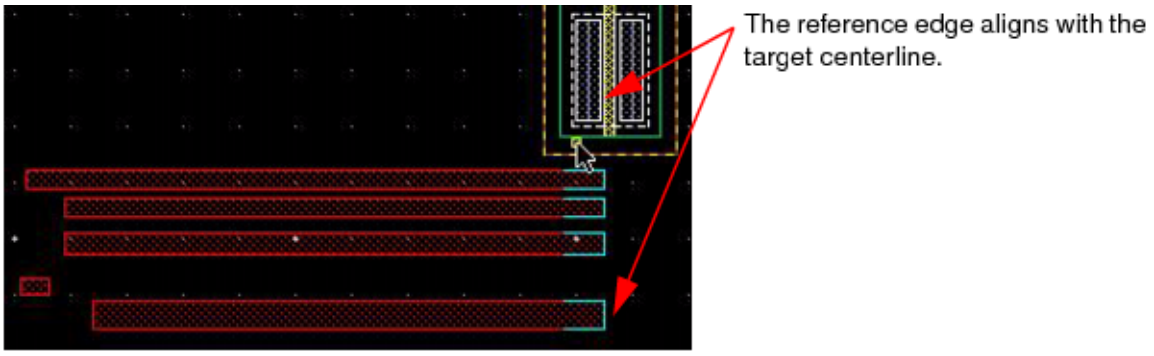
c. Double-click at a point when no target edge is highlighted.

The object with the reference edge stretches to align with the point. All the objects in the reference set stretch by varying amounts to align with the target point, as the

© 2023 Cadence Design Systems, Inc.

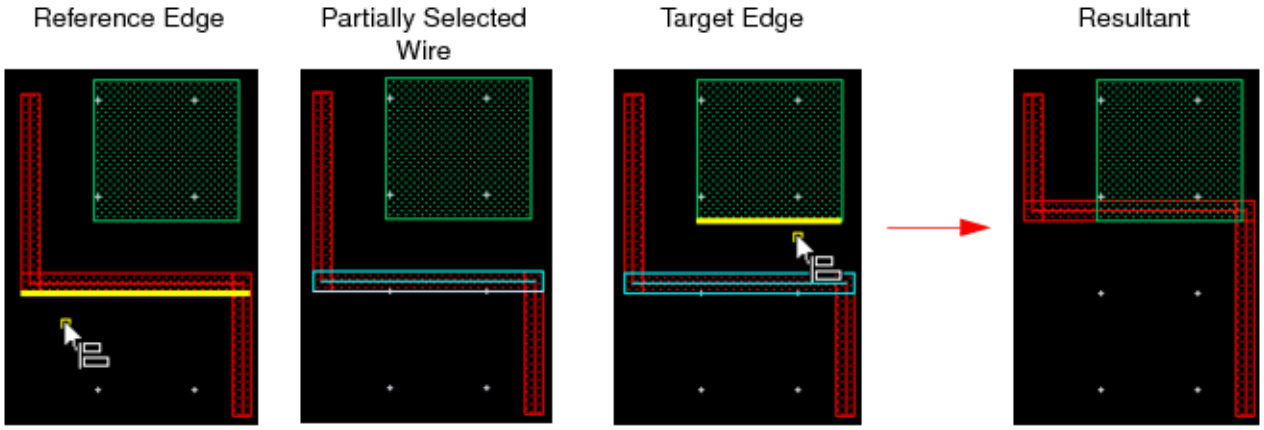
reference edge. The reference edge determines the stretch direction. The relative spacing amongst the reference set objects is not maintained.

The figure below shows the alignment of the reference set to the point.



The pre-selected reference set objects remain selected after the alignment.

If you quick align an edge of a wire (pathSeg) in partial selection mode to a target edge, the selected edge moves to align with the target edge and the remaining pathSegs of the wire stretch to move along with the aligned pathSeg, as shown below.



Related Topics

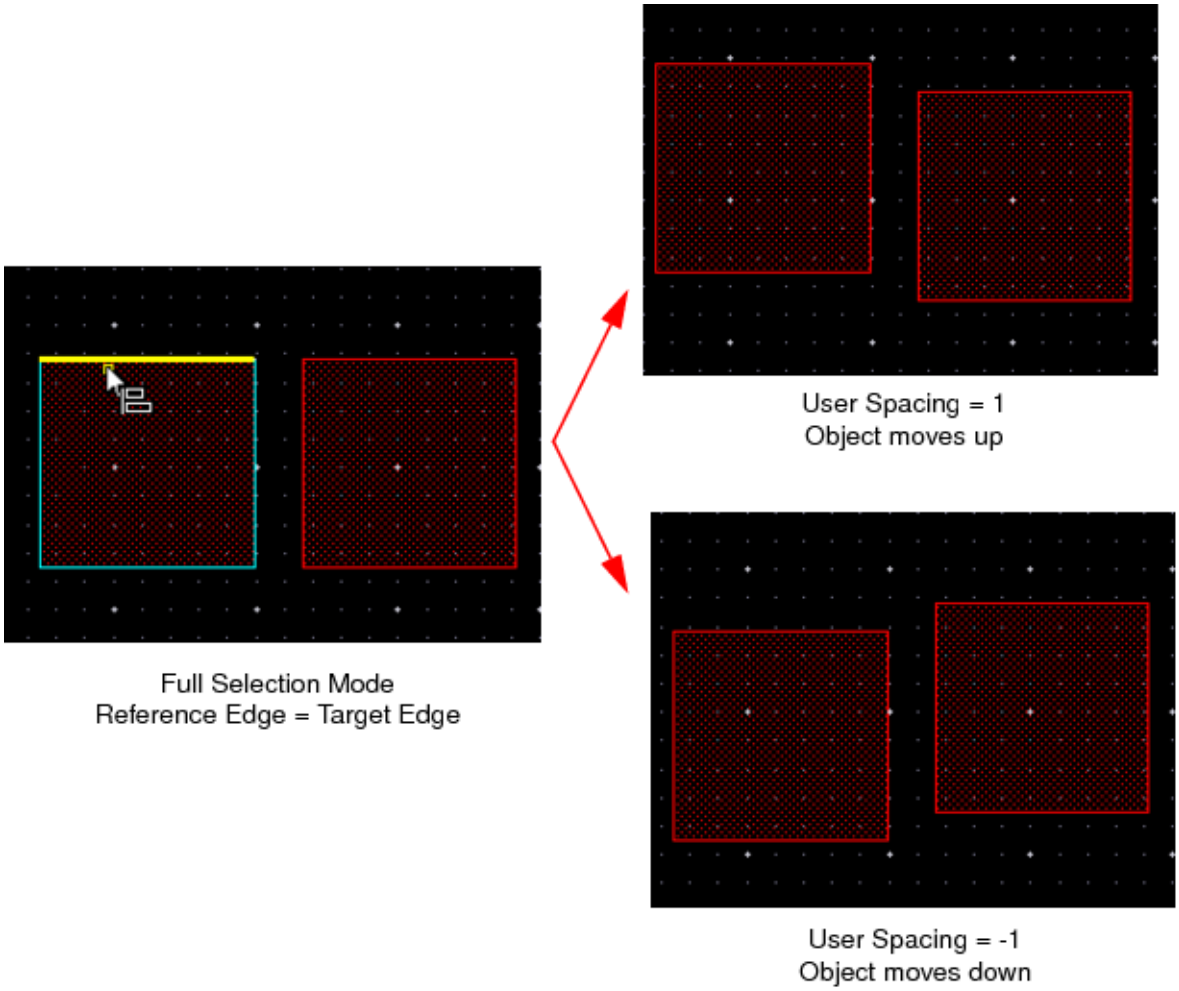
Quick Align Form

Object Alignment by Using the Quick Align Command

© 2023 Cadence Design Systems, Inc.

Alignment When Reference and Target Are Set on the Same Edge

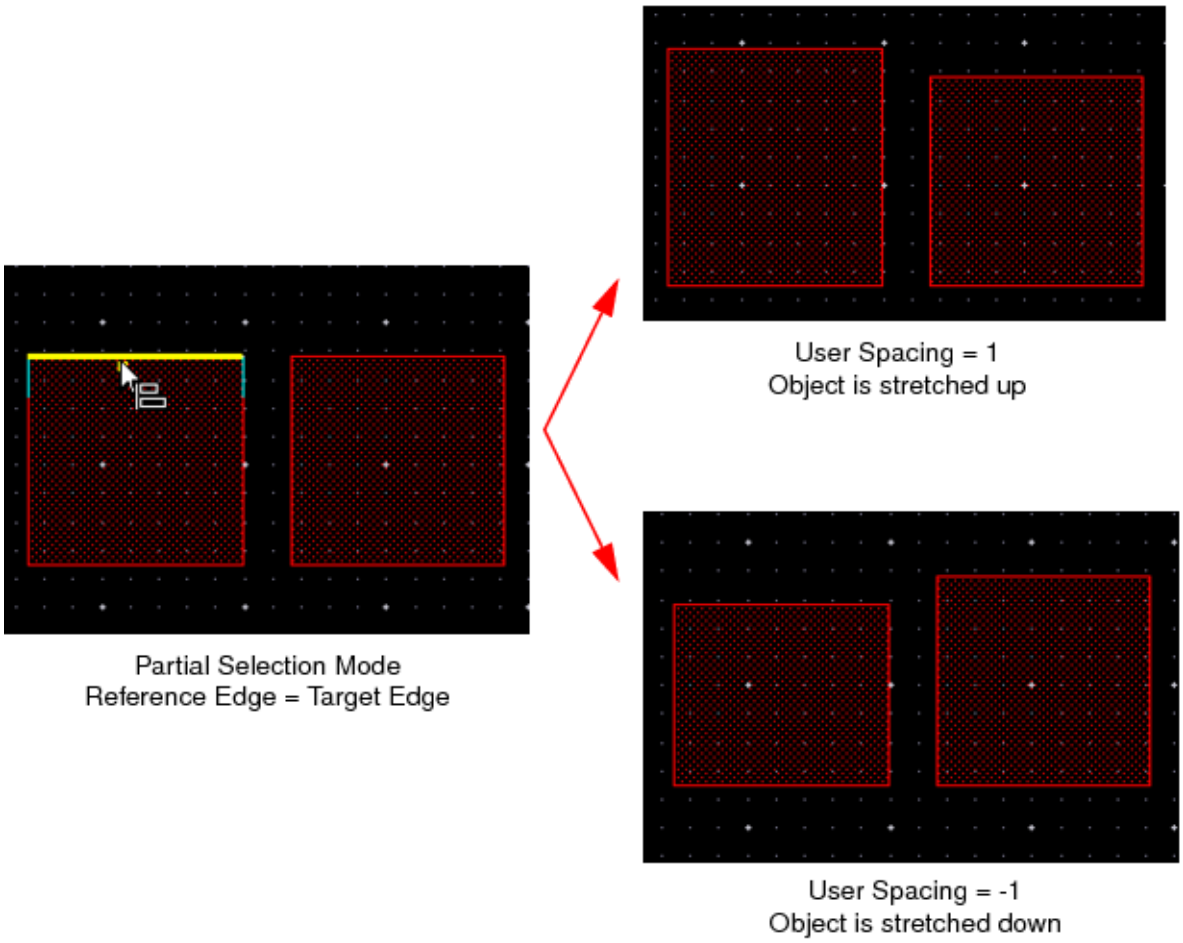
In full selection mode, if you set an edge of an object as the reference and then set the same edge as the target, the object moves based on the value specified in the *User Spacing* field. If you specify a positive value, the object moves in the positive direction. If you specify a negative value, the object moves in the negative direction. If *User Spacing* is `0`, no change is made to the object.



In partial selection mode, if you set an edge of an object as the reference and then set the same edge as the target, the selected edge stretches based on the value specified in the *User Spacing* field. If you specify a positive value, the edge is stretched in the positive

© 2023 Cadence Design Systems, Inc.

direction. If you specify a negative value, the edge is stretched in the negative direction. If *User Spacing* is `0`, no change is made to the object.



Related Topics

Quick Align Form

Object Alignment by Using the Quick Align Command

Aligning Points with a Target Point

You can align points to target points, edges, and centerlines. You can align points in full or partial selection mode.

Aligning Points with aTarget Point in Full Selection Mode

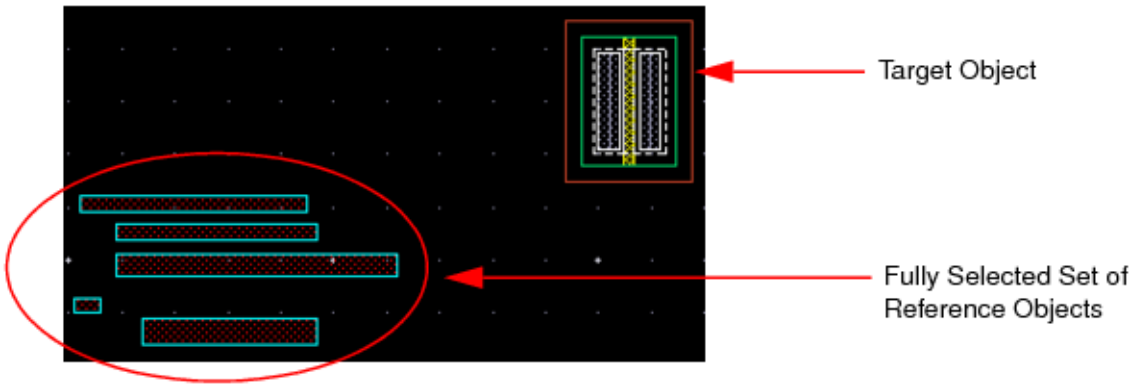
© 2023 Cadence Design Systems, Inc.

Aligning Points with a Target Point in Partial Selection Mode

Aligning Points with aTarget Point in Full Selection Mode

To align points of objects in a reference set with a target point in full selection mode:

1. On the *Options* toolbar in the layout window, ensure that the *Full Select* button is on. If not, press `F4` to enable it.
2. Select the reference set objects.



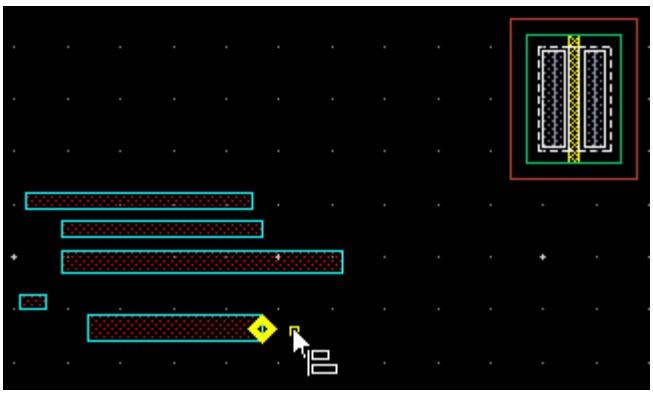
3. Choose *Edit – Quick Align* to start the *Quick Align* command.
4. Press `F3` or right-click and select *Options* from the Quick Align shortcut menu.

   The Quick Align form appears.
5. Set the options as required.
6. On the canvas, move the pointer around the selected reference set.

   The potential points, edges, and centerlines in the objects in the reference set highlight. By default, points, edges, and centerlines highlight. Ensure that the *Snap to Edges and Points* option is selected in the Quick Align shortcut menu.

© 2023 Cadence Design Systems, Inc.

7. Click to select a reference point.



Notice that the selected reference point is marked by a diamond shape.

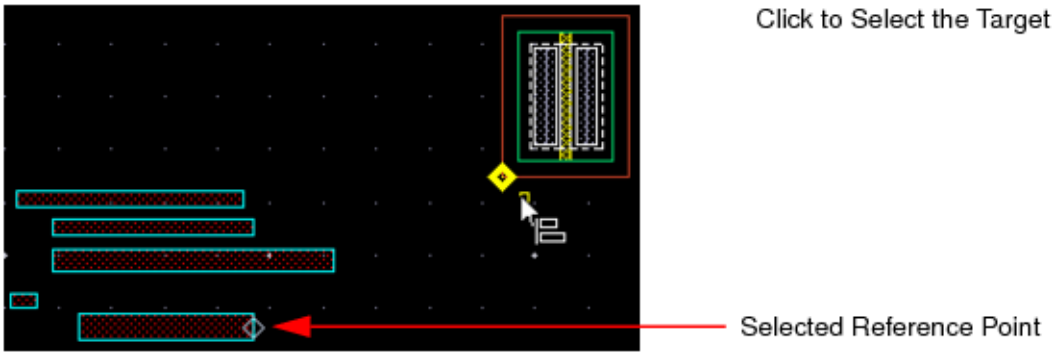
8. Move the pointer close to the target object.

The potential points and edges in the target object appear.

9. Do one of the following to select a highlighted target point:

   a. Click the highlighted target point.

   The object with the reference point moves to align with the selected target point. All the objects in the reference set move by the same distance. The relative spacing amongst the reference set objects is maintained.





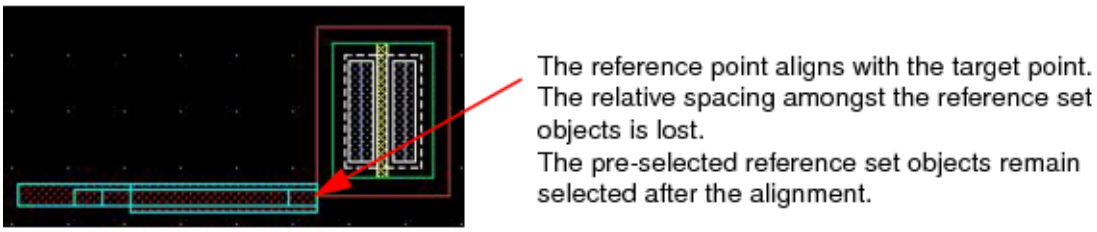
© 2023 Cadence Design Systems, Inc.

The pre-selected reference set objects remain selected after the alignment.

b. Double-click the highlighted target point.

The object with the reference point moves to align with the selected target point. All the objects in the reference set move to align with the reference and target point. The relative spacing amongst the reference set objects is lost.

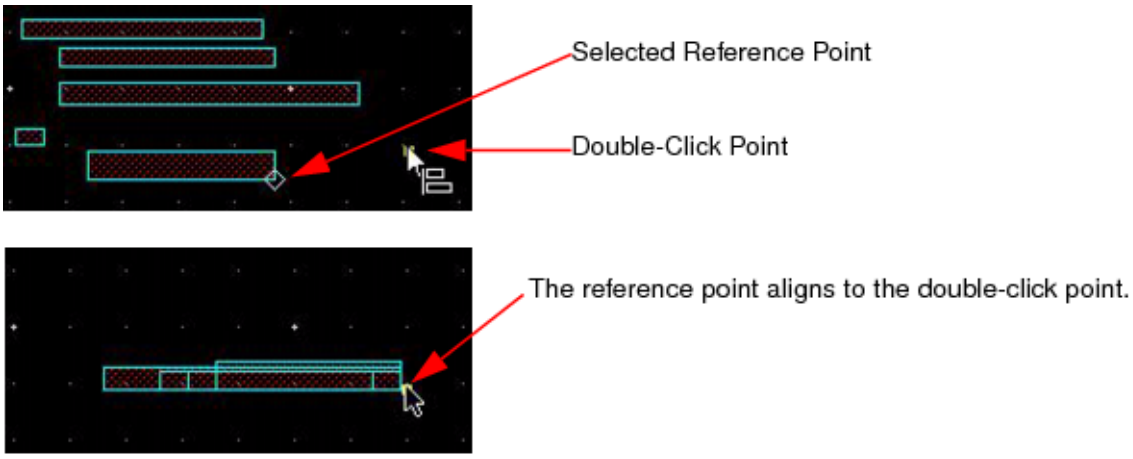
The points on the objects in the reference set that align with the reference and target points are relative to the selected reference point.



You need to double-click at a point when no target point is highlighted.

The object with the reference point moves to align with the point. All the objects in the reference set move to align with the reference and the double-clicked point. The relative spacing amongst the reference set objects is lost.

The points on the objects in the reference set that align with the reference and target points are relative to the selected reference point.



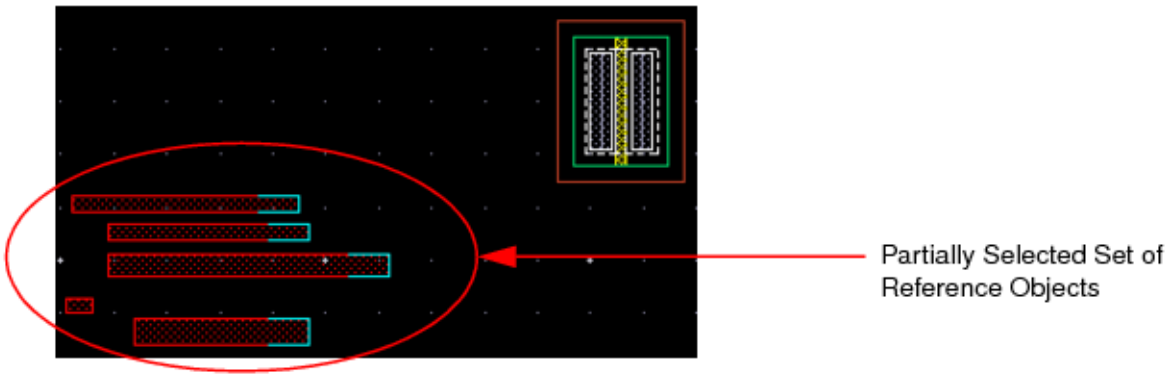
The pre-selected reference set objects remain selected after the alignment.

Aligning Points with a Target Point in Partial Selection Mode

To align points of objects in a reference set with a target point, in partial selection mode:

© 2023 Cadence Design Systems, Inc.

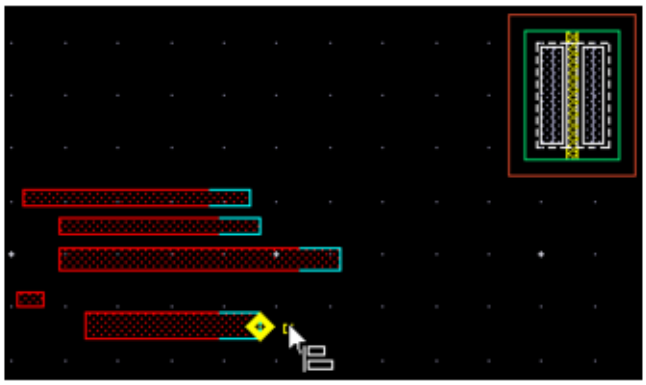
1. On the *Options* toolbar in the layout window, ensure that the *Partial Select* button  is on. If not, press `F4` to enable it.
2. Partially select the set of reference objects. Use `Shift` + click to add to the selection and `Ctrl` + click to remove from the selection.



3. Select *Edit – Quick Align* to start the *Quick Align* command.
4. Press `F3` or right-click on the canvas and select *Options* from the *Quick Align* shortcut menu.

   The *Quick Align* form appears.
5. Set the options as required.
6. On the canvas, move the pointer around the partially selected reference set.

   The partially selected objects determine the alignment axis. The potential points, edges, and centerlines in the objects in the reference set along one axis are highlighted. By default, points, edges, and centerlines are highlighted. Ensure that the *Snap to Edges and Points* option is selected from the *Quick Align* shortcut menu.
7. Click a highlighted point to set it as the reference point.



© 2023 Cadence Design Systems, Inc.

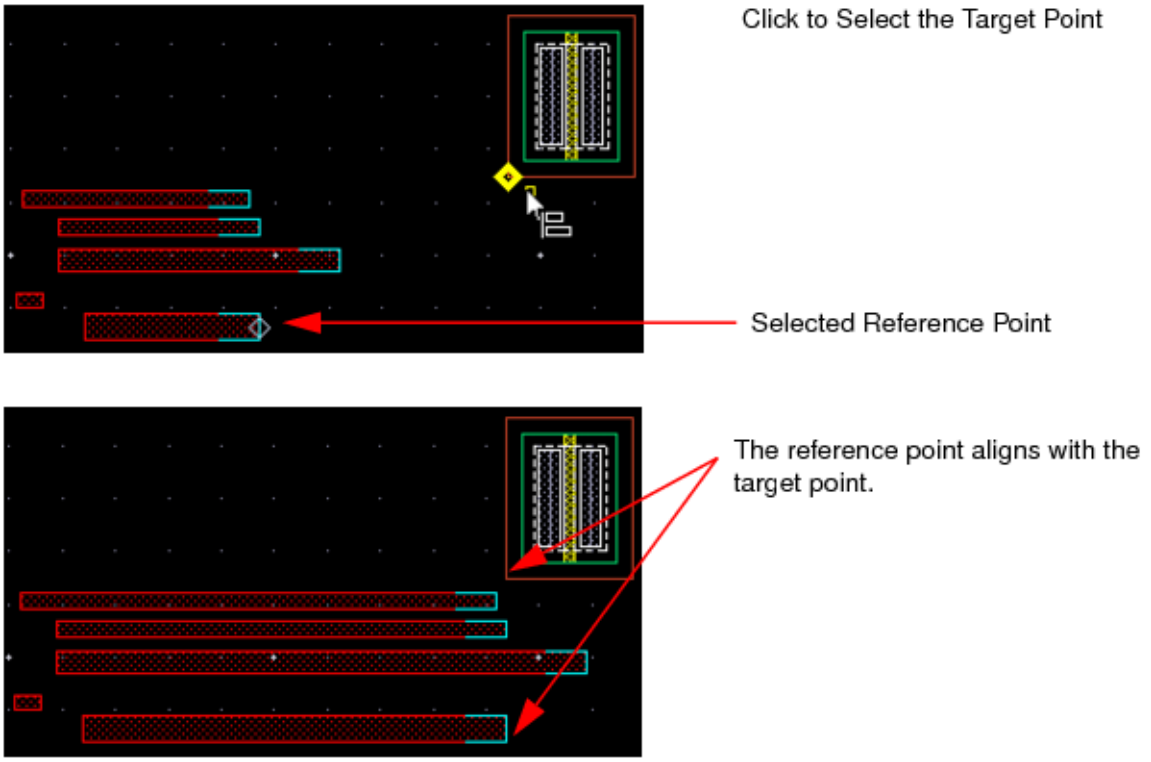
8. Move the pointer close to the target object.

   The potential points and edges in the target object are highlighted. Notice that the selected reference point is marked by a diamond shape.

9. Do one of the following to select a highlighted target point:

   a. Click the highlighted target point.

      The object with the reference point stretches to align with the selected target point. All the objects in the reference set stretch by the same amount. The relative spacing amongst the reference set objects is maintained.

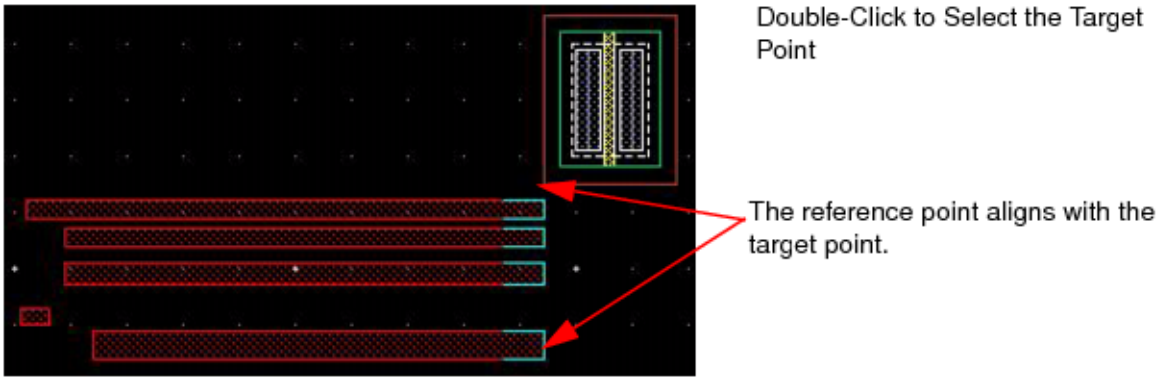


      The pre-selected reference set objects remain selected after the alignment.

   b. Double-click to select the target point.

© 2023 Cadence Design Systems, Inc.

The object with the reference point stretches to align with the target point. All the objects in the reference set stretch by varying amounts to align with the target point. The relative spacing amongst the reference set objects is not maintained.



The pre-selected reference set objects remain selected after the alignment.

Related Topics

Quick Align Form

Object Alignment by Using the Quick Align Command

Aligning Objects with a Target Axis

The support for aligning objects in a reference set to a target axis is available only in post-selection mode. You can align the selected objects using full or partial selection.

Aligning Objects with a Target Axis in Full Selection Mode

Aligning Objects with a Target Axis in Partial Selection Mode

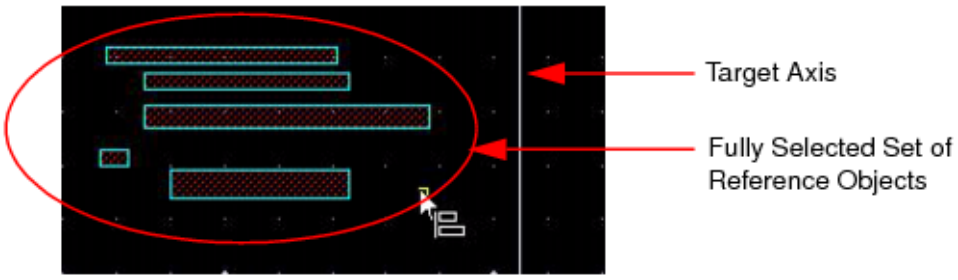
Aligning Objects with a Target Axis in Full Selection Mode

To align objects in a reference set with a target axis in full selection mode:

1. On the *Options* toolbar in the layout window, ensure that the *Full Select* button  is on. If not, press `F4` to enable it.
2. Select *Edit – Quick Align* to start the *Quick Align* command.
3. Right-click on the canvas and select *Vertical* or *Horizontal* from the *Target Axis* submenu on the *Quick Align* shortcut menu.

© 2023 Cadence Design Systems, Inc.

4. Click on the canvas to place the target axis.
5. Select the reference set objects.



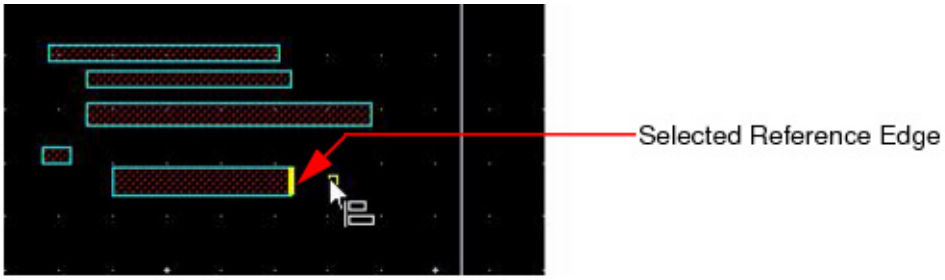
6. Move the pointer around the selected reference set.

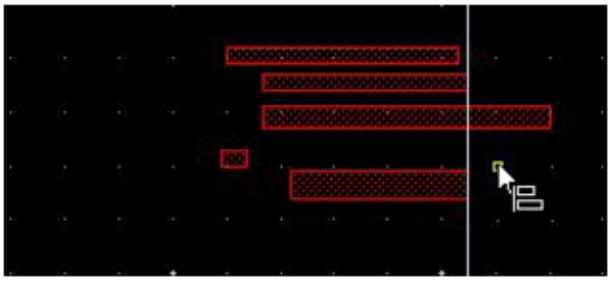
   The potential points, edges, and centerlines in the objects in the reference set are highlighted. By default, points, edges, and centerlines are highlighted. To highlight only the edges and centerlines, right-click on the canvas and select *Snap to Edges* from the *Quick Align* shortcut menu.

7. Do one of the following to align the reference set with the target axis:

   a. Click to select a reference edge, centerline, or point.

   The object with the reference edge moves to align with the target axis. All the objects in the reference set move by the same distance. The relative spacing amongst the reference set objects is maintained.



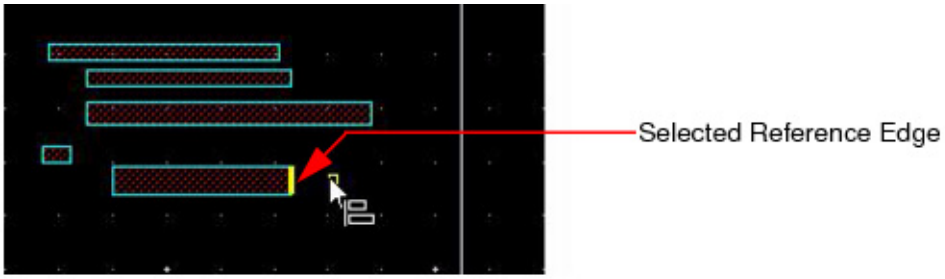


The reference edge aligns with the target axis.
The relative spacing amongst the reference set objects is maintained.
The post-selected reference set objects are de-selected after the alignment.

   b. Double-click to select a reference edge, centerline, or point.

© 2023 Cadence Design Systems, Inc.

The object with the reference edge moves to align with the target axis. All the objects in the reference set move by varying distances to align with the target axis, as the reference edge. The relative spacing amongst the reference set objects is lost.





The reference edge aligns with the target axis. The relative spacing amongst the reference set objects is lost.
The post-selected reference set objects are de-selected after the alignment.

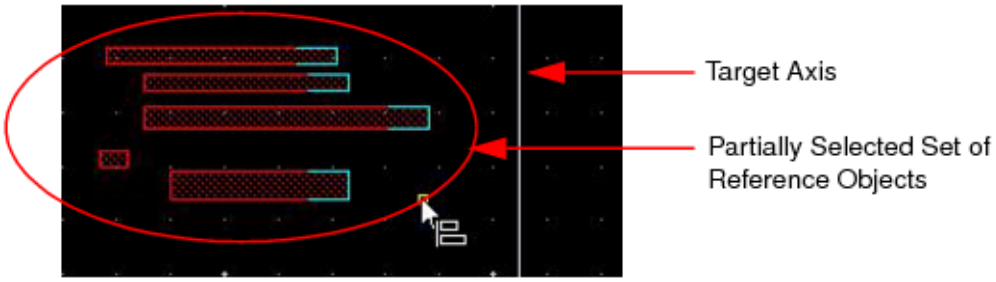
Aligning Objects with a Target Axis in Partial Selection Mode

To align objects in a reference set with a target axis in partial selection mode:

1. On the *Options* toolbar in the layout window, ensure that the *Partial Select* button  is on. If not, press `F4` to enable it.
2. Select *Edit – Quick Align* to start the *Quick Align* command.
3. Right-click on the canvas and select *Vertical* or *Horizontal* from the *Target Axis* submenu on the Quick Align shortcut menu.
4. Click on the canvas to place the target axis.

© 2023 Cadence Design Systems, Inc.

5. Partially select the set of reference objects. Use `Shift` + click to add to the selection and `Ctrl` + click to remove from the selection.



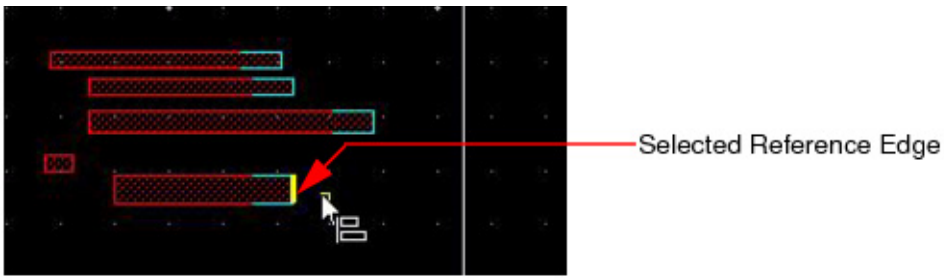
6. On the canvas, move the pointer around the selected reference set.

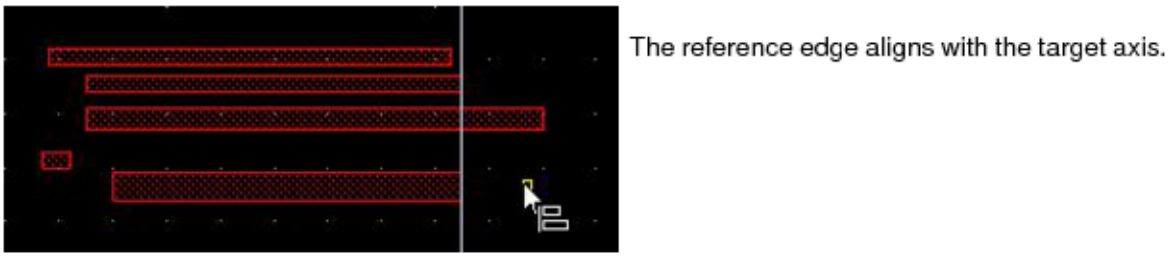
   The potential points, edges, and centerlines in the objects in the reference set highlight. By default, points, edges, and centerlines highlight. To highlight only the edges and centerlines, right-click and select *Snap to Edges* from the Quick Align shortcut menu.

7. Do one of the following to align the reference set with the target axis:

   a. Click to select a reference edge, centerline, or point.

      The object with the reference edge stretches to align with the target axis. All the objects in the reference set stretch by the same amount. The relative spacing amongst the reference set objects is maintained.



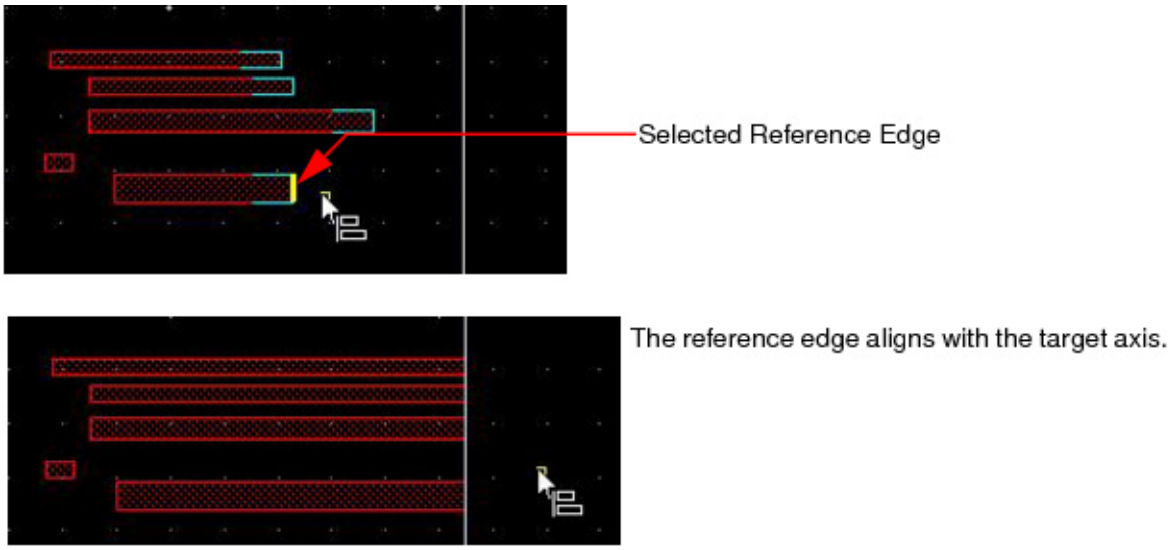


      The post-selected reference set objects are deselected after the alignment.

   b. Double-click to select a reference edge, centerline, or point.

© 2023 Cadence Design Systems, Inc.

The object with the reference edge stretches to align with the target axis. All the objects in the reference set stretch by varying amounts to align with the target axis, as the reference edge. The relative spacing amongst the reference set objects is lost.



The post-selected reference set objects are deselected after the alignment.

Related Topics

Object Alignment by Using the Quick Align Command

Object Alignment by Using the Align Command

The *Align* command lets you align objects or shapes in a cellview. You can align objects to either a point, another object, or a layer. You can also align a layer to another layer or other objects. When aligning objects in a chain, the entire chain gets aligned with the selected object. You can add objects to an already aligned group.

Caution

The Align command does not utilize spacing rules from the technology file and does not obey user-defined constraints with the exception of group constraints. As a result, rule and constraint violations can occur.

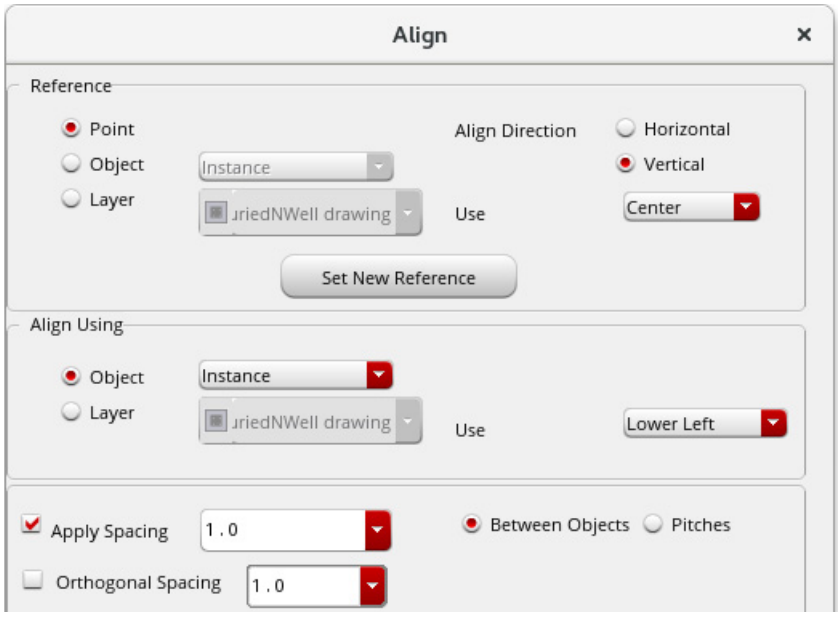
If nothing is selected in the cellview, the *Align* command prompts you to select one or more objects. You can area select or use the `Shift` + left mouse button to select additional

© 2023 Cadence Design Systems, Inc.

objects. Use the `Enter` key or double-click the mouse button to complete the command. To specify a different reference, use the *Set New Reference* button in the Align form. The layout editor prompts you to select a new reference to which you want to align objects.

You can quickly align an object by placing the pointer over the object and then pressing the `Enter` key or by double clicking. Conversely, to prevent an object from aligning when you press the `Enter` key or double-click, ensure that the pointer points at empty space on the design canvas. You can also use the `Enter` key or double-click for aligning multiple selected objects.

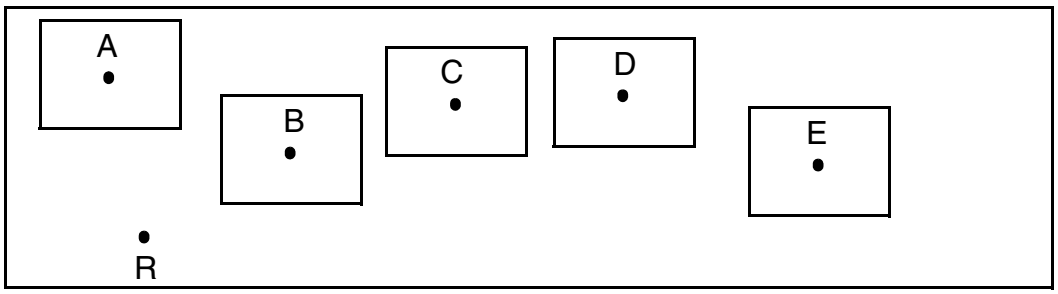
While aligning objects, you can specify spacing settings. In the Align form, you can set spacing as between objects or as pitches.



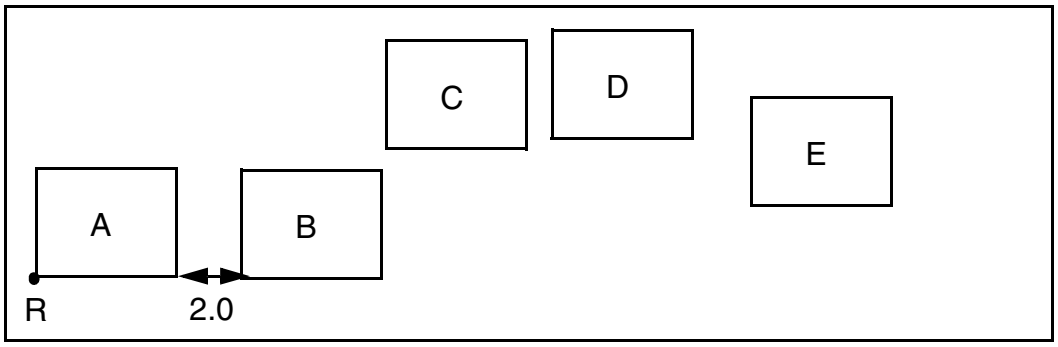
The *Between Objects* option specifies the distance, in user units, to be maintained between adjacent edges of objects to be aligned. The first object overlaps the selected reference point. The next object aligns at a distance specified in the *Apply Spacing* field from the adjacent edge of the previous object. If you specify a negative number, the aligned objects overlap.

© 2023 Cadence Design Systems, Inc.

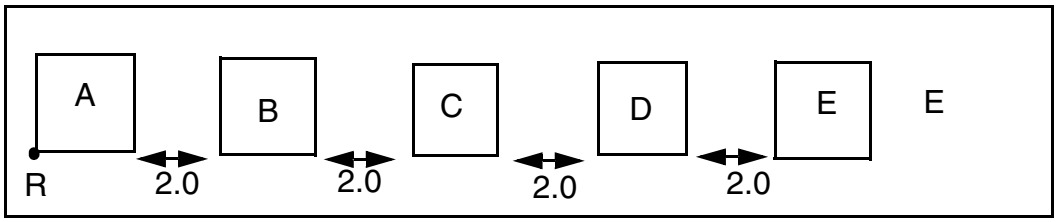
In the following example, point R is the selected reference point.



When the *Align Direction* is set to *Horizontal*, the *Use* field in the *Align Using* section is set to *Lower Left*, and the *Apply Spacing* value is specified as `2.0`, then when you apply the *Align* command, the lower-left point of the first rectangle, A, overlaps the reference point R. For aligning the next rectangle, B, the *Between Object* spacing distance of `2.0` is calculated from the outer edge of rectangle A to the lower-left point of rectangle B, in the horizontal direction, as shown in the following image.



The remaining rectangles are similarly aligned in the horizontal direction with the *Between Objects* spacing of 2.0, as shown in the following image.



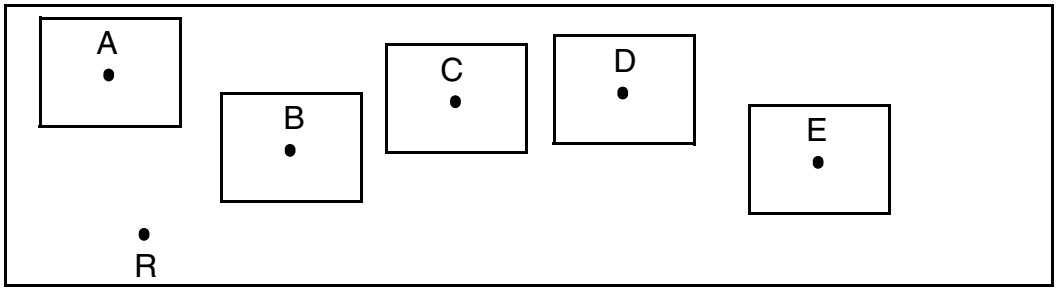
If the *Reference* section is set to *Object* or *Layer* and the *Apply Spacing* check box is selected, both *Between Objects* and *Pitches* options are disabled. However, the spacing of the aligned objects by default follows the *Between Objects* spacing behavior.

The *Pitches* option in the Align form specifies the distance, in user units, to be maintained across objects between the specified points (*Use* in the *Align Using* section). The first

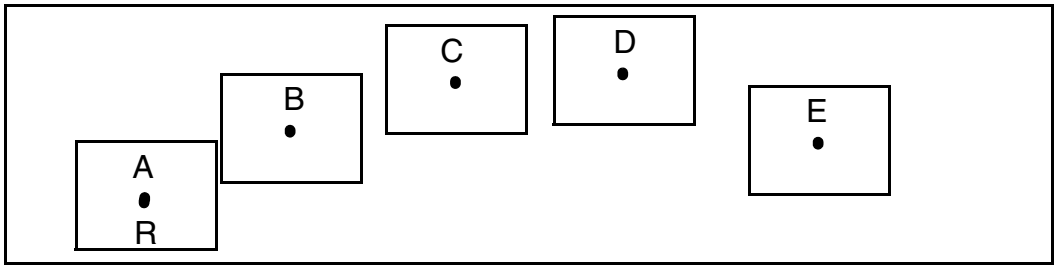
© 2023 Cadence Design Systems, Inc.

object overlaps the selected reference point. The next object aligns at a distance specified in the *Apply Spacing* field from the aligned point of the previous object. The *Pitches* option is available only if you select *Point* in the *Reference* section. The spacing distance is specified in the *Apply Spacing* field.

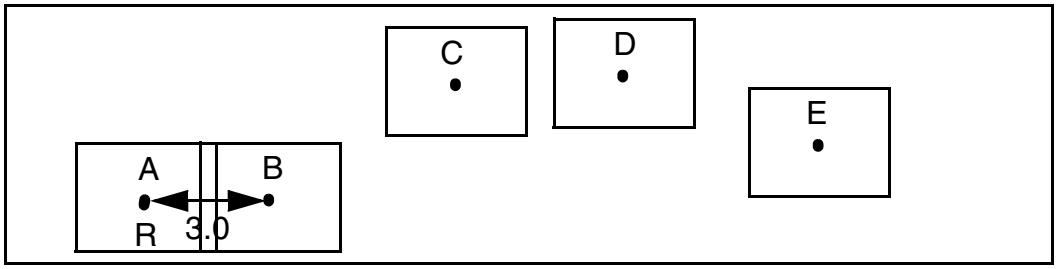
In the following example, point R is the selected reference point.



When the selected *Align Direction* is *Horizontal*, the *Use* cyclic field in the *Align Using* section is set to *Center Center*, and the *Pitches Apply Spacing* value is specified as 3.0, then when you apply the *Align* command, the center point of the first rectangle, A, overlaps the reference point R, as shown in the following image.

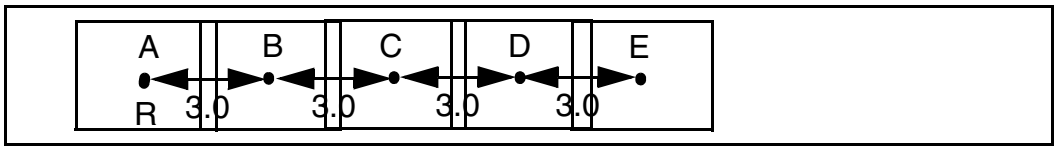


The center point of the next rectangle, B, then aligns to the center point of rectangle A at a distance of 3.0, in the horizontal direction, as shown in the following image.



© 2023 Cadence Design Systems, Inc.

Similarly, center points of subsequent objects align with the center point of the previous object at a distance of 3.0.



Related Topics

Align Form

Aligning Objects with a Point

Aligning Objects with a Reference Object

Aligning Objects with a Layer

Aligning a Layer with Another Layer

Aligning a Layer with Objects

Adding Objects with a Pre-Selected Aligned Group

Aligning Objects by Using the Align Toolbar

Aligning Objects with a Point

To align objects, you must always choose a reference point. The selected shapes align with the selected reference point as per the other options specified in the Align form. If you specify a spacing value for point alignment, the first object always overlaps the reference point and the spacing value is used to align the remaining objects. After alignment, all selected shapes are deselected.

To align objects with a point:

1. In the layout window, select *Edit – Advanced – Align*.

   The Align form opens.

2. Choose the *Align Direction* in the *Reference* section.
3. Choose *Point* in the *Reference* section to align objects to a point.

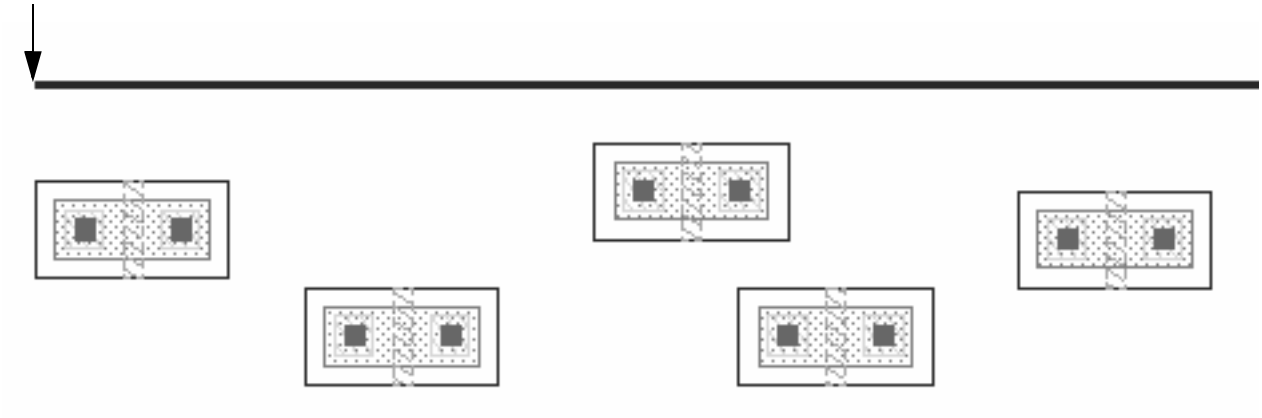
   You are prompted to point at a reference point for the alignment.

© 2023 Cadence Design Systems, Inc.

4. Click the point on the canvas or enter the coordinates of the point in the CIW. After you have chosen a point and an alignment direction, a pointer line appears indicating the beginning point of the alignment and the alignment direction. The following figure shows a sample pointer line when the direction is horizontal.

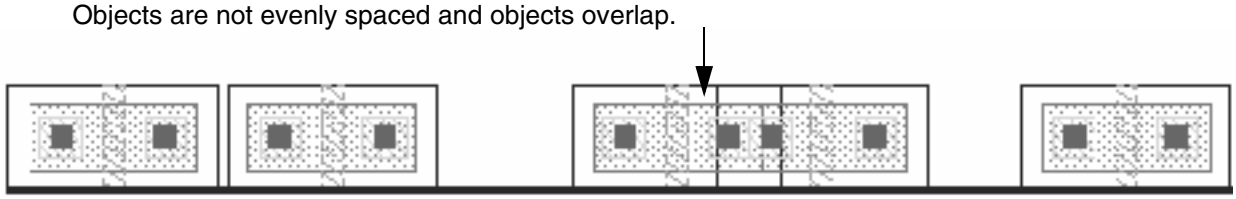
   You can use the *Set New Reference* button in the form to clear any existing reference point and specify a new reference point.

Pointer indicating beginning point and alignment direction.



5. In the *Align Using* section, set the object type of objects or layer to align with the point, object, or layer selected in the *Reference* section. For example, you can choose *Object* and set the object type as *Instance*.
6. From the *Use* list, select the edge of the object you want to align. For example, *Lower Left*.
7. To evenly space the objects in the specified direction, select the *Apply Spacing* check box and enter the value in user units by which you want to separate the objects. If you do not use *Apply Spacing*, the objects retain their original spacing. They are aligned, but not evenly spaced, as shown in the example below:

Objects are not evenly spaced and objects overlap.

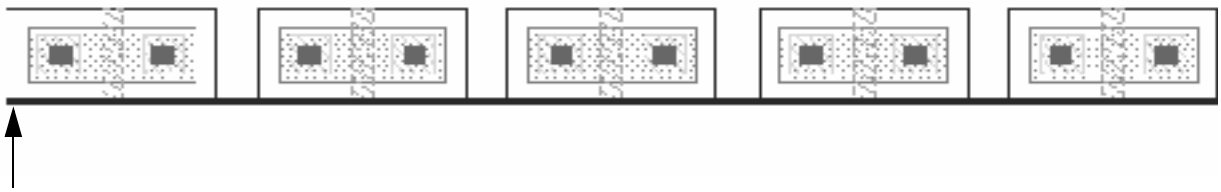


8. To space the objects perpendicular to the *Align Direction* as well, select the *Orthogonal Spacing* check box. This option is available only if *Apply Spacing* is selected.

© 2023 Cadence Design Systems, Inc.

9. Choose *Between Objects* or *Pitches*, depending on the spacing you want. For example, you can select *Between Objects* with a spacing of two user units.
10. From *Sort Objects By*, choose how you want the objects to be sorted.
11. On the canvas, select the objects to be aligned. You can select the objects by clicking each object one at a time while holding down the `Shift` key, or you can select the objects by drawing a selection box around them.
12. Click *Apply*.

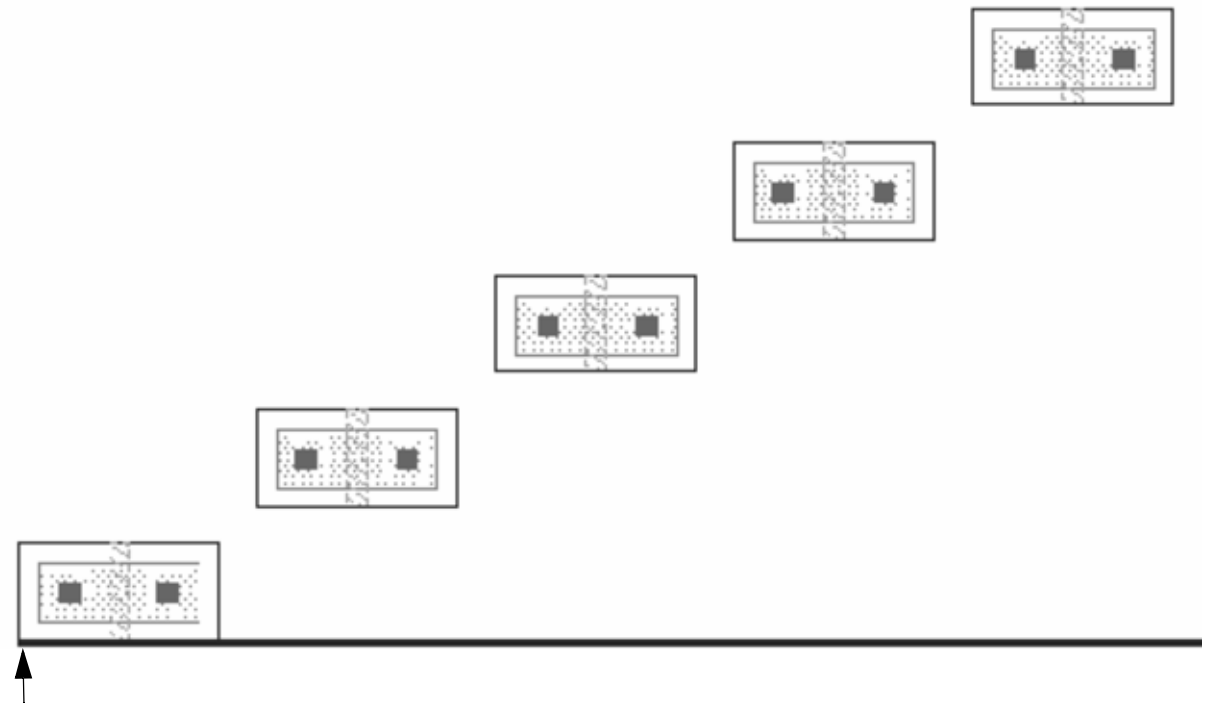
    The following figure shows an example when the reference point is -157:-12 and the *Align Direction* is *Horizontal*. The *Align Using* is *Object* and *Instance*, the *Use* list is set to *Lower Left*, *Apply Spacing* is set to `2.0` user units *Between Objects*, and the objects are sorted by *Instance Name*.



Pointer indicating beginning point and alignment direction.

© 2023 Cadence Design Systems, Inc.

If *Orthogonal Spacing* is also selected and set to 2.0 user units, objects are aligned along the Y-axis as well.



Related Topics

Align Form

Object Alignment by Using the Align Command

Aligning Objects with a Reference Object

The *Align* command supports top-level pins, shapes, instances, devices, rows, cluster boundaries, and groups. The command ignores area halos, layer halos, boundaries and blockages, if selected as part of a set of objects.

To align objects with a reference object:

1. In the layout window, choose *Edit – Advanced – Align*.

   The Align form opens.

2. In the *Reference* section:

© 2023 Cadence Design Systems, Inc.

a. Choose *Align Direction* as horizontal or vertical.

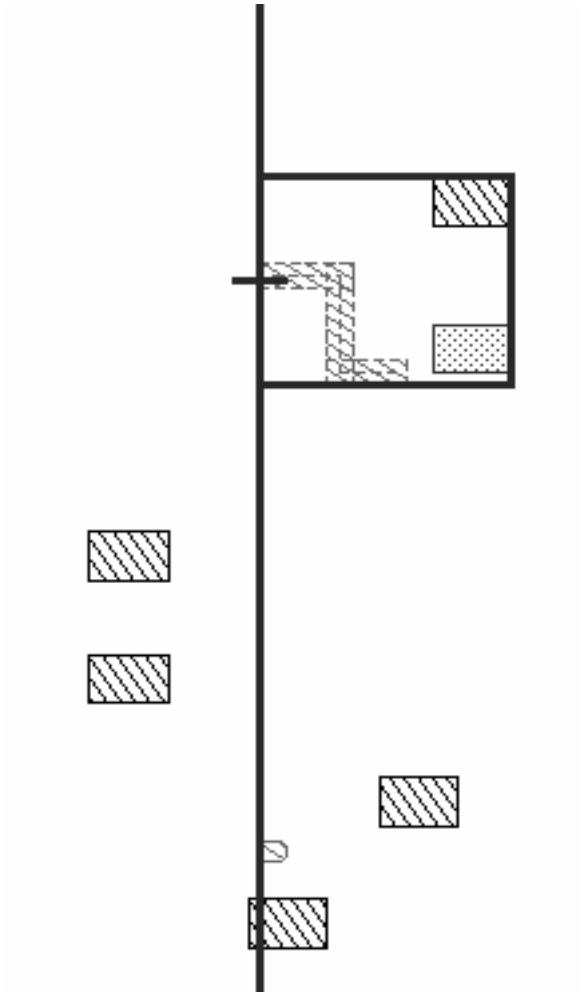
b. Choose *Object* as the reference to align items to an object.

You are prompted to select a reference object for the alignment.

c. Select the type of object you want to align with. For example, *Group*.

3. On the canvas, click the reference object that you want to align other objects to.

The alignment pointer appears.



4. In the *Align Using* section:

a. Choose the *Object* radio button to specify that objects need to align with the reference object.

© 2023 Cadence Design Systems, Inc.

b. Choose the type of objects you want to align with the reference object. For example, *Any Shape*.

c. From the *Use* list, choose the edge of the object you want to align. For example, *Left Edge*.

5. To evenly space the objects in the specified align direction, select the *Apply Spacing* check box and enter the value in user units by which you want to separate the objects. If you do not use *Apply Spacing*, objects retain their original spacing. They are aligned, but not evenly spaced.

6. To space the objects perpendicular to the *Align Direction* as well, select the *Orthogonal Spacing* check box. This option is available only if *Apply Spacing* is selected.

7. Choose Between *Objects* or *Pitches*, depending on the spacing you want. For example, you can choose *Between Objects* with a spacing of two user units.

8. From *Sort Objects By*, choose how you want the objects to be sorted.

You are prompted to select the objects to be aligned.

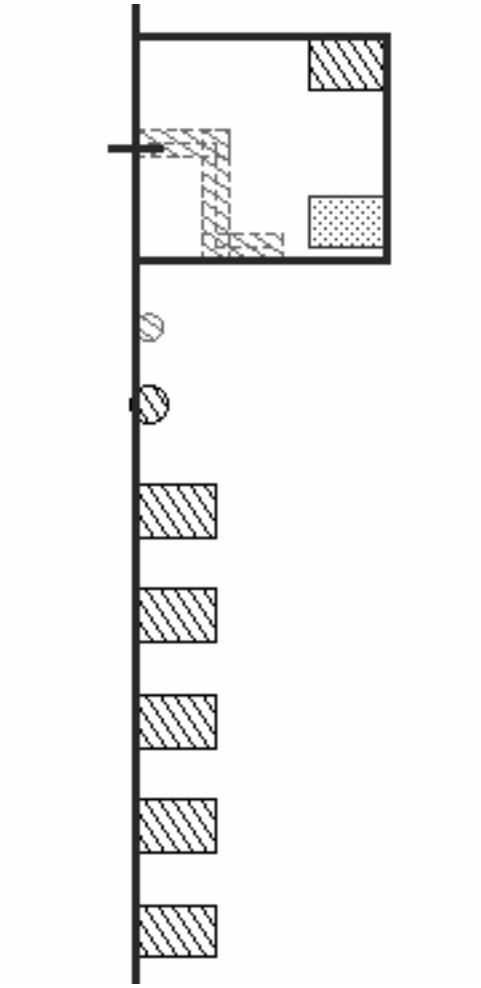
You can select the objects on the canvas by clicking on each object one at a time while holding down the `Shift` key, or you can select the objects by drawing a selection box around them.

9. Click *Apply*.

The following figure shows an example where the reference object is a group, the *Align Direction* is *Vertical*, and the *Use* field is set to *Left Edge*. The *Align Using* is *Any*

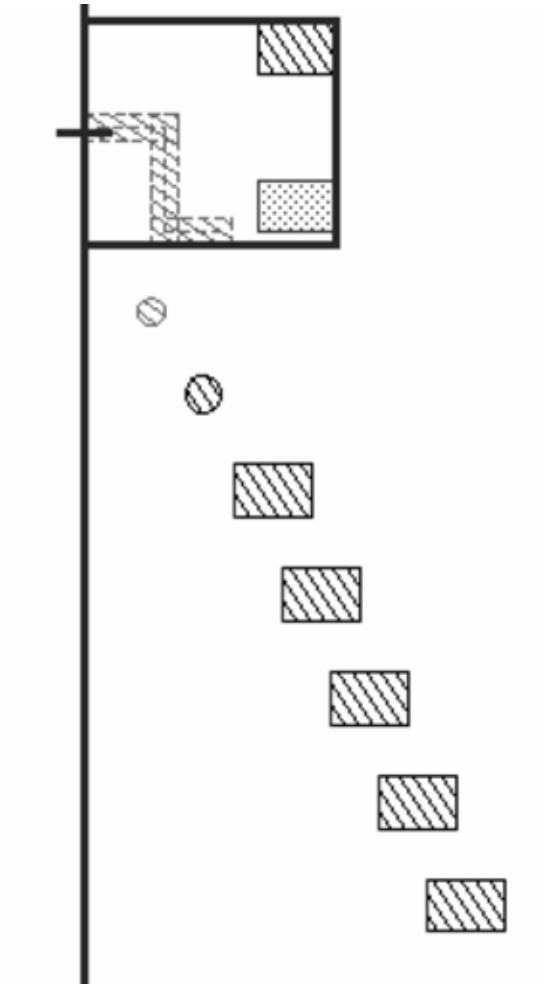
© 2023 Cadence Design Systems, Inc.

Shape, the *Use* list is set to *Left Edge*. *Apply Spacing* is set to 2.0 user units between objects, and the objects are sorted by *Order Selected*.



© 2023 Cadence Design Systems, Inc.

If *Orthogonal Spacing* is also selected and set to 2.0 user units, objects are aligned along the X-axis as well.



Related Topics

Align Form

Object Alignment by Using the Align Command

Aligning Objects with a Layer

The *Align* command supports top-level pins, shapes, instances, devices, rows, cluster boundaries, and groups. The command ignores area halos, layer halos, boundaries and blockages, if selected as part of a set of objects.

© 2023 Cadence Design Systems, Inc.

To align objects with a layer:

1. In the layout window, choose *Edit – Advanced – Align*.

   The Align form opens.

2. In the *Reference* section:

   a. Choose *Align Direction* as horizontal or vertical.

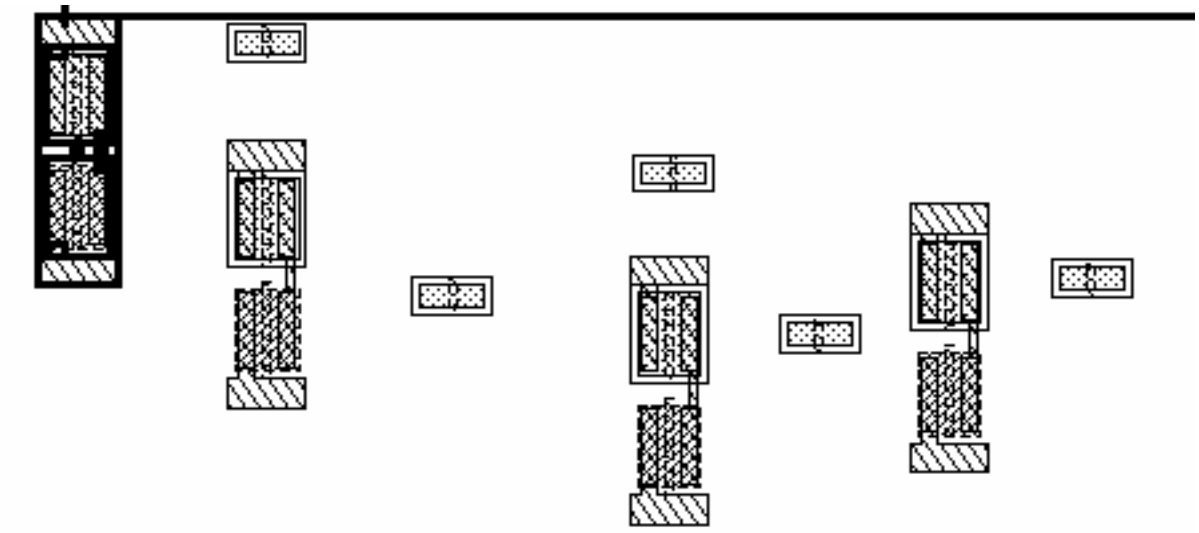
   b. Choose *Object* as the reference to align items to an object.

      You are prompted to select a reference object for the alignment.

   c. Select the type of object you want to align to. For example, *Instance*.

3. On the canvas, click the object you want to align other objects or layers to.

   The alignment pointer appears.



4. In the *Align Using* section:

   a. Choose the *Layer* radio button to specify that you want to align a layer to the reference object.

   b. Choose the layer you want to align with the reference object. For example, `metal1 drawing` layer.

   c. From the *Use* list, choose the edge of the object or layer you want to align. For example, *Top Edge*.

5. To evenly space the objects in the *Align Direction*, select the *Apply Spacing* check box and enter the value in user units by which you want to separate the objects. If you do not

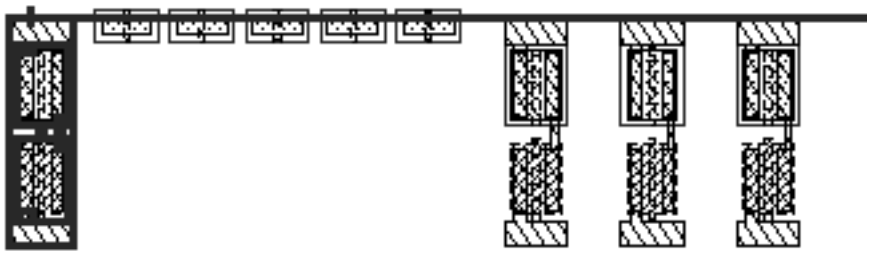
© 2023 Cadence Design Systems, Inc.

use the *Apply Spacing*, objects retain their original spacing. They are aligned, but not evenly spaced.

6. Select the *Orthogonal Spacing* check box to space the objects perpendicular to the *Align Direction* as well. This option is available only if *Apply Spacing* is selected.
7. Choose *Between Objects* or *Pitches*, depending upon the spacing you want. For example, you can choose *Between Objects* with a spacing of `5.0` user units.
8. Choose how you want the objects sorted, in the *Sort Objects By* cyclic field.

   You are prompted to select the objects to be aligned. You can select the objects by clicking on each object one at a time while holding down the `Shift` key, or you can select the objects by drawing a selection box around them.
9. Click *Apply*.

   The following figure shows an example where the reference object is an instance, the *Align Direction* is *Horizontal*, and the *Use* field is set to *Top Edge*. The *Align Using* is *Layer* `metal1 dg`, and the *Use* cyclic is set to `Top Edge`. The *Apply Spacing* is selected, with 5.0 user units *Between Objects* and objects are sorted by *Order Selected*.



Related Topics

Align Form

Object Alignment by Using the Align Command

Aligning a Layer with Another Layer

To align a layer with another layer:

1. In the layout window, choose *Edit – Advanced – Align*.

   The Align form opens.
2. In the *Reference* section:

© 2023 Cadence Design Systems, Inc.

a. Choose *Align Direction* as horizontal or vertical.

b. Choose *Layer* as the reference to align items to a layer.

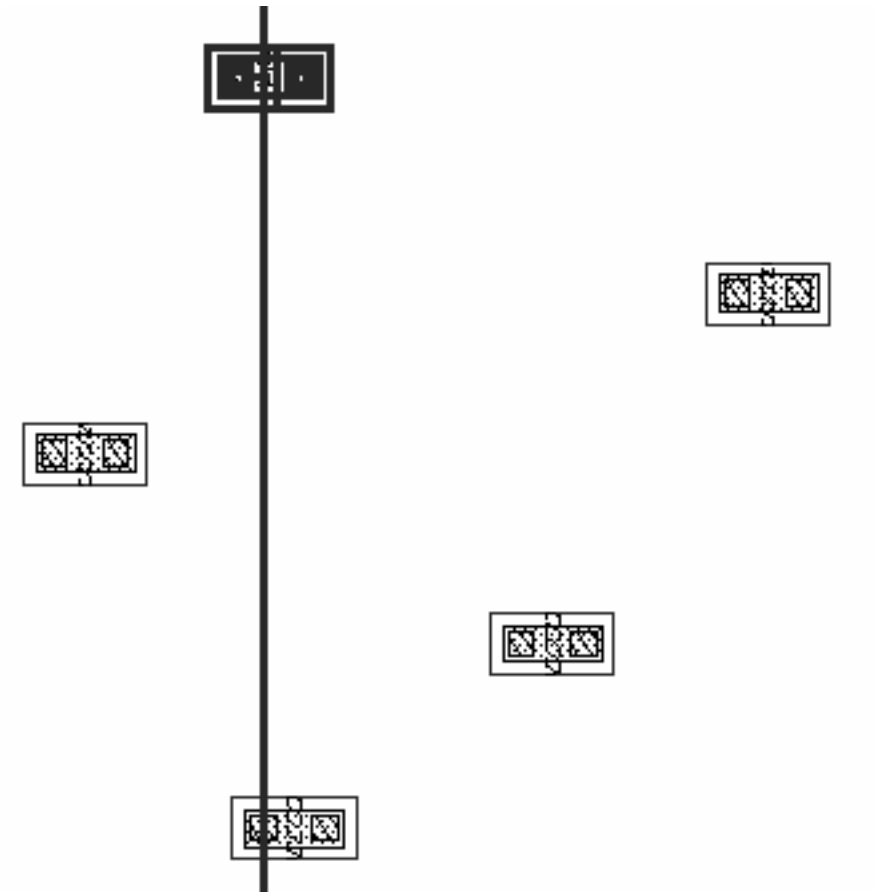
c. Select the layer you want to use to align the layer to.

d. Choose the layer edge you want the objects to align with, from the *Use* list.

You are prompted to select a reference object for the alignment.

3. On the canvas, click the object you want to align other objects or layers to.

The alignment pointer appears.



4. In the *Align Using* section:

a. Select the *Layer* radio button.

b. Select the layer you want to align with the reference layer.

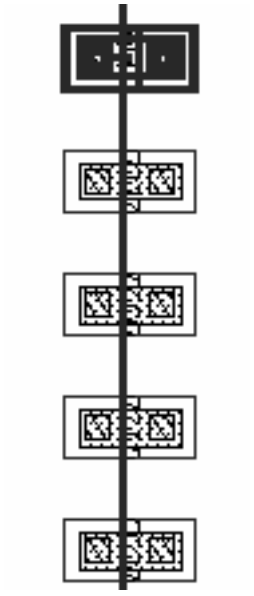
© 2023 Cadence Design Systems, Inc.

c. Select the edge of the object or layer you want to align, from the *Use* list. For example, *Left Edge*.

5. To evenly space the objects in the *Align Direction*, select the *Apply Spacing* check box and enter the value in user units by which you want to separate the objects. If you do not use the *Apply Spacing*, the objects retain their original spacing. They are aligned, but not evenly spaced.
6. Select the *Orthogonal Spacing* check box to space the objects perpendicular to the *Align Direction* as well. This option is available only if *Apply Spacing* is selected.
7. Select how you want the objects sorted, from the *Sort Objects By* list.

   You are prompted to select the objects to be aligned. You can select the objects by clicking on each object one at a time while holding down the `Shift` key, or you can select the objects by drawing a selection box around them.
8. Click *Apply*.

   The following figure shows an example where the reference is *Layer* of material `poly1 dg`, the *Align Direction* is *Vertical*, and the *Use* field is set to *Left Edge*. The *Align Using* is *Layer*, the *Use* cyclic is set to Left Edge. The Apply Spacing is selected, with 5.0 user units Between Objects. Objects are sorted by Instance Name.



Related Topics

Align Form

© 2023 Cadence Design Systems, Inc.

Object Alignment by Using the Align Command

Aligning a Layer with Objects

To align a layer with objects:

1. In the layout window, choose *Edit – Advanced – Align*.

   The Align form opens.

2. In the *Reference* section:

   a. Select *Align Direction* as horizontal or vertical.

   b. Select *Layer*.

   c. Select the layer from the list. For example, `pdiff dg`.

   d. Select the layer edge you want the objects to align to.

   You are prompted to select a reference object for the alignment.

3. On the canvas, click the object you want to align other objects or layers to.

   In the following example, `pdiff dg` is chosen. The alignment pointer appears.



4. In the *Align Using* section:

   a. Select the *Object* radio button.

© 2023 Cadence Design Systems, Inc.

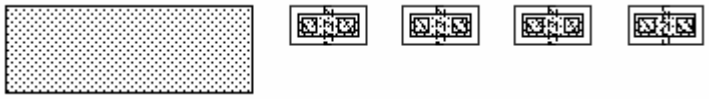
b. Select the object type, from the list, that you want to align with the reference layer.

c. Select the edge of the object or layer you want to align, from the *Use* list. For example, *Top Edge*.

5. To evenly space the objects in the *Align Direction*, click the *Apply Spacing* check box and enter the value in user units in the amount by which you want to separate the objects. If you do not use the *Apply Spacing*, the objects retain their original spacing. They are aligned, but not evenly spaced.
6. Select the *Orthogonal Spacing* check box to space the objects perpendicular to the *Align Direction* as well. This option is available only if *Apply Spacing* is selected.
7. From *Sort Objects By*, select how you want the objects to be sorted.

   You are prompted to select the objects to be aligned. You can select the objects by clicking on each object one at a time, while holding down the `Shift` key, or you can select the objects by drawing a selection box around them.
8. Click *Apply*.

   The following figure shows an example where the reference layer is `pdiff dg`, the *Align Direction* is *Horizontal*, and the *Use* field is set to *Top Edge*. The *Align Using* is *Instance*, the *Use* cyclic is set to *Top Edge*. The Apply Spacing is selected, with 5.0 user units Between Objects. Objects are sorted by Order Selected.



Related Topics

Align Form

Object Alignment by Using the Align Command

Adding Objects with a Pre-Selected Aligned Group

To add objects with a pre-selected aligned group:

1. On the layout canvas, use `Shift` + click to select the unaligned objects.
2. Choose *Edit – Advanced – Align* to open the Align form.
3. Click *Set New Reference*.

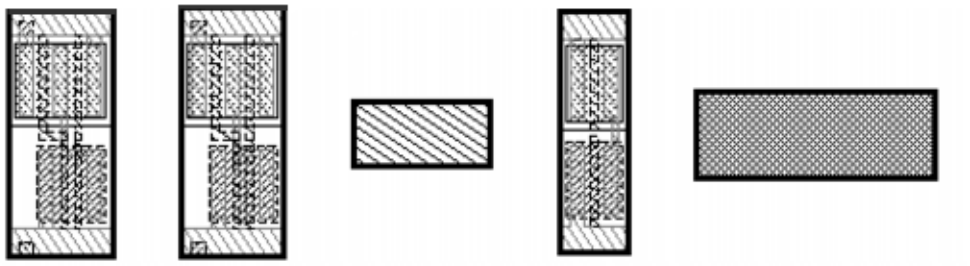
© 2023 Cadence Design Systems, Inc.

You are prompted to select a reference object for the alignment.

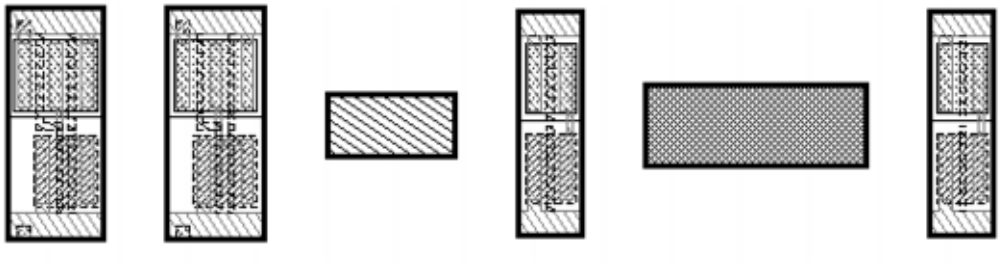
4. On the canvas, click at the point to which you want the objects to align.

   The unaligned devices align with the original selected alignment group.

   Before:



   After:



Related Topics

Align Form

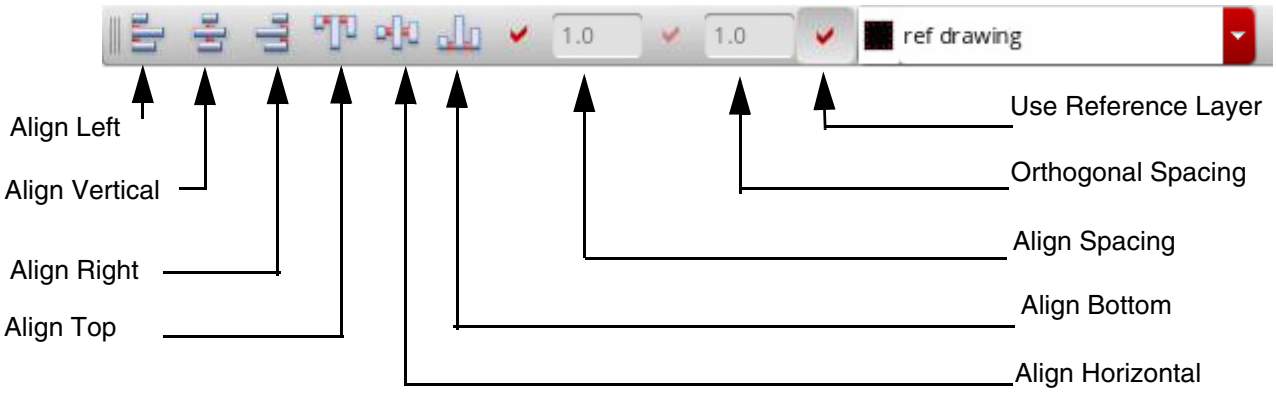
Object Alignment by Using the Align Command

Aligning Objects by Using the Align Toolbar

For performing quick alignment operations by using the *Align* command, use the *Window - Toolbars - Align* toolbar. Using the toolbar, you can perform left, right, top, and bottom

© 2023 Cadence Design Systems, Inc.

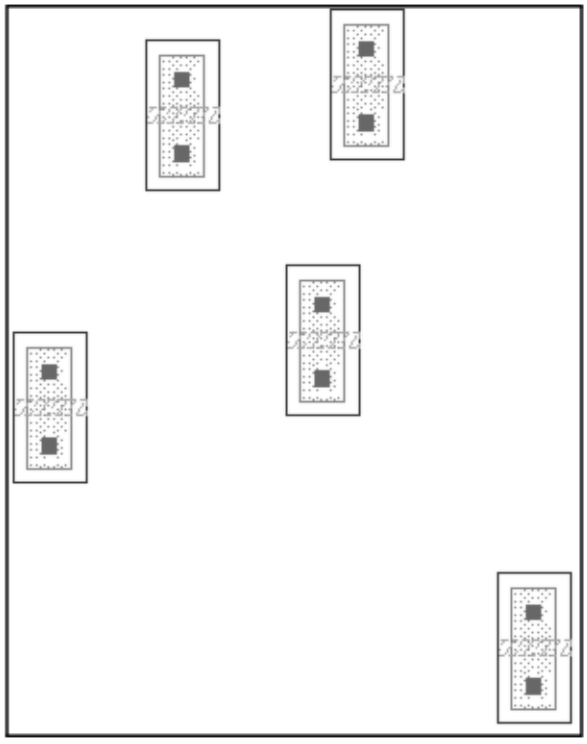
alignment of the selected objects. You can also align them vertically or horizontally. The toolbar also provides the option of specifying alignment and orthogonal spacing values.



To specify more advanced align options, such as the reference point, object, or layer, the align direction, and the sort order, you can use the Align form, accessed using *Edit – Advanced – Align*.

To align the left, right, top, or bottom edges of the selected objects:

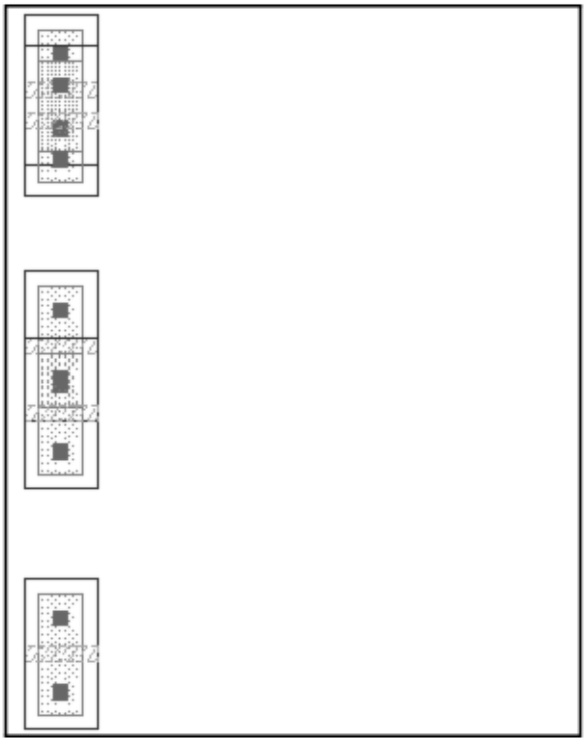
1. On the layout canvas, select the objects to be aligned.



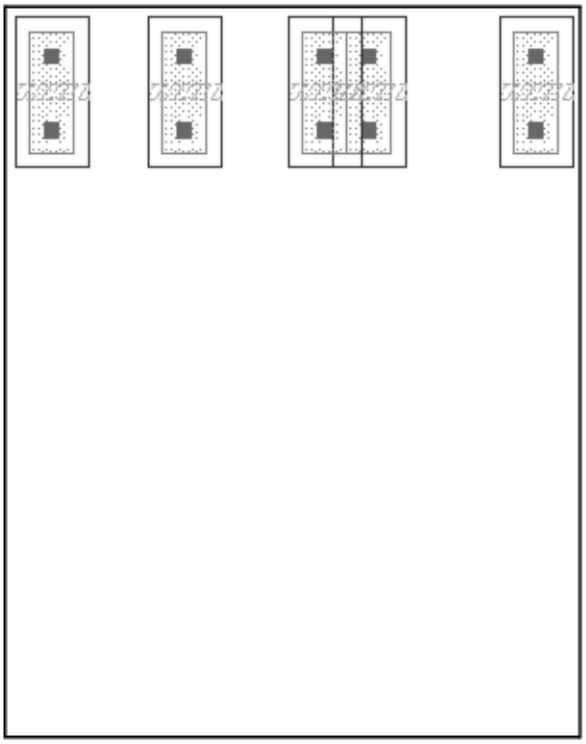
© 2023 Cadence Design Systems, Inc.

2. Click *Align Left*, *Align Right*, *Align Top*, or *Align Bottom* on the *Align* toolbar for the required alignment. The selected objects are aligned to the left, right, top, or bottom edge of the imaginary bounding box around the selected set.
3. To evenly space the objects, select *Align Spacing* and enter the value in user units of the amount by which you want to separate the objects. If you do not use the *Apply Spacing*, the objects retain their original spacing. They are aligned, but not evenly spaced.
4. To space the objects perpendicular to the *Align Direction* as well, select the *Orthogonal Spacing* option. This option is available only if *Apply Spacing* is selected.
5. To align objects of a specific reference layer, select the *Use Reference Layer* option and select a reference layer.

   In the following example, the selected objects have been aligned first to the left and then to the top. The objects have not been spaced.



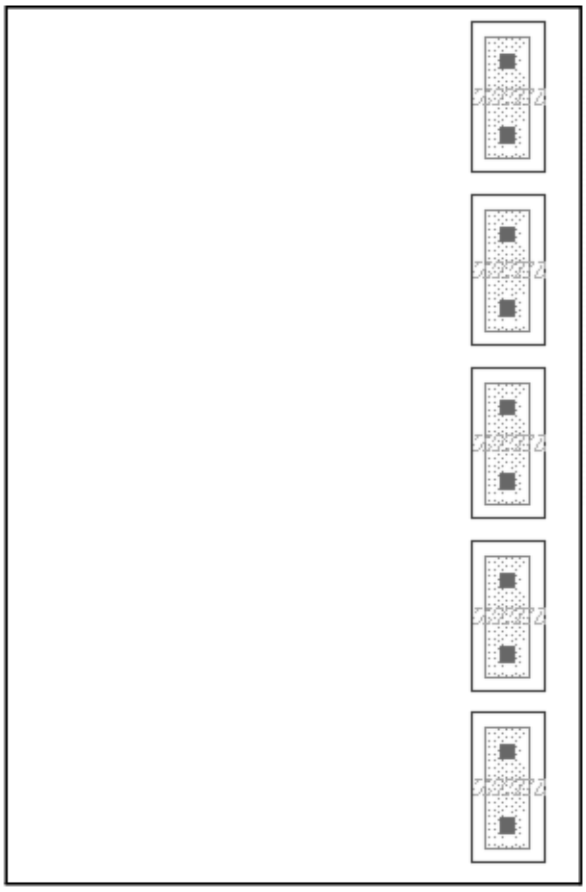
Result of Using *Align Left* from the Toolbar, *Align Spacing* = 0



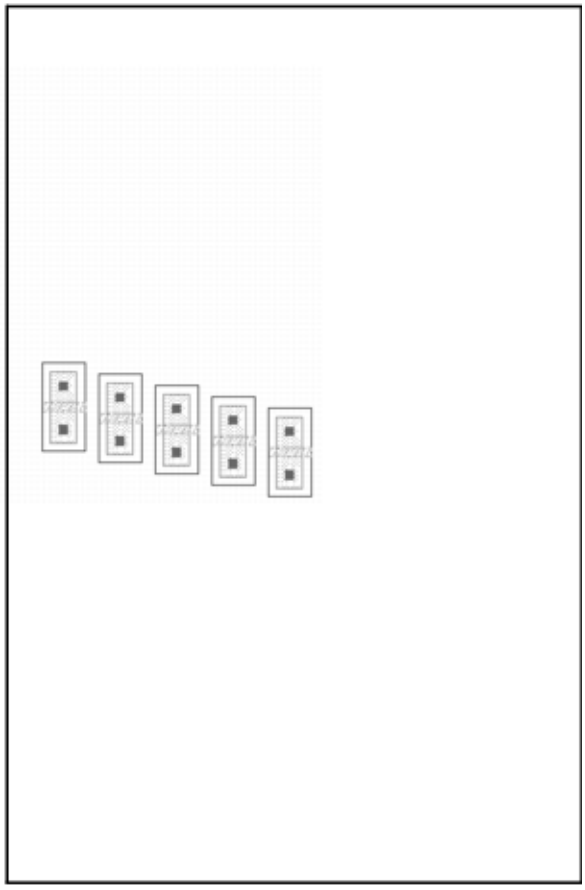
Result of Using *Align Top* from the Toolbar, *Align Spacing* = 0

© 2023 Cadence Design Systems, Inc.

Alignment in the following examples uses spacing.



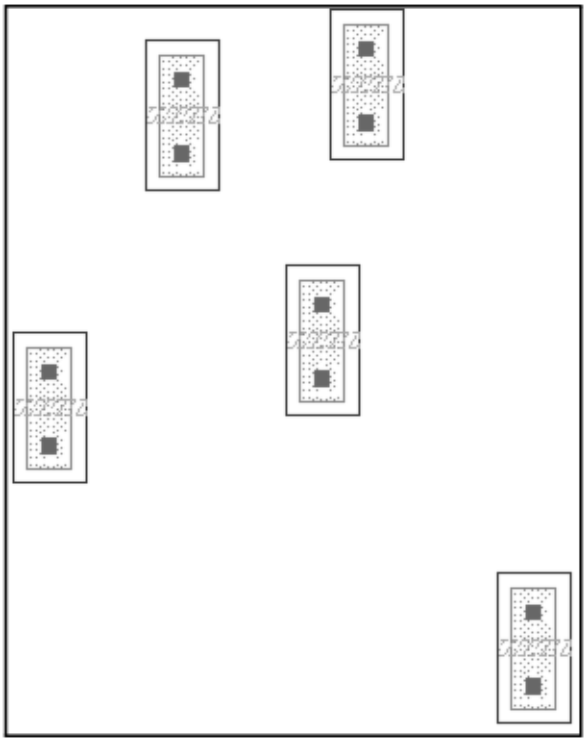
Result of Using *Align Right* from the Toolbar, *Align Spacing* = 1.0



Result of Using *Align Bottom* from the Toolbar, *Align Spacing* = 1.0, Orthogonal Spacing = 1.0

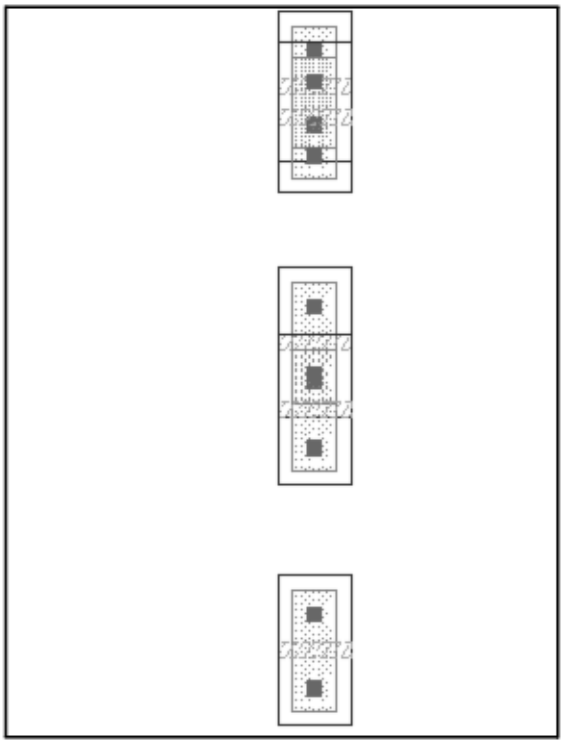
© 2023 Cadence Design Systems, Inc.

You can similarly align objects horizontally or vertically. The following example illustrates this functionality. The figure shows the objects that need to be aligned.

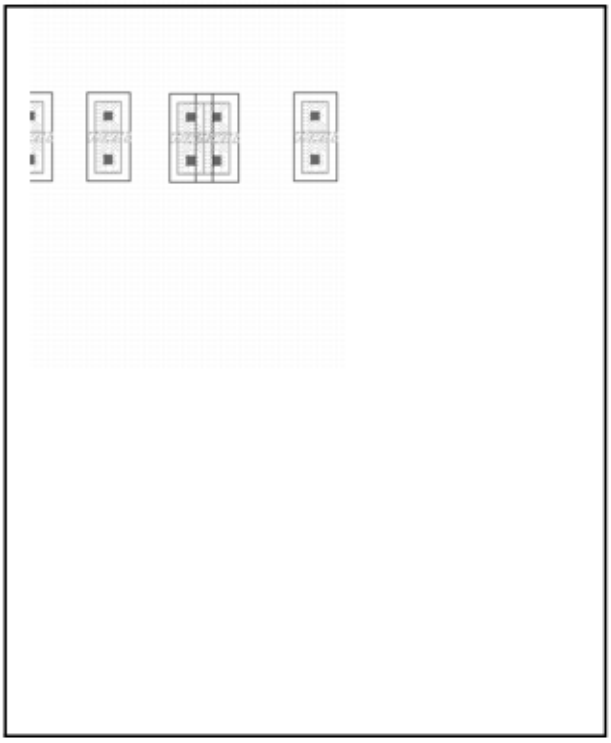


© 2023 Cadence Design Systems, Inc.

The following figure shows the objects aligned vertically and horizontally.



Result of Using *Align Vertical* from the Toolbar, *Align Spacing* = 0



Result of Using *Align Horizontal* from the Toolbar, *Align Spacing* = 0

If *Align Spacing* is not on, the center of the imaginary bounding box around the selected set is considered as the reference point. Objects are then aligned along the imaginary vertical or horizontal line passing through this center point.

If *Align Spacing* is on, the object closest to the center of the imaginary bounding box around the selected set is regarded as the reference object. Remaining objects are aligned vertically or horizontally with respect to the reference object, separated by *Align Spacing* distance and *Orthogonal Spacing*, if specified.

Related Topics

Object Alignment

Object Alignment by Using the Quick Align Command

Object Alignment by Using the Align Command

© 2023 Cadence Design Systems, Inc.

Editing Slots

The *Edit Slot* command applies only to slotted shapes that are created by using the *Create Slot*, *Create Rectangle*, or *Create Path* commands.

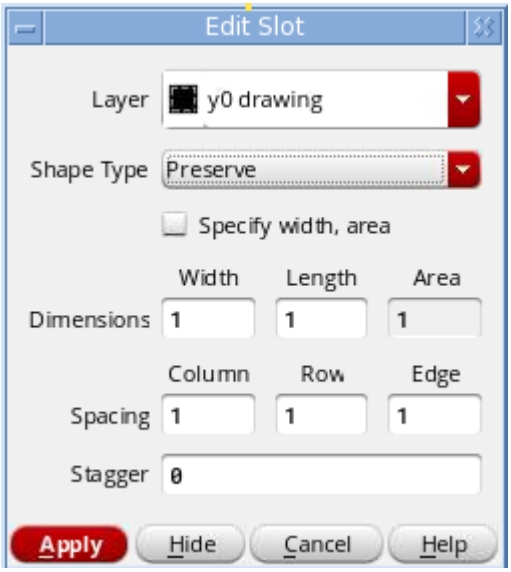
To edit slots in an existing rectangle or path:

1. On the layout canvas, click the rectangle or path containing the slots.

   You can select multiple shapes containing slots to simultaneously edit the slots in all the selected shapes. The selected shapes can be a mix of ROD objects and polygons.

2. Choose *Edit – Advanced – Slot*.

   The Edit Slot form opens and is populated with the values of the slots in the selected shapes.



   Alternatively, you can first open the form, and then select one or more shapes for editing the slots they contain.

3. From the *Shape Type* list, select *Preserve*, *Convert to Polygon*, or *Convert to ROD* based on the required resultant shape type.

4. Update the values in the different fields as required.

5. Click *Apply*.

   The values specified in the form are applied to the slots in the selected shapes.

6. Press `Esc` or click *Cancel* in the form to finish editing the slots.

© 2023 Cadence Design Systems, Inc.

Related Topics

Create Slot Form

Creating Slots

Creating a Rectangle

Creating a Path

Editing Groups

You can perform the following edit operations on a group:

- Adding Objects to a Group
- Adding Objects to a Group By Using the Select Target Group Form
- Ungrouping All Objects in a Group
- Removing Objects from a Group
- Editing Group Contents by Using Edit In Place
- Editing Group Contents by Using Transparent Group Mode

Related Topics

Creating a Group

Adding Objects to a Group

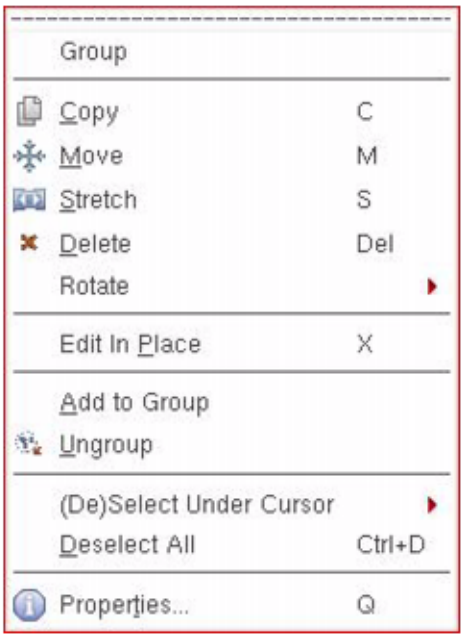
To add an object to a group:

1. In the layout window, choose *Edit – Group – Add to Group*.

   You are prompted to point at the target group.
2. On the canvas, click the group you want to edit.

© 2023 Cadence Design Systems, Inc.

Alternatively, you can select the group you want to edit and right-click to display the *Group* shortcut menu. Then, select *Add to Group* from the shortcut menu.



You are prompted to select the figures to be added to the group.

3. Click the objects you want to add to the group.

   The objects are added to the group. The solid line surrounding the group changes to include the new objects.

4. Press the `Esc` key to end the command.

   The dotted line surrounds all the group objects when you move the pointer over the group.

Related Topics

Creating a Group

Adding Objects to a Group By Using the Select Target Group Form

Ungrouping All Objects in a Group

Removing Objects from a Group

Editing Group Contents by Using Edit In Place

© 2023 Cadence Design Systems, Inc.

Editing Group Contents by Using Transparent Group Mode

Adding Objects to a Group By Using the Select Target Group Form

To add an object to a group by using the Select Target Group form:

1. In the layout window, choose *Edit – Group – Add to Group*.
2. Press the `F3` key.

   The Select Target Group form appears.
3. Enter the name of the group in the *Group Name* field.
4. Click the *Filter* button and the group name is added to the information box.

   You can continue to add groups to the information box by adding the group name, or you can click the *Filter* button to add all of the groups into the information box.
5. Click the name of the first group you want to zoom to.
6. Select the *Zoom To* group check box.
7. Move the *scale* slider to zoom in closer or zoom out from the group.

   The scale set at 1 is the equivalent of the fit edit command. The scale set at 50 is the farthest you can zoom out in the cellview.

   You are prompted to point at the target group.
8. Click the group name, in the form, that you want to edit.

   You are prompted to select the figures to be added to the group.
9. Click the *Select* button on the form or click the group on the canvas.
10. On the canvas, select the objects you want to add to the group.

    The objects are added to the group. The solid line surrounding the group changes to include the new objects.
11. Press `Esc` to end the command.

Related Topics

Creating a Group

Adding Objects to a Group

© 2023 Cadence Design Systems, Inc.

Ungrouping All Objects in a Group

Removing Objects from a Group

Editing Group Contents by Using Edit In Place

Editing Group Contents by Using Transparent Group Mode

Ungrouping All Objects in a Group

To ungroup all objects from a group:

1. In the layout window, choose *Edit – Group – Ungroup*.

   You are prompted to point at the group to ungroup it.

2. On the canvas, click the group you want to ungroup.

   Alternatively, you can select the group and right-click it. Then, select *Ungroup* from the *Group* shortcut menu.

   The group is ungrouped. The line surrounding the group is deleted, and no longer appears when you place the pointer over the objects that were in the group.

When you ungroup objects from a group, the child object of the group is not deleted after you ungroup.

Related Topics

Creating a Group

Adding Objects to a Group

Adding Objects to a Group By Using the Select Target Group Form

Removing Objects from a Group

Editing Group Contents by Using Edit In Place

Editing Group Contents by Using Transparent Group Mode

Removing Objects from a Group

To remove an object from a group:

© 2023 Cadence Design Systems, Inc.

1. In the layout window, choose *Edit – Hierarchy – Edit In Place*.

   Alternatively, you can right-click the group to display the *Group* shortcut menu and select *Edit In Place.*

   You are prompted to point at a shape in the cellview to be edited in place.

2. On the canvas, click the group you want to edit.

   The *Edit – Group – Remove from Group* remains grayed out until you have completed this step.

3. Choose *Edit – Group – Remove from Group*.

   You are prompted to select the figures to be removed from the current group.

4. Click the object you want to remove from the group.

   The object is removed from the group. The solid line surrounding the group changes to exclude the removed object. Similarly, you can continue to click other objects to remove them from the group.

5. Click the `Esc` key to end the command.

6. Choose *Edit – Hierarchy – Return to Level* or right-click anywhere on the design canvas and select *Return* from the *Layout* shortcut menu.

   The dotted line surrounds only the remaining objects when you move your pointer over the group.

Related Topics

Creating a Group

Adding Objects to a Group

Adding Objects to a Group By Using the Select Target Group Form

Ungrouping All Objects in a Group

Editing Group Contents by Using Edit In Place

Editing Group Contents by Using Transparent Group Mode

© 2023 Cadence Design Systems, Inc.

Editing Group Contents by Using Edit In Place

You can edit the contents of a group either by using the *Edit In Place* command or by switching to transparent group mode.

To edit group contents by using the *Edit In Place* command:

1. On the layout canvas, click the group whose contents you want to edit.
2. Choose *Edit – Hierarchy – Edit In Place*.

   Alternatively, you can right-click the group to display the *Group* shortcut menu and select *Edit In Place.*
3. Perform the desired edit operations on the objects within the group.
4. Choose *Edit – Hierarchy – Return to Level* or right-click anywhere on the design canvas and select *Return* from the *Layout* shortcut menu.

In a row type of FigGroup, you can edit or stretch the objects within the group when the selection mode is set to partial. By using this feature, you can stretch multiple rows or rails at the same time.

Related Topics

Creating a Group

Editing Group Contents by Using Transparent Group Mode

CDBA Figure Functions

Editing Group Contents by Using Transparent Group Mode

The transparent group mode enables you to edit a user-defined group without doing Edit-in-place (EIP) into the group. In this mode, the contents of user-defined groups become available for selection and editing at the top-level, while being retained as objects of their respective groups. You can edit objects in nested groups also at the top level itself, without performing multiple EIPs. You can enable this mode only for the active window. In transparent group mode, you can perform group-related commands, such as creating groups and adding, removing, and ungrouping objects in a group.

You can enable the bannerFields environment variable to indicate on the status banner whenever transparent group mode is activated.

© 2023 Cadence Design Systems, Inc.

To edit groups in transparent group mode:

1. In the layout window, activate transparent group mode in one of the following ways:
   - In the Display Options form, select the *Transparent Group* option in the *Display Controls* section.
   - On the *Options* toolbar, click the *Transparent Group*  icon. You can access the toolbar by choosing *Window - Toolbars - Options*.
   - Set the window-specific environment variable in the CIW:

     ```
     window = hiGetCurrentWindow()
     window~>transparentGroup = t
     ```

     This makes all the user-defined groups in the design editable. In transparent group mode, the pointer changes to white.
2. Perform the desired edit operations on the objects within a group.
3. After you are done with editing, you can leave the transparent group mode enabled for the current window session or you can disable it in one of the following ways:
   - In the Display Options form, deselect the *Transparent Group* option in the *Display Controls* section.
   - On the *Options* toolbar, toggle the *Transparent Group* icon to OFF.
   - Set the window-specific environment variable in the CIW:

     ```
     window = hiGetCurrentWindow()
     window~>transparentGroup = nil
     ```

If you intend to retain the transparent group setting ON for the remaining window session, Cadence recommends that you switch on the visibility (Vis) of *Groups* in the *Objects* panel of the Palette assistant. This will display a boundary around groups even in transparent group mode and will prevent you from inadvertently modifying the objects belonging to any group.

In transparent group mode, contents of user-defined groups are available for selection and editing without doing EIP into groups. You can, therefore, select objects belonging to multiple groups and edit them simultaneously. For example, you can move shapes belonging to two different groups at the same time: Activate the transparent group mode, select shapes irrespective of the group to which they belong, and move them by the desired value. Similarly, you can select objects belonging to different groups and chop them simultaneously.

You cannot merge objects belonging to different groups. You also cannot attach an object from one group as the owner of an object belonging to another group. If you run the *Make Cell* command on objects belonging to different groups, the selected objects are removed from their respective groups. The resultant instance is placed at the top level and is not included in any group.

© 2023 Cadence Design Systems, Inc.

Related Topics

Display Options Form

Editing Group Contents by Using Edit In Place

Edit Commands Supporting Snap Pattern Snapping

Snap pattern snapping is supported in the *Copy*, *Move*, *Stretch*, *Flip*, *Rotate*, *Yank* and *Paste*, and *Repeat Copy* commands.

You can enable the snapResizeWireObjects environment variable to snap, resize, and color all pathSegs and vias in the selection set during the *Move* and *Copy* commands. The pathSegs are snapped to the track by maintaining the relative spacing between them. The connectivity of pathSegs and vias is also maintained.

In Layout EXL or a higher tier, the *Move*, *Stretch*, and *Chop* commands support virtual hierarchies with area boundary. When you move, stretch, or chop virtual hierarchies with area boundaries, the area boundaries are snapped to the grid.

Related Topics

Snap Pattern Snapping for FinFET Devices in Layout Editor

Object Snapping to WSPs During Editing

Pin Snapping to Width Spacing Patterns

Snapping Mode Behavior in Edit Commands

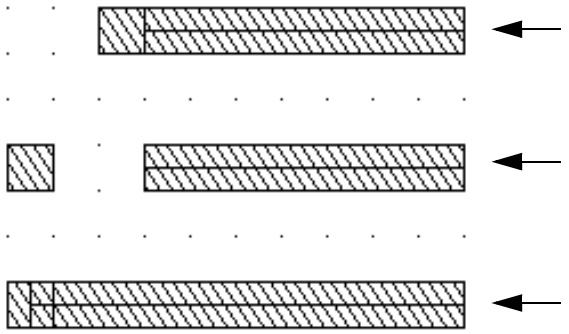
Pin-Path and Pin-PathSeg Connections Preserved

You can configure the layout editor to automatically reconnect path-to-pin and pathSeg-to-pin connections that are broken during any pin editing activity. You can maintain connections by doing one of the following:

- In Layout Editor Options form, select the *Maintain Connections* check box.
- Press `F6` to toggle *Maintain Connections* on and off.

© 2023 Cadence Design Systems, Inc.

A message in the CIW lets you know the status of *Maintain Connections*.



Maintain Connections is a background process that monitors the connectivity of pins that abut or overlap path or pathSeg ends on the same layer. When any editing activity causes the pin to move away from the path or pathSeg end, *Maintain Connections* automatically modifies the path or pathSeg so that it remains connected to the pin.

Maintain Connections searches all instances in the top cellview to find all the pins, then finds all the overlapping paths and pathSegs associated with each pin to check for broken connections. If the top-level design contains many instances and each instance contains many pins, opening a design can become more time consuming. When you open large designs, you should set *Maintain Connections* to off.

The editing activities that can cause a pin to change location include:

- Moving or stretching a pin
- Moving or stretching an instance containing a pin
- Editing pin properties by changing the pin shape or pin location
- Changing Pcell parameters

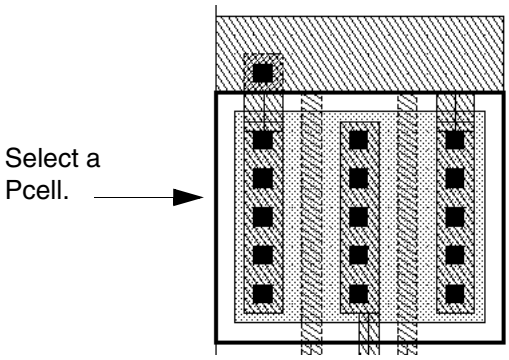
The *Maintain Connections* option operates on pins, paths, and pathSegs with the following characteristics:

- The pin and path or pathSeg are on the same or equivalent layer and have the same layer purpose.
- The path or pathSeg end overlaps or abuts the pin.
- The pin is rectangular.
- If the pin is an instance, it must be placed at the same level as the path or pathSeg.
- If the pin and path or pathSeg are ROD objects, they cannot be aligned.

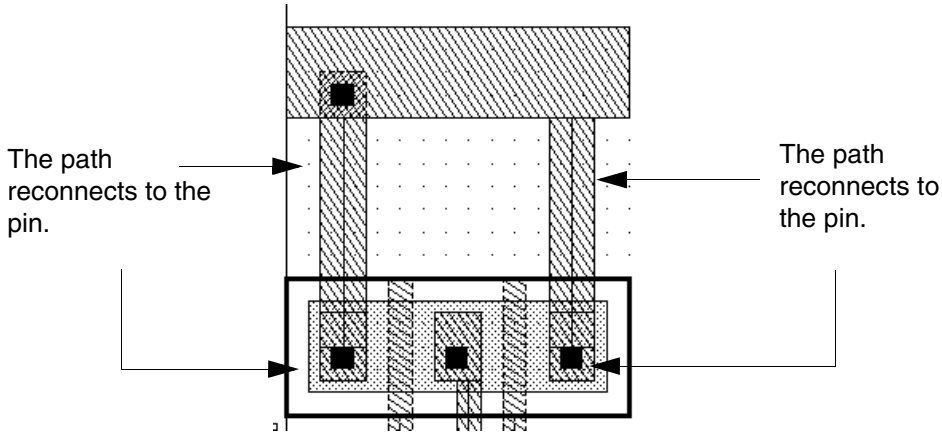
© 2023 Cadence Design Systems, Inc.

One way to see the effects of *Maintain Connections* is to change the parameters of an instance that contains pin-path or pin-pathSeg connections on the same layer. After you change the parameters, the instance changes shape to the new value, and the path or pathSeg and pin retain their connectivity.

To change the parameters of an instance that contains pin-path or pin-pathSeg connections, on the layout canvas, select a Pcell containing pin-path or pin-pathseg connections. Then, open the properties form by choosing *Edit – Basic – Properties*. On the *Parameter* tab, change the size of the Pcell and click *Apply*. The following image shows a Pcell selected.



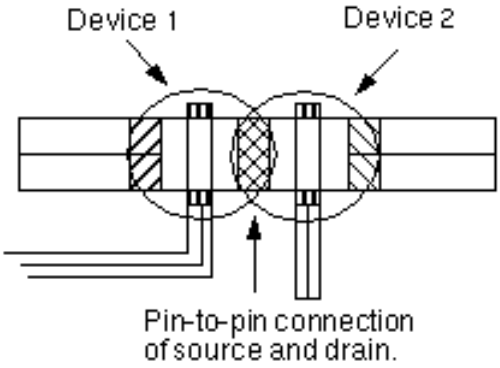
The instance changes shape to the new width value, and the paths/pathsegs reconnect to the pins.



Maintain Connections does not retain connectivity in the following relationships:

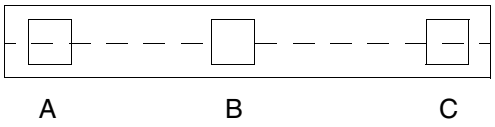
© 2023 Cadence Design Systems, Inc.

- Pin-to-pin connections



- Pins placed in the middle of paths

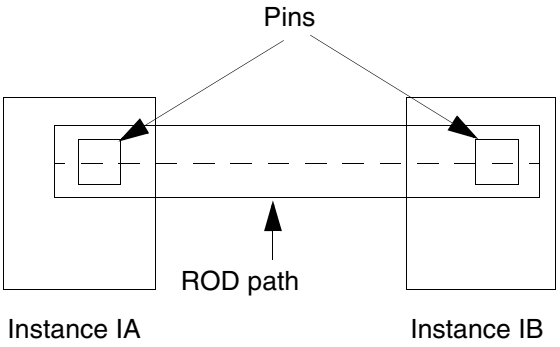
Pins A and C remain connected to the path, but pin B might not.



- Aligned ROD objects

  ROD alignments take precedence over maintaining connections.

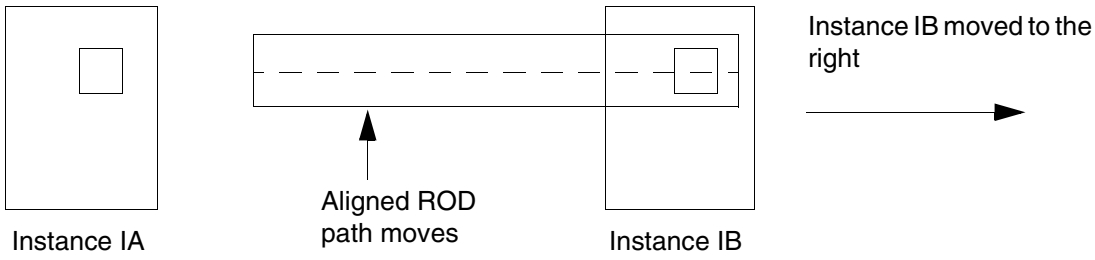
  The following image shows a ROD path before a move operation with *Maintain Connections*:



The pin in instance IB is aligned to the ROD path.

© 2023 Cadence Design Systems, Inc.

The following image shows a ROD path after a move operation with *Maintain Connections*:



Moving instance `IB` causes the whole ROD path to move. The aligned path does not stretch, so moving instance `IB` breaks the electrical connection between the two pins.

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

7

Multipart Paths

You can use the *Create Multipart Path* command to create named paths that have relative object design (ROD) information associated with them. You can create simple one-part paths or complex paths containing several parts, such as a guard ring, transistor, bus, or shielded path. A path that consists of more than one part is called a multipart path (MPP).

A multipart path consists of a single master path and one or more subparts. The master path is the defining object; all subparts are based upon and exist in relation to the master path. You can define any number of subparts for a multipart path. Subparts can be any or all of the following: offset sub-paths, enclosure sub-paths, and sets of sub-rectangles. Also, you can define connectivity for any or all parts of a multipart path. In the layout editor, a multipart path (master path and all its subparts) is treated as a single object.

A ROD multipart path offers advantages over a regular path. For example, you can access a ROD multipart path by name, even through levels of hierarchy, and you can chop it.

Related Topics

Flow for Creating Multipart Paths

Creating an MPP Guard Ring

Multipart Path Environment Variables

Multipart Path Selection

Moving Multipart Paths

Splitting Multipart Paths

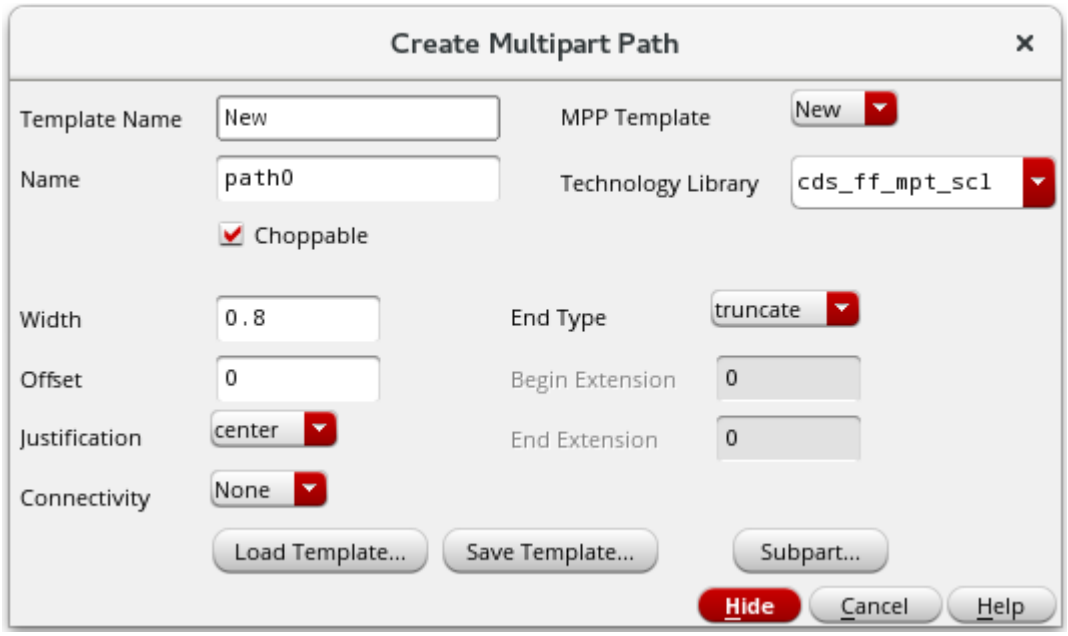
Reshaping a Multipart Path

How Chopping Affects Multipart Paths

© 2023 Cadence Design Systems, Inc.

Multipart Path Creation Overview

With the *Create Multipart Path* command active, you can define a multipart path by providing details in the Create Multipart Path form and the ROD Subpart form. After specifying information in the forms, you can click in the current layout cellview window to define the points for the master path of the MPP. You can change the values in the forms while you are clicking points, until you double-click to complete the MPP.



When you define a new type of MPP, you might want to save the values in the forms as a template so that you can create other MPPs that are same or similar. A template is the set of values that specify a particular multipart path, such as a guard ring. You can define as many MPP templates as you wish.

With the *Create Multipart Path* command, you can:

- Create a single-part path using the form defaults.
- Choose an MPP template using the *MPP Templates* field.
- Modify the values in the *Create Multipart Path* and *ROD Subpart* forms to create a different MPP.
- Save the current form values as a template in your technology file if you have write permission, or in an ASCII file.

© 2023 Cadence Design Systems, Inc.

- Load an ASCII MPP template file from your local hierarchy to add the templates to the list in the *MPP Templates* field. For additional ways to load ASCII MPP template files, use the `leDefineMPPTemplate` SKILL function.

As you draw an MPP in a cellview, you can make changes to the forms and see the results of your changes immediately, until you double-click to complete the MPP. You can make as many changes as you want to an MPP using the Create Multipart Path form and the ROD Subpart form prior to entering the last point of your MPP in the layout window. Once you enter the last point, you cannot use these forms to change the MPP. You can change a completed MPP using the Edit Properties form.

If you are creating a multipart path with abutted ends and the centerline of the master path intersects itself, the system will not create the path. A dialog box appears stating that no path was created because it would have been self-intersecting. The system does not allow you to create a path whose master path centerline intersects itself, and does not allow you to create a path with one or more self-intersecting sub-paths.

When you first start the layout editor software, the *Template Name* and *MPP Template* fields in the *Create Multipart Path* form are set to *New* and the values of the other fields are system defaults, unless the mppTemplate environment variable is set. If `mppTemplate` is set, *New* is replaced with a specific template name, and the other fields are set to the template values. The `mppTemplate` environment variable enables you to designate a particular MPP template as your default rather than use the system default values.

For some fields, the system looks at your technology file for the design rule for the layer and, if it is defined, uses it as the default. For example, the system looks for the `minWidth` rule for the current layer to use as the default for the *Width* field.

You can create an MPP in a layout cellview window using the default values without even opening the form, you can open the form and change the values, or you can use the values from an MPP template that was saved earlier. When you choose an MPP template from the *MPP Template* field, the system fills in the rest of the fields in the *Create Multipart Path* form and *ROD Subpart* form with the values specified in the template. The system assigns system default values to fields for which values are not specified in the template. You can use the template values as they are or modify them as desired. If you want to define connectivity for any part of an MPP, you must open the Create Multipart Path form, and specify the net name in the *Connectivity* section.

(Virtuoso RF Option) The *Create Multipart Path* command does not support curved shapes.

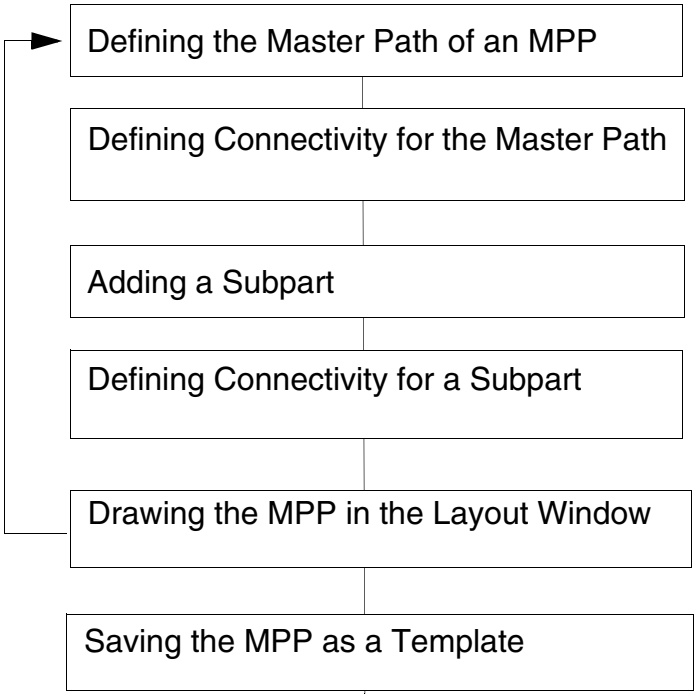
Related Topics

Flow for Creating Multipart Paths

© 2023 Cadence Design Systems, Inc.

Flow for Creating Multipart Paths

The following figure lists the tasks involved in creating a multipart path (MPP).



The following table describes the tasks involved in creating a multipart path.

Task	Description
Defining the Master Path of an MPP	Define the master path of your MPP.
Defining Connectivity for the Master Path	Optionally, define the connectivity for the master path of your MPP.
Adding a Subpath	Optionally, add subparts to the master path. You can define the following types of subparts: offset subpath, enclosure subpath, or subrectangle. You can choose to edit or delete a subpart as required while creating an MPP.
Defining Connectivity for a Subpart	Optionally, define the connectivity for any or all subparts.

© 2023 Cadence Design Systems, Inc.

Task	Description
Drawing a Multipart Path in the Layout Window	Draw the MPP on the layout canvas. As you draw an MPP, you can change the values in the Create Multipart Path and ROD Subpart forms and see immediate changes to the MPP. After you enter the last point to complete an MPP, you can no longer change that MPP using the create forms. You need to use the *Edit MPP Path Properties* form to change a completed MPP.
Saving the MPP as a Template	Optionally, save the MPP values as a template if you want to use the MPP again. You can save an MPP to an ASCII file in your local hierarchy or to the temporary technology file in the virtual memory.

Defining the Master Path of an MPP

You can use a template to create a multipart path (MPP) or create a new MPP from scratch. The templates already defined in your technology library are listed in the *MPP Template* field of the Create Multipart Path form. You can also load MPP templates from ASCII files to add them to the templates listed in the *MPP Template* field and to the temporary technology file in the virtual memory.

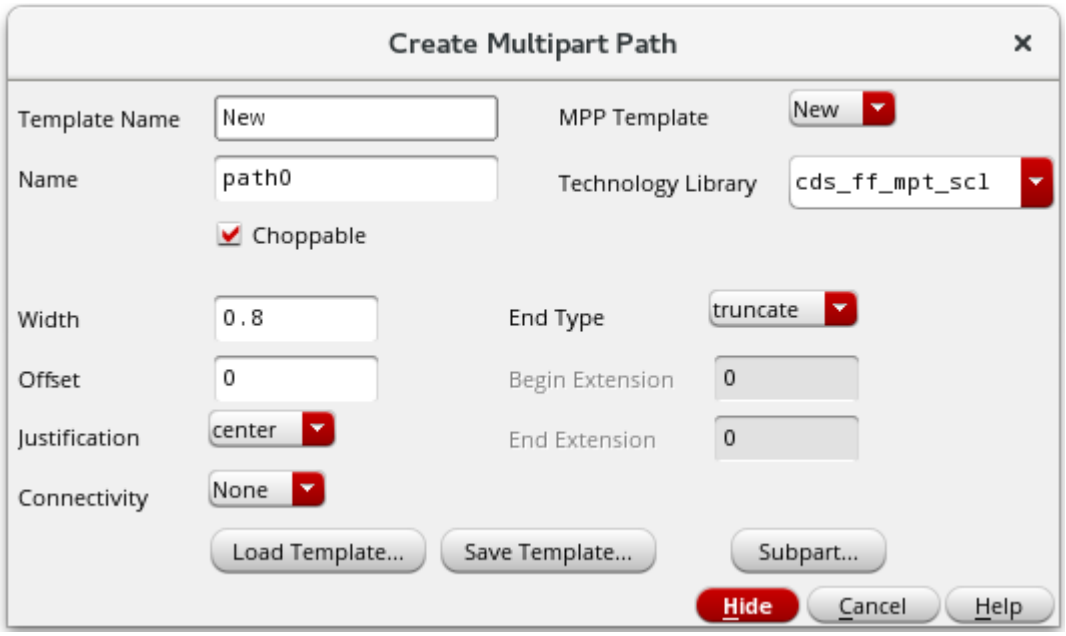
Templates do not contain net names because each MPP you create from the same template could be on a different net. When you choose a template that contains pin connectivity, the system prompts you to enter a valid net name. You must enter a name in the *Net Name* field for each part (master path and/or subparts) that has pin connectivity. If you click in the cellview window before entering the required net names, the system beeps.

To define the master path of an MPP:

1. In the layout window, choose *Create – Multipart Path*.
2. Press `F3`.

© 2023 Cadence Design Systems, Inc.

The Create Multipart Path form opens.



The first time you open form, the values of fields default to the values set in the technology file and can vary depending on which layer is current in the Layers panel. If there is no rule defined in the technology file, the system uses a system default. When you select *Create – Multipart Path*, you might see warning messages about no minimum width rule defined for the first valid layer listed in the *Layers* assistant.

3. To create a new MPP in the current layout window or to use an existing template, do one of the following.

   - To create a new MPP from scratch, without using an existing template:
     - When you do not want to save your MPP as a template, from the *MPP Template* field, select *New*.

       Choosing *New* resets all values in the Create Multipart Path and ROD Subpart forms to system defaults and sets the *Template Name* field to *New*.
     - When you want to save your MPP as a template, for *Template Name*, type a unique name, such as `myMPP`.
     - Check the *Layers* panel for the current layer and purpose for the master path and modify it, if desired.

       When you create a new MPP without using a template, you need to set the layer and purpose in the *Layers* panel of the Palette assistant to the value you want for the master path.

© 2023 Cadence Design Systems, Inc.

- To use an existing MPP template:
  - Select the name of the template you want from the *MPP Template* field.

    When you choose an existing MPP template, the system changes the current layer and purpose in the *Layers* panel to the layer and purpose defined in the MPP template for the master path.

    If the layer specified in the template is not a valid layer, the system displays a dialog box prompting if you want the system to add the master path layer to the *Layers* panel. If you do not, the system uses the current layer in the *Layers* panel.
  - If a dialog box prompts whether you want to add the template layer to the list of valid layers, click *OK* or *Cancel*, as required.

    If you want a different layer, you can change the layer in the *Layers* panel.
  - Check the *Layers* panel for the current layer and purpose for the master path, and modify it if desired.

Now you can modify the values in the *Create Multipart Path* form.

4. For *Name*, accept the default system-assigned name or type in a name that is unique in the current cellview.

   The *Name* field specifies a unique name for the ROD object information about the MPP in the layout window. You can access information about the MPP through multiple levels of hierarchy using this name. The default ROD object name consists of the prefix `path` followed by a number.

5. For *Technology Library*, select the library in which you want to save the MPP. The list displays all the technology libraries in the effective technology libraries, which are defined in the technologies database, that contain only the multipart path templates defined in the database. When a different technology library is selected in the *Technology Library* field, the *MPP Template* field is updated to list the templates contained in that library.

6. Select the *Choppable* check box if you want to make the master path and all subparts choppable. To make the master path non- choppable (subparts can be choppable or not), turn off *Choppable*.

7. For *Width*, do one of the following:
   - Type a positive integer or floating-point number.
   - To use the `minWidth` rule for the master path layer from the technology file, do not change the *Width* field.
   - If the `minWidth` rule is not defined in the technology file, and:

© 2023 Cadence Design Systems, Inc.

- This is the first time you have used this form in your current editing session and you do not specify a width, the system supplies a default and displays a warning saying that the default width is being used.
- You have used this form earlier in your current editing session, the previous value for this field is displayed and the system displays a warning.

8. For *End Type*, choose a value. If you want to specify a beginning or ending extension, you must choose *variable*.

The *Offset* and *Justification* fields are interdependent. *Justification* determines the part of the master path (master path centerline, left edge, or right edge) that is offset from the points you enter (the point list), relative to the direction of the master path. The direction of the master path is determined by the sequence in which you enter its points.

9. For *Justification*, select one of the following values:

- When *Offset* is `0`:
  - *center* to make the master path centerline coincident with the point list
  - *left* to make the right edge of the master path coincident with the point list
  - *right* to make the left edge of the master path coincident with the point list
- When *Offset* is not `0`:
  - *center* to offset the centerline of the master path from the point list
  - *left* to offset the left edge of the master path from the point list
  - *right* to offset the right edge of the master path from the point list

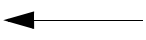
10. For *Offset*, do one of the following:

- To make the centerline of the master path coincident with the points you enter, leave *Offset* set to `0` and *Justification* set to *center*.
- To offset the master path from the point list, type a signed integer or floating-point number.

  The part of the master path that is offset from the point list is determined by the value of the *Justification* field. Whether you enter a positive or negative number depends

© 2023 Cadence Design Systems, Inc.

on where you want the master path in relation to the point list. The following table lists various combinations of Offset with respect to the position of the master path.

Direction of Point List Segment	Positive Offset	Negative Offset
Positive along X axis	Above	Below
Negative along X axis 	Below	Above
Positive along Y axis 	Left	Right
Negative along Y axis 	Right	Left

11. For *Begin Extension*, do one of the following:

- If you want the master path layer to begin at the first point you enter for the master path, leave *Begin Extension* set to `0`.
- If you want the master path layer to extend beyond the first point you enter for the master path, type a positive integer or floating-point number.

  You can specify an extension only if the *End Type* field is set to *variable*.

12. For *End Extension*, do one of the following:

- If you want the master path layer to end at the last point you enter for the master path, leave *End Extension* set to `0`.
- If you want the master path layer to extend beyond the last point you enter for the master path, type a positive integer or floating-point number.

13. For *Connectivity*, do one of the following:

- If you do not want to define connectivity, leave *Connectivity* set to *None* and you can define the subparts as described in Editing a Subpart of an MPP.
- If you want to associate the master path with a net or specify it as a pin, see Defining Connectivity for the Master Path.

Related Topics

Create Multipart Path Form

© 2023 Cadence Design Systems, Inc.

Flow for Creating Multipart Paths

Examples of Offsetting the Master Path

Defining Connectivity for the Master Path

You can choose to associate the master path with a net, and you can designate the master path as a pin.

Before you define the connectivity for a master path, ensure you have completed the procedure described in Defining the Master Path of an MPP.

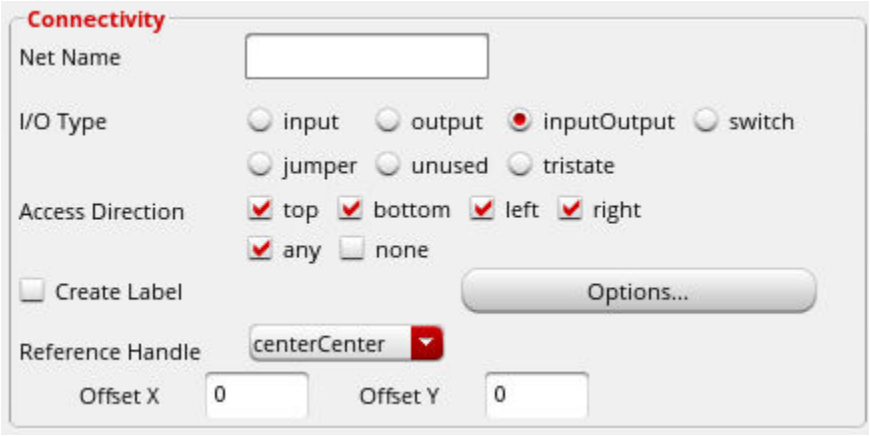
To define connectivity for a master path, do one of the following in the Create Multipart Path form:

- To assign the master path to a net, do the following:
  - **a.** From *Connectivity*, choose *Net*.

    The form expands to show the *Connectivity* section.
  - **b.** In *Net Name*, type the name of the net.

    If you want to assign the master path to a pin, continue with the next step; otherwise, you have completed assigning the master path to a net.
- To assign the master path to a pin, do the following:
  - **a.** From *Connectivity*, choose *Pin*.

    The form expands to show the net and pin connectivity fields in the *Connectivity* section.



© 2023 Cadence Design Systems, Inc.

b. In *Net Name*, type the name of the net.

When you choose *Pin*, you must enter a net name or the system beeps when you click in the cellview window to begin creating the MPP.

c. For *I/O Type*, choose one of the following: *input*, *output*, *inputOutput*, *switch*, or *jumper*.

d. For *Access Direction*, choose the parts of the pin to which routing can be connected. For example, top, bottom, left, and right.

e. Select *Create Label* if you want to create a text-display object to display the net name for the pin.

If pin names are not visible in the cellview, turn on *Pin Names* in the Display Options form.

f. For *Reference Handle*, choose where you want to align the pin name in relation to the master path or subpart.

You can offset the pin name from the object in the direction of the X and/or Y axis.

g. For *Offset X*, type a positive or negative floating-point number to offset the origin point of the pin name along the X axis from the point specified by *Reference Handle.*

h. For *Offset Y*, type a positive or negative floating-point number to offset the origin point of the pin name along the Y axis from the point specified by *Reference Handle.*

i. Use the *Options* button if you want to set pin name display options, such as font, height, layer, and justification on the Set Pin Label Text Style form.

The system uses the values of the following fields to calculate the position of the text-display object: *Reference Handle*, *Offset X*, *Offset Y*, and of the *Justification* field on the Set Pin Label Text Style form.

The changes you make to the Create Multipart Path form immediately affect the master path in the layout window.

For information about the overall steps involved in creating an MPP, see Flow for Creating Multipart Paths.

Related Topics

Create Multipart Path Form

© 2023 Cadence Design Systems, Inc.

Subparts of a Multipart Path

You can add any number of subparts to the current master path of a multipart path (MPP) in the layout window.

To create a subpart, you use the ROD Subpart form, which can be accessed by clicking the *Subpart* button in the Create Multipart Path form. You can choose to create an offset subpath, an enclosure subpath, or a set of subrectangles.

- Adding an Offset Subpath
- Adding an Enclosure Subpath
- Adding a Set of Subrectangles Subpart

If any subparts of the type you selected already exist, the ROD Subpart form displays one line of information for each subpart in the box at the top of the form.

While creating an MPP, you can choose to edit or delete a subpart as required.

- Editing a Subpart of an MPP
- Deleting a Subpart of an MPP

If you want to use the subpart values again, you need to save them as a template along with the master path.

Related Topics

Create Multipart Path Form

ROD Subpart Form

Flow for Creating Multipart Paths

Adding an Offset Subpath

You can define a subpath coincident with an edge of the master path, on the left or right side of the master path, or overlapping the master path. This is called an *offset subpath*.

You can specify the width of the offset subpath or let it default to the `minWidth` rule for the subpath layer from the technology file. The location of an offset subpath in relation to the master path depends on the values of the *Separation* and *Justification* fields, in relation to the direction of the master path.

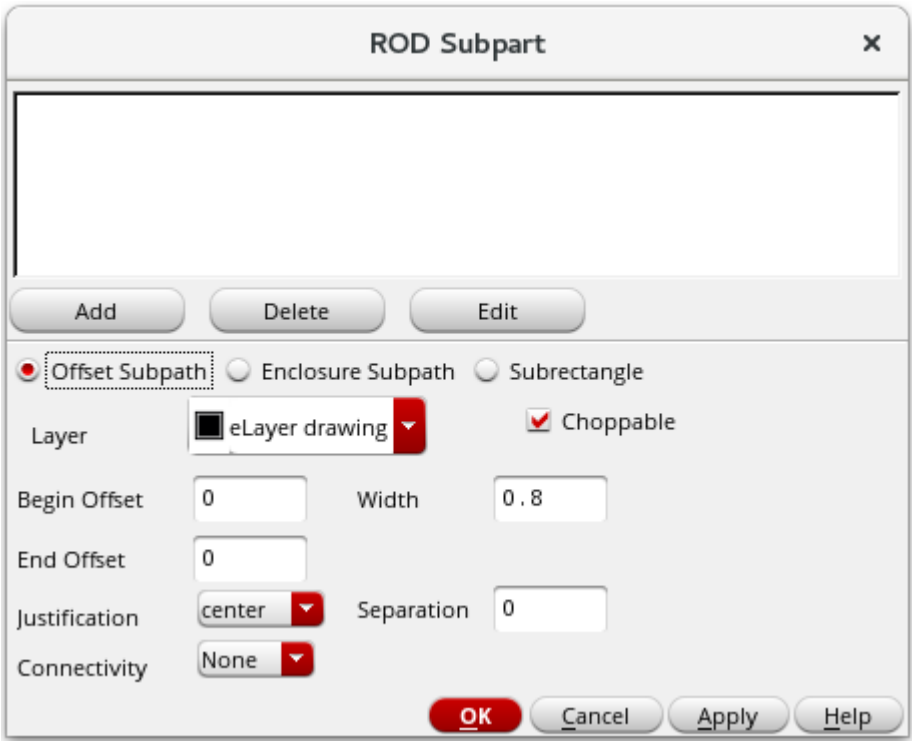
© 2023 Cadence Design Systems, Inc.

To add an offset subpath to the master path while creating a multipart path:

1. In the Create Multipart Path form, click the *Subpart* button.

   The ROD Subpart form appears.

2. Choose *Offset Subpath*.



3. For *Layer*, choose the layer for the offset subpath.
4. Select the *Choppable* check box if the master path is set to choppable; otherwise set the offset subpath to choppable or not choppable, as desired.
5. For *Begin Offset*, type a signed integer or floating-point number to specify the starting edge of the subpath in relation to the starting edge of the master path. A positive number extends the end of the subpath beyond the end of the master path; a negative number retracts the end of the subpath from the end of the master path.
6. For *End Offset*, type a signed integer or floating-point number to specify the ending edge of the subpath in relation to the ending edge of the master path. A positive number extends the end of the subpath beyond the end of the master path; a negative number retracts the end of the subpath from the end of the master path.
7. For *Width*, do one of the following:

   - ❑ Type a positive integer or floating-point number.

© 2023 Cadence Design Systems, Inc.

- If you want to use the `minWidth` rule for the subpath layer from the technology file, do not change the *Width* field.

  If the `minWidth` rule is not defined in the technology file, you must specify a width; otherwise, the width remains zero and no subpath is created.

8. For *Separation* and *Justification*, type a signed integer or floating-point number for *Separation* and choose a setting for *Justification*.

   Whether you enter a positive or negative number depends on where you want to define the subpath in relation to the master path.

9. For *Connectivity*, do one of the following:

   - If you do not want to define connectivity, leave *Connectivity* set to *None*.
   - If you want to associate the subpart with a net or specify it as a pin, see Defining Connectivity for a Subpart.

10. To add the new offset subpath to the window at the top of the ROD Subpart form, click *Add*.

    The system displays a line of data for the new offset subpath in the box.

11. To save the new offset subpath as part of the MPP, click *OK* or *Apply* in the *ROD Subpart* form.

    You must click *OK* or *Apply* to make the new offset subpath part of the MPP.

    You have defined a new enclosure subpath only for the MPP in the current layout window. If you want to add the new subpath to an MPP template, you will need to save your changes to the template.

You can also use the shortcut menu in the *Property Editor* assistant to add an offset subpath.

For information about the overall steps involved in creating an MPP, see Flow for Creating Multipart Paths.

Related Topics

Subparts of a Multipart Path

Create Multipart Path Form

ROD Subpart Form

© 2023 Cadence Design Systems, Inc.

Offset Subpaths

Examples of Offset Subpaths

Adding an Enclosure Subpath

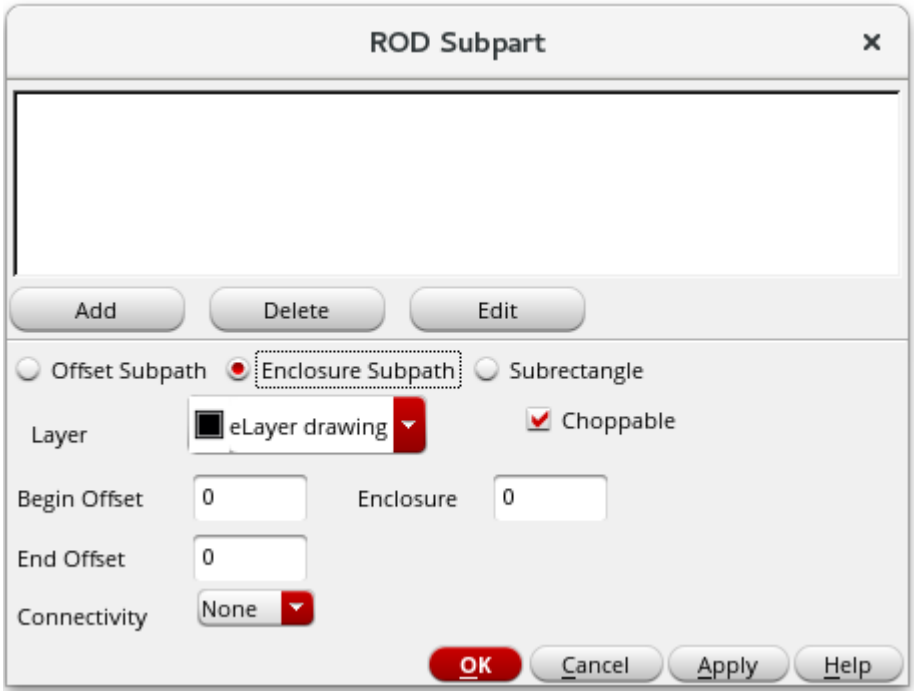
You can define a subpath with its centerline on the centerline of the master path and its width calculated using the width of the master path plus a positive or negative enclosure value. This is called an enclosure subpath. The enclosure determines by how much the subpath is enclosed by the master path or by how much the master path is enclosed by the subpath.

To add an enclosure subpath to the master path while creating a multipart path:

1. In the Create Multipart Path form, click the *Subpart* button.

   The ROD Subpart form appears.

2. Choose *Enclosure Subpath*.



3. For *Layer*, choose the layer for the enclosure subpath.
4. Select *Choppable* if the master path is set to choppable; otherwise, set the enclosure subpath to choppable or not choppable, as you wish.

© 2023 Cadence Design Systems, Inc.

5. For *Begin Offset*, type a signed integer or floating-point number to start the edge of the subpath before or after the starting edge of the master path.
6. For *End Offset*, type a signed integer or floating-point number to end the edge of the subpath before or after the ending edge of the master path.

   To calculate the width of an enclosure subpath, the system subtracts two times the enclosure value from the width of the master path:

   ```
   Width of Enclosure Subpath = Width of Master Path - (2 * Enclosure Value)
   ```
7. For *Enclosure*, type an integer or floating-point number for the enclosure of the subpath in relation to the master path.
   - To define a subpath that is narrower than the master path, type a positive number.
   - To define a subpath that is wider than the master path, type a negative number.
   - To use the `minExtensionDistance` rule for the subpath layer from the technology file, keep the default value.

     If `minExtensionDistance` for the subpath layer is not defined in your technology file and you do not type a value, the system sets the value to zero.
8. For *Connectivity*, do one of the following:
   - If you do not want to define connectivity, leave *Connectivity* set to *None*.
   - If you want to associate the subpart with a net or specify it as a pin, see Defining Connectivity for a Subpart.
9. To add the new enclosure subpath to the box at the top of the ROD Subpart form, click *Add*.

   The system displays a line of data for the new enclosure subpath in the box.
10. To save the new enclosure subpath as part of the MPP in the cellview, click *OK* or *Apply*.

    You must click *OK* or *Apply* to make the new enclosure subpath part of the MPP.

    You have defined a new enclosure subpath only for the MPP in the current layout window. If you want to add the new subpath to an MPP template, you need to save your changes to the template.

You can also use the shortcut menu in the *Property Editor* assistant to add an enclosure subpath.

For information about the overall steps involved in creating an MPP, see Flow for Creating Multipart Paths.

© 2023 Cadence Design Systems, Inc.

Related Topics

Create Multipart Path Form

ROD Subpart Form

Enclosure Subpaths

Examples of Enclosure Subpaths

Subparts of a Multipart Path

Sets of Subrectangles

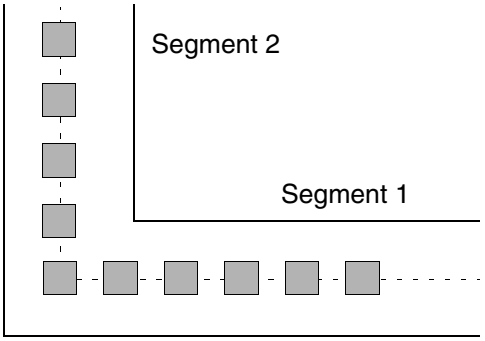
You can define one or more sets of subrectangles in relation to the centerline of any master path. A set can contain one or more subrectangles.

You can specify the width and length of the subrectangles or let them default to the `minWidth` rule for the subrectangle layer from the technology file. You can offset the centerline of the subrectangles from the master path centerline, and you can offset the edge of the first and last subrectangles from the ends of the master path.

You can define connectivity for a set of subrectangles. When you specify that a set of subrectangles is a pin, each rectangle in the set becomes a pin. In the layout editor, the whole multipart path (master path and all subparts) is treated as a single object.

The location of a set of subrectangles in relation to the master path depends on the values of the *Separation* and *Justification* fields, in relation to the direction of the master path.

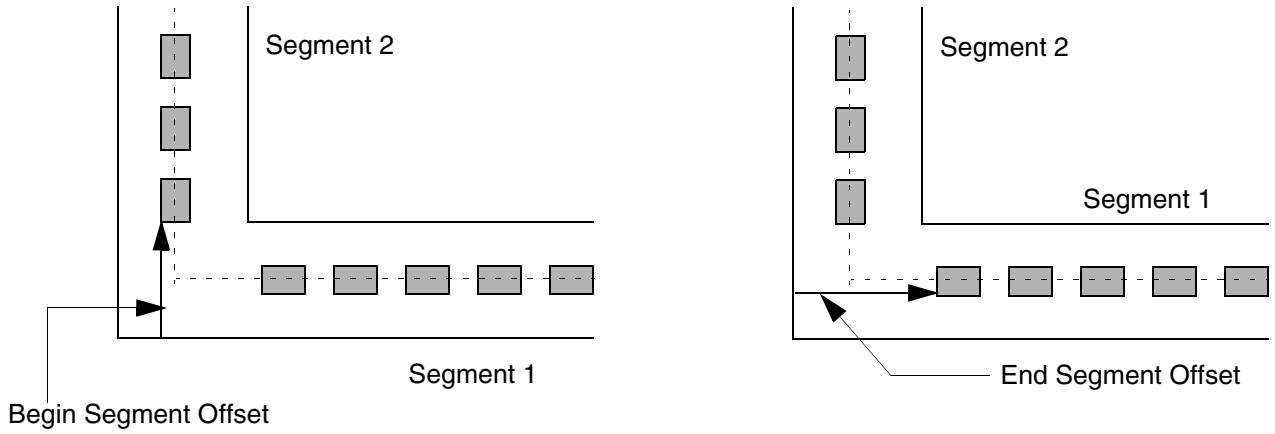
When you create a subpart consisting of a set of subrectangles, the system creates a subrectangle in the corner formed by each segment of the subrectangle subpath, and then fills in the rest of the segment with subrectangles, following the spacing you specify.



© 2023 Cadence Design Systems, Inc.

There are several reasons that you might want to keep rectangles out of the corners of your subrectangle subpaths. For example, when you specify multiple subrectangle subpaths, placing rectangles in the corners of segments might cause individual subrectangles to be out of alignment vertically or horizontally with the subrectangles in the other subrectangle subpaths.

If you do not want rectangles in the corners of segments, you can keep them out by specifying the *Begin Segment Offset* and *End Segment Offset* options in the ROD Subpart form. The system measures the segment offset values from the edge of the master path, as shown below.



When setting values for the segment offset options, keep in mind the following:

- The *Begin Segment Offset* value does not apply to the first segment; the position of the first rectangle in the first segment is determined by the value of the *Begin Offset* option.
- The *End Segment Offset* value does not apply to the last segment; the position of the last rectangle in the last segment is determined by the value of the *End Offset* argument.
- The segment offset arguments work in combination with the *Begin Offset* and *End Offset* arguments.
- If the segment begins before the full distance specified by the *Begin Segment Offset* option, then the *Begin Segment Offset* is not applied, and a subrectangle could be placed at the corner of the subrectangle subpath.
- If the segment ends before the full distance specified by the *End Segment Offset* option, then the *End Segment Offset* is not applied, and a subrectangle could be placed at the corner of the subrectangle subpath.

© 2023 Cadence Design Systems, Inc.

- When applying the Offset options, the system will not create a spacing error. If the first rectangle in a subrectangle subpath segment would be too close to the last rectangle in the previous segment, the system discards the last rectangle in the previous segment.

When you specify the offset options, you need to check the results to make sure the corners are as desired. If they are not, you should change the values of the offset arguments and try again.

Related Topics

ROD Subpart Form

Sets of Subrectangles

Flow for Creating Multipart Paths

Subparts of a Multipart Path

Adding a Set of Subrectangles Subpart

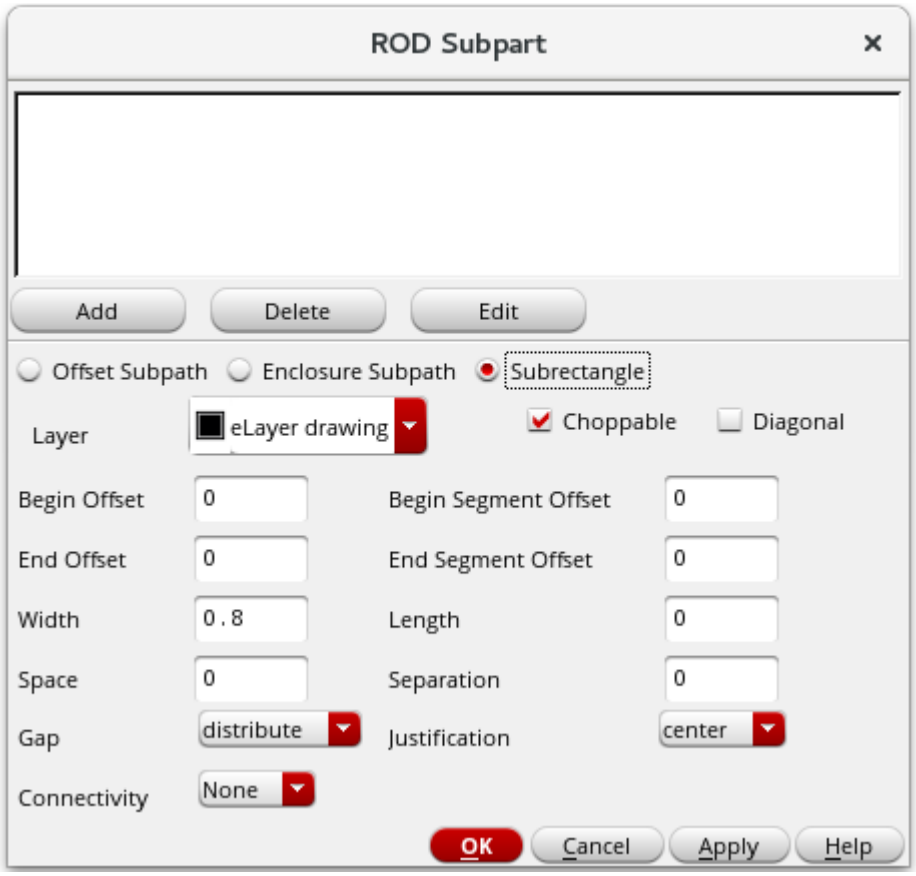
To add one or more sets of subrectangle subparts to the master path while creating a multipart path (MPP):

1. In the Create Multipart Path form, click the *Subpart* button.

   The ROD Subpart form appears.

© 2023 Cadence Design Systems, Inc.

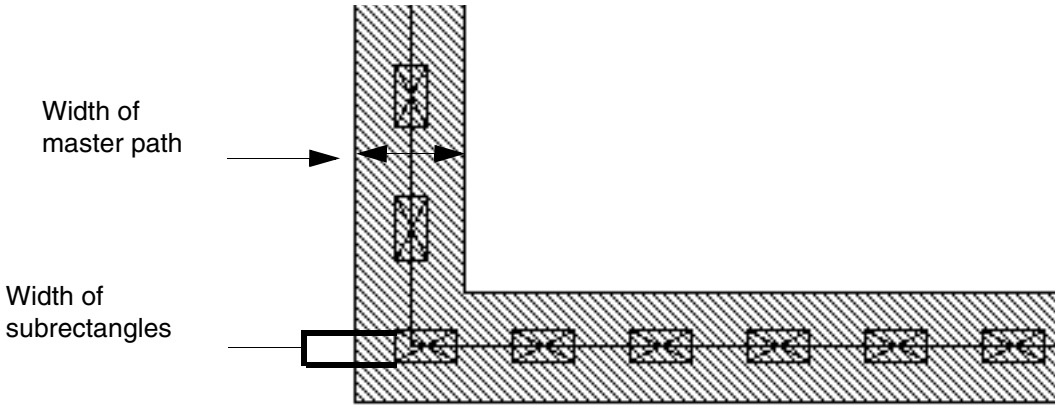
2. Choose *Subrectangle*.



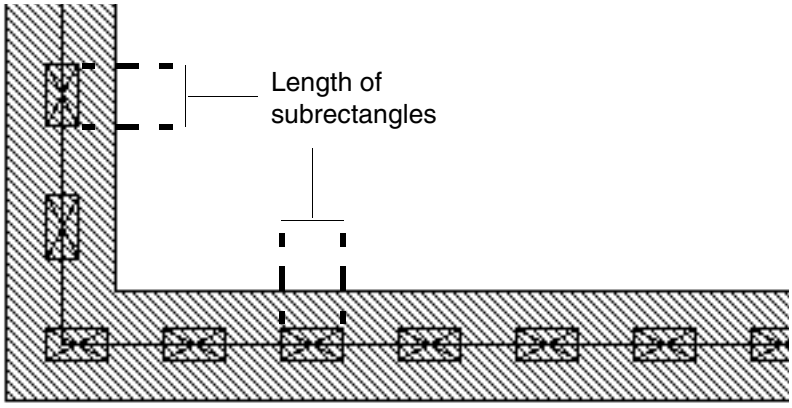
3. For *Layer*, choose the layer for the set of subrectangles.
4. Select *Choppable* if the master path is set to choppable; otherwise, set the subrectangles to choppable or not choppable, as desired.
5. Select *Diagonal* if you want to create subrectangles on 45-degree path segments. This option works only when all subrectangles on the path are square.
6. For *Begin Offset*, type a signed integer or floating-point number to start the first subrectangle before or after the starting edge of the master path.
7. For *End Offset*, type a signed integer or floating-point number to end the last subrectangle before or after the ending edge of the master path.
8. If you want to keep subrectangles out of the corners of segments, assign positive numbers for the *Begin Segment Offset* and *End Segment Offset*. Ensure you have read the concepts described in Sets of Subrectangles.

© 2023 Cadence Design Systems, Inc.

For an MPP, the width of a subrectangle is parallel to the width of the master path.



The length of a subrectangle is parallel to the master path centerline.



9. For *Width*, type a positive integer or floating-point number. Retain the default value if you want to use the `minWidth` rule for the subrectangle layer from the technology file.

10. For *Length*, type in a positive integer or floating-point number. To let the length default to the `minWidth` rule for the subrectangle layer from the technology file, do not change the *Length* field.

For each segment, you control the space between rectangles and after the last rectangle with the two fields *Gap* and *Space*. *Space* determines the minimum distance between rectangles. *Gap* controls where the system places any space that remains after the rectangles are placed the minimum distance apart.

11. For *Space*, type a positive integer or floating-point number for the distance between the edges of adjoining rectangles. To use the `minSpacing` rule for the subrectangle layer from the technology file, keep the default.

© 2023 Cadence Design Systems, Inc.

12. For *Gap*, choose *distribute* or *minimum*.

13. Type a signed integer or floating-point number for *Separation*, and choose a setting for *Justification*. Whether you enter a positive or negative number depends on where you want to define the set of subrectangles in relation to the master path.

When you specify that a set of subrectangles is a pin, each rectangle in the set becomes a pin. In the layout editor, the whole multipart path (master path plus all subparts) is treated as a single object.

14. For *Connectivity*, do one of the following:

- If you do not want to define connectivity, leave *Connectivity* set to *None*.
- If you want to associate the subpart with a net or specify it as a pin, see Defining Connectivity for a Subpart.

15. To add the new subrectangle to the box at the top of the ROD Subpart form, click *Add*.

The system displays a line of data for the new set of subrectangles in the box.

16. To save the new set of subrectangles as part of the MPP in the cellview, click *OK* or *Apply*.

You must click *OK* or *Apply* in the ROD Subpart form to make the new set of subrectangles part of the MPP.

You have defined a set of subrectangles only for the MPP in the current layout window. If you want to add the new set of subrectangles to an MPP template, you need to save your changes to a template.

You can also use the shortcut menu in the *Property Editor* assistant to add a subrectangle to MPP.

For information about the overall steps involved in creating an MPP, see Flow for Creating Multipart Paths.

Related Topics

Create Multipart Path Form

ROD Subpart Form

Examples of Sets of Subrectangles

Subparts of a Multipart Path

© 2023 Cadence Design Systems, Inc.

Defining Connectivity for a Subpart

You can define connectivity for any multipart path (MPP) subpart by associating it with a net. You can also designate a subpart as a pin. The changes you make do not affect the MPP in the layout window until you click *OK* or *Apply* in the ROD Subpart form.

You must type the name of the net in the ROD Subpart form for each subpart that has connectivity. If you do not type a name, the subpart is not associated with a net, and the system beeps when you click to start creating the MPP in a cellview window.

Before you add the connectivity for a subpart, ensure you have defined a subpart. For information, see Subparts of a Multipart Path.

To define the connectivity for a subpart, do one of the following in the ROD Subpart form:

- To assign the subpart to a net:
  1. For *Connectivity*, choose *Net*.
  2. The form expands to show the *Connectivity* section.
  3. In *Net Name*, type the name of the net.
  4. To apply the net assignment to the subpart of the MPP in the layout window, click *OK* or *Apply*.
- To assign the subpart to a pin:
  1. For *Connectivity*, choose *Pin*.

     The form expands to show both the net and pin connectivity fields.
  2. For *Net Name*, type the name of the net.

     You must enter a net name for every subpart that has pin connectivity or the system beeps when you click in the cellview window to begin creating the MPP.
  3. For *I/O Type*, choose one of the following: *input*, *output*, *inputOutput*, *switch*, or *jumper*.
  4. For *Access Direction*, choose the parts of the pin to which routing can be connected.
  5. Select *Display Terminal Name* if you want to create a text-display object to display the net name for the pin.

     If pin names are not visible in the cellview, enable the *Pin Names* option in the Display Options form.

© 2023 Cadence Design Systems, Inc.

The system uses the values of four fields to calculate the position of the text-display object: *Reference Handle*, *Offset X*, *Offset Y* in the Rod Subpart form and the *Justification* field on the Set Pin Label Text Style form.

f. Select *Display Terminal Name Option* if you want to set pin name display options, such as font, height, layer, and justification on the Set Pin Label Text Style form. To set the options, use the *Options* button.

g. For *Reference Handle*, choose where you want to align the pin name in relation to the master path or subpart.

You can offset the pin name from the object in the direction of the X and/or Y axis.

h. For *Offset X*, type a positive or negative floating-point number to offset the origin point of the pin name along the X axis from the point specified by *Reference Handle.*

i. For *Offset Y*, type a positive or negative floating-point number to offset the origin point of the pin name along the Y axis from the point specified by *Reference Handle.*

j. To save the subpart values as part of the MPP in the layout window, click *OK* or *Apply* in the ROD Subpart form.

To make all the current values in the ROD Subpart form, including the connectivity values, part of the MPP in the layout window, you must click *OK* or *Apply*.

The system changes the MPP in the layout window to include the subpart connectivity.

For information about the overall steps involved in creating an MPP, see Flow for Creating Multipart Paths.

Related Topics

Create Multipart Path Form

ROD Subpart Form

Subparts of a Multipart Path

Editing a Subpart of an MPP

The subparts of a multipart part (MPP) are listed at the top of the ROD Subpart form. You can change an existing subpart by changing the values for the selected subpart in the form.

Remember, you must make all edits to subparts before entering the last point of your MPP on the layout canvas. You cannot change an MPP after clicking on its last point; instead, you

© 2023 Cadence Design Systems, Inc.

must delete the whole MPP and re-create it. It is safest to save your form values as a template before drawing an MPP in your cellview. Then, if you need to edit a subpart, you can re-create the MPP from the template and use the ROD Subpart form to edit the subpart.

To edit a subpart while you are creating an MPP:

1. In the Create Multipart Path form, click the *Subpart* button.

   The ROD Subpart form appears.

   Existing subparts are listed in the scroll window at the top of the form.

2. Select the subpart you want to edit.

   The system fills in the fields of the ROD Subpart form with the values of the subpart you selected.

3. Change the values as desired.

4. To make your changes reflect in the data listed for the subpart in the box at the top of the ROD Subpart form, click *Edit*.

   The system changes the selected subpart data to match the new form values.

5. To make your subpart changes reflect in the MPP in the cellview, click *OK* or *Apply* in the *ROD Subpart* form.

   So far, your changes affect only the MPP in the current layout window. If you want your changes to affect an MPP template, you need to save your changes to the template.

Related Topics

Create Multipart Path Form

ROD Subpart Form

Subparts of a Multipart Path

Flow for Creating Multipart Paths

Deleting a Subpart of an MPP

You can delete any subpart of a multipart path (MPP). Subparts are listed at the top on the ROD Subpart form.

© 2023 Cadence Design Systems, Inc.

Remember, you must make all deletions to subparts before entering the last point of your MPP on the layout canvas. You cannot change an MPP after clicking on its last point; instead, you must delete the whole MPP and re-create it. It is safest to save your form values as a template before drawing an MPP in your cellview. Then, if you need to delete a subpart, you can re-create the MPP from the template and use the ROD Subpart form to delete the subpart.

To delete a subpart while creating an MPP:

1. In the Create Multipart Path form, click the *Subpart* button.

   The ROD Subpart form appears.

   Existing subparts are listed in the box at the top of the form.

2. Select the subpart you want to delete.
3. To delete the selected subpart from the data listed at the top of the ROD Subpart form, click *Delete*.

   The subpart is deleted from the subparts listed in the form.

4. To delete the subpart from the MPP in the cellview, click *OK* or *Apply* in the ROD Subpart form.

   The selected subpart is deleted from the MPP in the current cellview.

   Your changes affect only the MPP in the current layout window. If you want to remove the deleted subpart from an MPP template, you need to save the current form values to the MPP template, overwriting the old values.

You can also use the shortcut menu in the *Property Editor* assistant to delete a subpath.

Related Topics

Create Multipart Path Form

ROD Subpart Form

Subparts of a Multipart Path

Flow for Creating Multipart Paths

© 2023 Cadence Design Systems, Inc.

Drawing a Multipart Path in the Layout Window

You can you draw a multipart path (MPP) on your layout canvas based on the template selected or the values specified in the Create Multipart Path and ROD Subpart forms.

MPP templates do not contain net names because each MPP you create from the same template could be on a different net. When you choose an MPP template that has pin connectivity defined, the system displays a message at the bottom of the cellview window prompting you to enter a valid net name. If the MPP template does not have pin connectivity, and you want to associate the MPP with a net, you need to type the net name in the *Net Name* field for the master path or for any subpart, as desired.

When there is pin connectivity, you must enter the name of the net in the *Net Name* field for each part (master path and/or subparts) before you begin to create the MPP in the layout window. If you click in the layout window before entering the net names, the system beeps.

As you draw an MPP, you can change the values in the Create Multipart Path and ROD Subpart forms and see immediate changes to the MPP, until you double-click to enter the last point. After you enter the last point to complete an MPP, you can no longer change that MPP using the Create forms. You will need to use the Edit MPP Path Properties form to modify a completed MPP.

To draw an MPP in the layout window:

1. Ensure you have specified the required values in the Create Multipart Path and ROD Subpart forms.
2. Click on the canvas to enter the first point.
3. Move the pointer to the next point and click on the canvas.
4. Continue to move the pointer and click to enter points.
5. To enter the last point of the path, either double-click on the canvas or press `Enter`.

You have completed drawing the MPP in your layout window.

Related Topics

Create Multipart Path Form

Flow for Creating Multipart Paths

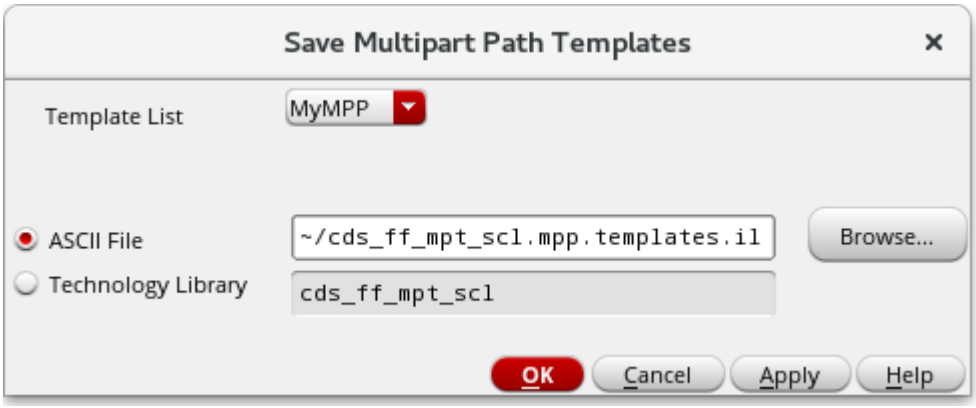
© 2023 Cadence Design Systems, Inc.

Multipart Path Values Saved as a Template

Whenever you create a new type of multipart path (MPP) that you might want to use again, you can save the values currently in the Create Multipart Path and ROD Subpart forms as a template with a unique name. This lets you create the same or similar MPPs. You can save MPPs as templates in the following ways:

- Save the MPP template to an ASCII file in your local hierarchy.
- Save the MPP template to the temporary technology file in the virtual memory.

  If you have the write permission, you can also save the MPP to your technology library on the disk.



Instead of using a graphical user interface to define an MPP, you can create an ASCII file to define one or more MPP templates by using the `leDefineMPPTemplate` function.

Related Topics

Saving MPP Templates to an ASCII File

Saving MPP Templates to Your Technology Library

Saving MPP Templates to Your Technology Library on the Disk

Loading an ASCII Multipart Path Template File

Flow for Creating Multipart Paths

© 2023 Cadence Design Systems, Inc.

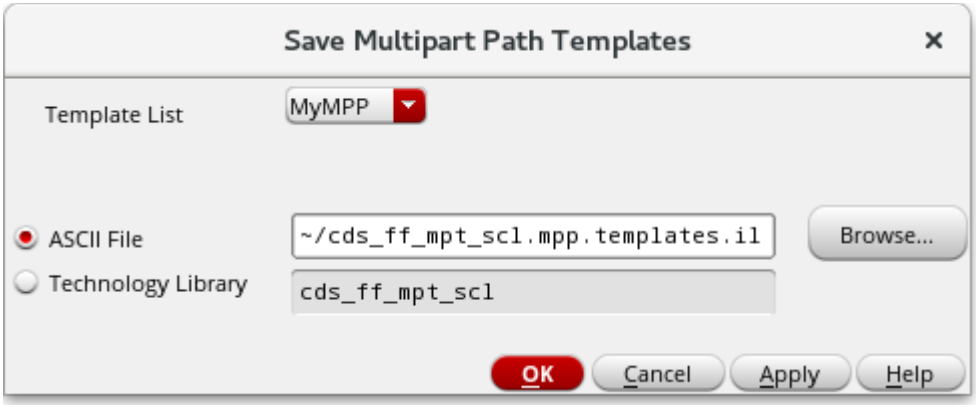
Saving MPP Templates to an ASCII File

When you save a multipart path (MPP) to an ASCII file, you can enter a new file name or overwrite an existing file. The system does not append to an existing ASCII file.

To save to one or all MPP templates to an ASCII file:

1. In the Create Multipart Path form, click the *Save Template* button.

   The Save Multipart Path Templates form appears.



2. From *Template List*, choose the name of the MPP template you want to save or choose *All Templates* to save every template in the list.
3. Select *ASCII File*.

   If you use the name of an existing ASCII file, it is overwritten.
4. Type the ASCII file name, or path and file name, or keep the default.

   If you want to designate a particular value as the default for the *ASCII File* field, you can set the mppASCIIFileName environment variable.
5. Click *OK*.

   If the system overwrites an existing ASCII file, the following message appears in the CIW:

   `Template templateName is replacing an existing template by the same name.`

When you save an MPP to an ASCII file, the system also updates the temporary technology file in the virtual memory.

Whenever you load multipart path (MPP) templates from ASCII files, they overwrite the templates in the technology file in the virtual memory that have matching names. Later, when you save changes to your technology library on the disk, the templates in the virtual memory overwrite templates with matching names in the technology library on the disk. If you do not

© 2023 Cadence Design Systems, Inc.

want this to happen, see Problem: Warning in CIW: MPP template name is replacing an existing MPP template.

Related Topics

Create Multipart Path Form

Multipart Path Environment Variables

Flow for Creating Multipart Paths

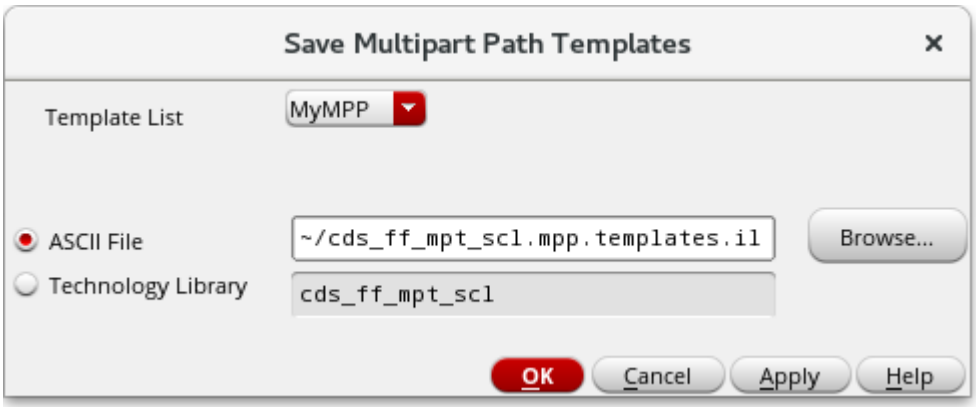
Saving MPP Templates to Your Technology Library

The Save Multipart Path Templates form allows you to save the template definitions to an ASCII file or directly to the technology library you select in the *Technology Library* field. If you do not have write permission to the technology library, and you choose the *Technology Library* option, the field is grayed out and will not allow you to save the templates. Choose the *ASCII File* option instead.

To save to one or all templates to the temporary technology file in virtual memory:

1. In the Create Multipart Path form, click the *Save Template* button.

   The Save Multipart Path Templates form appears.



2. From *Template List*, choose the name of the MPP template you want to save or choose *All Templates* to save every template in the list.
3. Select the *Technology Library* radio button.
4. Click *OK*.

© 2023 Cadence Design Systems, Inc.

To make the template part of your binary technology library on disk, you must save your technology library changes before you exit the editing session.

Related Topics

Create Multipart Path Form

Save Multipart Path Templates Form

Flow for Creating Multipart Paths

Saving MPP Templates to Your Technology Library on the Disk

If you have not saved the changes to your binary technology library during your editing session, the system displays a dialog box when you exit the software prompting whether you want to save the changes. You can save your template changes to the disk while exiting the session or you can save them to the disk now.

To save the changes to your binary technology library on the disk:

1. In the CIW, choose *Tools – Technology File Manager*.

   The Technology File Manager window appears.

2. Click *Save*.

   The Save Technology File form appears.



3. From the *Technology Library* list, choose the name of the library to which you want to save and click *OK*.

   The Technology File Save dialog box opens, prompting you to confirm whether you want to save the specified technology library to the disk.

4. Click *Yes*.

© 2023 Cadence Design Systems, Inc.

Related Topics

Loading an ASCII Multipart Path Template File

Flow for Creating Multipart Paths

Loading an ASCII Multipart Path Template File

You can access multipart part (MPP) templates that were saved in ASCII files by loading the ASCII files from the Create Multipart Path form. The templates in each file you load are added to the MPP templates in the technology file in the virtual memory and added to the list of template names in the *MPP Template* field of the Create Multipart Path form.

To load an ASCII MPP template file:

1. In the Create Multipart Path form, click *Load Template*.

   The Load Multipart Path Template form appears.



2. In the *Template File* field, type the path and file name for the ASCII MPP Template file you want to load and click *OK*.

   If you enter an invalid technology file name, the system displays a warning message in the CIW and no templates are loaded. Use the *Browse* button if you need assistance locating the template file name.

The system does not provide a default for the *Template File* field on the Load Multipart Path Template form. However, you can designate a file name or path and file name as the default for the *Template File* field on the Load Multipart Path Template form by setting the mppASCIIFileName environment variable.

When you load an ASCII MPP template file, the system checks the names of the ASCII MPP templates against the names of MPP templates that already exist in the technology file in the virtual memory. If there is a name conflict, the system displays a message in the CIW stating that an MPP template in the ASCII file is replacing an existing MPP template. The existing template is overwritten in the virtual memory only. If you do not want to overwrite the original MPP template in your binary technology library on the disk, do not save changes to the technology file. You can also avoid overwriting existing templates by changing the duplicate

© 2023 Cadence Design Systems, Inc.

names in your ASCII file. For more information, see Problem: Warning in CIW: MPP template name is replacing an existing MPP template.

Related Topics

Create Multipart Path Form

Load Multipart Path Template Form

Multipart Path Values Saved as a Template

Multipart Path Environment Variables

© 2023 Cadence Design Systems, Inc.

Creating an MPP Guard Ring

The following tasks are involved in creating a multipart path (MPP) guard ring:

- Loading a technology file into your local directory.
- Creating an MPP template.

  An MPP guard ring template must contain at least one subrectangle, and it should not have any connectivity in it.
- Using the MPP template in the Guard Ring form and fine-tuning other settings.

To create an MPP guard ring when you have an MPP template available:

1. In the layout window, choose *Create – MPP Guard Ring.*

   The Create Guard Ring form appears.
2. From *Guard Ring Template, s*elect the MPP template you want to use to create the guard ring.
3. Select the *Technology Library* in which the selected template is defined.
4. Specify the connectivity information for the MPP guard ring in the *Net Name* field.
5. Adjust the value in the *Enclose by* field to set the distance between the MPP guard ring and the selected objects.
6. Select *Rectangular* or *Rectilinear* shape for the MPP guard ring.
7. On the canvas, select the objects around which the MPP guard ring will be created. A preview of the guard ring appears around the selected objects.
8. Click *Apply* to place the MPP guard ring around the selected objects.

Related Topics

Guard Ring Form

Loading an ASCII Multipart Path Template File

Defining the Master Path of an MPP

Flow for Creating Multipart Paths

© 2023 Cadence Design Systems, Inc.

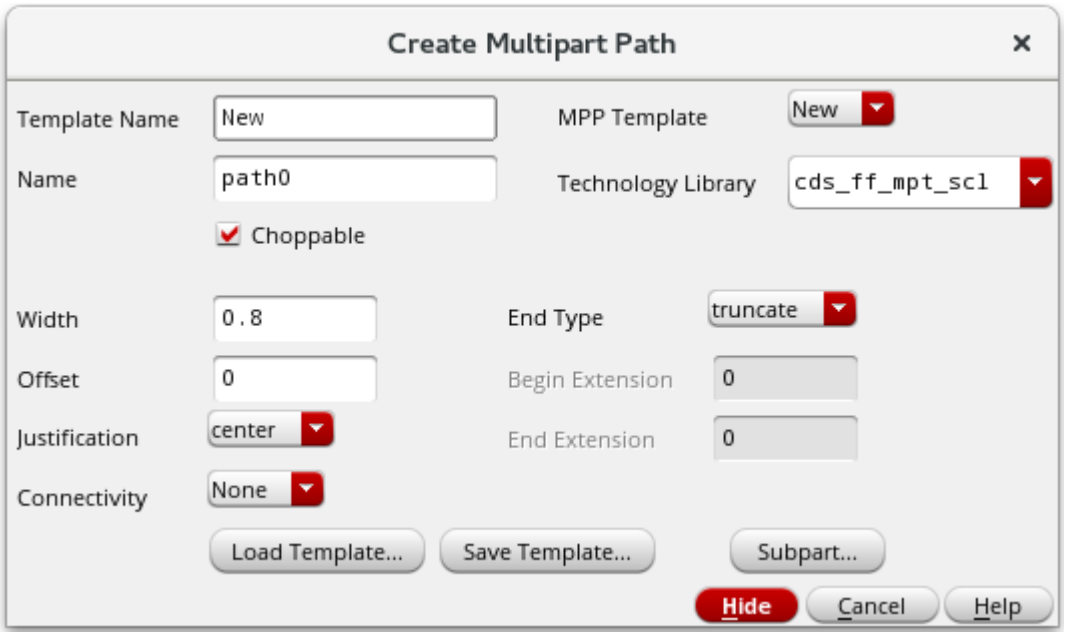
Multipart Path Environment Variables

You can set the following environment variables to work with multipart paths:

- mppTemplate Environment Variable
- mppASCIIFileName Environment Variable

mppTemplate Environment Variable

You can set mppTemplate to control the default values for the fields in the Create Multipart Path form the first time the form is opened during an editing session. The system default for `mppTemplate` is *New*. When you open the Create Multipart Path form, the system sets the *Template Name* and *MPP Template* fields to *New* and the remaining fields to their system default values. Whenever you click *New* in the *MPP Template* field, the system resets all the fields in the Create Multipart Path and ROD Subpart forms to the system defaults.



To preset the *Template Name* and *MPP Template* fields to the name of an MPP template you have already saved, set the `mppTemplate` environment variable to the template name. Then when you open the Create Multipart Path form for the first time during an editing session, the system sets the *Template Name* and *MPP Template* fields to the name of your template and uses the template values as defaults for the remaining fields.

When you choose a different value for the *MPP Template* field in the Create Multipart Path form, the system automatically resets the values of all the fields in the Create Multipart Path

© 2023 Cadence Design Systems, Inc.

and ROD Subpart forms and updates the `mppTemplate` environment variable to match the template name you chose. For example:

- If the *MPP Template* field is set to *New*, which is the default, and you change it to `guardring`, the system resets all form fields to the values defined for the `guardring` template and updates the `mppTemplate` environment variable to `guardring`.
- Conversely, if the value of the *MPP Template* field is set to `guardring` and you change it to *New*, the system resets all fields to the system defaults and updates the `mppTemplate` environment variable to *New*.

mppASCIIFileName Environment Variable

You can set the mppASCIIFileName editor environment variable to control default values for the first time you open the following forms during an editing session:

- The *ASCII File* field in the Save Multipart Path Templates form
- The *Template File* field in the Load Multipart Path Template form

You can set this environment variable to a filename or to a path and filename. When you do not specify a path, the system uses your current directory.

The system lets you set `mppASCIIFileName` to a filename that does not exist yet. When you save a template to an ASCII file, the system creates the file. However, if you try to load before the file is created, the system displays a warning.

When the `mppASCIIFileName` environment variable is undefined or set to `nil`,

- For the *ASCII File* field in the Save Multipart Path Templates form, the first time you open the form, the system supplies a default value that is based on your technology library name followed by `.mpp.templates.il`. For example, `myTechLib.mpp.templates.il`.

  You can change the *ASCII File* field by typing in it. If you do not type the `.il` suffix, the system appends it.
- For the Load Multipart Path Template form, the system does not supply a default value.

When you set the `mppASCIIFileName` environment variable to a specific filename or path and filename, the system supplies your value as the default for both the *ASCII File* field in the Save Multipart Path Templates form and the *Template File* field in the Load Multipart Path Template form.

When you change the value of the *ASCII File* field in the Save Multipart Path Templates form, the system automatically resets the value of the *Template File* field in the Load Multipart Path

© 2023 Cadence Design Systems, Inc.

Template form to match and also updates the `mppASCIIFileName` environment variable to match the value. However, if you change the *Template File* field in the Load Multipart Path Template form, the system does not change the value of the *ASCII File* field in the Save Multipart Path Templates form and does not change the `mppASCIIFileName` environment variable.

Related Topics

Create Multipart Path Form

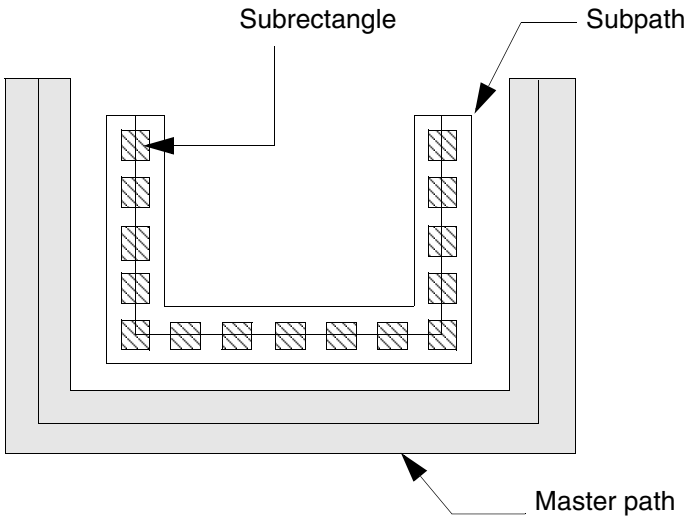
Load Multipart Path Template Form

Save Multipart Path Templates Form

Multipart Path Selection

A multipart path (MPP) is a single object composed of multiple parts on the same or different layers. The parts are a single master path and one or more subparts. The master path is the primary part; subparts exist in relation to the master path. A subpart can be an offset subpath, an enclosure subpath, or a set of subrectangles. Individual subrectangles are not individual shapes but are part of a specific set of subrectangles.

The following figure shows an MPP that has one subpath and one set of subrectangles. Both the subpath and the set of subrectangles are offset from the master path.



© 2023 Cadence Design Systems, Inc.

Selecting multipart paths is similar to selecting regular paths: you point and click or create a selection box in a similar manner. In general, you cannot select subparts separately from the master path; however, you can select the chopped end of subpaths to stretch them.

- In full selection mode, when you select any part of a multipart path, the whole multipart path is selected. The master path is highlighted on the current selection layer, while subparts are highlighted on a different layer. This lets you see which part is the master path and which parts are subparts.
- In partial selection mode, you can select an end, segment, or vertex of the master path. In this mode, you can select the chopped ends of one or more subpaths and stretch them. When you stretch a chopped subpath, all other choppable subparts are stretched too.

You cannot directly select a set of subrectangles, but selecting and stretching the chopped end of a subpath causes all chopped parts in the multipart path to be regenerated, including chopped sets of subrectangles. The system regenerates subrectangles along orthogonal segments only.

Related Topics

Selecting a Whole Multipart Path

Selecting Multipart Path Ends

Selecting Chopped Subpath Ends in Multipart Paths

Selecting Multipart Path Segments

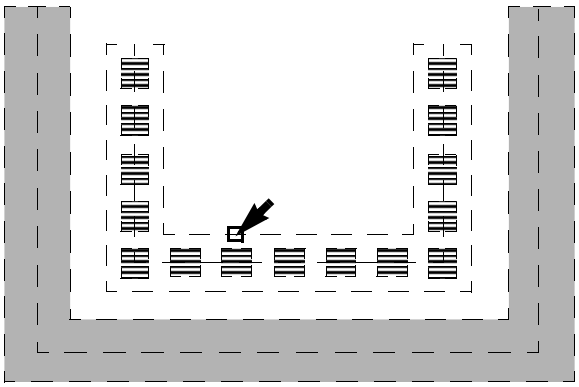
Selecting a Multipart Path Vertex

Selecting a Whole Multipart Path

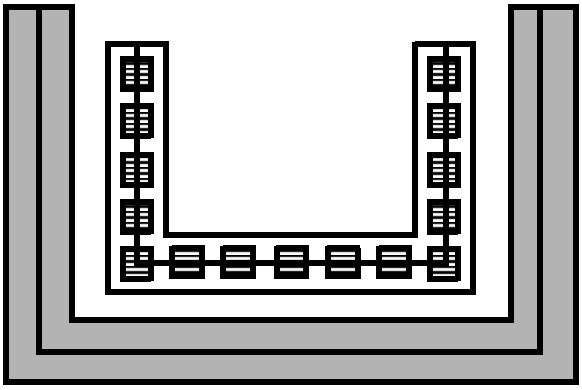
In full selection mode, selecting multipart paths is similar to selecting regular paths. When you select any part of a multipart path, the whole multipart path is selected. The master path is highlighted on the current selection layer, while subparts are highlighted on a different layer. This lets you see which part is the master path and which parts are subparts.

© 2023 Cadence Design Systems, Inc.

To select a whole multipart path in full selection mode, point to any part of the multipart path and click on it. As you point, the whole path is highlighted with dotted lines.



Point to any part of the multipart path and click.



Highlighting shows the whole multipart path is selected.

Related Topics

Multipart Path Selection

Stretching Multipart Paths

Selecting Multipart Path Ends

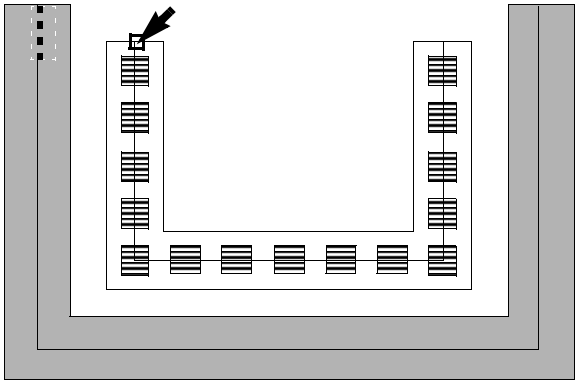
In partial selection mode, you can select an end of a master path in a multipart path. You cannot select the end of a subpart, except for chopped subpath ends. If you attempt to select the non-chopped end of a subpart, the nearest end of the master path is selected.

To select an end of a master path, do one of the following in partial selection mode:

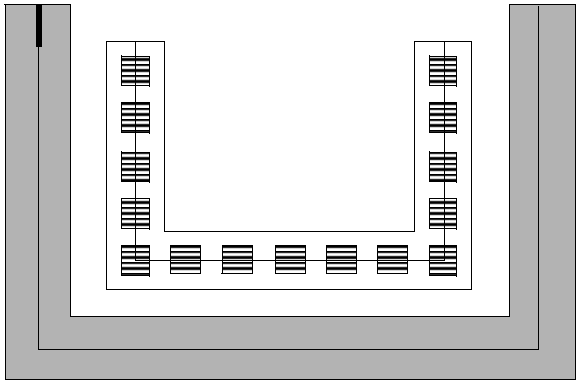
- Point to an end of the master path or an end of a subpath and click on it.

© 2023 Cadence Design Systems, Inc.

As you point, the nearest end of the master path is highlighted with dotted lines.

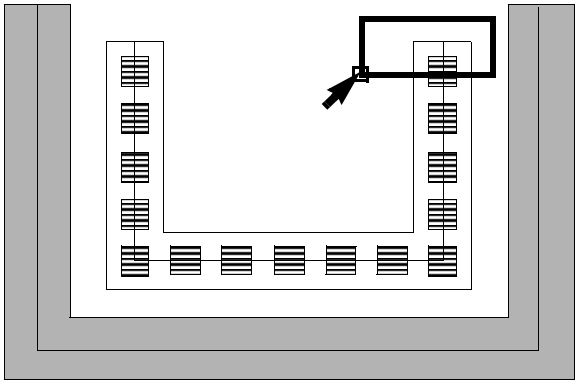


Point to an end and click on it.

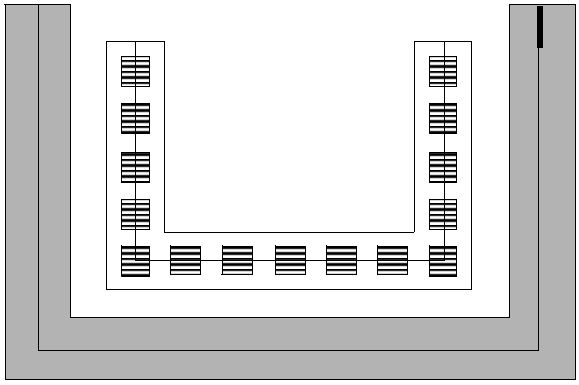


Highlighting shows the nearest master path end is selected.

- Create a selection box around an end of the master path or around an end of a subpath.



Create a selection box around an end of the master path or subpath.



Highlighting shows the nearest master path end is selected.

Even in partial selection mode, if you select one or more whole subrectangles, the whole MPP is selected.

Related Topics

Multipart Path Selection

© 2023 Cadence Design Systems, Inc.

Selecting Chopped Subpath Ends in Multipart Paths

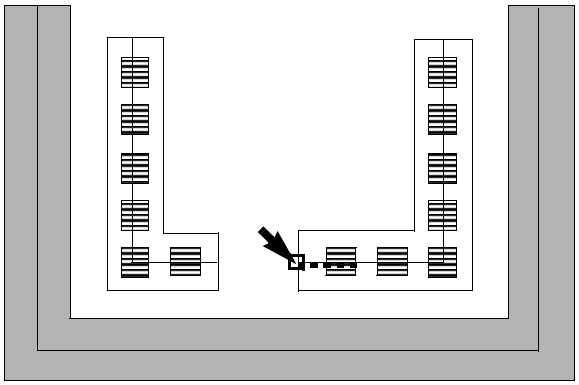
In partial selection mode, you can select the ends of one or more chopped subpaths in a multipart path and stretch them.

You cannot directly select a set of subrectangles, but selecting and stretching the chopped end of a subpath causes all chopped parts in the multipart path to be regenerated, including chopped sets of subrectangles. The system regenerates subrectangles along orthogonal segments only.

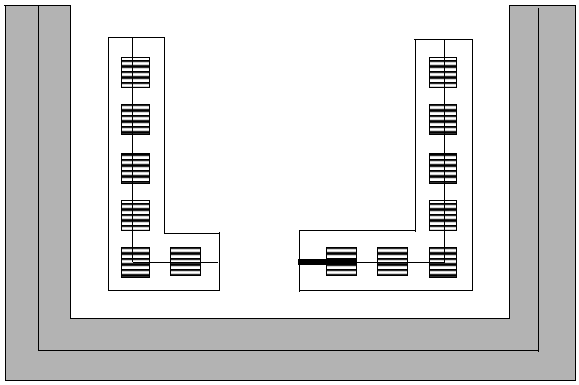
To select the end of one chopped subpath, do one of the following in partial selection mode:

- In partial selection mode, point to a chopped end and click on it.

  As you point, the chopped end is highlighted with dotted lines.



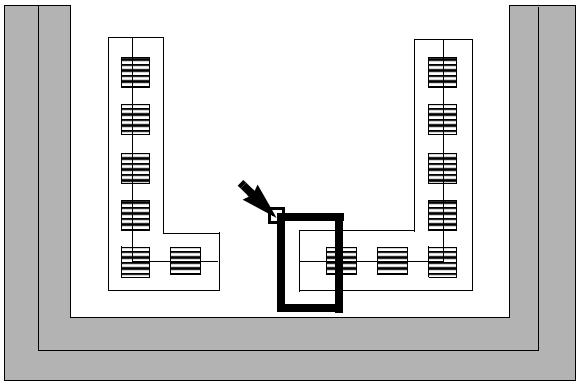
Point to a chopped subpath end and click.



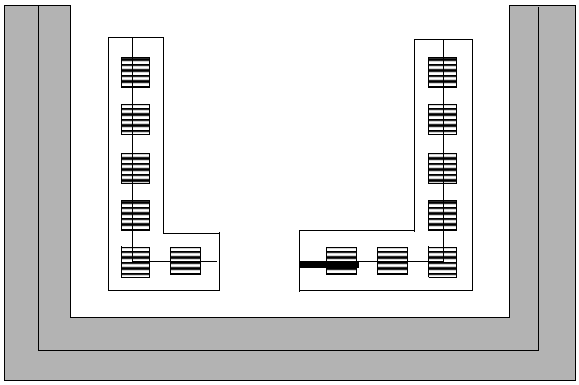
Highlighting shows the chopped subpath end is selected.

© 2023 Cadence Design Systems, Inc.

- Create a selection box around the chopped end of a subpath.



Create a selection box around a chopped end of a subpath.



Highlighting shows the chopped subpath end is selected.

Related Topics

Multipart Path Selection

Stretching Multipart Paths

Selecting Multipart Path Segments

In partial selection mode, you can select one or more segments of the master path in a multipart path. You cannot select segments of a subpart. If you attempt to select a subpart segment, the nearest segment of the master path is selected.

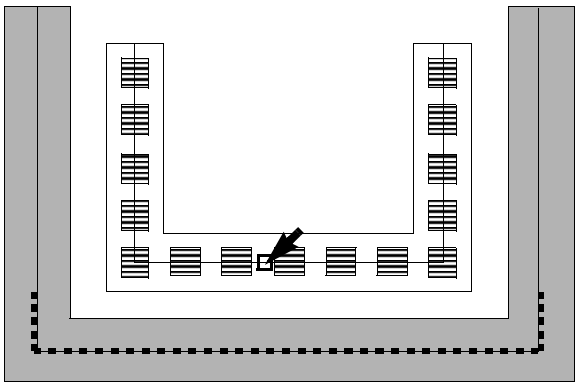
In partial selection mode, if you click the edge of a subrectangle, the nearest master path segment is selected.

To select a single master path segment, do one of the following in partial selection mode:

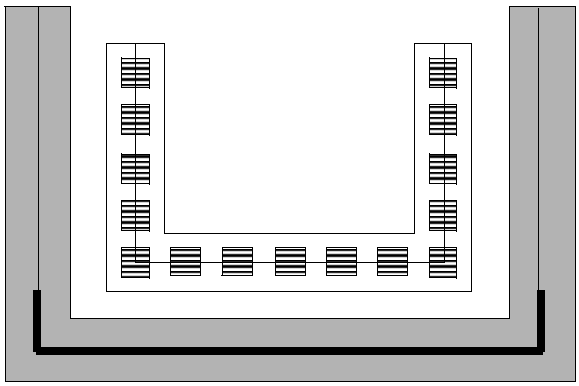
- Point to the middle of the centerline in a master path segment or subpath segment and click on it.

© 2023 Cadence Design Systems, Inc.

As you point, the nearest master path segment is highlighted with dotted lines.

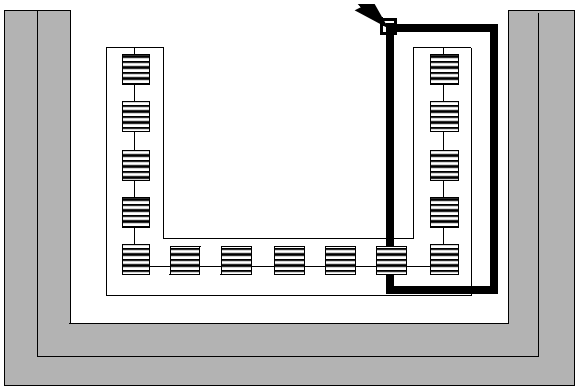


Point to the centerline in a path segment and click.

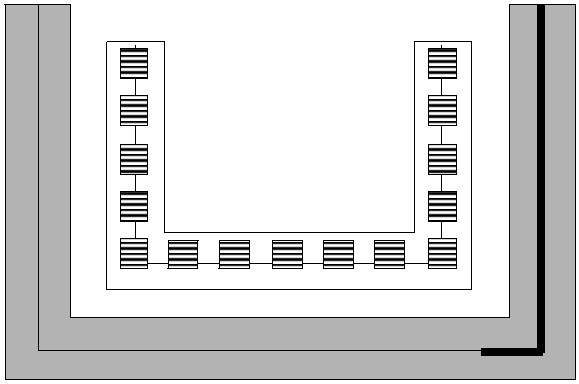


Highlighting shows the (nearest) master path segment is selected.

- Create a selection box around a master path or subpath segment. Make sure both end points of the segment centerline are inside the box.



Create a selection box around the centerline of a path segment.

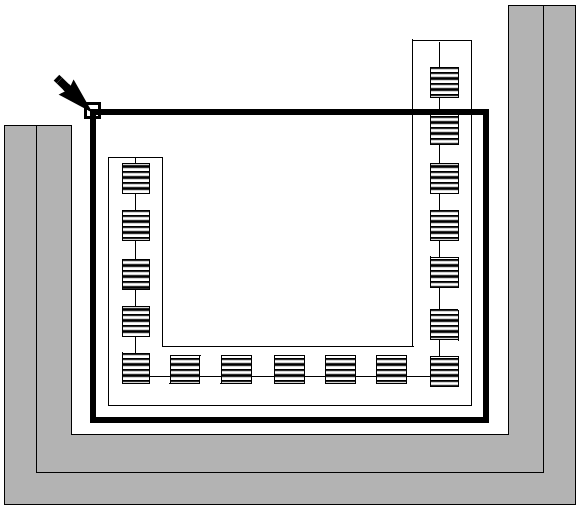


Highlighting shows the (nearest) master path segment is selected.

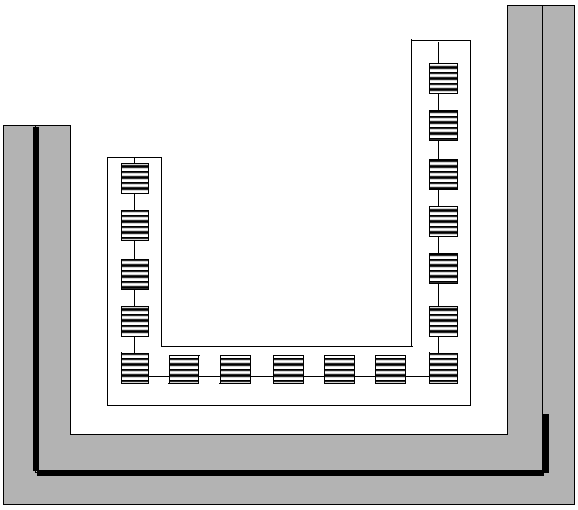
To select more than one master path segment, do one of the following in partial selection mode:

© 2023 Cadence Design Systems, Inc.

- Create a selection box around multiple master path or subpath segments. Make sure the end points of each segment centerline are inside the box.



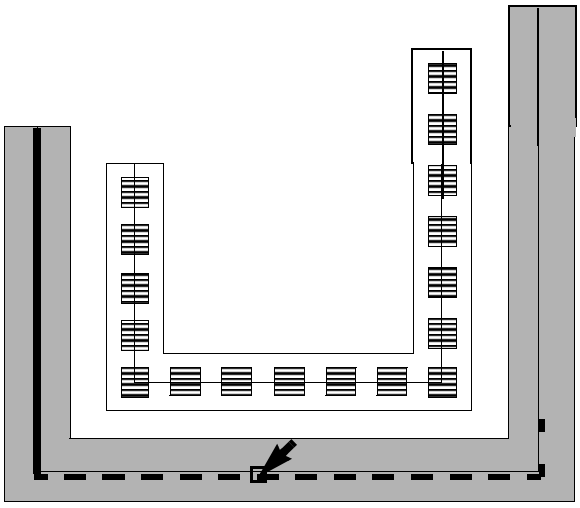
Create a selection box around the centerline of multiple segments.



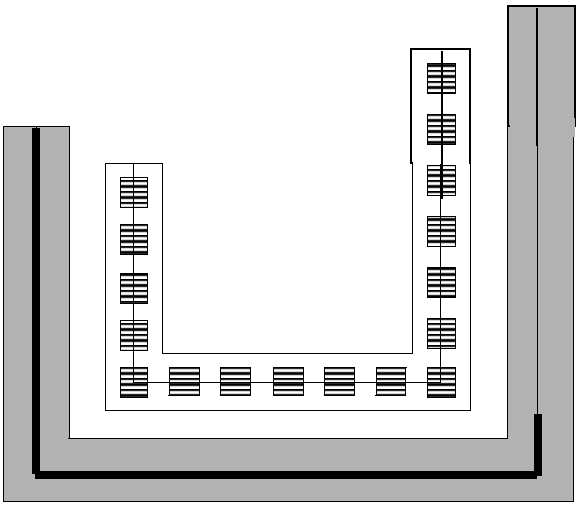
Highlighting shows the nearest master path segments are selected.

- Point to the middle of the centerline in a master path or subpath segment and click on it. Press `Shift` and click to select additional segments.

  As you point, the centerline of the nearest master path segment is highlighted with dotted lines.



Point to a segment centerline and click, then press `Shift` and click to select another segment.



Highlighting shows the path segments are selected.

© 2023 Cadence Design Systems, Inc.

Related Topics

Multipart Path Selection

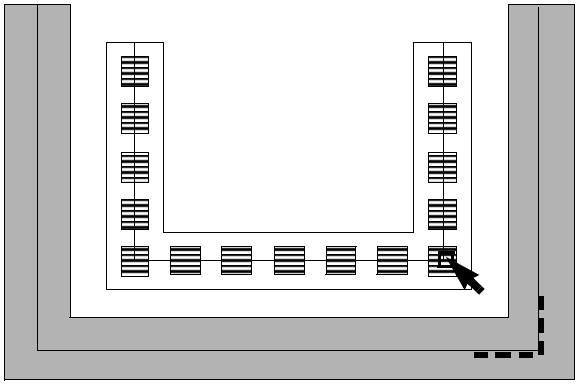
Selecting a Multipart Path Vertex

For multipart paths, a vertex is a point on the master path centerline where two segments join. You can select a vertex of the master path in a multipart path. To do so, you either select the master path vertex directly or select the nearest vertex of a subpath.

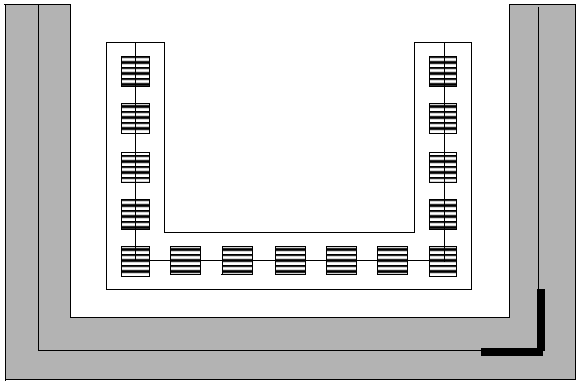
To select a vertex of the master path, do one of the following in partial selection mode:

- Point to a vertex on the centerline of the master path or centerline of a subpath and click on it.

  As you point, the nearest master path vertex is highlighted with dotted lines.



Point to a vertex on the master path or subpath and click.

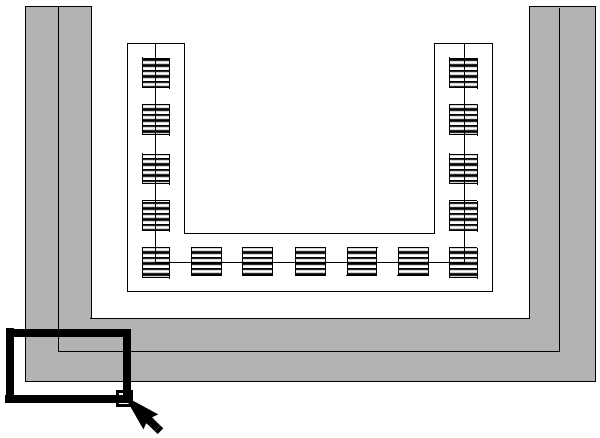


Highlighting shows the nearest master path vertex is selected.

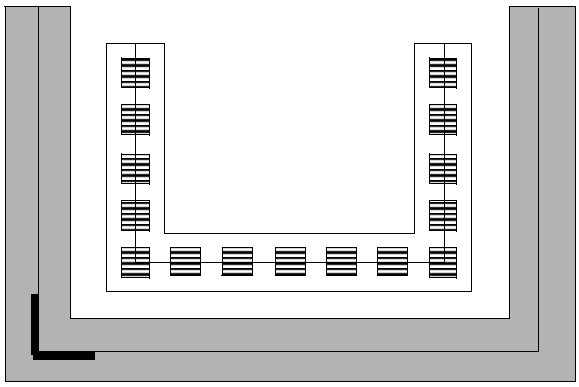
  If you are not able to click a centerline vertex, the current setting of the *Gravity* option is too restrictive. In the Layout Editor Options form, turn off *Gravity*.

© 2023 Cadence Design Systems, Inc.

- Create a selection box around a vertex on the centerline of a master path or subpath. If you click the corner of a subrectangle, the nearest master path vertex is selected.



Create a selection box around a vertex of the master path or a subpath.



Highlighting shows the nearest master path vertex is selected.

Related Topics

Multipart Path Selection

Multipart Paths

rodCreatePath

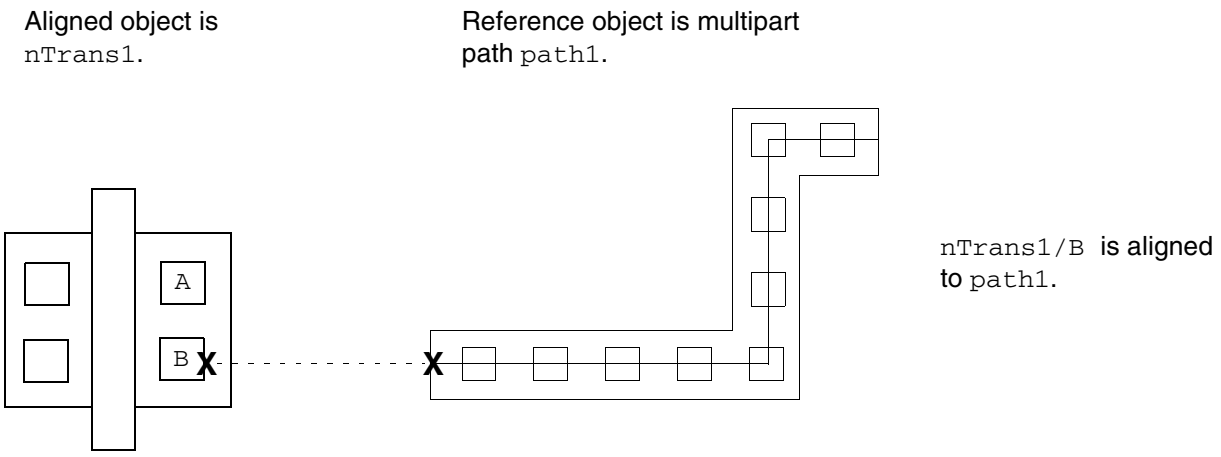
Moving Multipart Paths

When you move a multipart path within a cellview, its master path and subparts move together. You cannot move the master path separately from its subparts, nor can you move subparts separately from the master path.

When you move a multipart path, objects aligned to the multipart path move with it. To retain alignments when you move a multipart path to a new cellview, all objects aligned with the multipart path must be part of the selected set. If aligned objects are not selected, they are not moved and the alignments is broken. In the following example, the multipart path `path1` is the reference object. Via `B` in instance `nTrans1` is aligned to `path1`, with a separation of

© 2023 Cadence Design Systems, Inc.

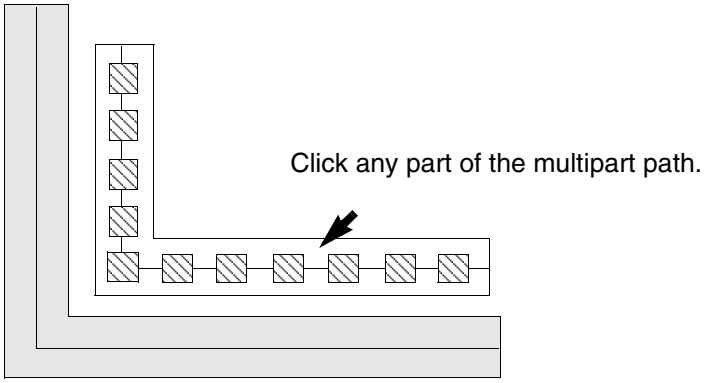
-10 units in the direction of the X axis. Therefore, `nTrans1` behaves as if it is aligned to `path1`. When you move `path1`, `nTrans1` moves with it.



The system automatically recalculates the alignment of objects when you open a layout cellview in edit mode or edit an aligned object in any way.

To move a multipart path:

1. In the layout window, choose *Edit – Move*.
2. In full selection mode, choose the multipart path by clicking on the master path or any subpart.



3. Click at the location where you want to place the multipart path; the aligned objects follow.

   The multipart path and all objects aligned to it appear in the new location.

© 2023 Cadence Design Systems, Inc.

Related Topics

Aligning Named Objects

Multipart Paths

rodCreatePath

Multipart Path Selection

Stretching Multipart Paths

You can stretch the ends, segments, and/or corners of a multipart path in the same way you stretch single-part paths; the master path and its subparts stretch together. You cannot stretch the master path separately from its subparts nor can you stretch subparts, except chopped subpaths, separately from the master path.

If you attempt to stretch a subpart, the system applies the stretch to the whole multipart path. The only exception is that you can stretch the chopped ends of subpaths. When you stretch a chopped subpath, all other chopped subparts are stretched too.

You cannot directly select a set of subrectangles, but selecting and stretching the chopped end of a subpath causes all chopped parts in the multipart path to be regenerated, including chopped sets of subrectangles. The system regenerates subrectangles along orthogonal segments only.

To stretch one segment in a multipart path:

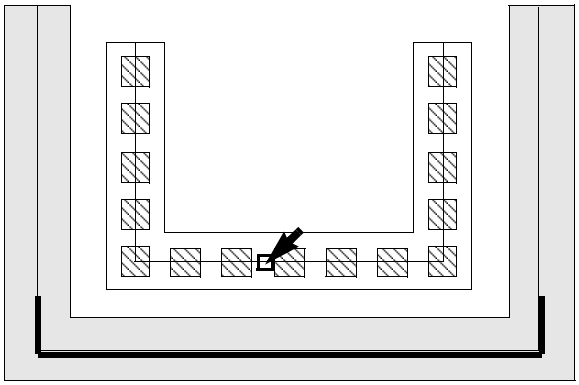
1. In the layout window, choose *Edit – Stretch*.

   Stretch automatically changes to partial selection mode and prompts you to select the object you want to stretch.

2. Select the segment you want to stretch by doing one of the following:

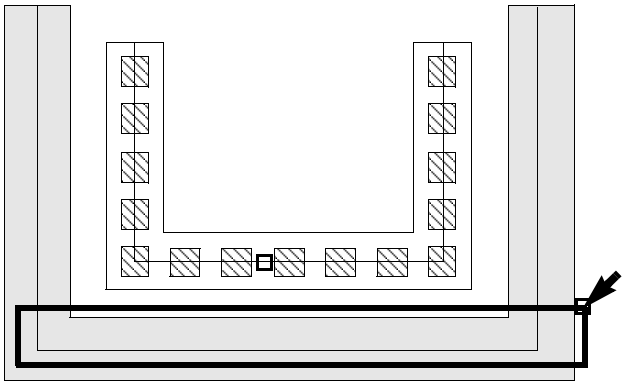
© 2023 Cadence Design Systems, Inc.

- Click the middle of the centerline of the master path segment or the comparable subpath segment.



The nearest master path segment is selected.

- Create a selection box around the centerline of a segment.

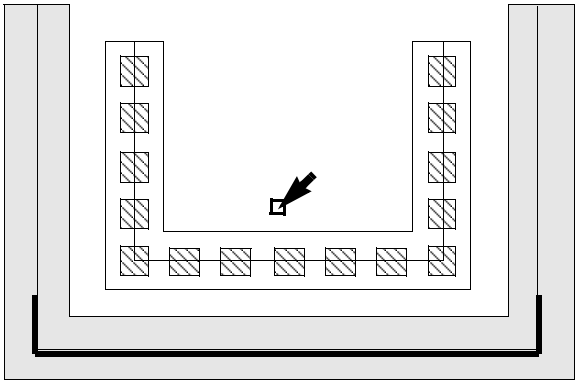


Be sure to include the ends of the segment centerline.

When you use a selection box, *Stretch* prompts you for a reference point.

3. If you are prompted for a reference point, click where you want the stretch to begin.

The reference point does not have to be directly on the multipart path.

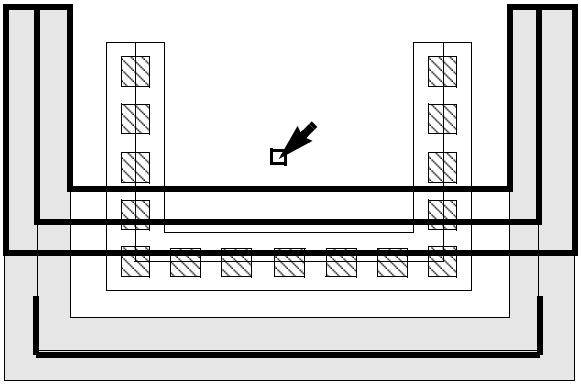


© 2023 Cadence Design Systems, Inc.

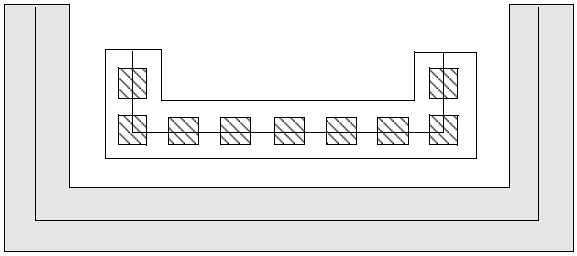
Stretch prompts you to point to the new location.

4. Point to a new location for the stretch.

   As you point, an outline of the master path segment shows the new location.



5. Click to select the new location for the selected multipart path segment.



Related Topics

Stretching Chopped Subpath Ends in Multipart Paths

Chopping a Multipart Path

Stretching Chopped Subpath Ends in Multipart Paths

You can select and stretch the ends of chopped subpaths in a multipart path. You might want to do this to reconnect the ends. When you stretch the chopped ends of one or more subpaths, all chopped subparts stretch with the selected subpaths.

You cannot directly select a set of subrectangles, but selecting and stretching the chopped end of a subpath causes all chopped parts in the multipart path to be regenerated, including chopped sets of subrectangles. The system regenerates subrectangles along orthogonal segments only.

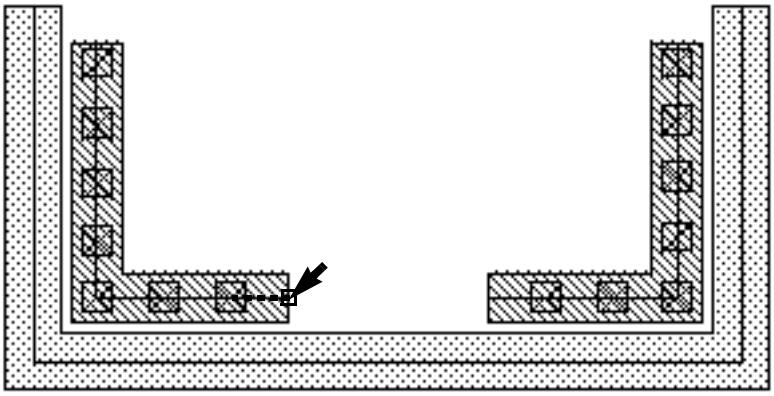
© 2023 Cadence Design Systems, Inc.

To stretch the chopped end of a subpath:

1. In the layout window, choose *Edit – Stretch*.

   Stretch automatically changes to partial selection mode and prompts you to select the object you want to stretch.

2. Select a chopped subpath end with a selection box or by clicking on it.



   When you use a selection box, *Stretch* prompts you for a reference point.

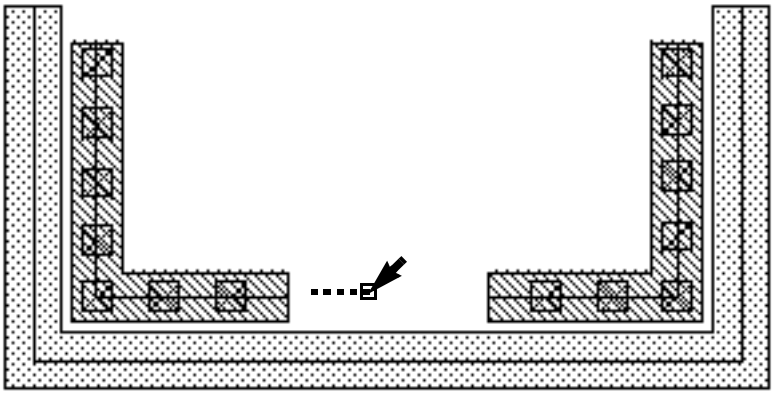
3. If you are prompted for a reference point, click where you want the stretch to begin.

   The reference point does not have to be directly on the path end.

   Stretch prompts you to point to the new location.

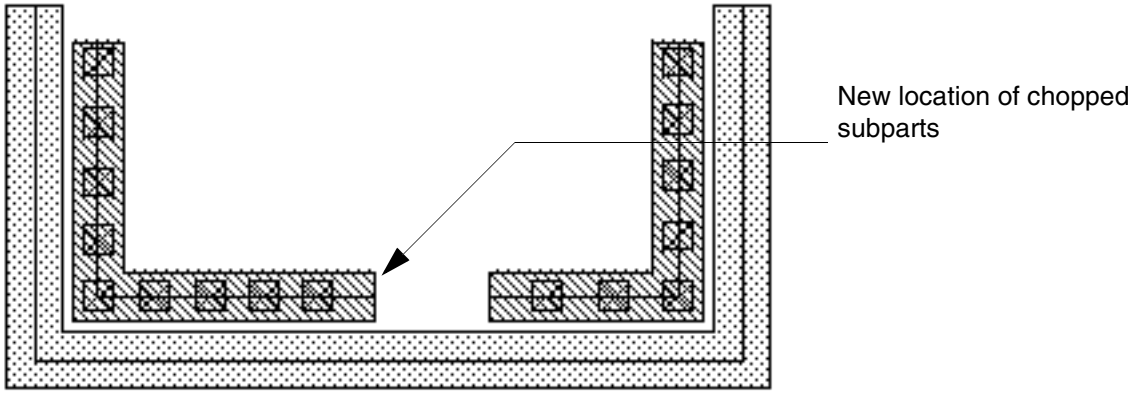
4. Point to a new location for the stretch.

   As you point, an outline shows you the new location of the subpath end.



© 2023 Cadence Design Systems, Inc.

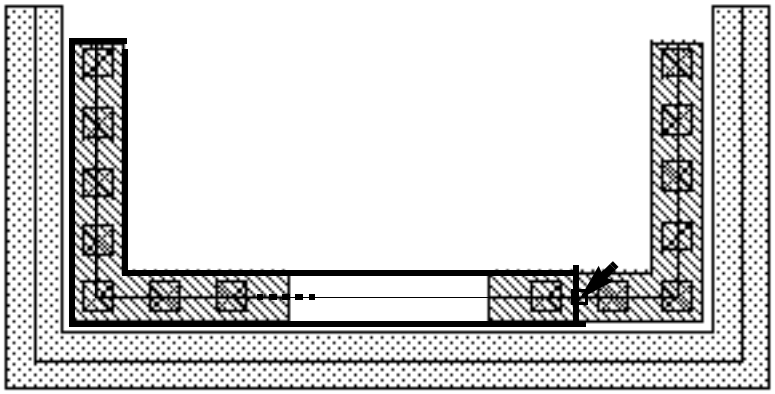
5. Click to select the new location for the chopped subpath end.



You can reconnect chopped subparts by stretching one subpath end over to the other subpath end. Even if you overlap the ends, the system regenerates the subrectangles correctly.

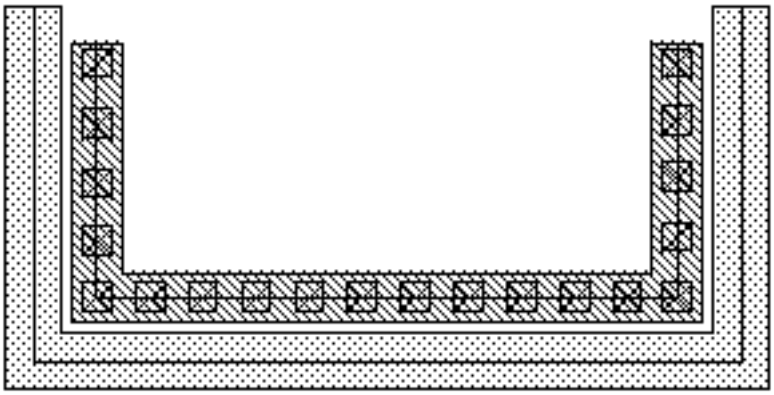
6. Select the same chopped subpath end and point to select the new location so that it overlaps the other chopped subpath end.

As you point, an outline shows you the new location of the subpath end.



© 2023 Cadence Design Systems, Inc.

The system reconnects the subpaths and regenerates subrectangles to fill the reconnected segment.



Related Topics

Stretching Multipart Paths

Splitting Multipart Paths

You can add a jog to a section of a multipart path in the same way you add a jog to single-part paths; the jog affects the master path and all of its subparts. You might want to split and stretch a multipart path to direct it around an object or to add new connecting segments.

Any objects aligned with or attached to the part of the multipart path that moves, move with the stretch. You can split paths only in full selection mode.

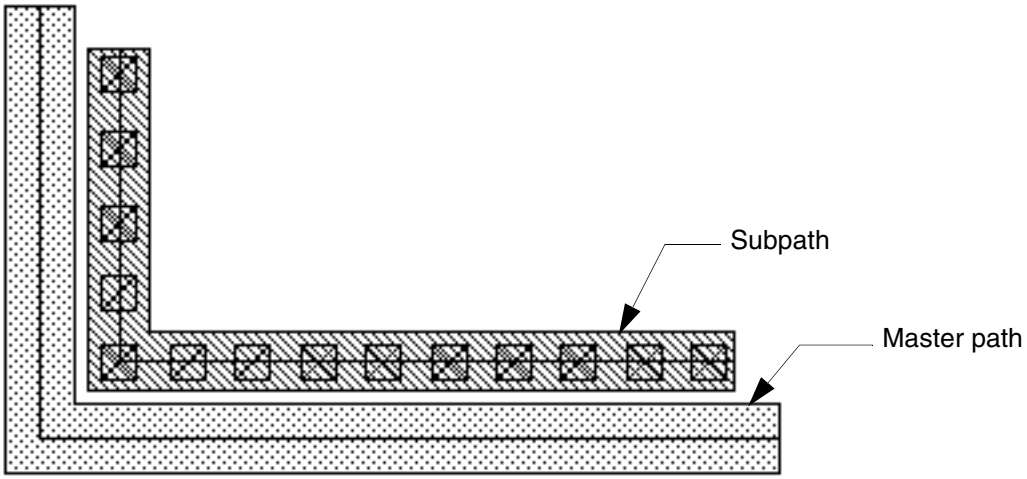
You cannot split or stretch the master path separately from its subparts, nor can you split or stretch subparts separately from the master path. If you attempt to split a subpart, the system applies the split to the whole multipart path.

To stretch a section of a multipart path segment:

1. In the layout window, choose *Edit – Advanced – Split*.

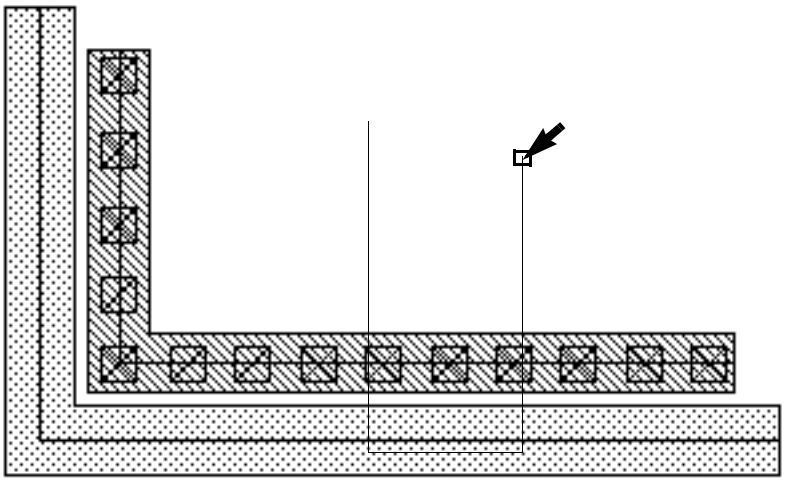
© 2023 Cadence Design Systems, Inc.

2. Select the object to be split by clicking anywhere on the multipart path.



Split prompts you to draw a split line by clicking on points. To define the section you want to stretch, your split line must cross the centerline of the master path in two places.

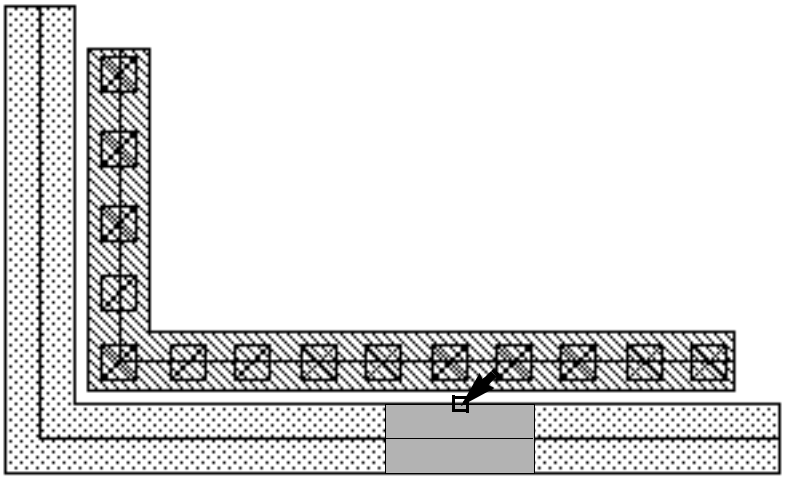
3. To create a split line, click four points as shown below, then press `Enter` to end the split line.



Split prompts you for a reference point.

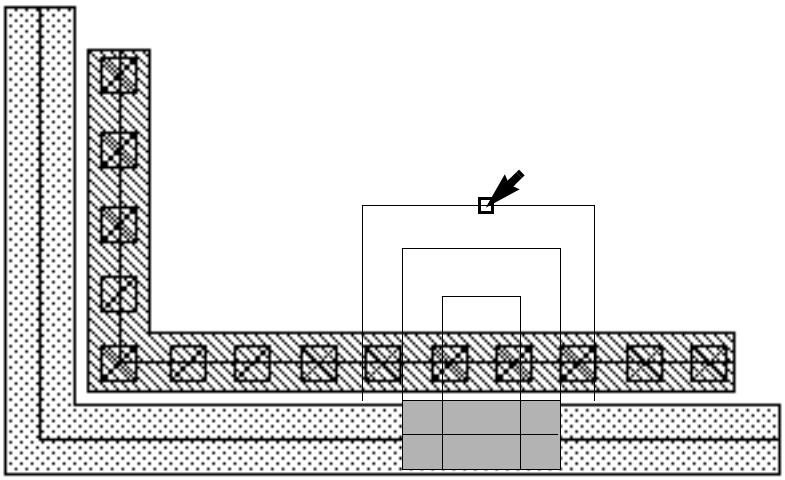
© 2023 Cadence Design Systems, Inc.

4. Click a starting point for the stretch.



Split prompts you for the new location for the stretch.

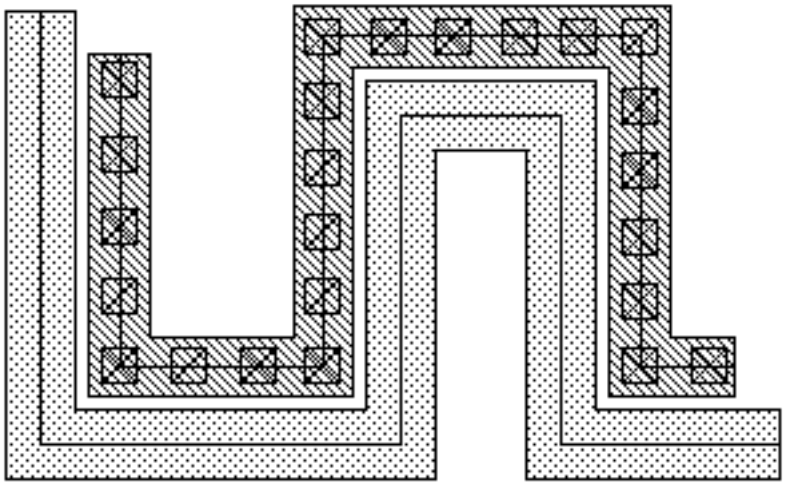
5. Point to a new location for the stretch.



An outline of the segment shows you the position of the stretch.

© 2023 Cadence Design Systems, Inc.

6. Click to select the new location for the section being stretched.



Related Topics

Multipart Path Selection

Reshaping a Multipart Path

You can reshape a segment of a multipart path in the same way you reshape a segment of a single-part path; however, you must start the reshaped segment on the master path centerline.

The master path and its subparts reshape together. You cannot reshape the master path separately from its subparts, nor can you reshape subparts separately from the master path.

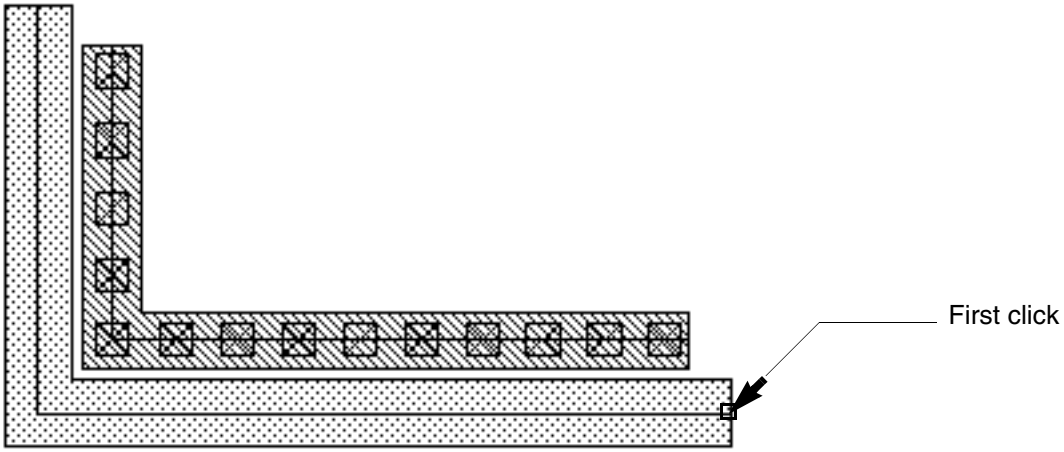
To reshape one segment in a multipart path:

1. In the layout window, choose *Edit – Advanced – Reshape*.
2. Press `F3` to open the Reshape form.
3. Select *Mode* as *line*.
4. Select a multipart path on the canvas.

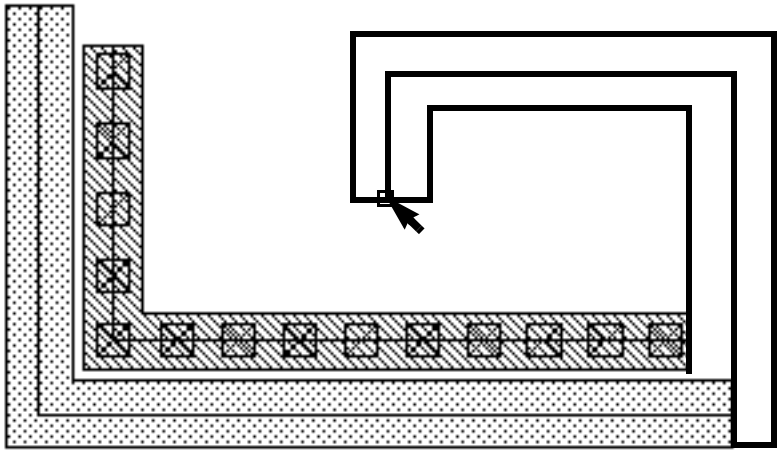
   You are prompted to enter the first point.

© 2023 Cadence Design Systems, Inc.

5. Click the centerline of the master path to enter the first point.



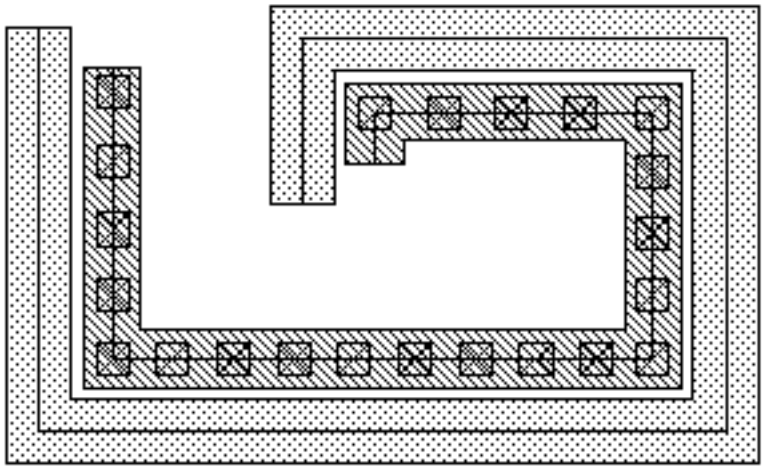
As you move the pointer, an outline of the master path shows the new shape.



6. Double-click on the canvas when you are finished entering points.
7. Right-click to toggle between highlighting the new path and highlighting both the old and new path.

© 2023 Cadence Design Systems, Inc.

8. When the shape you want is highlighted, click on the canvas.



Related Topics

Reshape Form

Multipart Path Selection

Selecting a Whole Multipart Path

Multipart Paths

rodCreatePath

How Chopping Affects Multipart Paths

When you use the *Chop* command on a multipart path, the result depends on what parts of the multipart path are choppable. You specify whether a part is choppable when you create the multipart path.

- If you specify the master path as choppable, all of its subparts must be choppable too. When you specify the master path as not choppable, you can specify each of its subparts as choppable or not.
- If the master path is choppable, you can chop the whole multipart path into two or more separate multipart paths by chopping all the way through the master path at 90 degrees.
- If the master path is choppable and you chop at an angle other than 90 degrees and/or chop only part of the way through the master path, the chop changes all objects in the

© 2023 Cadence Design Systems, Inc.

multipart path into ordinary paths, polygons, and rectangles. If this is not what you wanted, you can undo the chop using the *Edit – Undo* command.

- If the master path is not choppable, you can chop all subparts that are specified as choppable by chopping all the way through the master path at 90 degrees. All choppable subparts are chopped where you chop over the master path.
- When a chop affects sets of subrectangles that are choppable, the system regenerates subrectangles along orthogonal segments only.
- If the master path is not choppable, the system does not allow you to chop at an angle other than 90 degrees or to chop only part of the way through the multipart path.

After you chop a multipart path, you can select and stretch the chopped ends of subpaths. Although you cannot directly select, stretch, or chop a set of subrectangles, all choppable sets of subrectangles are also chopped when you chop a subpath.

When you chop a multipart path into one or more separate multipart paths, the system assigns the name of the original multipart path to the first new multipart path. The system assigns unique names to the other new multipart paths, starting with `path0`, `path1`, and so on.

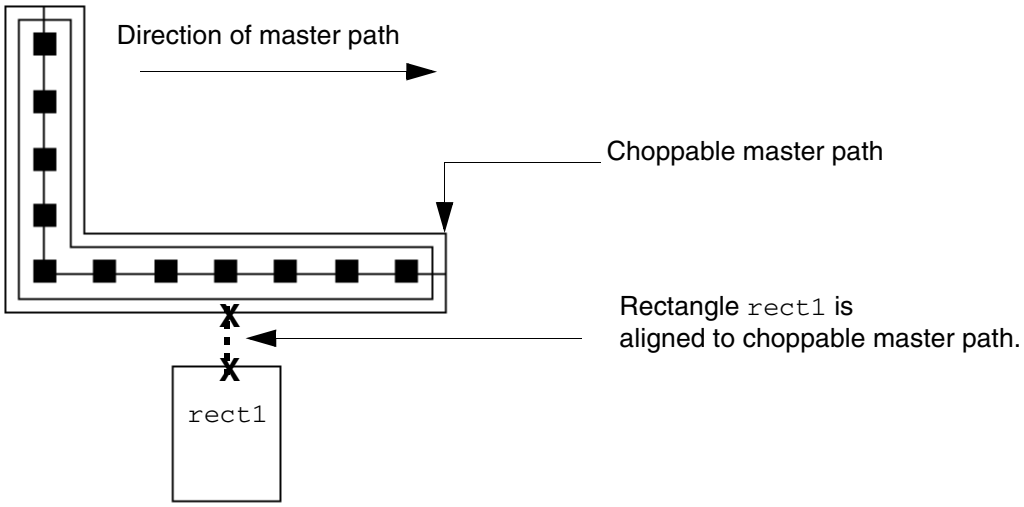
If you want to chop and rotate a multipart path, rotate the multipart path first, then chop it. Avoid rotating a chopped multipart path as the chop holes might not rotate correctly.

Chopping Multipart Paths with Aligned Objects

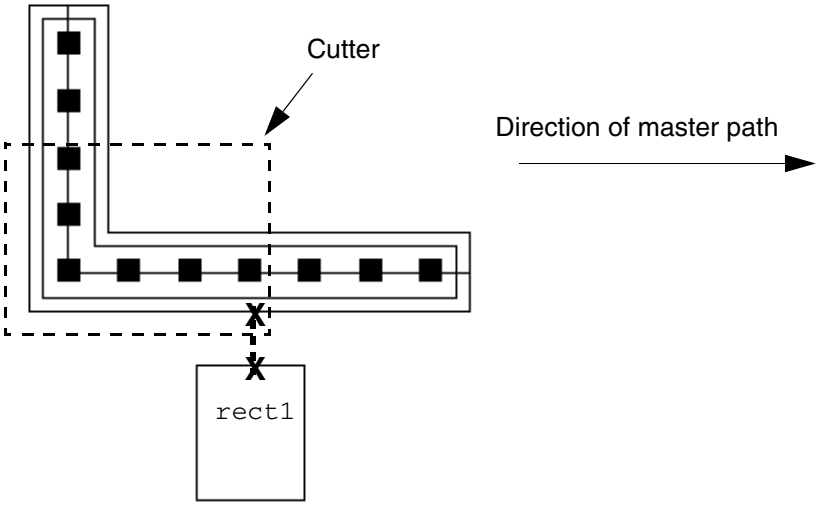
When you chop through the choppable master path of a multipart path, alignments to other objects are lost.

© 2023 Cadence Design Systems, Inc.

The following example shows what happens to an aligned object after cutting away a section of a choppable master path.

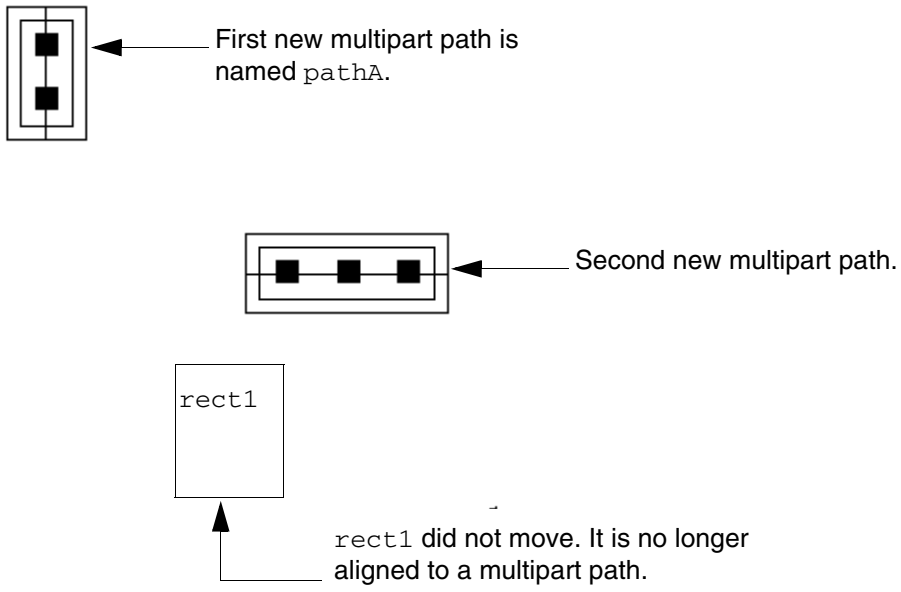


You can use the cutter to chop out part of the master path as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

The result is two new, shorter multipart paths, neither of which are aligned to `rect1`.



The system keeps the name `pathA` with the first new multipart path and assigns a unique name in the format of `path`*n* to the second new multipart path.

Related Topics

Multipart Path Selection

Chopping a Multipart Path

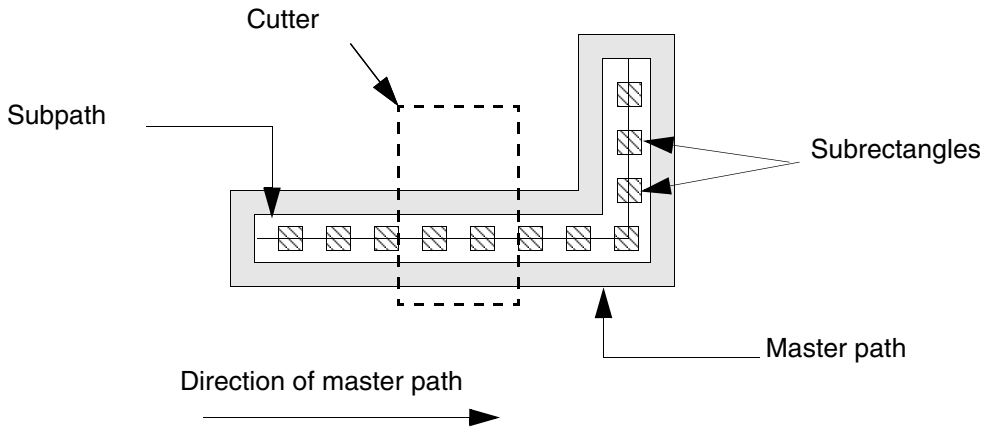
Chopping a Multipart Path

To cut away part of a multipart path that has one subpath and one set of subrectangles:

1. In the layout window, choose *Edit – Basic – Chop*.
2. Select the multipart path.

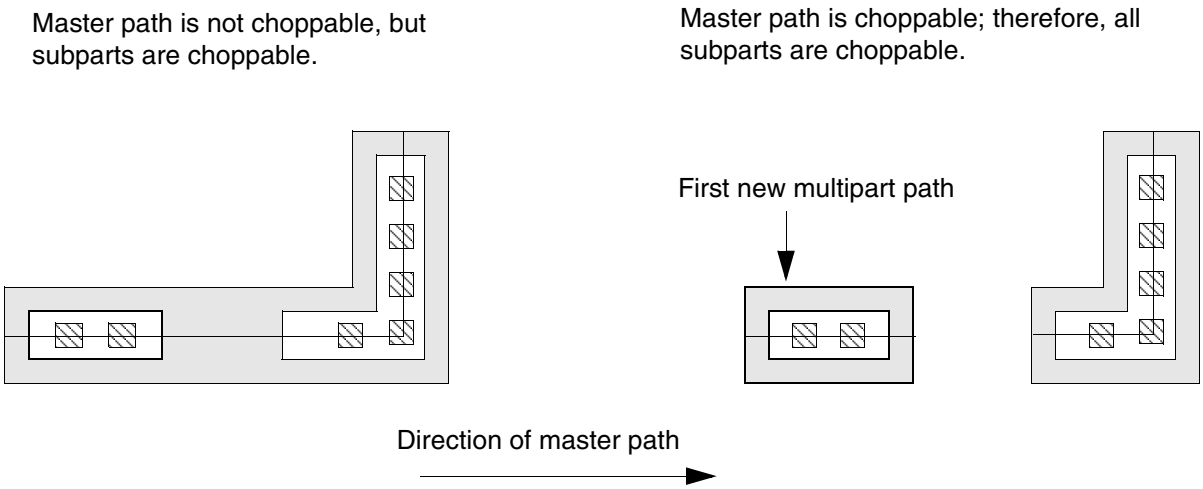
© 2023 Cadence Design Systems, Inc.

3. Create a chop rectangle around the master path to identify the parts you want to cut away.



To chop a multipart path or its choppable subparts, you must chop all the way through the master path. It is not necessary to chop through subparts.

The results of the chop depend on whether the master path is choppable or not, as shown below.



Related Topics

Selecting a Whole Multipart Path

Multipart Paths

rodCreatePath

© 2023 Cadence Design Systems, Inc.

How Chopping Affects Multipart Paths

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

8

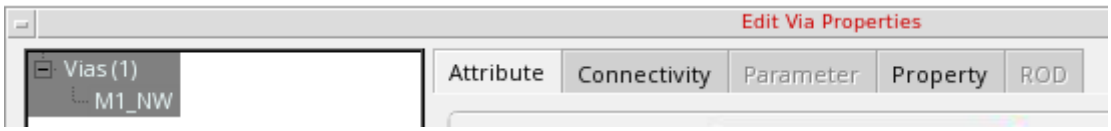
Object Properties

Every object in your database has information associated with it. You can view information about an object by using its Edit Properties form, also referred as the property editor. The Edit Properties form is a multipurpose, multi-tool form that enables you to edit the properties of whatever you create. Object information is displayed across the following tabs of the form:

- *Attribute* defines the object. Each type of object has specific, built-in attributes. For example, a polygon always has a layer and a set of coordinates that define vertices. You can view and change attributes of an object.
- *Connectivity* shows information on how this object relates to a net and/or a terminal and associated net expressions.
- *Parameter* controls values associated with a cell. Only parameterized cells (Pcells) can have parameters.
- *Property* displays user-created information that further defines the object. However, some applications also add properties.
- *ROD* displays the relative object design (ROD) name, handle, and alignment information of the object or instance.

You can access the Edit Properties form by using the *Edit - Basic - Properties* command or by pressing the Q key before or after selecting an object. You can change the size of the Edit Properties form by using the environment variables propEditorWidth and propEditorHeight.

The title and contents of the form vary depending on the object you select. For example, if you select a via, the name of the form changes to Edit Via Properties. If you select a rectangle, the name of the form changes to Edit Rectangle Form. If you select multiple, different objects, the name of form remains Edit Properties.



© 2023 Cadence Design Systems, Inc.

You can also edit object properties by using the *Property Editor* assistant.

Related Topics

Edit Properties Form

Displaying Object Properties

Editing Object Attributes

Editing Object Connectivity

Via Editing

Editing Cut Patterns

Inherited Connections and Net Expressions

Property Editor Assistant

Interactively Setting Color on a Selected Object

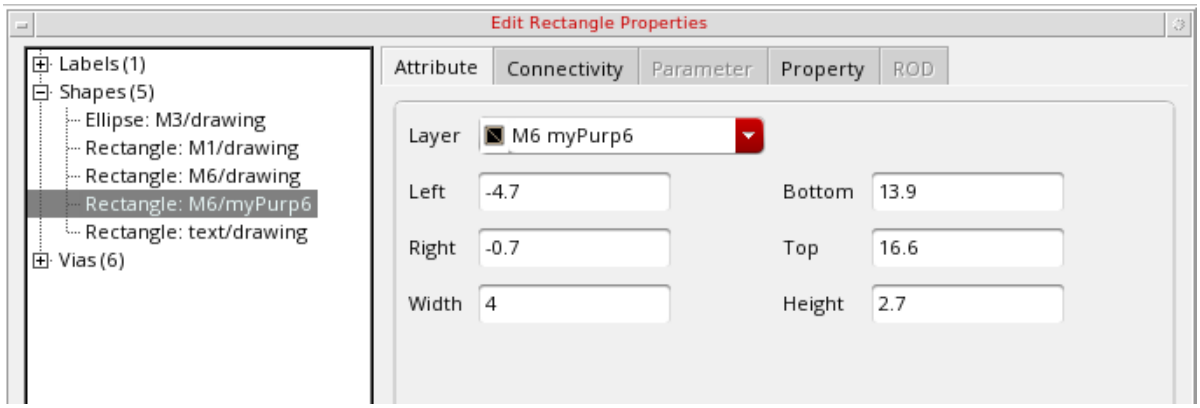
Displaying Object Properties

To display the properties of an object:

1. In the layout window, choose *Edit – Basic – Properties*. Alternatively, you can press the Q key or click the *Edit Properties* icon on the *Edit* toolbar. You can also right-click at any free space on the design canvas to display the *Layout* shortcut menu and select *Properties.*
2. Select one or more objects on the canvas.

© 2023 Cadence Design Systems, Inc.

The Edit Properties form opens, displaying the selected objects. Markers are also listed in the selection tree.



3. Click the appropriate tab in the form to view attributes, connectivity, parameters, or user-defined properties.

4. Click the appropriate icons on the form to perform actions on objects. You have the option to hide or show the selection tree, deselect highlighted objects from the canvas, keep highlighted objects and deselect others from the canvas, go to the previous or the next object in the tree, or select all objects in the tree.

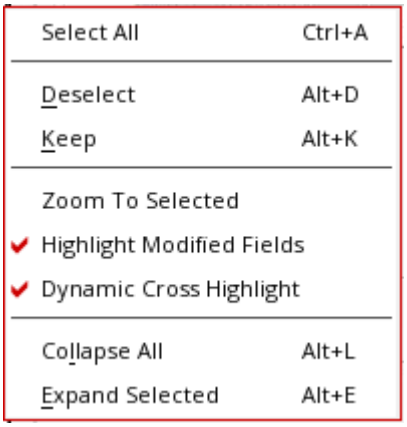
   You can use the Up and Down arrow keys on the keyboard to go to the next or previous object in the selection tree. The title and contents of the form vary depending on the object you select.

   The number of objects selected on the canvas and in the selection tree is also displayed on the form 8/26.

5. Right-click in the selection tree to display property editor shortcut menu. You have the option to select or deselect all objects in the selection tree, zoom to the selected object

© 2023 Cadence Design Systems, Inc.

on the canvas, highlight the modified fields in the form, and expand or collapse nodes in the selection tree.



6. Click *Cancel* to close the form.

Related Topics

Edit Properties Form

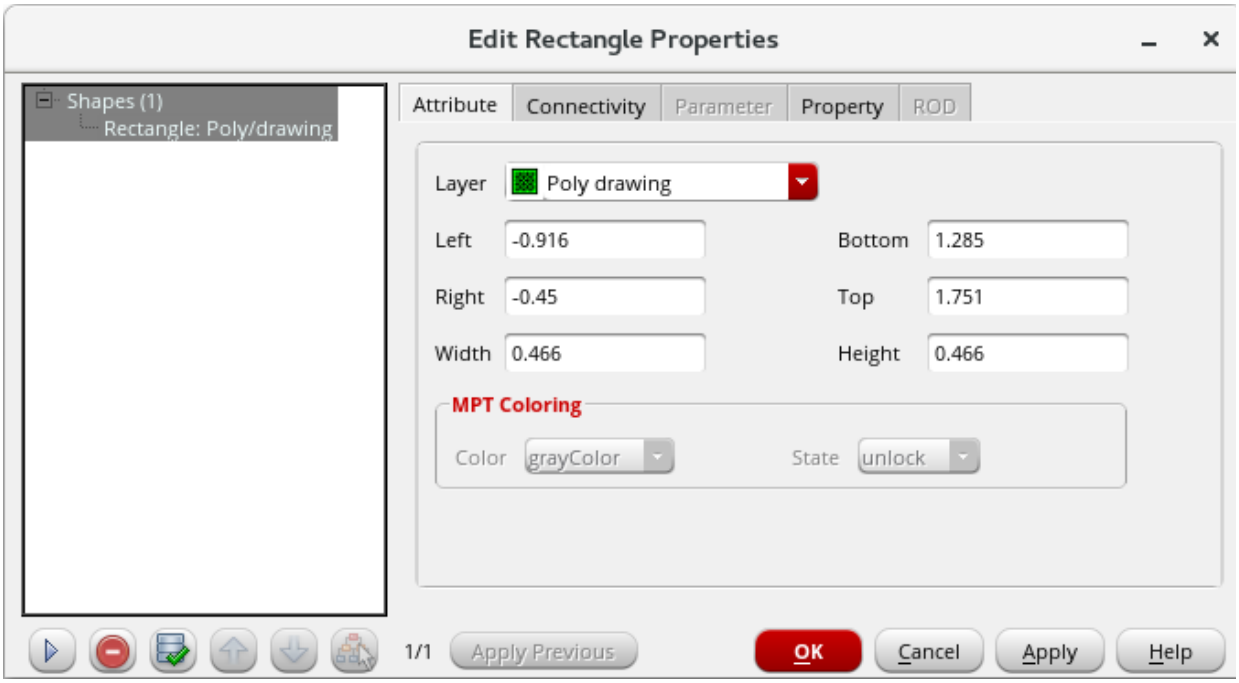
Editing Properties of an Object

To edit the properties of an object:

1. In the layout cellview, select the object on the canvas that you want to edit.
2. Choose *Edit – Basic – Properties*. Alternatively, you can press the Q key.

© 2023 Cadence Design Systems, Inc.

The Edit Properties form for the selected object opens.



3. Specify the new values in the required fields on the appropriate tabs of the form.

   The edited fields are highlighted in the form. If the values of the non-editable fields get updated because of the change in the value of an editable field, such fields are highlighted in a darker shade of blue and gray color. It helps in distinguishing the fields that are modified and non-modified due to changes.

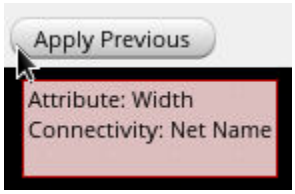
4. Click *Apply* or *OK*.

   The object is updated on the canvas according to the new values.

After you have edited the properties of an object, you can apply the same changes to another object by using the *Apply Previous* button in the property editor. This button applies the previously applied data from the property editor cache of the current session. When you hover over the button, the previously applied data, if any, is displayed as a tooltip. The button appears disabled if there is no previously applied data. *Apply Previous* applies only the properties that are applicable to the selected object. For example, suppose you modified the width of a rectangle on the *Attribute* tab of the property editor and the net name on the *Connectivity* tab. The *Apply Previous* tooltip lists the names of the tabs and modified fields as shown in the image. If you apply this data to a polygon, the same net is assigned to the polygon but the width change is not applied because *Width* is not a valid

© 2023 Cadence Design Systems, Inc.

attribute of a polygon.



Related Topics

Edit Properties Form

Editing Properties of a Group of Objects

To edit properties of a group of objects:

1. In the layout cellview, select all the objects that you want to edit.
2. Choose *Edit – Basic – Properties* to open the Edit Properties form.
3. Click the *Select all objects in the tree* icon or the *Select All* option from the shortcut menu in the selection tree.

   The values for all properties common to the selected objects appear. If a property does not apply to all or is different for some of the selected objects, its value appears as *AS IS*, meaning this property cannot be changed on any of the selected objects.
4. Type or select new values for the objects.
5. Click *OK*.

   The following example shows that a new value is specified in the *Width* field to change the width of all selected paths.





Original paths

Edited paths, where both are now 2 microns wide

© 2023 Cadence Design Systems, Inc.

Related Topics

Edit Properties Form

Searching and Replacing Object Properties

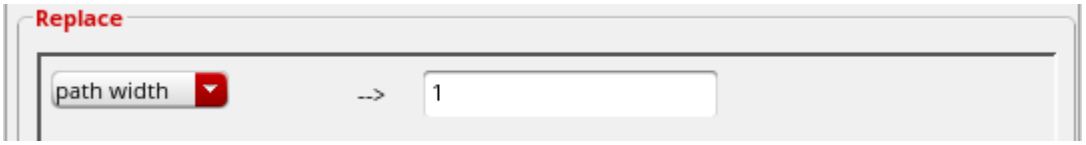
You can search for one or more objects that have identical properties and replace those properties with a new value. This enables you to quickly update a large number of objects at the same time.

To search for objects and replace their properties:

1. In the layout window, choose *Tools – Find/Replace* to display the Find/Replace form.
2. Choose the object that you want to search for and set the required search criteria.



3. In the *Replace* section, from the drop-down list, choose the type of property you want to replace.
4. Type or select the new value.



5. Click *Select All*.

   The layout editor highlights on the canvas all the objects that it finds. All the objects are now in a search group. The first object in the group is highlighted in a different color.
6. Click *Previous* or *Next* to select different objects in the search group.
7. To change the properties, do one of the following:
   - ❑ Click *Replace* to change the property for the current object in the search stack.
   - ❑ Click *Replace All* to change properties for all of the highlighted objects.

© 2023 Cadence Design Systems, Inc.

8. Click *Cancel* when you are done updating the objects.

Related Topics

Find/Replace Form

Adding a New Cellview Property

A cellview property is the information that you or a Cadence application assigns to the cellview. This information can be used to override the default information or to add information for further processing by other applications. For example, when post-processing cells, you could have a SKILL routine that uses a property to identify which cells to process.

To define a new property for a cellview:

1. In the layout window, choose *File – Properties*.

   The Edit Cellivew Properties form opens.

2. Click the *Property* tab.

   The properties of the cellview appear. Most of these properties correspond to settings in the Display Options form.

3. Click *Add*.

   The Add Property form appears.

4. Specify the required values for the new property.

5. Click *OK*.

   Your new property appears at the bottom of the Edit Cellview Properties form.

Related Topics

Edit Cellview Properties Form

Display Options Form

Add Property Form

© 2023 Cadence Design Systems, Inc.

Defining a New Object Property

An object property is the information that you assign to an object such as a via, circle, donut, ellipse, instance, label, path, pin, polygon, or rectangle. This information can be used to identify an object or to add information for further processing by other applications.

To define a new property for an object:

1. In the layout window, choose *Edit – Basic – Properties*.
2. Select one or more objects on the canvas.

   The Edit Properties form appears, displaying the properties of the first selected object.
3. Click the *Property* tab.

   The properties of the object appear.
4. Click *Add*.

   The Add Property form appears.
5. Specify the required values for the new property.
6. Click *OK*.

   Your new property appears at the bottom of the Edit Properties form.

Related Topics

Edit Properties Form

Add Property Form

Modifying a Property

To change a user-created property:

1. In the layout window, do one of the following:
   - To change a cellview property, choose *File – Properties*.
   - To change the property of an object, select the object on the canvas and then choose *Edit – Basic – Properties*.
2. Click the *Property* tab on the form that opens.

© 2023 Cadence Design Systems, Inc.

The properties of the cellview or the object appear.

3. Click the property label that you want to change.
4. Click *Modify*.

   The Modify Property form appears.
5. Make your changes to the property.
6. Click *OK*.

Related Topics

Edit Properties Form

Edit Cellview Properties Form

Modify Property Form

Deleting a Property

To delete a property:

1. In the layout window, do one of the following:
   - If you want to delete a cellview property, choose *File – Properties*.
   - If you want to delete an object property, select the object and then choose *Edit – Basic – Propertie*s.

   The Edit Cellview Properties or the Edit Properties form of the object appears.
2. Click the *Property* tab.

   The properties of the object or cellview appear.
3. Click the property label you want to delete.
4. Click *Delete*.

   The property is deleted from the form.

Related Topics

Edit Properties Form

© 2023 Cadence Design Systems, Inc.

Edit Cellview Properties Form

Editing Object Attributes

The characteristics that define an object are known as the object's attributes. All design objects have default attributes, such as color.

To change the attributes of an object:

1. In the layout window, choose *Edit – Basic – Properties*.
2. Select the object on the canvas.

   The Edit Properties form for that object appears.
3. Click the *Attribute* tab.
4. Change any of the attributes as required.
5. Click *Apply*.

Related Topics

Edit Properties Form

Attribute Tab of Various Objects

Editing Object Connectivity

The connectivity information defines how an object relates to a net and/or a terminal and the associated net expressions.

To change the connectivity of an object:

1. In the layout window, choose *Edit – Basic – Properties*.
2. Select the object on the canvas.

   The Edit Properties form for that object appears.
3. Select the *Connectivity* tab.
4. Specify the *Net Name* and *Net Criticality* information.
5. Click *Apply*.

© 2023 Cadence Design Systems, Inc.

If you select multiple instances on the canvas, you can set the net on a particular terminal for all the selected instances.

Related Topics

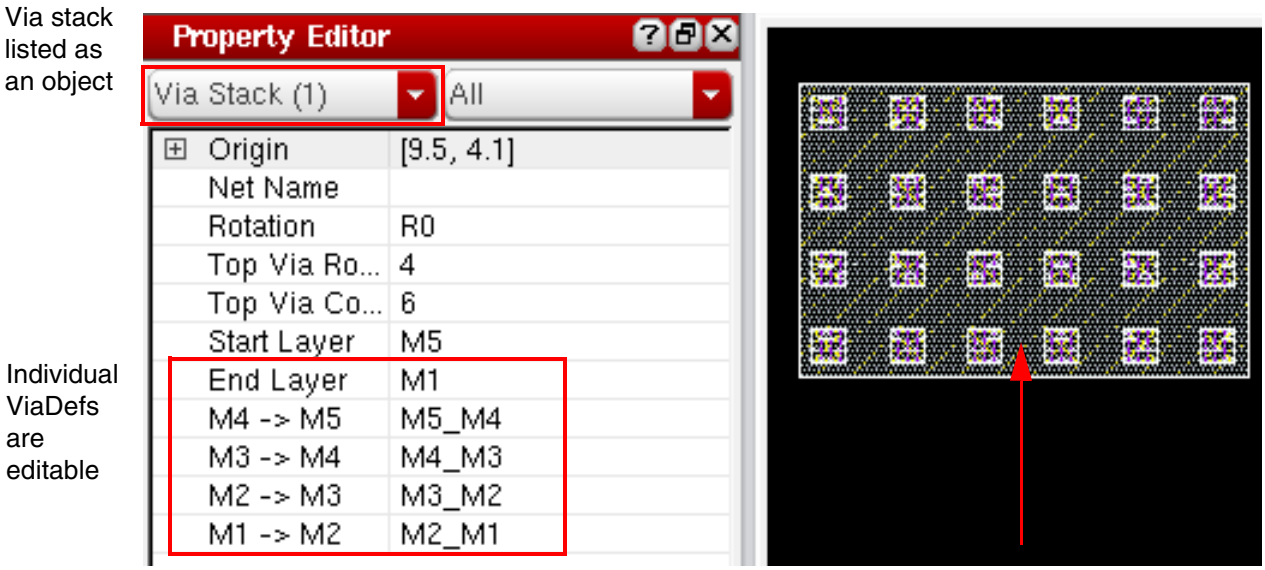
Edit Properties Form

Inherited Connections and Net Expressions

Via Editing

You can edit vias by using the *Property Editor* assistant. You can also use the property editor - the Edit Via Properties and Edit Via Stack Properties forms. You can modify cdsVia parameters, including the metal layer and via layer purposes.

The *Property Editor* assistant displays a selected via stack as a single entity. From the *Property Editor* assistant, you can edit the start or end layers of the via stack as well as update viaDefs used for each of the layer pairs. You can also modify the number of top rows and columns in the via stack, the net name, and the rotation of the via stack. Right-clicking a via stack displays the *Via* shortcut menu.

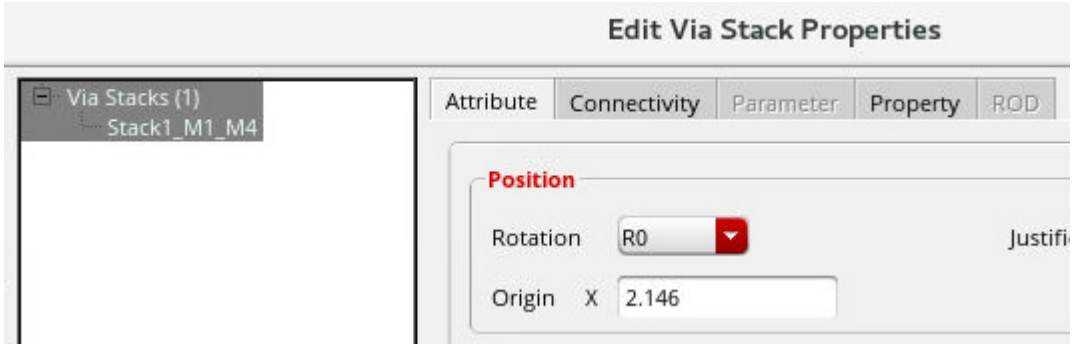


Via stack of 4 rows and 6 columns

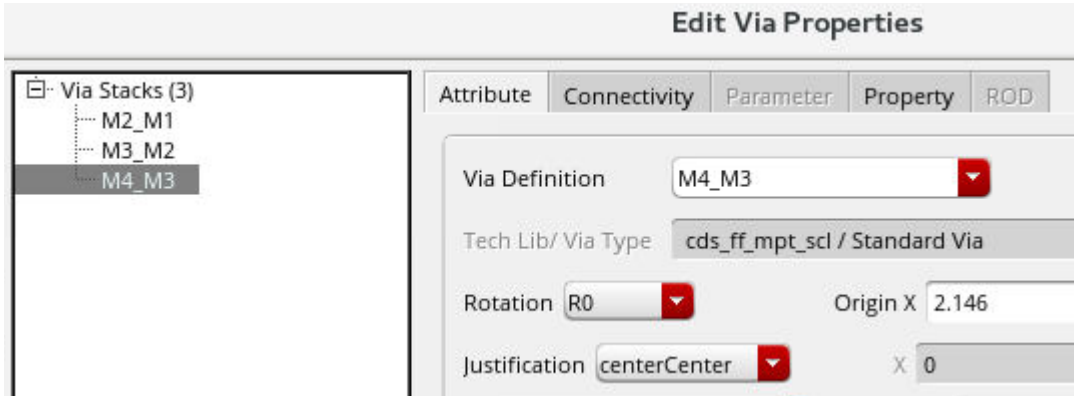
© 2023 Cadence Design Systems, Inc.

To edit individual vias of a via stack, you need to select *Via* from the object category selector list in the *Property Editor* assistant. This displays properties of all the vias in the via stack. You can expand the required property node and update properties on any of the individual vias.

In the property editor, a selected via stack is displayed as a single entity when the Via Stack mode is enabled in the Selection Options form.



When the Via Stack mode is disabled, individual vias of a via stack are listed, as shown in the following figure.



When you edit a via stack, the parameters for lower vias are generated to have the same footprint and not necessarily the same parameters values. If you want exactly same values on all vias of a via stack, then in the property editor you need to select all the vias in the selection tree and edit the required property. The new values are applied to all selected vias.

A cdsFixedVia via cannot be modified. You cannot use the property editor or the find and replace functionality to modify a cdsFixedVia. Also, you cannot chop or stretch a cdsFixedVia.

© 2023 Cadence Design Systems, Inc.

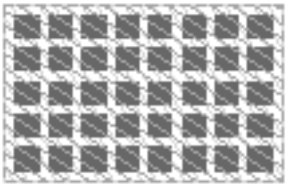
Related Topics

Creating a cdsVia

Editing Cut Patterns

To edit a cut pattern:

1. In the layout window, select a via array. For example, a via array of five rows and eight columns.



2. Choose *Edit – Basic – Properties*.

   The *Edit Via Properties* form opens.

3. Click the *Edit Cut Pattern* button in the *Cut* section.

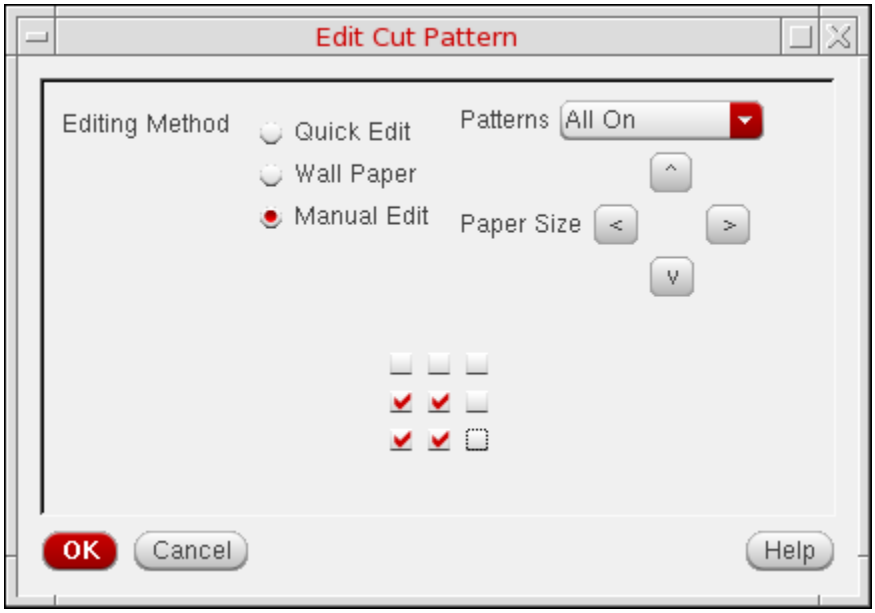
   The Edit Cut Pattern form appears.

4. Select *Wall Paper* as the editing method.

5. Use the *Paper Size* left and down arrow buttons to reduce the rows and columns, for example, to three and three.

© 2023 Cadence Design Systems, Inc.

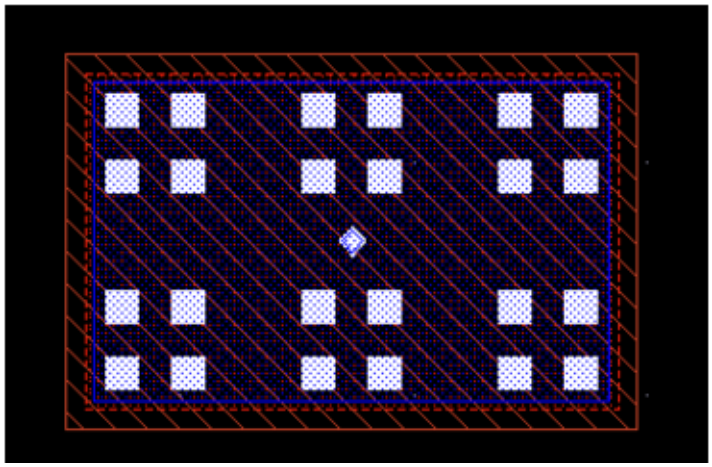
6. Edit cut-by-cut by clicking on the squares to turn off the top row and the far-right column, as shown below:



7. Click *OK*.
8. Click *Apply* on the *Edit Via Properties* form.

   The pattern then repeats across the entire cut array, starting from the lower left corner of the array and repeating upwards and to the right.

   The following image shows the cut pattern result.



© 2023 Cadence Design Systems, Inc.

Related Topics

Edit Properties Form

Edit Cut Pattern Form

Attributes in the Edit Via Properties Form

Inherited Connections and Net Expressions

- Inherited Connections
- Net Expressions

Inherited Connections

Inherited connections are an extension to the connectivity model that allow you to create global signals and override their names for selected branches of the design hierarchy. This flexibility allows you to use:

- Multiple power supplies in a design
- Overridable substrate connections
- Parameterized power and ground symbols

Net Expressions

A net expression is a special property placed on a net or terminal to define its connectivity. A net expression consists of a property name and a default net name. You use a net expression to override the connection made by a signal or terminal. This allows you to use, for example, multiple power supplies in your design. The signal is redefined due to the value assigned to it. Redefining the signal eliminates the problem of global nets being merged into a single, electrically-equivalent signal, which occurs when the signal traverses the design hierarchy.

You can create, edit, and delete net expressions for pins by changing the net expression information on the *Connectivity* tab of the Edit Properties form. Once the net expression information is applied to a terminal, the net expression is displayed in the text labels of pins. The following must be true for a net expression to display:

- The pin is created with the *Create Label* option enabled in the Create Pin form.
- The *Net Expressions* option is enabled in the Display Options form.

© 2023 Cadence Design Systems, Inc.

If *Net Expressions* is disabled, the terminal name is displayed.

Net expressions are not interpreted by the display code in the layout editor. Net expressions are displayed as is, but other tools such as the Layout Versus Schematic (LVS) program use and interpret net expressions. The net expressions on layout terminals must match those found in schematic views.

When you view an instance containing a pin whose terminal has a net expression, the label display is of the terminal name, not the net expression. To see the net expression in a hierarchical design, you must descend into the cellview containing the net expression to see the full text display of the net expression.

Related Topics

Inherited Connections

Creating and Editing a Net Expression

To create or edit a net expression:

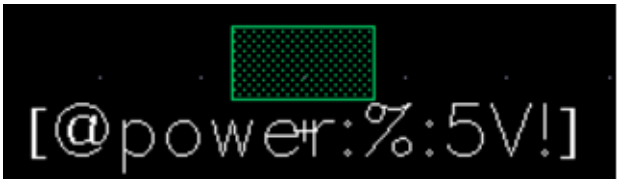
1. In the layout window, select the pin to whose terminal you want to assign a net expression.
2. Choose *Edit – Basic – Properties*.

   The Edit Properties form for the pin opens.
3. Click the *Connectivity* tab.
4. Type the net expression override property name in the *Net Expression* field.
5. Type the default net name in the *Default* field.

   The default net name is used when no override property is defined in the hierarchy above this point in the schematic view. If you do not enter a default net name, the terminal name is used.
6. Click *Apply.*

© 2023 Cadence Design Systems, Inc.

The pin label display changes to show the net expression property preceded by the @ symbol and the default net name preceded by the % symbol.



The *Net Expressions* field in the *Display Controls* section of the Display Options form must be enabled for the net expressions to display in your cellview.

Clearing the *Net Expression* field in the Edit Properties form deletes the net expression from your design.

If several pins in your design have the same terminal name, you need update only one pin and the rest will update automatically. You can see the edited text labels after you use *View – Redraw*.

Related Topics

Edit Properties Form

Display Options Form

Property Editor Assistant

The *Property Editor* assistant enables you to view and edit property values of objects in your design. By default, each property is displayed in a separate row in the *Property Editor*. You

© 2023 Cadence Design Systems, Inc.

access the *Property Editor* assistant by choosing *Window – Assistants – Property Editor.*

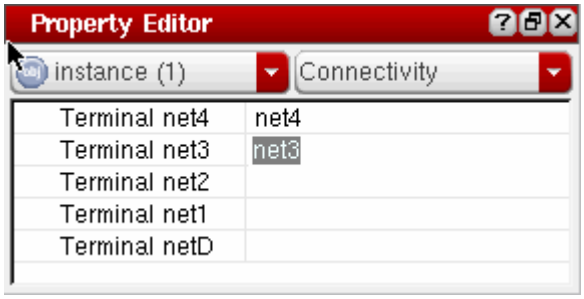


Related Topics

Property Editor Assistant

Net Setting on Instance Terminals

You can use the *Property Editor* assistant to set up the nets on the terminals of a selected instance. Select *Connectivity* from the property category selector list and specify the nets in the property value column for each terminal on the instance.



By using the *Property Editor* assistant, you can set up multiple terminals of different instances at the same time.

Related Topics

Property Editor Assistant

© 2023 Cadence Design Systems, Inc.

Route Status of Objects

To determine the route status of vias, paths, and pathsegs, you can use the following two attributes in the *Property Editor* assistant:

- *Route Status*: It is editable and can have the following values:
  - *Routed* indicates normal routing status.
  - *Fixed* indicates a route that should not be edited automatically, though it can be edited manually.
  - *Locked/Cover* indicates a route that should not be edited either by automatic tools or manually. The route forms a part of the base array. To edit the route, change the routing status to *Routed* or *Fixed*.
- *Part of a Route Status:* It is a non-editable attribute. It is set to `FALSE` if the selected via, path, or pathSeg is not part of a route and set to `TRUE` if the via, path, or pathSeg is part of a route.

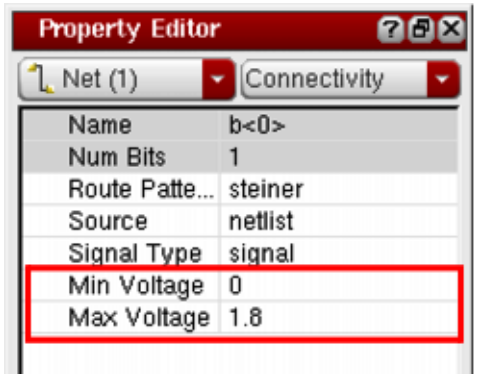
The *Create Wire* command in Layout XL creates vias and pathSegs that are in route.

Related Topics

Property Editor Assistant

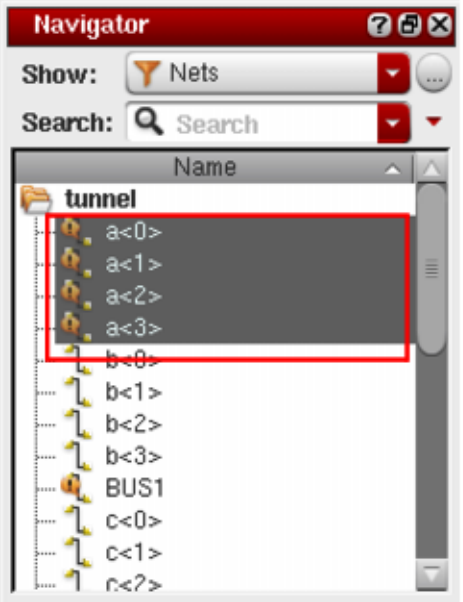
Voltage Range Information

In Virtuoso, the voltage range information on a net is used to verify or enforce voltage dependent constraints, such as voltage dependent spacings, `minVoltageSpacing`. By using the *Property Editor* assistant, you can view and edit the voltage range information of nets, `minVoltage` and `maxVoltage`.



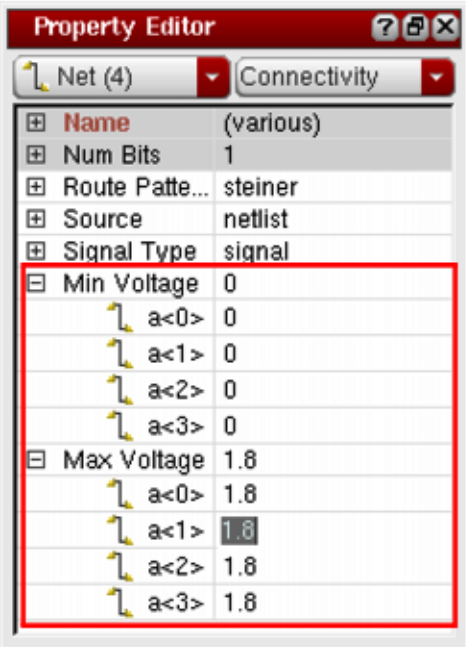
© 2023 Cadence Design Systems, Inc.

You can select one or more nets in the *Navigator* assistant as shown in the following figure.



Selected Nets

In the *Property Editor* assistant, the *Min Voltage* and *Max Voltage* are displayed for the selected nets.



Min Voltage and Max Voltage for Individual nets can be edited

You can edit the *Min Voltage* and *Max Voltage* values for any or all of the nets.

© 2023 Cadence Design Systems, Inc.

Related Topics

Property Editor Assistant

Interactively Setting Color on a Selected Object

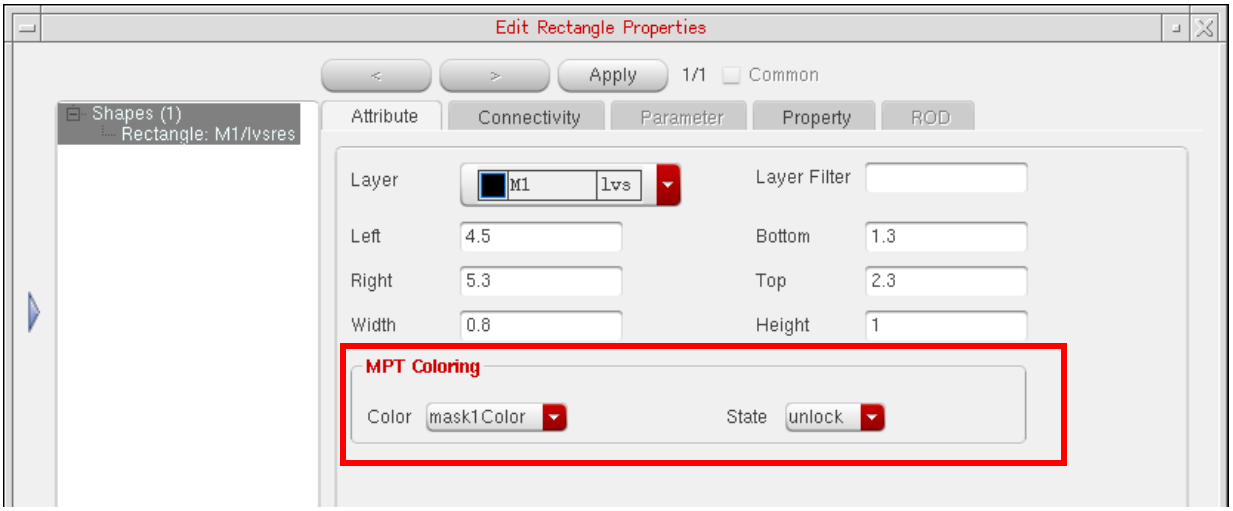
You can interactively set the color for the selected MPT shape objects and vias through the Property Editor and assistant.

- Setting Color on Shape Objects
- Setting Color on Vias

Setting Color on Shape Objects

For a selected MPT shape, you can modify the color and color state using the Property Editor and assistant.

Use the *Color* and *State* fields in the *MPT Coloring* section on the *Attribute* tab of the Edit Properties form of the shape to modify the color information.



Property Editor determines the available colors through the #Masks (number of masks) defined in the technology database. The table below summarizes the display colors based on

© 2023 Cadence Design Systems, Inc.

the association between the #Masks found in the technology database and *MPT Coloring* section.

#Masks	MPT Coloring Section	Available Colors
NoColor	Gray Out | Hidden	N/A
2	Enabled	grayColor | mask1Color | mask2Color
3	Enabled	grayColor | mask1Color | mask2Color | mask3Color

The values of the *State* field for the available colors are summarized in the table below:

Color	Color State
grayColor	Gray Out | Hidden
mask1Color	unlock | lock
mask2Color	unlock | lock
mask3Color	unlock | lock

The default value of the *Color* field is defined by the `setDefaultColorOnMPTLayers` environment variable. The table below summarizes the default values of the *Color* and *State* fields in the Property Editor, based on the value of the `setDefaultColorOnMPTLayers` environment variable.

setDefaultColorOnMPTLayers	Default Color Value	Default State Value
grayColor	grayColor	unlock
mask1Color	mask1Color	unlock
mask2Color	mask2Color	unlock
mask3Color	mask3Color	unlock

Setting Color on Vias

For a selected via, you can modify the color using the *MPT Coloring* section of the Edit Via Properties form or the Property Editor assistant. You can set the colors for Layer1, CutLayer, and Layer2.

© 2023 Cadence Design Systems, Inc.

The table below summarizes the MPT coloring fields that are available for the layers.

Via Layer	Available Colors
Layer1	grayColor | mask1Color | mask2Color | mask3Color
CutLayer	grayColor | mask1Color | mask2Color | mask3Color
Layer2	grayColor | mask1Color | mask2Color | mask3Color

Similar to shapes, you can set the lock state for vias. When multiple vias are selected, which implies the Common check box is selected in the Edit Via Properties form, and the lock state is different for some of the selected vias, the AS IS option is displayed to reflect the lock state of the vias.

 Video

You can view the enhancements related to MPT Coloring in the videos, *Coloring Utilities in Property Editor* and *Coloring Utilities Dynamic Selection Assistant*.

Related Topics

Object Properties

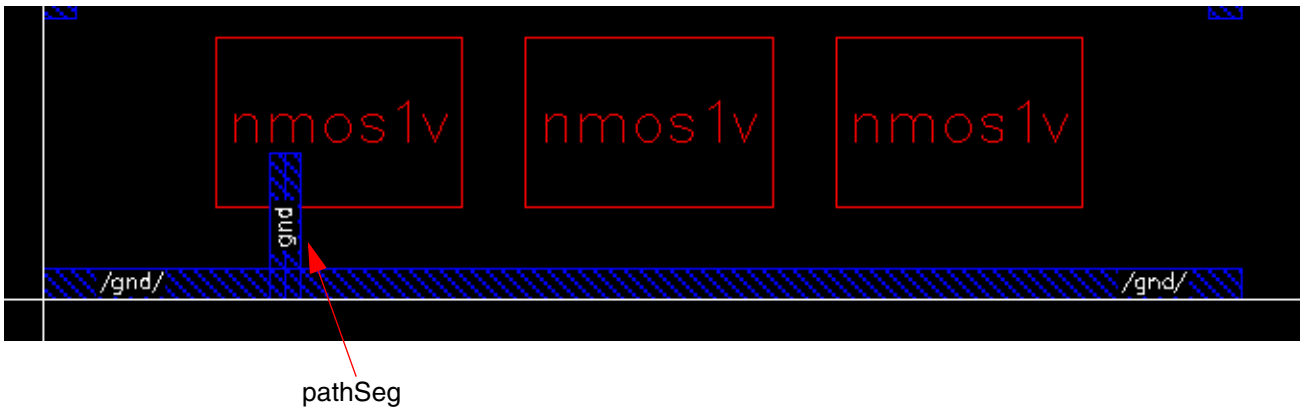
© 2023 Cadence Design Systems, Inc.

9

Group Arrays

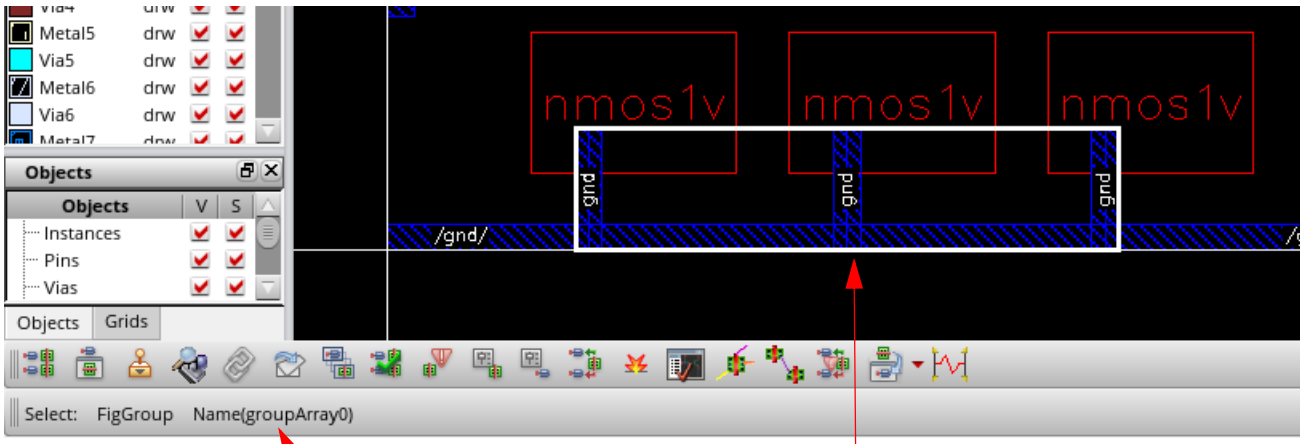
Certain custom and digital designs, such as those containing memory arrays and routing channels, might require patterns of objects to be repeated across a design. You can use group arrays to create such repeated patterns in your designs.

The following image shows a design with three devices and a power rail, with routing created for the first device using a pathSeg.



© 2023 Cadence Design Systems, Inc.

You can create routing for the entire design by using a group array. The following image shows the pathSeg copied to create a 1x3 group array, with 1 row and 3 columns, to create routing for all devices.



Status bar indicating a group array

pathSeg copied as a 1x3 group array

To create repeated design patterns, you can specify the following attributes for a group array:

- Number of rows and columns, which represent the number of cells
- Spacing between cells
- Orientation pattern of each cell.

The following are some of the key benefits that group arrays offer:

- After you have created a group array, you can easily customize any of its attributes. For example, you can increase or decrease the number of cells in a group array.
- Group arrays support synchronous editing. Changes made to one cell of a group array are replicated across all other cells. You can make changes to a cell by editing it in place.
- Group arrays do not introduce any additional level of hierarchy in a design, thus enabling you to keep your designs simple.

Related Topics

Group Array Creation

Group Array Considerations

© 2023 Cadence Design Systems, Inc.

Predefined Orientation Patterns for a Group Array

Creating a Group Array by Using the Copy Form

Specifying Group Array Orientation Using the Array Pattern Form

Saving a Custom Orientation Pattern for a Group Array

Group Array Editing

Group Array Creation

You can create a group array in any of the following ways:

- While copying objects by using the Copy form
- While generating selected schematic instances and pins directly in a layout view by using the Generate Selected Components form
- While generating clones by using the Generate Clones form
- Using the lxCreateGroupArray SKILL function

Related Topics

Copy Form

Generate Selected Components Form

Generate Clones Form

Group Arrays

Group Array Considerations

While working with group arrays, ensure that you adhere to the following considerations:

- Rulers and markers are not included in a group array. When you create a group array by selecting objects that include rulers and markers, the group array is created ignoring rulers and markers.
- A place-and-route boundary (PR boundary) cannot be included in a group array. If your selection includes a PR boundary, creation of the group array fails.

© 2023 Cadence Design Systems, Inc.

- Group arrays cannot be created by using the *Copy* command interactively. You must specify spacing mode and values within the Copy form.

Related Topics

Group Arrays

Copy Form

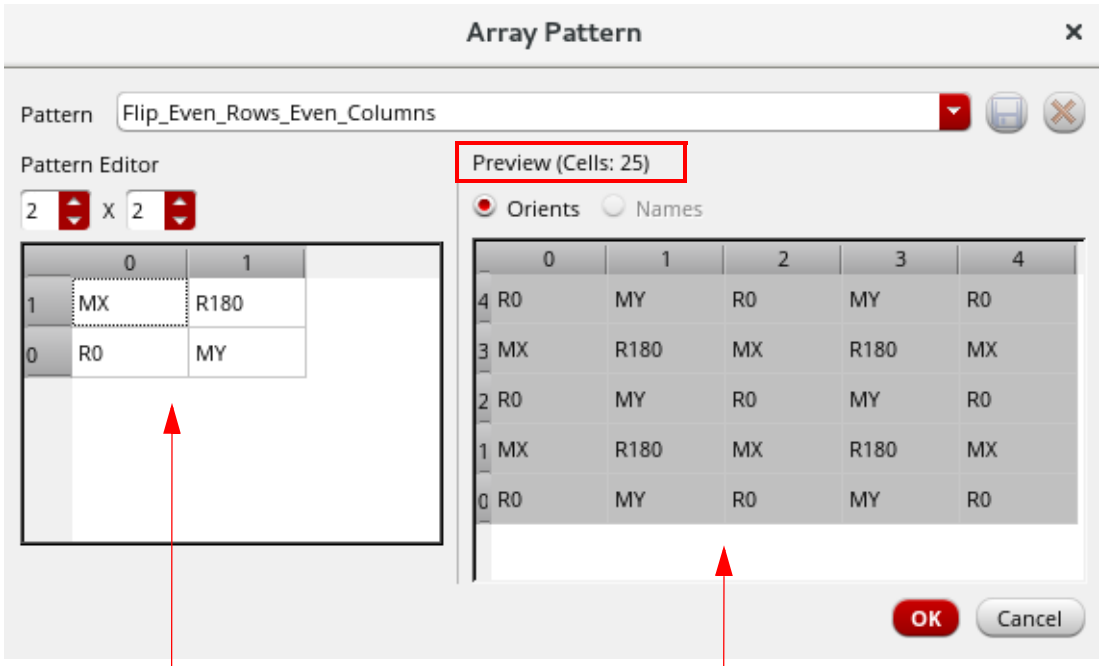
Holes in a Group Array

In a group array, you can choose to leave some cells empty to denote that there are no geometries underneath. Such cells are referred as holes.

While creating or editing a group array, you can specify the orientation pattern of each cell. To denote an empty cell or a hole in a pattern, you need to specify dash (-) as the orientation value for the cell. Empty cells are not considered for calculating the cell count of a group array.

© 2023 Cadence Design Systems, Inc.

The following image shows a 5x5 group array with an orientation pattern applied. The selected pattern is repeated across the group array, as shown in the *Preview* section of the Array Pattern form. Note that the *Preview* section shows the cell count as 25.

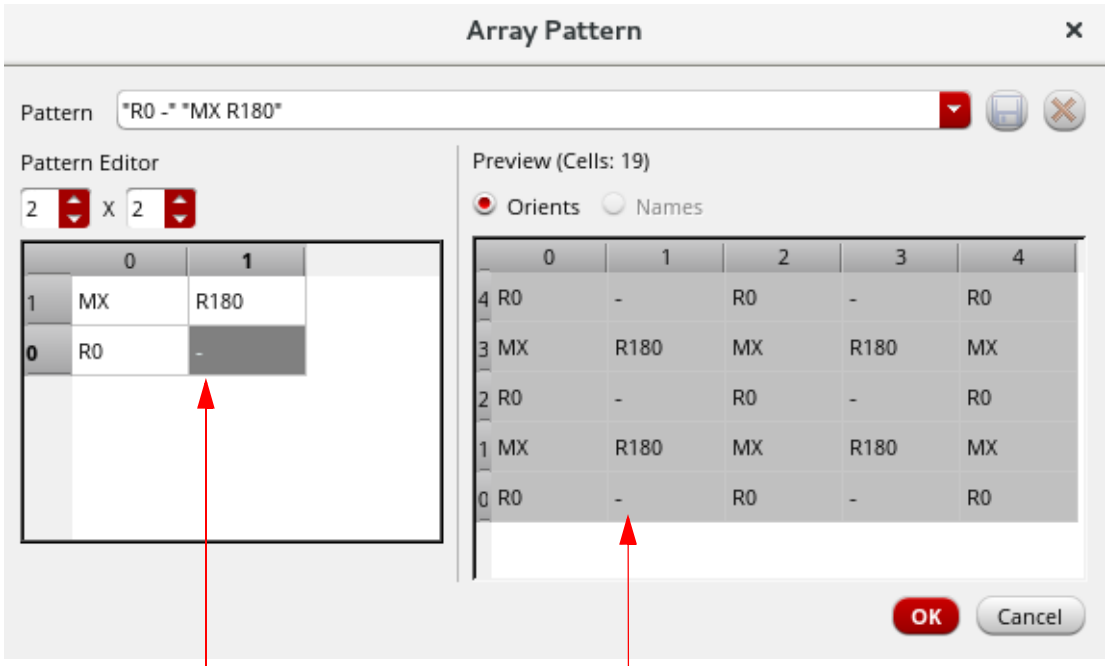


Pattern selected for the group array

Preview of the group array with the pattern applied

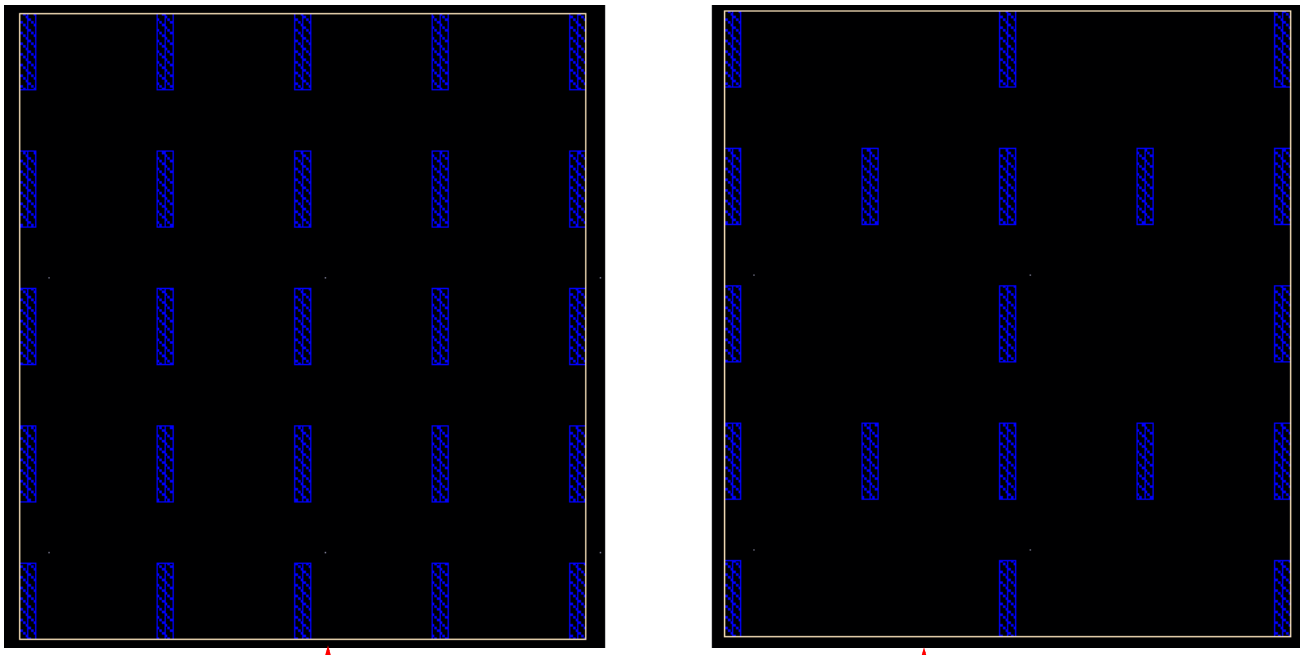
© 2023 Cadence Design Systems, Inc.

Now when you add a hole to a cell in the pattern, the preview displays the updated pattern applied to the group array, with the pattern repeated across cells. Because holes have been added, the *Preview* section now shows the cell count as 19.



© 2023 Cadence Design Systems, Inc.

The following image shows a 5x5 group array and how it appears after holes are added.



Related Topics

Creating Holes in a Group Array

Group Arrays

Array Pattern Form

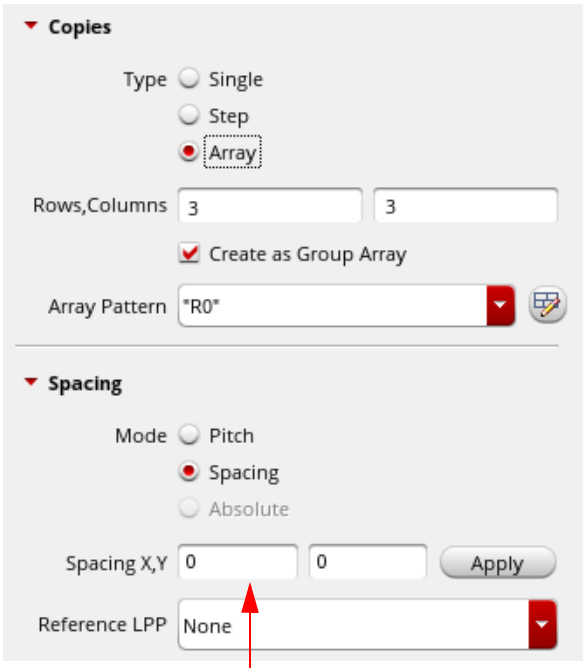
Predefined Orientation Patterns for a Group Array

Locked Group Arrays

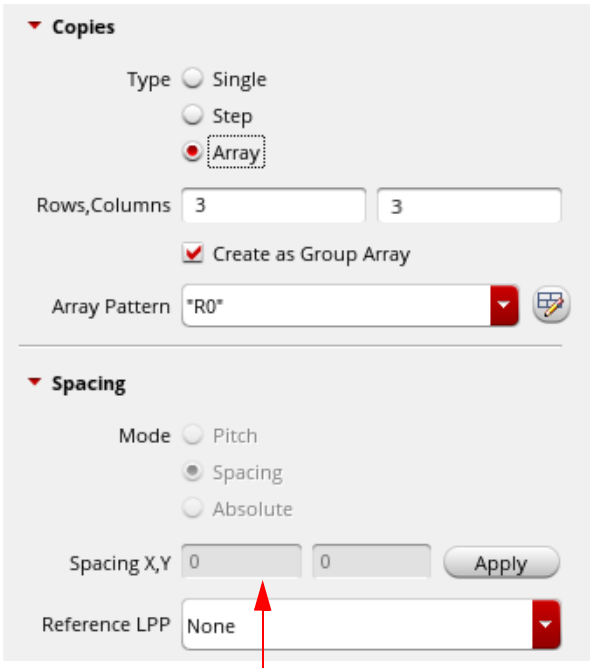
To ensure design integrity, while creating group arrays, designers can be prevented from specifying spacing mode and values by enabling the `copyGroupArrayLockXY`

© 2023 Cadence Design Systems, Inc.

environment variable. When `copyGroupArrayLockXY` is set to `t`, the options for specifying spacing mode and values for a group array appear disabled in the Copy form.



Spacing is editable when `copyGroupArrayLockXY` is `nil`

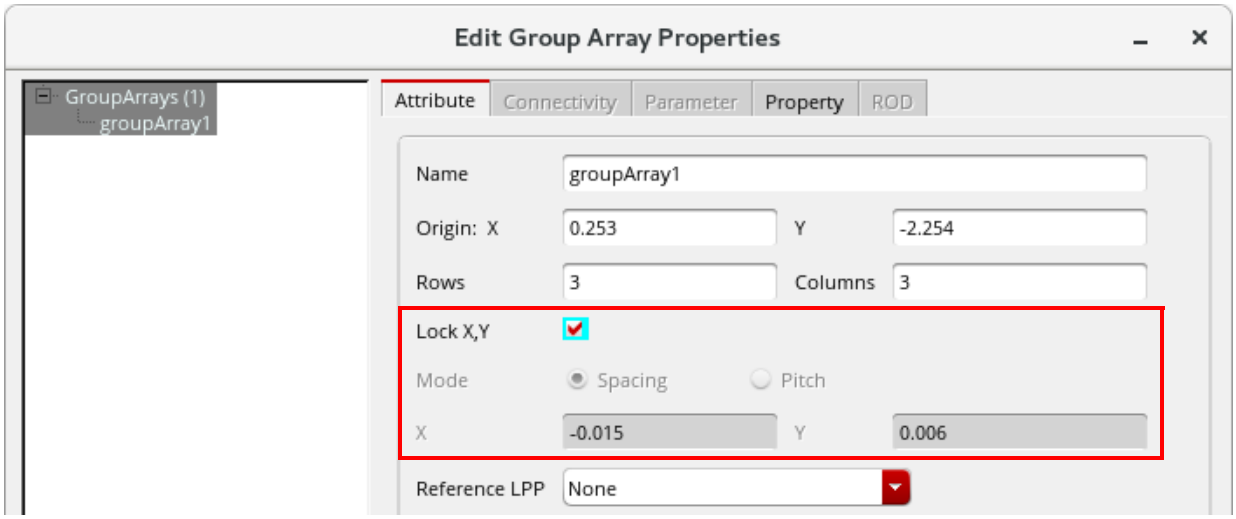


Spacing is not editable when `copyGroupArrayLockXY` is `t`

For an existing group array that has spacing defined, you can choose to lock the spacing settings so that users cannot modify them. You can lock an existing group array by selecting the *Lock X,Y* check box on the *Attributes* tab of the Edit Group Array Properties form. This

© 2023 Cadence Design Systems, Inc.

disables the *Mode* options and *X* and *Y* fields, indicating that the spacing mode and spacing values for the group array cannot be edited.



Related Topics

Locking Group Array Spacing

Group Arrays

Copy Form

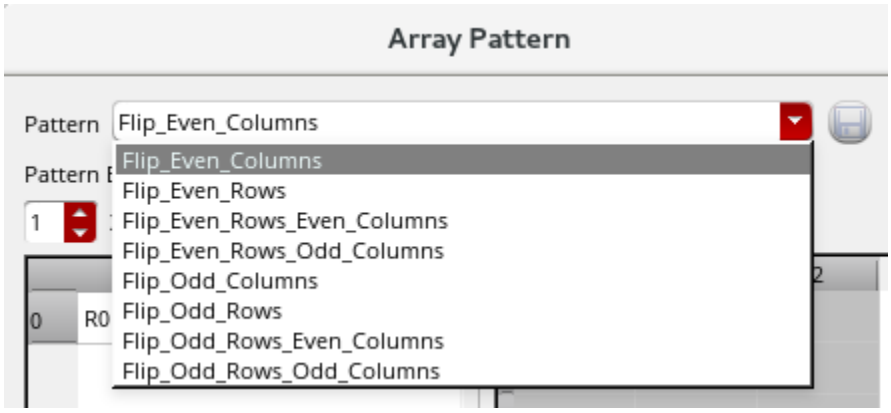
Attributes in the Edit Group Array Properties Form

Predefined Orientation Patterns for a Group Array

To define the orientation pattern for individual cells of a group array, you can choose a predefined pattern or create your own pattern. Predefined patterns cannot be deleted but you can delete a custom pattern if required.

© 2023 Cadence Design Systems, Inc.

The following image displays the list of predefined orientation patterns available for a group array in the Array Pattern form.



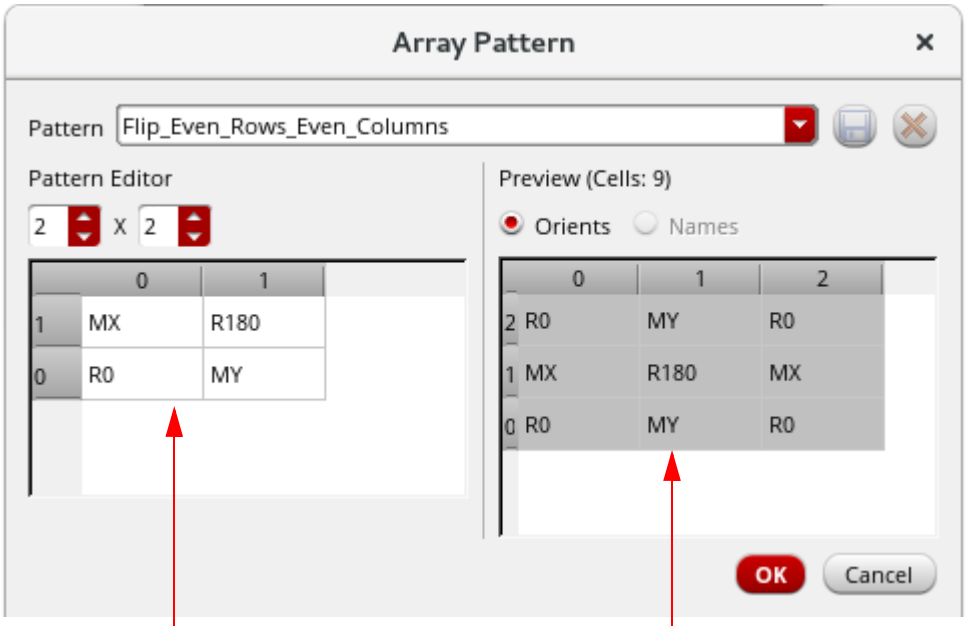
The following table describes the various predefined orientation patterns. For each predefined pattern, the table displays the orientation value for each individual cell and the total number of rows and columns available in the pattern.

Predefined Pattern	Orientation Value	Number of Rows/ Columns in a Pattern
Flip_Even_Columns	`"R0 MY"`	1x2
Flip_Even_Rows	`"R0" "MX"`	2x1
Flip_Even_Rows_Even_Columns	`"R0 MY" "MX R180"`	2x2
Flip_Even_Rows_Odd_Columns	`"MY R0" "R180 MX"`	2x2
Flip_Odd_Columns	`"MY R0"`	1x2
Flip_Odd_Rows	`"MX" "R0"`	2x1
Flip_Odd_Rows_Even_Columns	`"MX R180" "R0 MY"`	2x2
Flip_Odd_Rows_Odd_Columns	`"R180 MX" "MY R0"`	2x2

In the following image, the Array Pattern form displays the *Flip_Even_Rows_Even_Columns* pattern selected, which is a 2x2 pattern with `"R0 MY"` `"MX R180"` orientation values. The *Preview* section displays how the selected pattern is

© 2023 Cadence Design Systems, Inc.

applied to the group array. It shows the actual number of rows and columns defined for the group array and the orientation value of each cell.



Orientation values and number of rows/ columns in the pattern

Orientation values and number of rows/ columns in the group array

Your group array can be smaller or larger than a predefined pattern. Depending on the size of your group array, the selected pattern is applied accordingly. For example, when you are creating a group array that is larger than the selected pattern, the pattern is repeated across cells.

Related Topics

Group Arrays

Array Pattern Form

Creating a Group Array by Using the Copy Form

Specifying Group Array Orientation Using the Array Pattern Form

Saving a Custom Orientation Pattern for a Group Array

© 2023 Cadence Design Systems, Inc.

Group Array SKILL Functions

You can use SKILL functions to create and manage group arrays. You can use the following SKILL functions to work with group arrays:

- lxCreateGroupArray
- lxGetGroupArrayParams
- lxGroupArrayCreatePreset
- lxHiConvertMosaicToGroupArray
- lxIsGroupArray
- lxIsGroupArrayLocked
- lxIsGroupArrayRegular
- lxSetGroupArrayLocked
- lxSetGroupArrayRegular
- lxUpdateGroupArrayParams

Related Topics

Group Arrays

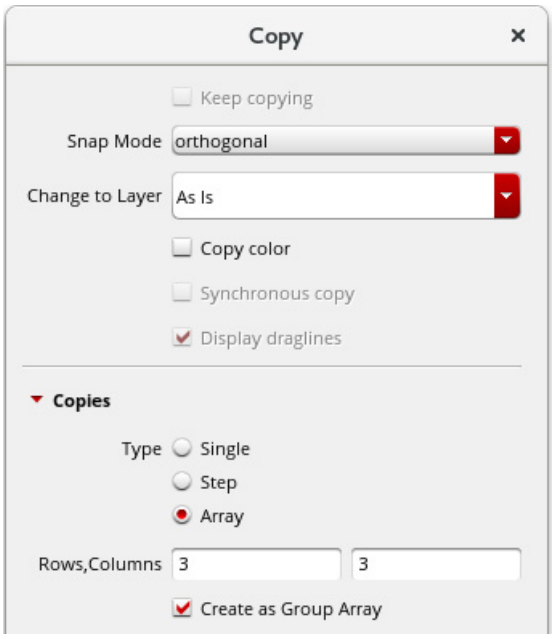
Creating a Group Array by Using the Copy Form

To create a group array by using the Copy form:

1. In the layout design window, select the objects that you want to repeat as a pattern in your design.
2. Choose *Edit – Copy* and then press `F3` to open the Copy form.
3. In the *Copies* section, select the type of copy you want to create. Depending on the option you select, specify the number of copies or rows and columns in the required fields.

© 2023 Cadence Design Systems, Inc.

4. Select the *Create as Group Array* check box.



5. In *Array Pattern*, specify the orientation value of individual cells of the group array by using any of the following methods:

   - Enter the orientation values directly in the text field by enclosing them in quotation marks.
   - Select a pattern from the drop-down list, which shows predefined and custom patterns.
   - Use the Array Pattern form, which you can access by clicking the *Open Array Pattern editor* button .

     For information, see Specifying Group Array Orientation Using the Array Pattern Form.

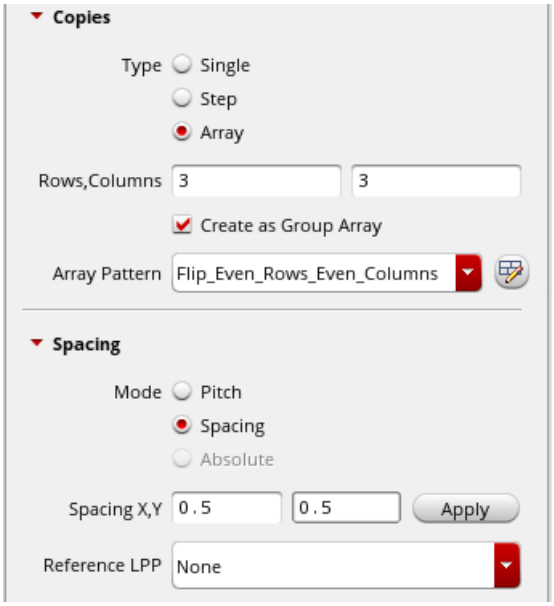
6. In the *Spacing* section, select a mode and specify the spacing values:

   - *Pitch*: Specify the spacing in the *Delta X,Y* fields.
   - *Spacing*: Specify the spacing in the *Spacing X,Y* fields.

   If `copyGroupArrayLockXY` is set to `t`, you cannot specify spacing mode and spacing values for a group array.

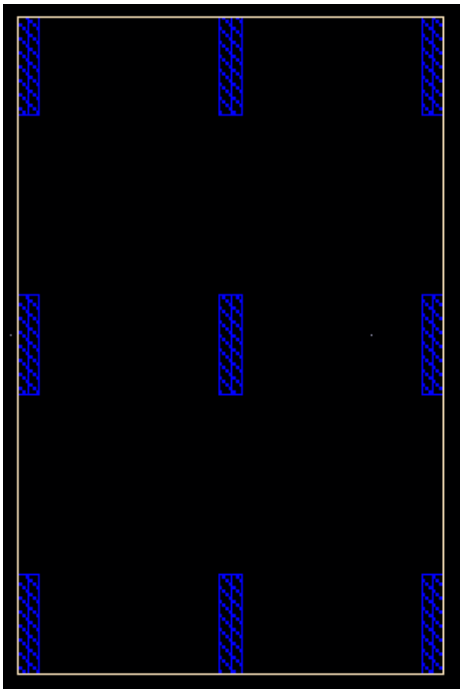
© 2023 Cadence Design Systems, Inc.

7. Optionally, in *Reference LPP*, select the reference layer-purpose pair that needs to be used for computing the bounding box of the cells of the group array.



8. Specify other settings in the Copy form as required.
9. Click *Apply* in the *Spacing* section of the form.

   The group array with the specified parameters is created.



© 2023 Cadence Design Systems, Inc.

Related Topics

Copy Form

Group Arrays

Array Pattern Form

lxCreateGroupArray

Predefined Orientation Patterns for a Group Array

Saving a Custom Orientation Pattern for a Group Array

Group Array Creation

Specifying Group Array Orientation Using the Array Pattern Form

You can define the orientation of individual cells of a group array. While creating or editing a group array, you can specify the orientation by entering values directly in the *Array Pattern* field, by selecting an existing pattern from the drop-down list, or by using the Array Pattern form.

To specify the orientation pattern to be applied to a group array by using the Array Pattern form:

1. In the layout design window, open the required form.
   - If you are creating a group array, ensure the required objects are selected on the canvas and the relevant command form is open; for example, the Copy form. Also, ensure you have specified the required settings in the form.

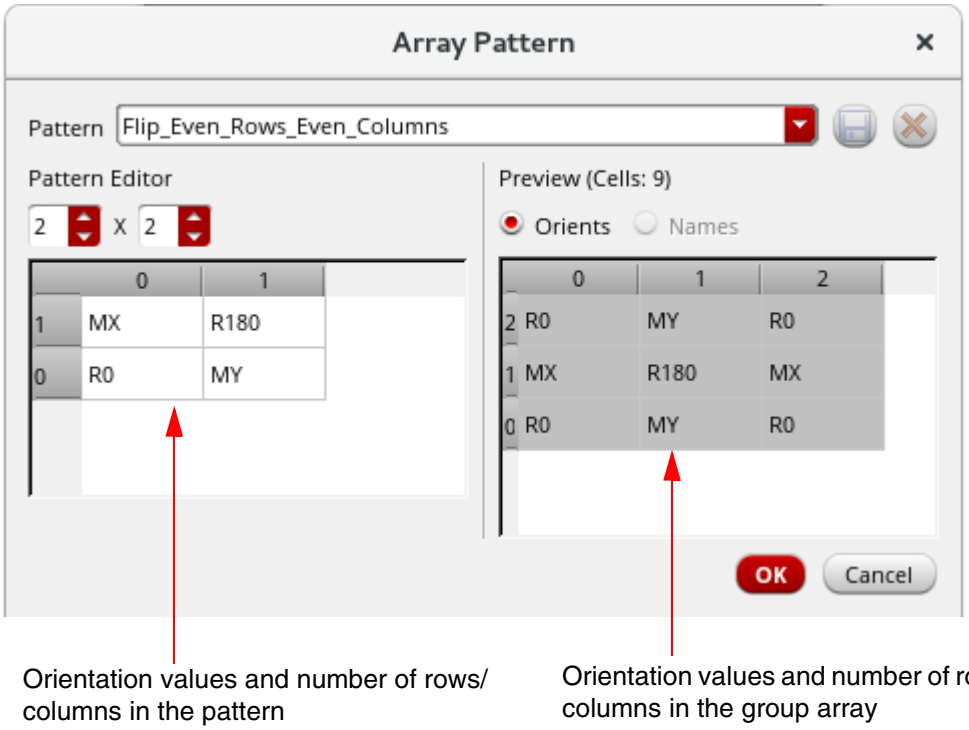
     A group array can be created by using the Copy, Generate Selected Components, and Generate Clones forms.
   - If you are editing a group array, ensure the Edit Group Array Properties form is open for the group array.
2. Click the *Open Array Pattern editor* button  , which is next to the *Array Pattern* field.

   The Array Pattern form opens.
3. Select a pattern from the *Pattern* list.

© 2023 Cadence Design Systems, Inc.

The selected pattern is populated in the *Pattern Editor* section and a preview is displayed in the *Preview* section.

Note that the table in the *Pattern Editor* section displays the orientation values and the number of rows and columns available in the selected pattern. Whereas, the *Preview* section displays how the selected pattern is applied to the actual number of cells in your group array. The actual number of cells is derived from the number of rows and columns you specify in the Copy form or the Edit Group Array Properties form.

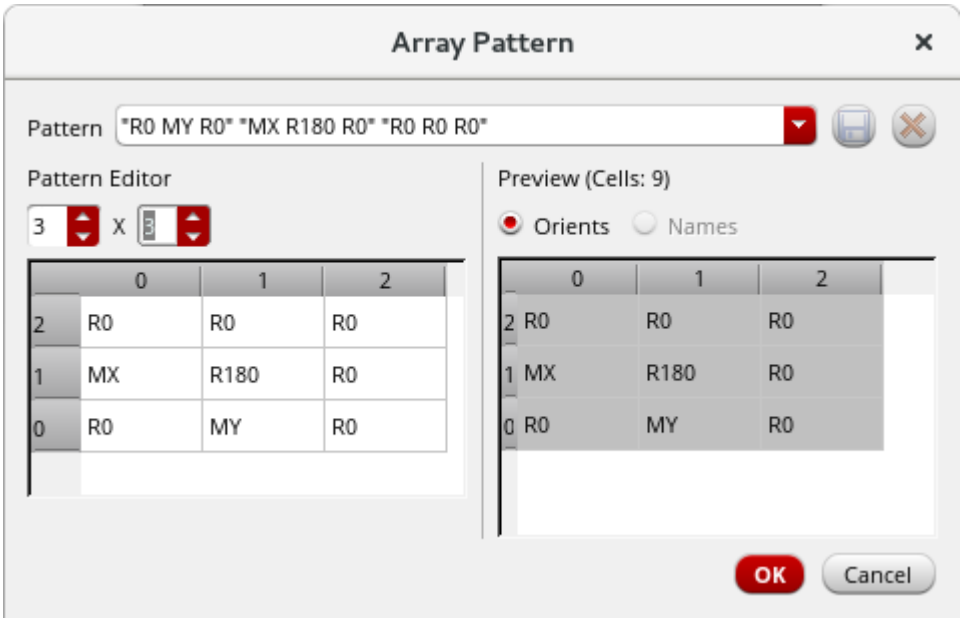


4. Optionally, use the spin boxes in the *Pattern Editor* section to modify the row and column count in the pattern.

   When you change the row and column count, the updated pattern gets populated in the *Pattern Editor* section. Also, the updated orientation values are listed in the *Pattern*

© 2023 Cadence Design Systems, Inc.

field of the form and the *Preview* section displays the updated pattern applied to the group array, as shown in the following image.



5. Click *OK*.
6. Click *OK*. If you are using the Copy form, click *Apply*.

   The specified orientation pattern is applied to the group array.

Related Topics

Copy Form

Array Pattern Form

Attributes in the Edit Group Array Properties Form

Predefined Orientation Patterns for a Group Array

Saving a Custom Orientation Pattern for a Group Array

Generate Selected Components Form

Generate Clones Form

© 2023 Cadence Design Systems, Inc.

Saving a Custom Orientation Pattern for a Group Array

While creating or editing a group array, you can define and save your own orientation patterns. You can reuse these patterns for group arrays across cellviews and sessions. If required, you can delete the custom orientation patterns.

To define and save a custom orientation pattern for a group array:

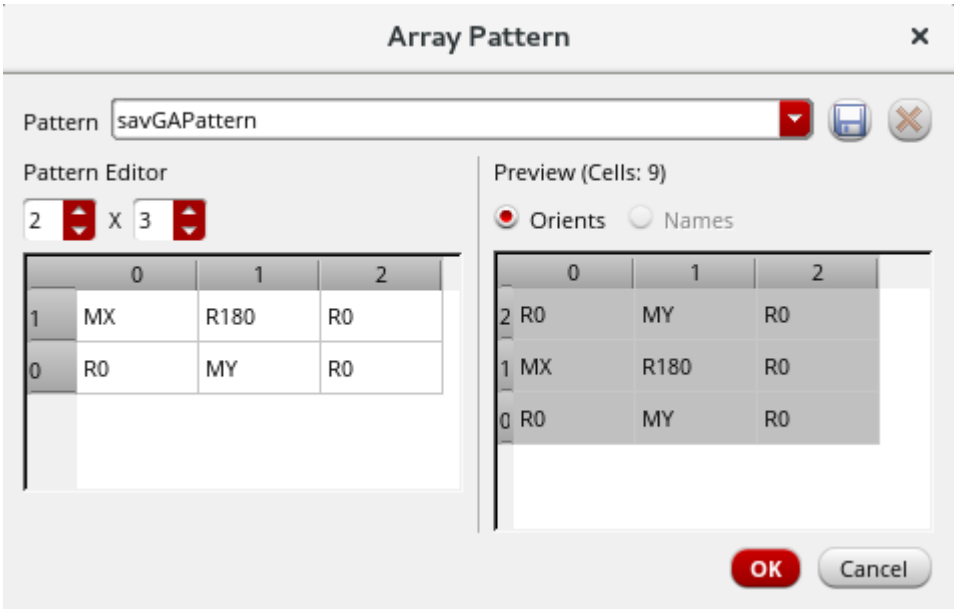
1. In the layout design window, open the required form.
   - If you are creating a group array, ensure the required objects are selected on the canvas and the relevant command form is open; for example, the Copy form. Also, ensure you have specified the required settings in the form.

     A group array can be created by using the Copy, Generate Selected Components, and Generate Clones forms.
   - If you are editing a group array, ensure the Edit Group Array Properties form is open for the group array.
2. Click the *Open Array Pattern editor* button  , which is next to the *Array Pattern* field.

   The Array Pattern form appears.
3. In the *Pattern Editor* section, use the spin boxes to modify the row and column count for the pattern.
4. In the orientation pattern table in the *Pattern Editor* section, double-click any cell and select the desired orientation from the drop-down list.
5. Similarly, select the orientation for other cells of the group array as per your requirement.
6. In the *Pattern* field, delete the orientation values and enter a name for the custom pattern.

© 2023 Cadence Design Systems, Inc.

As you specify a name, the *Save orient pattern* button is enabled.



7. Click to save the custom orientation pattern.

   The custom orientation pattern is listed in the *Pattern* drop-down list.

   The *Delete orient pattern* button is now enabled, which means that you can delete this pattern if required. Note that you can save and delete only the custom orientation patterns and not the predefined orientation patterns.

8. Click *OK*.

   The specified orientation pattern is created.

Related Topics

Copy Form

Array Pattern Form

Attributes in the Edit Group Array Properties Form

Predefined Orientation Patterns for a Group Array

Specifying Group Array Orientation Using the Array Pattern Form

Group Array Creation

© 2023 Cadence Design Systems, Inc.

Generate Selected Components Form

Generate Clones Form

Group Arrays

Group Array Editing

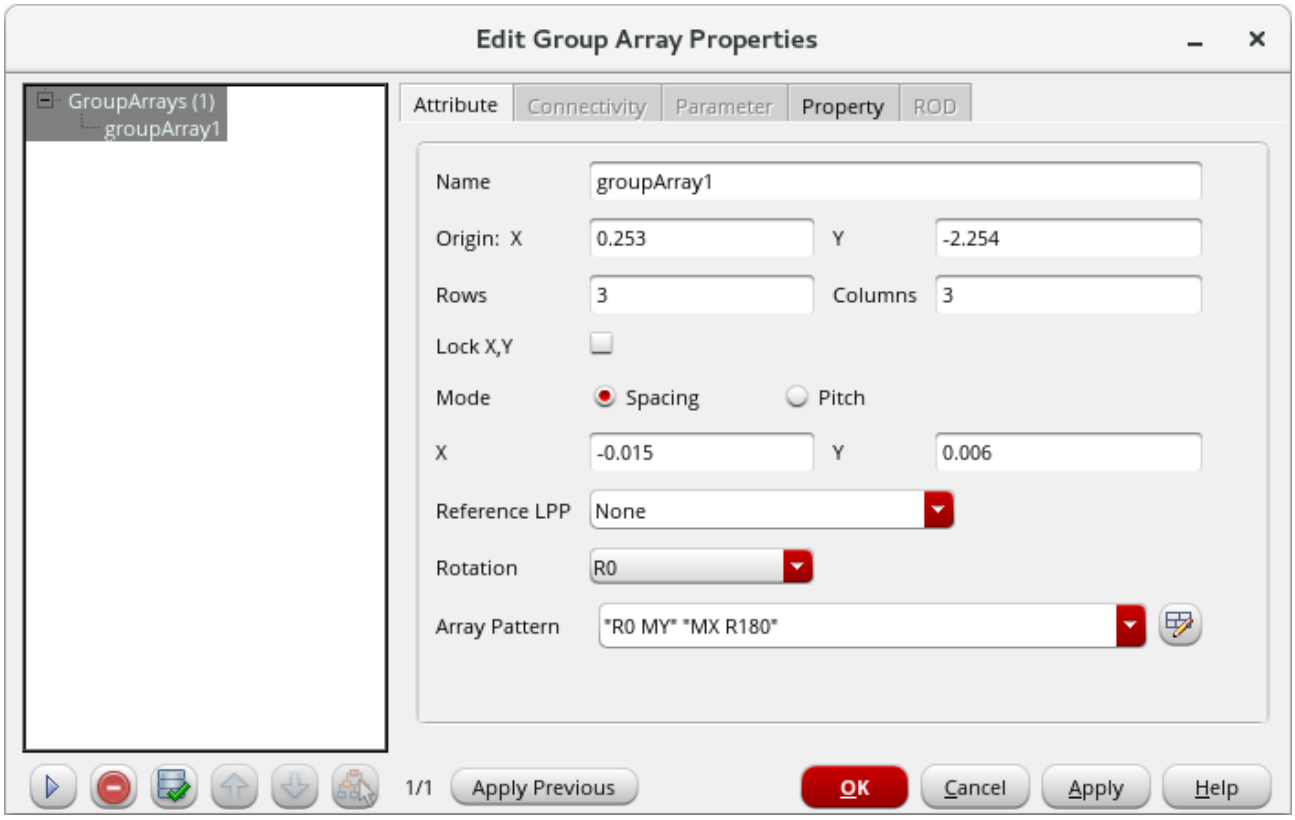
While working with a group array, you can perform basic functions such as stretching, moving, chopping, or ungrouping the group array. The drag mode set in the `groupArrayDragMode` environment variable is used while stretching a group array.

You can edit a group array by using the Edit Group Array Properties form, the Property Editor assistant, or the `lxUpdateGroupArrayParams` SKILL function. You can modify the following attributes of a group array:

- Number of rows and columns
- Spacing mode and values
- Orientation of individual cells
- Holes in the pattern
- Lock status of the group array, whether users can modify spacing mode and values

© 2023 Cadence Design Systems, Inc.

The following image shows group array properties displayed in the Edit Group Array Properties form.



Related Topics

Modifying the Number of Cells in a Group Array

Creating Holes in a Group Array

Locking Group Array Spacing

Converting a Mosaic to a Group Array

Object Stretching

Object Moving

Array Chopping

Group Arrays

© 2023 Cadence Design Systems, Inc.

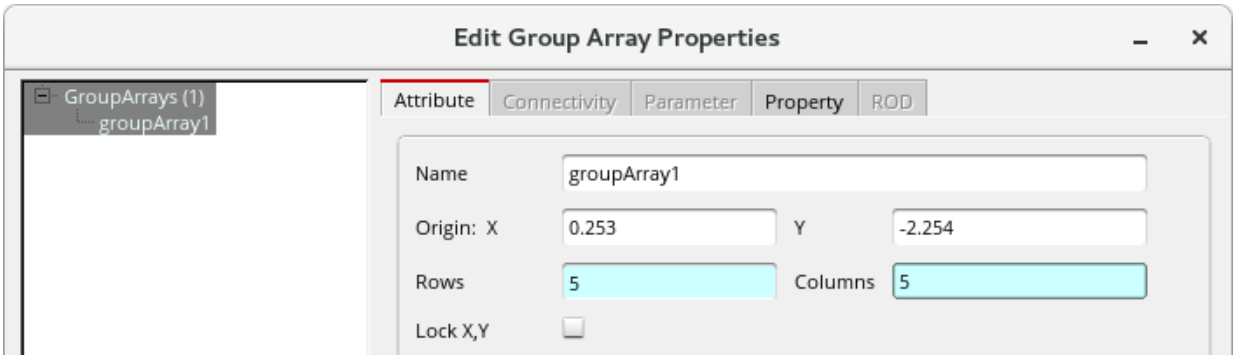
Modifying the Number of Cells in a Group Array

You can increase or decrease the number of cells in a group array by modifying the number of rows or columns. You can use the Property Editor assistant, Edit Group Array Properties form, or the *Stretch* command. With the *Stretch* command, you can modify the number of cells in a group array interactively by using the mouse pointer.

To increase or decrease the number of cells in a group array by using the Edit Group Array Properties form:

1. In the layout design window, select the group array you want to modify.
2. Choose *Edit – Basic – Properties*.

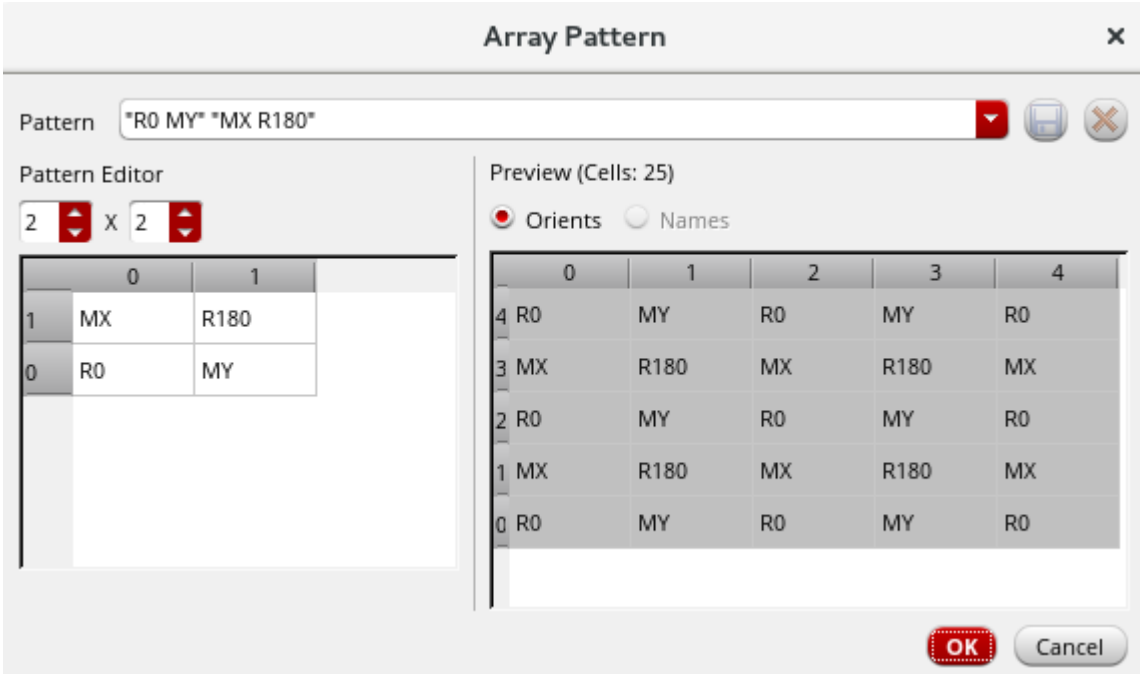
   The Edit Group Array Properties form appears.
3. On the *Attribute* tab, in the *Rows* field, specify the number of cell rows you want in the group array.
4. In the *Columns* field, specify the number of cell columns you want in the group array.



5. To preview the group array, click the *Open Array Pattern editor* button , which is next to the *Array Pattern* field.

© 2023 Cadence Design Systems, Inc.

In the following image, the *Preview* section in the Array Pattern form shows a sample 5x5 group array, with the updated number of rows and columns.



6. Click *OK*.
7. Click *OK*.

   The number of rows and columns in the group array are updated.

Related Topics

Attributes in the Edit Group Array Properties Form

Group Array Editing

Object Stretching

Group Arrays

lxGetGroupArrayParams

lxUpdateGroupArrayParams

© 2023 Cadence Design Systems, Inc.

Creating Holes in a Group Array

In a group array, you can choose to leave some cells empty to denote that there are no geometries underneath. Such cells are referred as holes. You can add a hole by specifying dash (-) as the orientation value of a cell while creating or editing a group array.

To create holes in a group array by using the Array Pattern form:

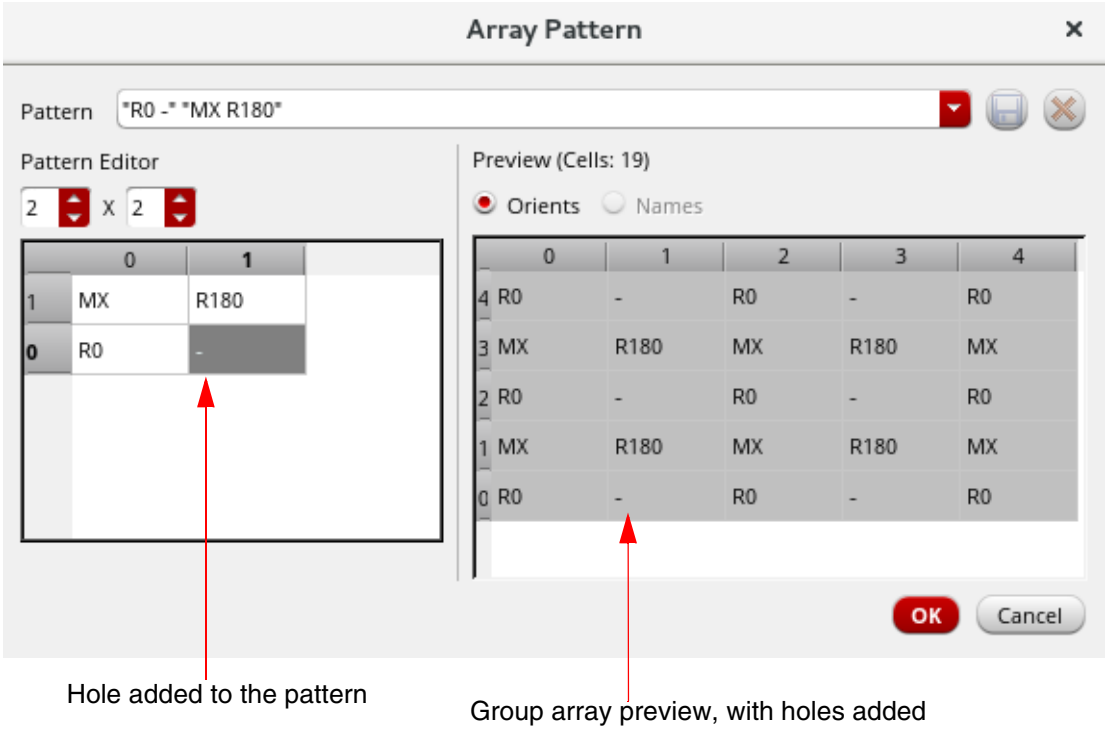
1. In the layout design window:
   - To add holes while creating a group array, choose *Edit – Copy* and then press `F3` to open the Copy form.
   - To create holes in an existing group array:
     - Select the group array you want to modify.
     - Choose *Edit – Basic – Properties.*
     - Press `F3` to open the Edit Group Array Properties form.
2. Click the *Open Array Pattern editor* button  , next to the *Array Pattern* field, to specify the orientation pattern.

   The Array Pattern form appears.
3. Optionally, select an existing pattern from the *Pattern* list if you want to modify the orientation pattern.

   The selected array pattern is populated in the *Pattern Editor* section.
4. Optionally, use the spin boxes in the *Pattern Editor* section to modify the row and column count of the pattern.
5. In the orientation pattern table of the *Pattern Editor* section, double-click the cell that you want to keep as empty. Then, from the drop-down list, select dash (`-`) as the orientation value.

© 2023 Cadence Design Systems, Inc.

The value is updated in the orientation tables, both in the *Pattern Editor* and *Preview* sections.



6. Repeat step 5 if you want to add more holes to the group array.

   Instead of clicking each cell and selecting the orientation value, you can specify values in the *Pattern* text field directly.

7. Click *OK* in the Array Pattern form.

8. Click *OK*.

   The holes are added to the group array.

You can also add holes by specifying the required orientation pattern directly in the *Array Pattern* field in the Copy form or the Edit Group Array Properties form.

Related Topics

Array Pattern Form

Attributes in the Edit Group Array Properties Form

Holes in a Group Array

© 2023 Cadence Design Systems, Inc.

Group Array Editing

Group Arrays

Copy Form

lxUpdateGroupArrayParams

Locking Group Array Spacing

By default, you can specify spacing mode and spacing values while creating a group array using the Copy form. If the `copyGroupArrayLockXY` environment variable is set to `t`, you cannot define spacing settings during group array creation.

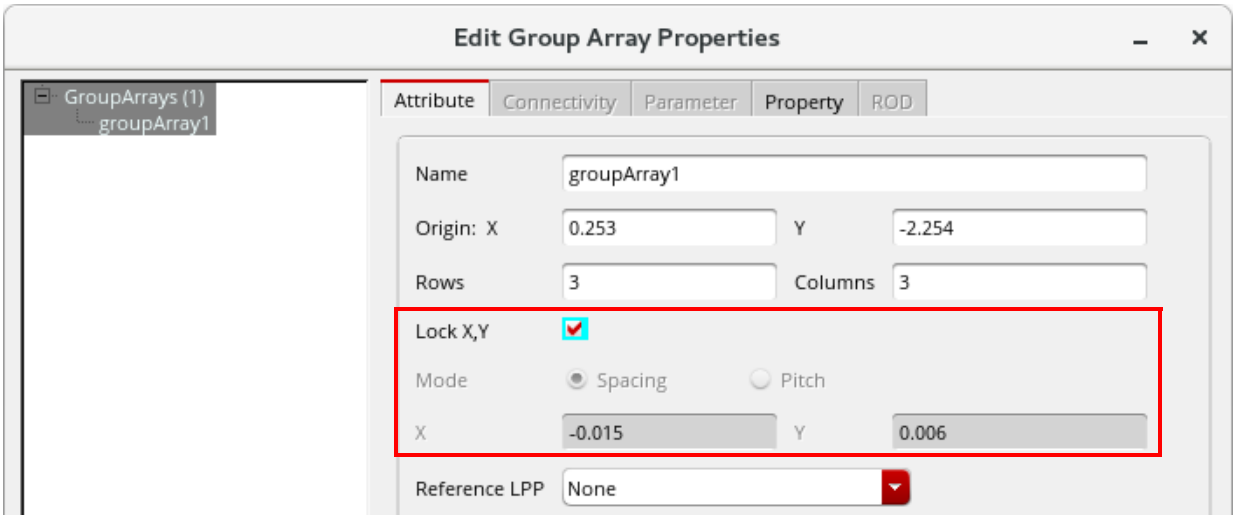
For an existing group array that has custom spacing defined, you can choose to lock the spacing settings so that users cannot modify them.

To lock spacing mode and values of an existing group array:

1. In the layout design window, select the group array you want to lock.
2. Choose *Edit – Basic – Properties*.

   The Edit Group Array Properties form appears.
3. Select the *Lock X,Y* check box.

   This disables the *Mode* options and the *X* and *Y* fields, indicating that spacing mode and spacing values for the group array cannot be edited.



© 2023 Cadence Design Systems, Inc.

4. Click *OK*.

Related Topics

Attributes in the Edit Group Array Properties Form

Locked Group Arrays

Group Array Editing

Group Arrays

lxIsGroupArrayLocked

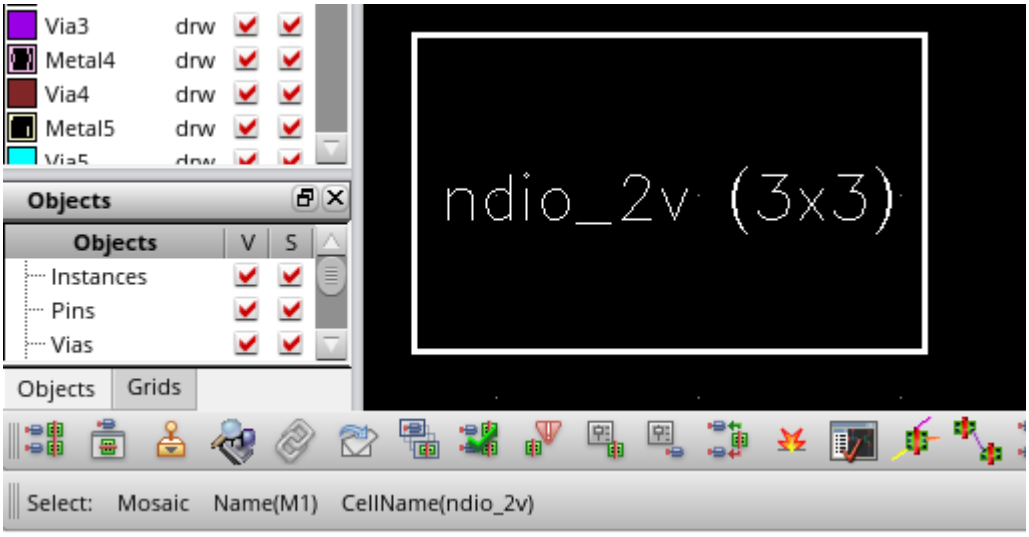
lxSetGroupArrayLocked

Converting a Mosaic to a Group Array

To convert a mosaic to a group array:

1. In the layout design window, select the mosaic you want to convert.

   The following image shows a 3x3 mosaic selected. Note the message on the status bar indicates that it is a mosaic.

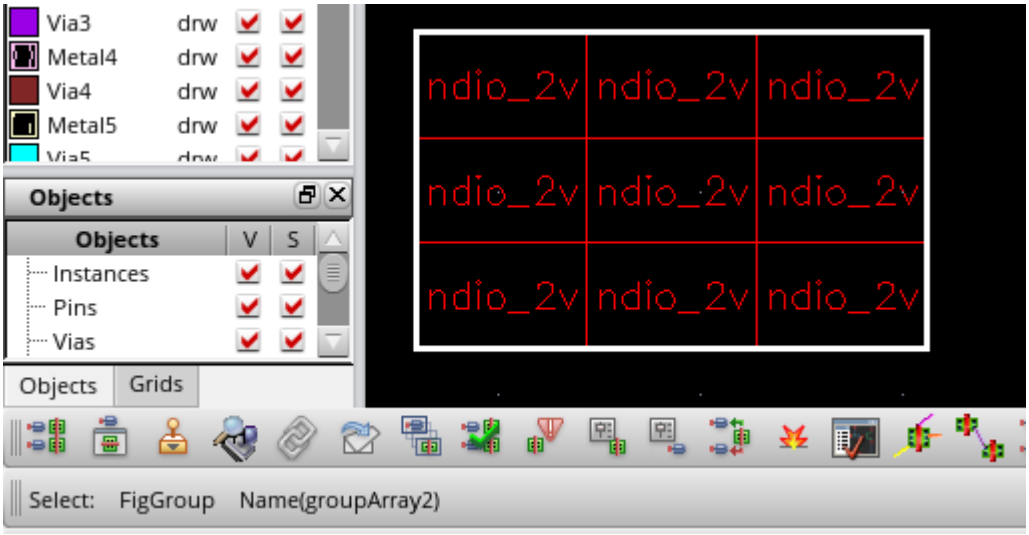


2. Choose *Edit – Convert – To GroupArray*.

   Alternatively, you can select the *Convert To GroupArray* command from the shortcut menu.

© 2023 Cadence Design Systems, Inc.

The following image shows that the mosaic is converted to a 3x3 group array. Note the message on the status bar indicates that the selection is a figGroup with the name *groupArray2*.



Related Topics

Group Arrays

Group Array Editing

lxHiConvertMosaicToGroupArray

© 2023 Cadence Design Systems, Inc.

10

Connectivity in Layout XL

Basic connectivity in Layout XL includes defining the following:

- Pin connectivity: You can define the level at which to connect a pin or group of pins in a net, internally or externally.
- Net propagation: You can map terminals on an instance or a homogenous mosaic to the nets in a cellview.
- Net and shape assignment: You can add selected shapes to nets that are attached to pins. Also, you can assign and unassign nets to shapes, instance pins, and routed objects like vias and pathSegs.
- Net tracing: You can visually trace the physical connectivity of nets in a layout design without having to use a schematic.
- Object probing: You can find and highlight certain types of objects from among all the objects present in a design and highlight the logical connectivity between objects.

Related Topics

Pin Connectivity

Propagating Nets

Adding and Deleting Shapes on Nets

Assigning and Unassigning Nets on Shapes

Ways to Select Shapes Based on Nets

Net Tracing Using the Mark Net Command

Probing Nets, Instances, Pins, and Shapes

© 2023 Cadence Design Systems, Inc.

Pin Connectivity

Any object with pins is a device that includes transistors, resistors, or capacitors. By default, pins are connected internally. The *Connectivity – Pins* command lets you specify to the router the level at which to connect a pin or group of pins in a net, internally or externally.

- *Connectivity – Pins – Must Connect* command: Makes the router to connect the selected pin or group of pins in a net externally, at a higher level. When you connect pins externally, you do not need to route between the pins inside the cell.
- *Connectivity – Pins – Strongly Connected* command: Makes the router to connect the selected pin or group of pins in a net internally, within the device. When you connect pins internally (strong connect), at least one pin in the group must be connected externally.
- *Connectivity – Pins – Weakly Connected* command: Makes the router to connect the selected pin or group of pins in a net externally. Similar to strongly connected pins, all weakly connected pins must also be connected together internally. Weakly connected pins have limited external connection to avoid specific internal connections, typically ones with high-resistance paths.
- *Connectivity – Pins – Pseudo Parallel Connect* command: Specifies to the router that the selected instance terminals within an instance do not need to be connected, even though they are on the same net.

Related Topics

Benefits of Defining Pin Connectivity

Defining Must-Connect Pins

Defining Strongly Connected Pins

Defining Weakly Connected Pins

Defining Pseudo Parallel Connected Nets

Benefits of Defining Pin Connectivity

The ability to define connection types for pins can be useful if:

- You are a library developer designing a library device for use in a block or circuit design and need to provide layout designers with maximum flexibility by defining some

© 2023 Cadence Design Systems, Inc.

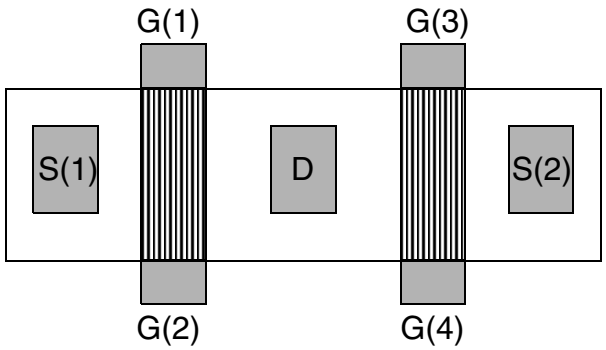
connections internally and leaving other connections to be completed at the next level of design.

- You are a layout designer using a device designed by others in a layout and need to specify, later in the layout cycle, which pins to connect externally.

The following examples show how the ability to define pins as must connects, strongly connected, weakly connected, or pseudo parallel connects can make the design process more efficient for both these job functions.

For Library Developers

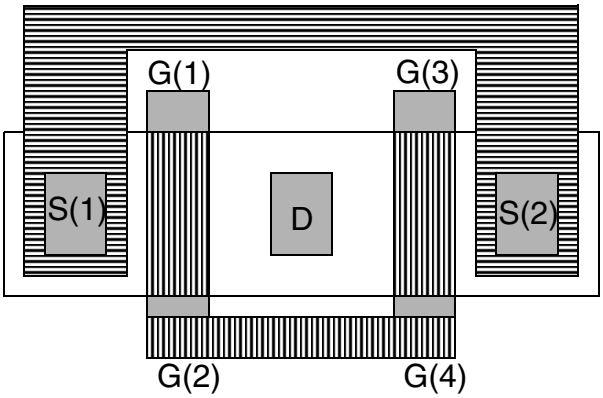
If you are working as a library developer, a typical device might be a multi-fingered FET as shown in the following figure.



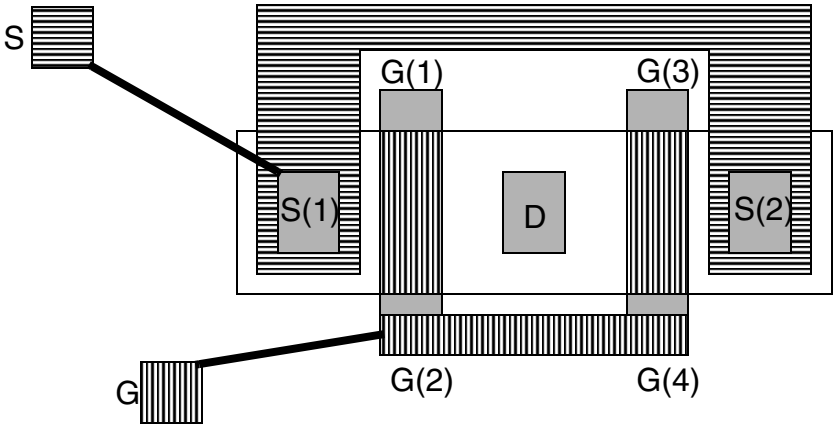
- G(1) and G(2) are weakly connected (inside the device)
- G(3) and G(4) are weakly connected (inside the device)
- The pair G(1)/G(2) must be externally connected to the pair G(3)/G(4)
- S(1) must be externally connected to S(2)

© 2023 Cadence Design Systems, Inc.

To use this device in a design, if no external (must-connect) connections are defined, you must make internal (strongly connected) connections so the cell looks like this:



Then, when this device is used in a design, the flight lines look as shown in the following figure:

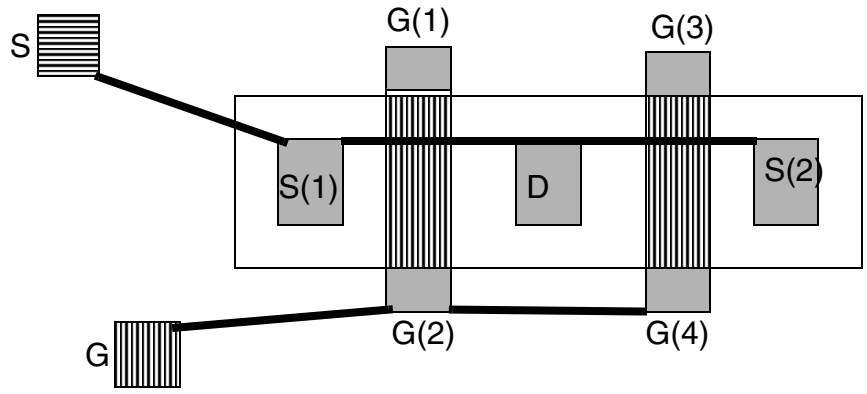


If the layout designer makes connections between G and G(2) or G(4), the layout editor accepts the connection as complete. If the layout designer makes connections between S and S(1) or S(2), the layout editor accepts the connection as complete.

If you, as a library developer, do not want to complete the connections to the source or gate inside the device, but want the connections made at the next level of hierarchy, you do not

© 2023 Cadence Design Systems, Inc.

wire internal connections. Then, when the cell is used in a design, the flight lines look as shown in the following figure:



If the layout designer makes a connection between S and S(1), the software does not accept the connection as complete and displays a flight line to S(2). If the layout designer makes a connection between G and G(2), the software does not accept the connection as complete and displays a flight line to G(4).

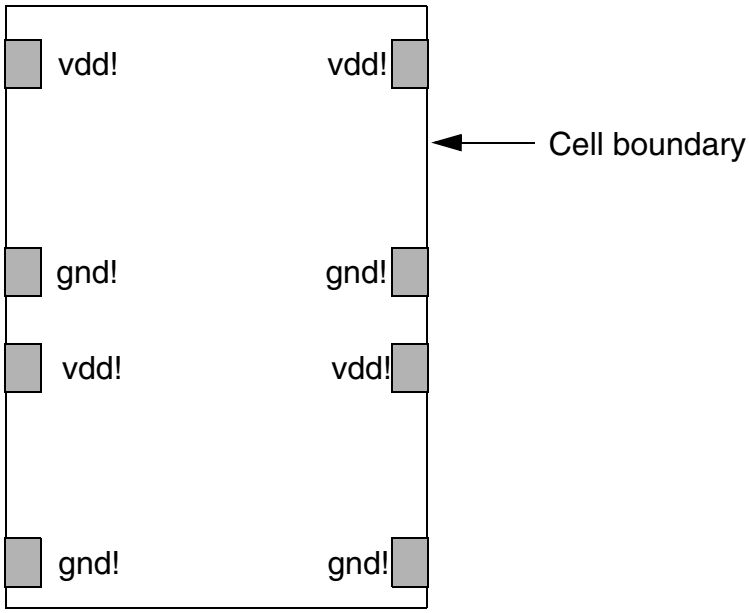
If you make a connection between G and G(2) and G(4), the software accepts the connection as complete and displays no more flight lines on this net, because G(1) and G(3) are weakly connected to G(2) and G(4) respectively.

If you make a connection between G and G(2), and between G(1) and G(3), this is considered an error and is indicated by a marker. The software would not accept the internal weak connection between G(1) and G(2) to complete the net.

© 2023 Cadence Design Systems, Inc.

For Layout Designers

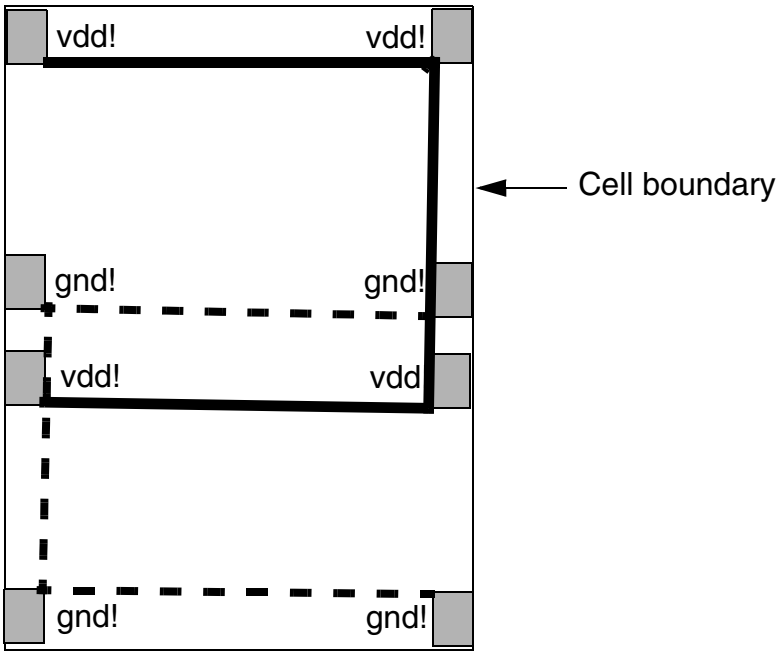
If you are working as a layout designer, a typical design might be a two-row-high standard cell layout as shown below.



Floorplanning information regarding external connections is provided on the boundary of the design by an outside source, such as a circuit designer or library. Given that information early, you can identify which pins must be must connects, strong connects, or weak connects, and can use that information for device placement and wiring.

© 2023 Cadence Design Systems, Inc.

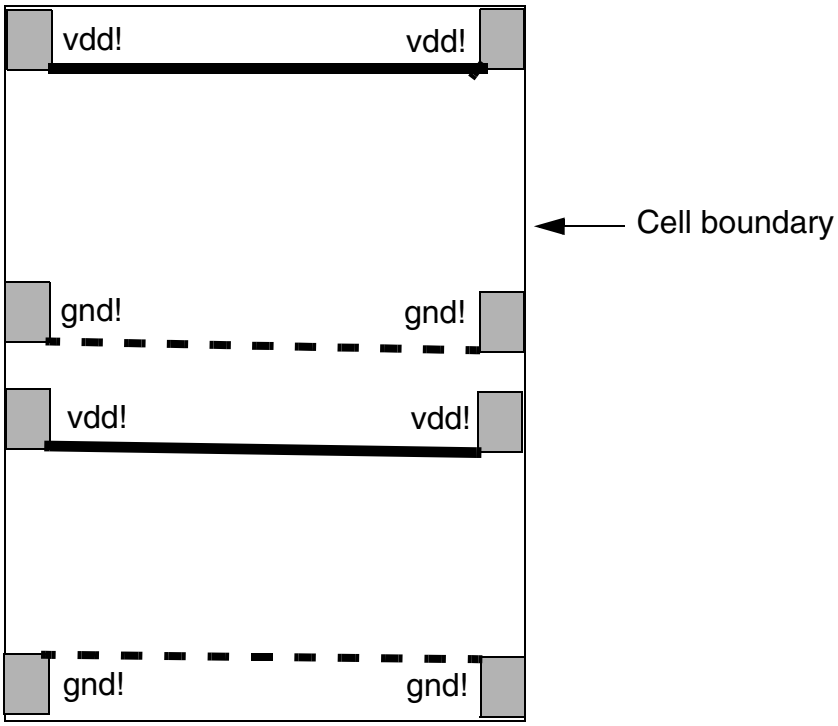
If you define no external connections, the flight lines look as shown below:



The library developer usually does not connect the top two vdd! pins to the bottom two vdd! pins but does connect the top two vdd! pins together. The same is true for the gnd! pins. So, the library developer defines the top two vdd! pins as being externally connected to the

© 2023 Cadence Design Systems, Inc.

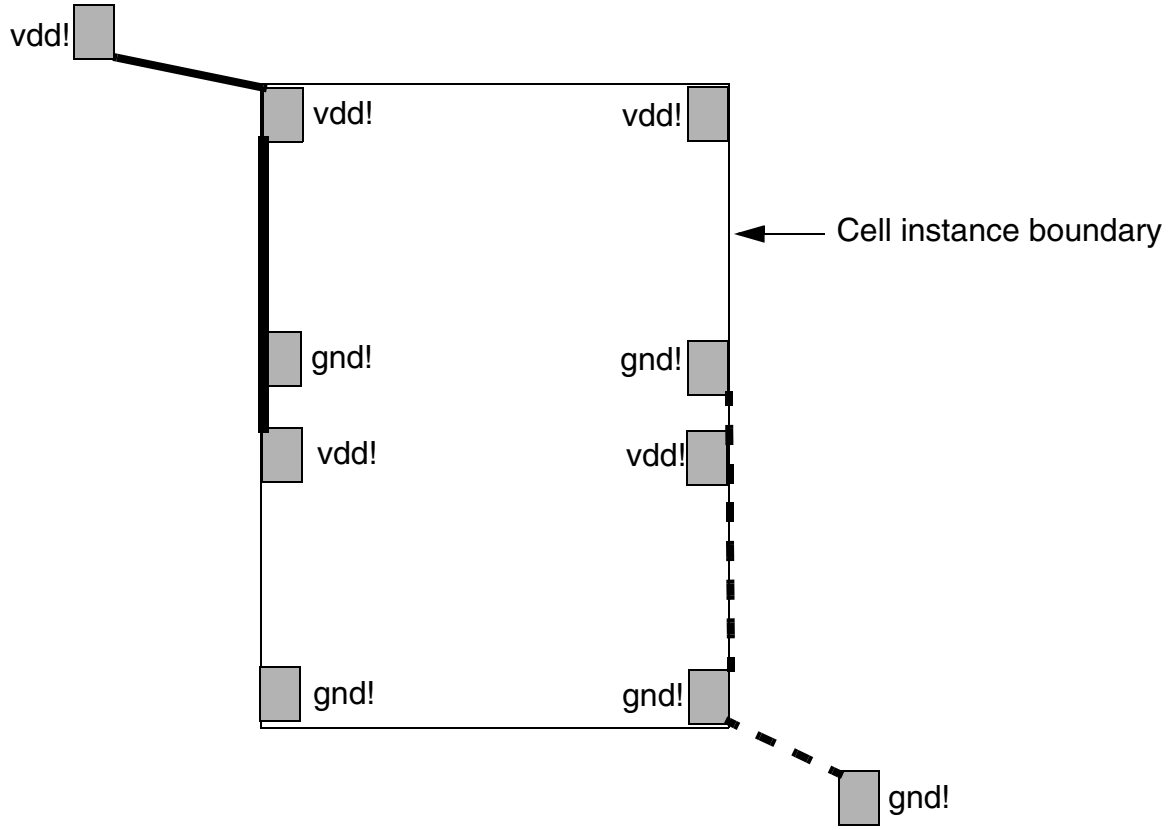
bottom two vdd! pins and the same for the gnd! pins. Then, when you are working inside the cell, the flight lines look as shown below:



When you use the cell in the next level of hierarchy, the flight lines look as shown in the following image. You must make the connections as shown by the flight lines. You are

© 2023 Cadence Design Systems, Inc.

provided information regarding pin connections by an outside source, such as a circuit designer or library.



Related Topics

Pin Connectivity

Defining Must-Connect Pins

To define a set of pins on a net to be connected externally:

1. In the layout window, choose *Connectivity – Pins – Must Connect*.

   The window prompts you to click a pin figure to select a net.

2. To select the net that includes the pins you want to connect externally, click a shape on that net.

   You can select a net by selecting a shape in the layout before you choose the *Must Connect* command.

© 2023 Cadence Design Systems, Inc.

Flight lines appear, connecting only the pins of the net you choose. The flight lines are drawn between pins that still need to be internally (strongly) connected. If incomplete nets are being displayed, the other net flight lines disappear. Flight lines are visible in Layout XL and higher tiers.

Selecting a net does not make a connection change.

The window prompts you to select one or more pins to be connected externally.

3. To select the pins you want to connect externally, click one or more pins on the selected net.

   You can click a pin, `shift`-click subsequent pins, or select pins by area selection by clicking and dragging with the right mouse button.

   Clicking the pins of the selected net makes the connection change.

   - If you select a single pin, all other pins on the net are deselected and that pin is connected externally to all other pins on the net.
   - If you select multiple pins, those pins are connected strongly (internally) as a set and are connected as must connects (externally) to all other pins on the net.

   When you define the connection of each pin, the flight line display of the net immediately changes to show the connection.

   You can also use the Define Must Connected Pins form to select nets.

4. Press `F3` to open the Define Must Connected Pins form, while the *Must Connect* command is active.



5. Type the name of a net in the *Net* field or click the net names in the list box of the form to select them.

   To select more than one net at a time to work on, you can:

© 2023 Cadence Design Systems, Inc.

- Choose multiple nets in the form.
- Use drag by area when the command line prompts you to select a net.

You must define external pin connectivity within a net. You cannot define external pin connectivity between nets.

6. When you are finished, press `Esc` to end the command.

   When you define pins as must-connects, the router routes them at the next level of the hierarchy.

You can change from the *Must Connect* command to the *Strongly Connected* command, to the *Weakly Connected* command, to the *Pseudo Parallel Connect* command with the right mouse button. Must Connect pins must be changed to Strongly Connected before being changed to Weakly Connected.

Related Topics

Define Net/Pin Connections Form

Flight lines

Defining Strongly Connected Pins

To define a set of pins on a net to be connected strongly (internally):

1. In the layout window, choose *Connectivity – Define Pins – Strongly Connected*.

   The window prompts you to click a pin figure to select a net.

2. To select the net that includes the pins you want to connect, click a shape on that net.

   You can select a net by selecting a shape in the layout before you choose the *Strongly Connected* command.

   Flight lines appear, connecting only the pins of the net you chose. The flight lines are drawn between pins that still need to be strongly connected. If incomplete nets are being displayed, the other net flight lines disappear.

   Selecting a net does not make a connection change.

   The window prompts you to select two or more pins to be connected strongly.

3. To select the pins you want to connect strongly, click one or more pins on the selected net.

© 2023 Cadence Design Systems, Inc.

You can click a pin, `Shift`+click subsequent pins, or select pins by area by clicking and dragging with the right mouse button.

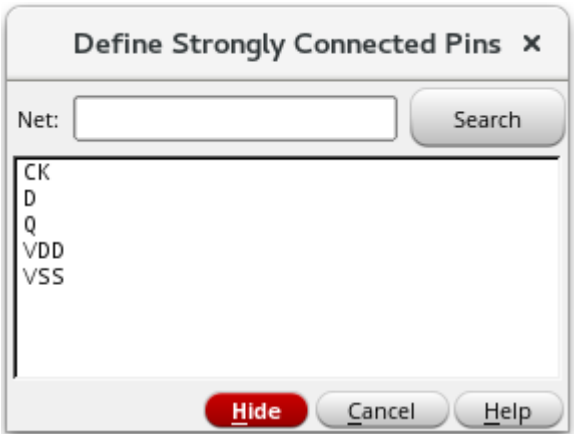
Clicking on the pins of the selected net makes the connection change.

- If you select a single pin, the other pins are not deselected, and that pin is connected strongly to the current selection of strongly connected pins.
- If you select multiple pins, those pins are connected strongly to all other strongly connected pins on the net.

When you click pins to be strongly connected, the net flight line display immediately changes to show the new information.

You can also use the Define Strongly Connected Pins form to select a net.

4. Press `F3` to open the Define Strongly Connected Pins form, while the *Strongly Connected Pins* command is active.



5. Type the name of a net in the *Net* field or click net names in the list box of the form to select them.
6. On the layout canvas, select the pins you want to define.

   When you add or remove pins to be strongly connected, the net flight line display changes immediately to show the new information.

   To select more than one net at a time to work on, you can:

   - Choose multiple nets in the form.
   - Use drag by area when the command line prompts you to select a net.

   You must define strong pin connectivity within a net. You cannot define strong pin connectivity between nets.

© 2023 Cadence Design Systems, Inc.

7. When you have finished, press `Esc` to end the command.

   When you define pins as strongly connected, the router routes them within the design.

You can change from the *Strongly Connected* command to the *Must Connect* command, to the *Weakly Connected* command, to the *Pseudo Parallel Connect* command with the right mouse button.

Related Topics

Define Net/Pin Connections Form

Flight lines

Defining Weakly Connected Pins

To define a set of pins on a net to be connected externally because the internal physical connection has a significant resistance:

1. In the layout window, choose *Connectivity – Define Pins – Weakly Connected*.

   The window prompts you to click a pin figure to select a net.

2. To select the net that includes the pins you want to connect externally, click a shape on that net.

   You can select a net by selecting a shape on the canvas before you choose the *Connectivity – Pins – Weakly Connected* command.

   Flight lines appear, connecting only the pins of the net you chose. Dotted flight lines are used to show the internal high-resistance path between the pins. If incomplete nets are being displayed, the other net flight lines disappear. Flight lines are visible in Layout XL and higher tiers.

   Selecting a net does not make a connection change.

   The window prompts you to select two or more pins to be connected.

3. To select the pins you want to connect, click one or more pins in the selected net.

   You can click a pin, `Shift`+click subsequent pins, or select pins by area by clicking and dragging with the right mouse button.

   Clicking on the pins of the selected net makes the connection change.

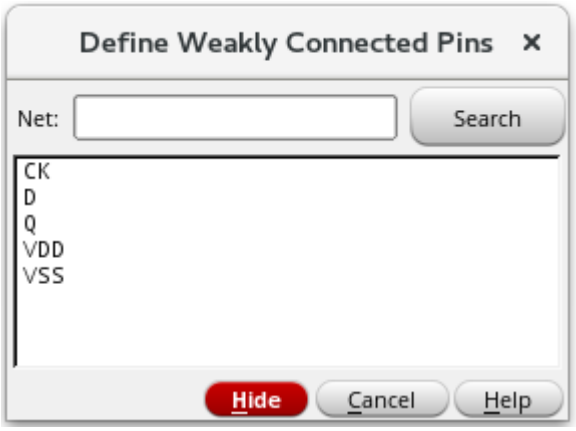
© 2023 Cadence Design Systems, Inc.

- If you select a single pin, the other pins are not deselected, and that pin is connected externally to the current selection of externally connected pins.
- If you select multiple pins, those pins are connected externally to all other externally connected pins on the net.

When you click pins to be externally connected as weak-connects, the net flight line display immediately changes to show the new information.

You can also use the Define Weakly Connected Pins form to select a net.

4. Press `F3` to open the Define Weakly Connected Pins form, while the *Weakly Connected Pins* command is active.



5. Type the name of a net in the *Net* field or click net names in the list box of the form to select them.
6. On the canvas, click the pins you want to define.

   When you add or remove pins to be weakly connected, the net flight line display changes immediately to show the new information.

   To select more than one net at a time to work on, you can:

   - Choose multiple nets in the form.
   - Use drag by area when the command line prompts you to select a net.

   You must define weak pin connectivity within a net. You cannot define weak pin connectivity between nets.
7. When you have finished, press `Esc` to end the command.

   When you define pins as weakly connected, the router routes them within the design.

© 2023 Cadence Design Systems, Inc.

Using the right mouse button, you can switch from the *Weakly Connected* command to the *Must Connect* command, to the *Strongly Connected* command, to the *Pseudo Parallel Connect* command. Must Connect pins must be changed to Strongly Connected before being changed to Weakly Connected.

Related Topics

Define Net/Pin Connections Form

Flight lines

Defining Pseudo Parallel Connected Nets

Pseudo parallel connections are like must connects within an instance and represent cases where instance terminals do not need to be connected because current does not pass between them.

To define a set of instance terminals on a net for pseudo parallel connection:

1. In the layout window, choose *Connectivity – Pins – Pseudo Parallel Connect*.

   The window prompts you to click to select a net.

   You can select a net by selecting a shape in the layout before you choose the *Pseudo Parallel Connect* command.

2. To select the net that includes the instance terminals you want to connect, click a shape on that net.

   Flight lines appear, connecting only the pins of the net you choose. The flight lines are drawn between pins that can be pseudo parallel connected. If incomplete nets are being displayed, the other net flight lines disappear.

   Selecting a net does not make a connection change.

   If you did not select a net or instance terminal before you chose the command, the Layout window prompts you to select an instance terminal, instTerm.

3. To select the instance terminals you want to connect in pseudo parallel connection, click one or more pins in the selected net.

   You can click an instance terminal, `Shift`+click subsequent pins, or select terminals by area by clicking and dragging with the right mouse button.

   Clicking on the instance terminals of the selected net makes the connection change.

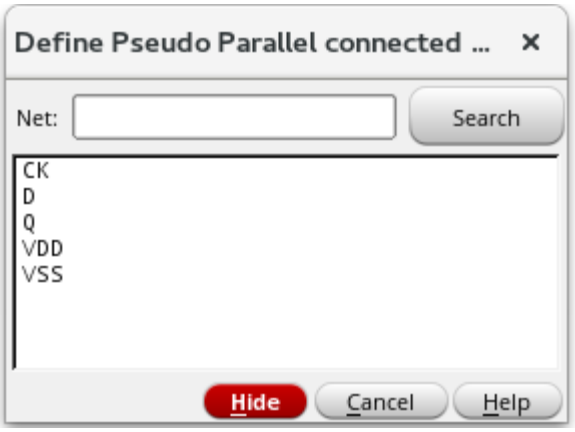
© 2023 Cadence Design Systems, Inc.

When you define the connection of each terminal, the flight line display of the net changes immediately to show the connection.

You can also use the Define Pseudo Parallel Connected Pins options form to select nets.

4. Press `F3` to open the Define Pseudo Parallel Connected Pins form, while the *Pseudo Parallel Connect* command is active.

   The form lists all nets without I/O pins, except le_ex_# nets.



5. Type the name of a net in the *Net* field or click the net names in the list box of the form to select them.

   To select more than one net at a time to work on, you can:

   - Choose multiple nets in the form.
   - Use drag-by-area when the command line prompts you to select a net.

   You must define pseudo parallel connectivity within a net. You cannot define pseudo parallel connectivity between nets.

6. When you are finished, press the `Esc` key to cancel the command.

   When you define pins as pseudo parallel connects, the router does not route them.

Using the right mouse button, you can change from the *Pseudo Parallel Connect* command to the *Must Connect* command, to the *Weakly Connected* command, to the *Strongly Connected* command.

Related Topics

Define Net/Pin Connections Form

© 2023 Cadence Design Systems, Inc.

Flight lines

Pseudoparallel Connections

Checking Connectivity of Pins

To see whether pins on a net are defined for external connection:

1. In the layout window, choose *Connectivity – Pins – Must Connect*.
2. Press `F3` to open the *Connectivity – Must Connected Pins* form.

   The form appears, showing all the nets that include terminals, which are possible connections to devices external to the device.
3. Click any of the nets shown in the form.

   Flight lines for the net appear on the canvas, showing whether the terminals are defined as must-connect, strong-connect, or weak-connect.

Related Topics

Define Net/Pin Connections Form

Flight lines

Propagating Nets

The *Propagate Nets* command maps terminals on an instance or a homogenous mosaic that you select to the nets in a cellview; that is, it defines how the instance or the instances in the homogenous mosaic are logically connected to the cellview. Using this command, you can assign or reassign a top-level net name for each terminal in the selected instances or the instances in the homogenous mosaic. For each such terminal, the command creates a top-level net, if necessary, and creates an instTerm on that net for the instance terminal. You use this command when you do not have a schematic for your layout cellview.

Routers use this logical connectivity information to physically connect the pins of the instance to the rest of the layout.

To promote net information from an instance in your layout:

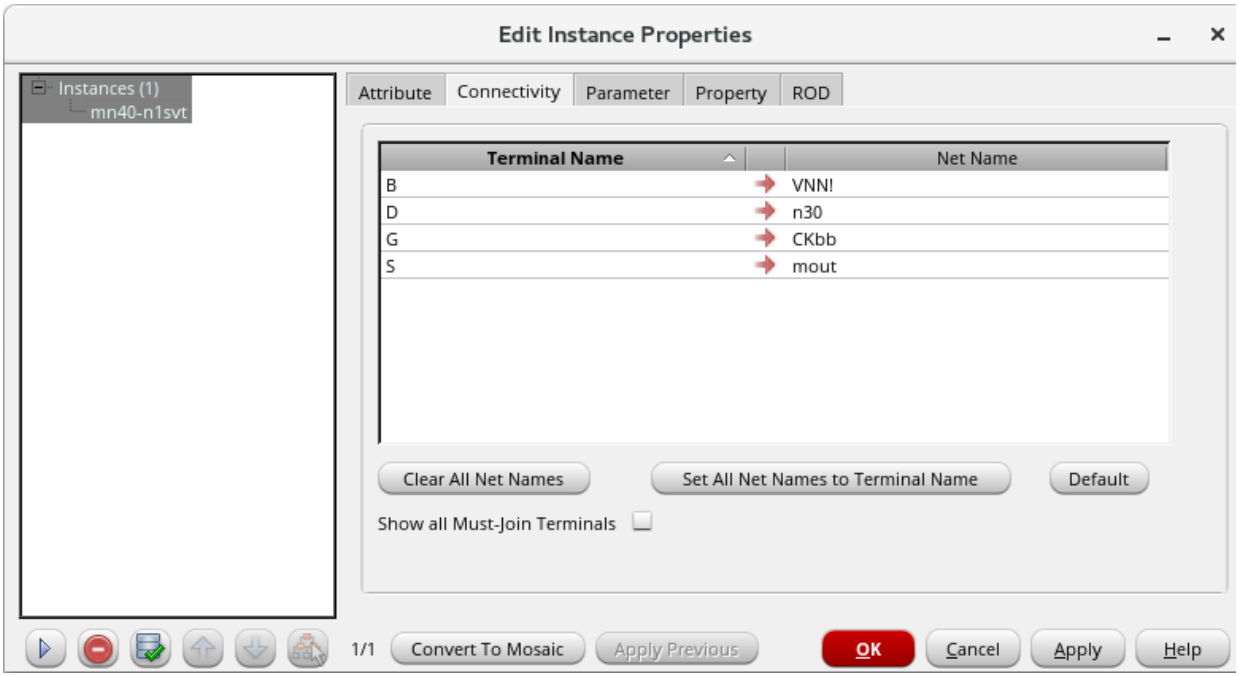
1. In the layout window, select one or more instances.
2. Choose *Connectivity – Nets – Propagate*.

© 2023 Cadence Design Systems, Inc.

A dialog box appears informing that the Propagate Nets form is now obsolete and instead the Connectivity tab of the property editor will be opened for the selected instances.

3. Select the *Do not show again* check box if you want to disable the display of this message in future. Then, click *Close*.

   The Connectivity tab of the Edit Instance Properties form opens, showing the terminals of the instance and the top-level nets they are mapped to, if any. Top level refers to the cellview in which the instance is placed.



4. Hover over a terminal name on the Connectivity tab. The terminal gets highlighted on the canvas.

   Similarly, when you hover over a net name on the tab, the net gets highlighted on the canvas.

5. For each terminal, in *Net Name*, specify the top-level net to which you want to map the terminal.

   If a *Net Name* field is left blank, the corresponding terminal is not mapped to a top-level net. *Net Name* fields can be left blank, but the pins on the corresponding terminals will not be routed.

   - ❑ If you want the net to have the same name as the terminal name, type or select the terminal name or click the → button.

© 2023 Cadence Design Systems, Inc.

- If you want all the top-level nets to have the same name as the terminal name, click *Set All Net Names to Terminal Name*.
- If you want to undo all the changes, click *Defaults*. The net names revert to their current value.
- If you want to clear all the net names, click the *Clear All Net Names* button.

The form keeps track of all the changes you have made. You do not need to click *Apply* after each change before editing other instances.

6. Click *OK* or *Apply* to apply all the changes.

   All the changes you have made are saved to the database. If you want to view all the nets in your design, you can enable the *Nets* option in the Display Options form and redraw the view.

Related Topics

Connectivity Tab

Adding and Deleting Shapes on Nets

The *Add Shape To Net* command lets you add selected shapes to nets that are attached to pins. An instance or mosaic can be added to a net as long as there is at most one terminal on the net. An instance cannot be added to a net if it contains instTerms. A shape cannot be added to more than one net. Terminals and nets are not deleted when the physical pin shape associated with them is deleted.

The *Delete Shape From Net* command lets you delete selected shapes from a net. Terminals and nets are not deleted when the physical pin shape associated with them is deleted.

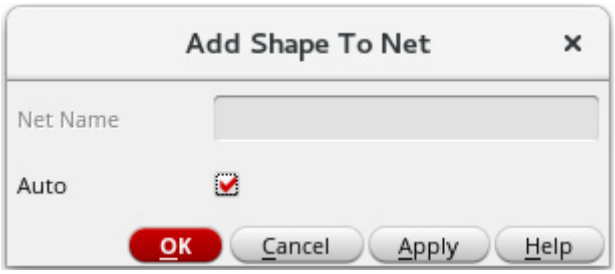
Adding Shapes to a Net

To add a shape to a net:

1. On the layout canvas, select one or more shapes.
2. Choose *Connectivity – Nets – Add Shape*.

© 2023 Cadence Design Systems, Inc.

The Add Shape to Net form appears.



3. Do one of the following:
   - To specify the net based on the selection, select the *Auto* check box. For each shape, the software looks for a pin that overlaps the shape and adds the shapes to the net.
   - To specify the net manually, turn *Auto* off and type the net name.
4. Click *OK* to close the form and add the shapes to the nets.

When there are zero-area dot pins on the edge of the shape to be added, the pin is used only if one of its access directions points toward the center of the shape.

Removing Shapes from a Net

To remove a shape from a net:

1. Select one or more shapes to be removed from a net.
2. Choose *Connectivity – Nets – Remove Shape*.

   The Remove Shape From Net form appears.
3. If the net name shown is not the net you want the shapes to be deleted from, type the correct net name.
4. Click *OK* to close the form and delete the shapes from the specified net.

Related Topics

Add Shape to Net Form

Remove Shape From Net Form

© 2023 Cadence Design Systems, Inc.

Assigning and Unassigning Nets on Shapes

You can assign and unassign nets to shapes, instance pins, and routed objects like vias and pathSegs by using the `leHiAssignNet` and `leHiUnassignNet` functions, respectively. You can assign nets to a shape by tapping an already assigned shape, by selecting a net from the drop-down list, or by typing an existing net name in the Assign Net form. The functions work in both pre-selection and post-selection modes. This functionality works only in Layout XL for instance pins.

- Assigning Nets to Shapes in Pre-Selection Mode
- Assigning Nets to Shapes in Post-Selection Mode
- Unassigning Nets from Shapes in Pre-Selection Mode
- Unassigning Nets from Shapes in Post-Selection Mode

Assigning Nets to Shapes in Pre-Selection Mode

To assign a net to a shape in pre-selection mode:

1. In the layout window, select the shapes to which you want to assign a net.
2. Run the `leHiAssignNet` command.
3. Do one of the following:
   - Tap an already assigned shape to pick the net.

     The net of the assigned shape is assigned to all the selected shapes.
   - Open the Assign Net form by pressing `F3`. From the *Net Name* drop-down list, select a net or type an existing net name, and click *Hide*.

     The net is assigned to all the selected shapes.
4. Press `Esc` to exit the command.

Assigning Nets to Shapes in Post-Selection Mode

To assign a net to a shape in post-selection mode:

1. Run the `leHiAssignNet` command.
2. In the layout window, do one of the following:
   - Tap an already assigned shape to pick the net.

© 2023 Cadence Design Systems, Inc.

- Open the Assign Net form by pressing `F3`. From the *Net Name* drop-down list, select a net or type an existing net name, and click *Hide*.

3. Select the shapes to which you want to assign a net.

   The net is assigned to all the selected shapes.

4. Press `Esc` to exit the command.

You can use the `t` bindkey to assign another net to the shapes.

Unassigning Nets from Shapes in Pre-Selection Mode

To unassign a net from a shape in pre-selection mode:

1. Select the shapes.
2. Run the `leHiUnassignNet` command.

   The net is unassigned from all the selected shapes.

Unassigning Nets from Shapes in Post-Selection Mode

To unassign a net from a shape in post-selection mode:

1. Run the `leHiUnassignNet` command.
2. Select the shapes.

   The net is unassigned from all the selected shapes.

Related Topics

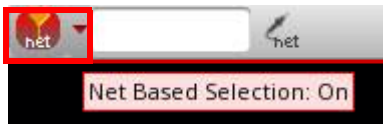
Assign Net Form

Ways to Select Shapes Based on Nets

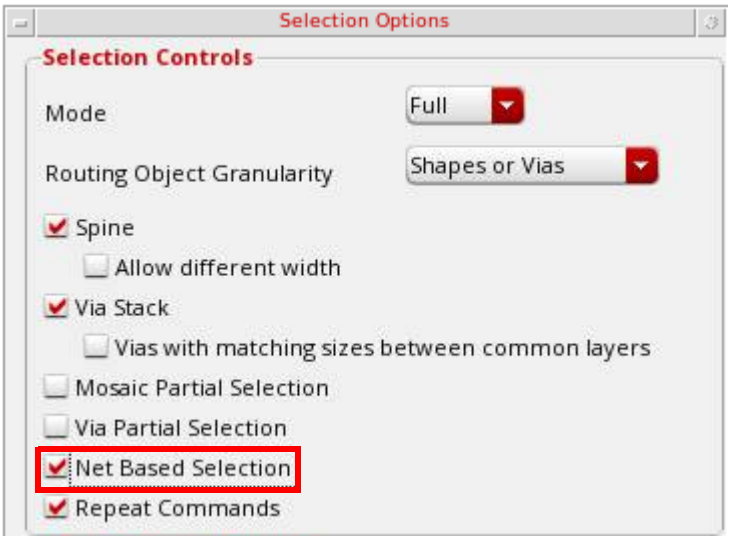
You can use the net-based selection feature to select shapes based on their nets. You can enable this feature in the following ways:

© 2023 Cadence Design Systems, Inc.

- Using the *Net Based Selection* toolbar: Select the *Net Based Selection* toolbar from the *Window – Toolbars* menu. Then, use the Net Based Selection icon on the toolbar.



- Using the *Net Based Selection* option on the Selection Options form.



You can also use the environment variable `netBasedSelectionOn` to enable the net-based selection feature.

When you enable the net-based selection feature, the following are the ways in which you can specify the nets to select figures based on that net:

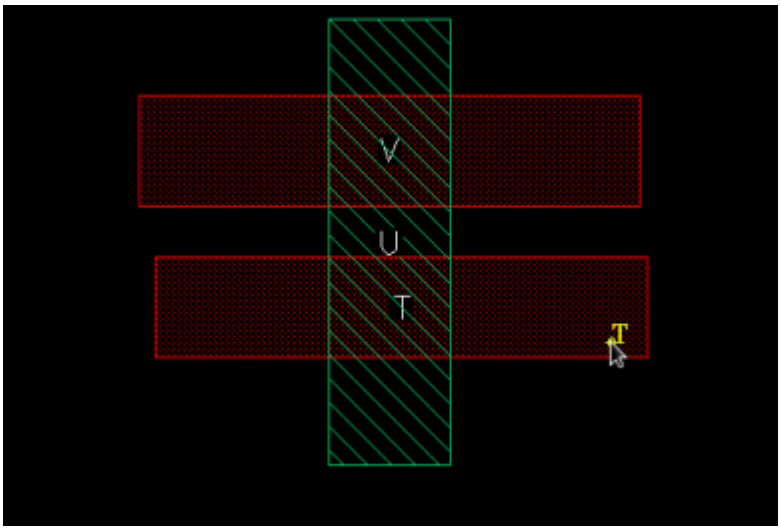
- Specify the nets in the text field on the *Net Based Selection* toolbar.



© 2023 Cadence Design Systems, Inc.

- Select the *Tap Net* icon on the *Net Based Selection* toolbar to populate nets by tapping an assigned shape in the canvas.





  Press `Ctrl` and click to remove the net of the tapped shape from the list of nets. Press `Shift` and click to add the net of the tapped shape to the list of nets.

- Select nets using the *Navigator* assistant.

When you select the *Reverse* option on the *Net Based Selection* toolbar, shapes, vias, and instances with a net other than the net specified in the text field can be selected in the canvas. You can also use the environment variable netBasedSelectionReverse to specify the *Reverse* option.



Related Topics

Navigator Assistant

Selection Options Form

© 2023 Cadence Design Systems, Inc.

Net Tracing Using the Mark Net Command

The *Mark Net* command lets you visually trace the physical connectivity of nets in a layout design without having to use a schematic. The command extracts the metal, via, poly, and diffusion layer information from the technology file and highlights these in the layout as the net passes from one layer to the next through the hierarchy. The trace is not a selectable object: it is the highlighted net. The *Mark Net* command is available both in read-only and edit modes.

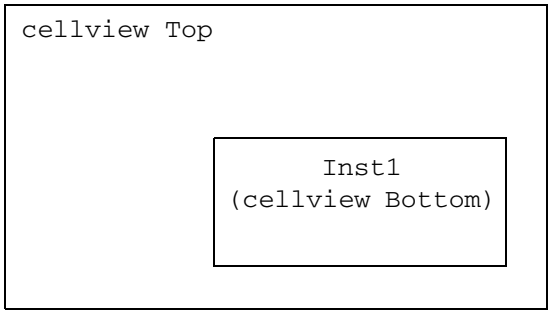
You can choose which and how many levels of hierarchy to traverse for highlight generation. After the hierarchy selection range is set, you cannot select a shape that is outside the start-stop hierarchical range. You can start the *Mark Net* command from any level of the hierarchy, for example from *Edit in Place*, and the command will correctly trace the net throughout the designated hierarchical range. *Mark Net Stop Level*, specified in the MarkNet Options form, can be lower than the *Display Level*, specified in the Display Options form. However, if you set the *Mark Net Start Level* to greater than zero, it is possible to get incomplete net tracing. When the interconnect is interrupted by a shape on a hierarchical depth outside the specified range, the rest of the net is not highlighted even if the subsequent sections of the net are within the specified range.

A traced net remains highlighted as long as the cellview window remains open, unless you use the *Unmark Net* command to remove the highlighting. If you close the cellview window, the highlighting is deleted. However, as long as the cellview window is open, the nets remain highlighted even after the cellview window is redrawn, after any changes are made to the edit context, or after you exit the command. When the *Mark Net* command is re-invoked, it does not remove the existing highlighting. This allows you to incrementally mark nets, mark nets from any level of the hierarchy, and retain nets marked as you move throughout the hierarchy. You can mark nets in multiple cellview windows. You can also view the highlighted nets in the plot of a cellview. You can unmark a net later if required.

The highlighting is created relative to the current top-level cellview. If you highlight nets in separate cellview windows, the *Unmark Net* command removes all highlighting associated with the current top-level cellview. For example, `cellview Top` contains an instance of `cellview Bottom` and both the cellviews are open at the same time and you have used *Edit In Place* into the instance of `cellview Bottom` in `cellview Top`. The order in which you mark nets in the two cellview windows determines if *Unmark Net* removes the highlighting from one or both the cellview windows. If you mark the nets first in `cellview Bottom`, then this is the current top-level cellview. The marked nets will be highlighted in `cellview Top` as well. However, if you further mark nets in `cellview Top` and then *Unmark All Nets* in `cellview Bottom`, then it will unmark only those nets that were

© 2023 Cadence Design Systems, Inc.

marked in `cellview Bottom`. You need to return to `cellview Top` and use *Unmark All Nets* to delete the additional highlighting in that window.



The final highlighted net is created by generating one large polygon representing the shapes on the net, so the highlight shows the outline of all the marked shapes; individual shapes on the net are not highlighted.

If the number of shapes on the net is 10,000 or less, a one-line message appears in the CIW, as in the following example:

```
Number of shapes processed by Mark Net: 5
```

If the number of shapes on the net is greater than 10,000, the output message in the CIW displays each time 10,000 shapes are processed. The traced nets are also displayed incrementally, in sets of 10,000 shapes. When the last shape is processed, the final number of shapes processed is displayed, as in the following example:

```
Number of shapes processed by Mark Net: 10000...
Number of shapes processed by Mark Net: 20000...
Number of shapes processed by Mark Net: 30000...
Number of shapes processed by Mark Net: 30030
```

The *Mark Net* command does not highlight elliptical nets or circular vias.

For the *Mark Net* command to work, your technology file must list the interconnect layers in the `viaDefs` section of the technology file. Via definitions are of two types, `customViaDefs` and `standardViaDefs`.

You can control the thread mode, single, auto or custom, of the Mark Net command using the environment variable, markNetNumThreads.

Related Topics

MarkNet Options Form

Display Options Form

© 2023 Cadence Design Systems, Inc.

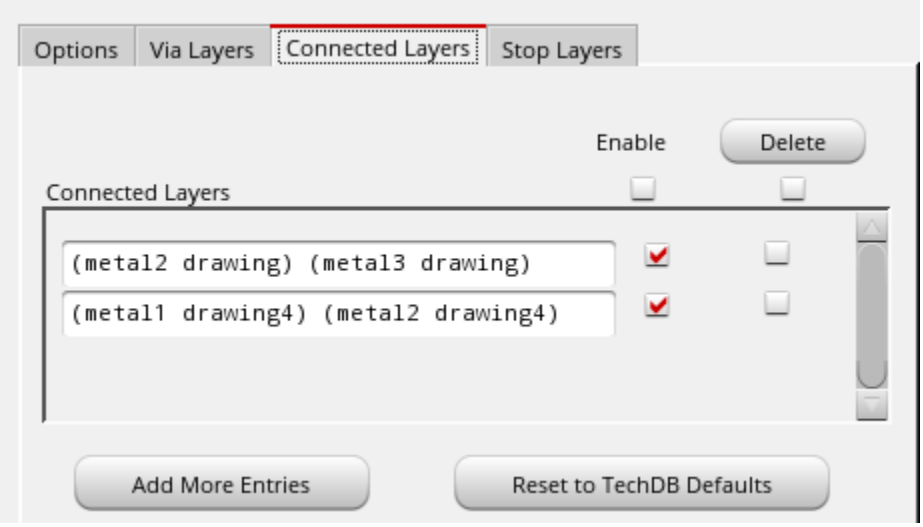
Via Definitions

Via Specifications

About Virtuoso Technology Data

Layers Specification for Tracing a Net

You can specify the via layers and the connected layers through which you want to trace a net. You can either use the via layers defined in the technology file or add your own via layers. To add a via layer, specify it as a (*layer1 viaLayer layer2*) combination on the Via Layers tab of the MarkNet Options form. Both the layers in the via layer combination can be specified as layer-purpose pairs (LPPs). To define connected layers, specify them in the *(layer1 purpose) (layer2 purpose)* format on the Connected Layers tab of the MarkNet Options form.



You can also specify the layers through which the *Mark Net* command must not trace a shape. These resistive layers, referred to as stop layers, help to block tracing through devices such as poly resistor and MIM capacitor. Specify the stop criterion in the *layer stopLayer* format on the Stop Layers tab of the MarkNet Options form. You can specify both *layer* and *stopLayer* as LPPs.

Note: To stop tracing a net at a particular via layer, Cadence recommends that you remove or disable that via layer entry on the Via Layers tab, instead of defining it as a stop layer on the Stop Layers tab. Having the entry only on one of the tabs prevents ambiguity and undesirable results.

© 2023 Cadence Design Systems, Inc.

In a new Virtuoso session, if the *Customized Via Layers*, *Customized Connected Layers*, and *Enable Stop Layers* check boxes are selected, the settings for the via, connected, and stop layers are read from the `markNetOptions` file stored at the following location in the technology library directory:

technologyLibraryName`/.cadence/dfII/markNet/markNetOptions`

After the *Mark Net* command completes its run or when you load new settings from a previously saved file, the via, connected, and stop layer settings are automatically stored in the `markNetOptions` file stored at the following location in your current working directory:

currentWorkingDirectory`/.cadence/dfII/markNet/`*technologyLibraryName*`/markNetOptions`

Subsequently, layer settings read from the `markNetOptions` file in your current working directory override the settings read from the file stored in the technology library directory.

If you want a file containing your customized layer settings to be read each time you start a new Virtuoso session, use the markNetOptionsFile environment variable to specify the path or name of this file. The layer settings stored in the file specified using the `markNetOptionsFile` environment variable override the settings stored in the other two `markNetOptions` files.

The format of the `markNetOptions` file is as follows:

```
connectedLayers(
; (layer1 purpose) (layer2 purpose) ... enabled
)
stopLayers(
; (layer purpose) (stopLayer purpose) enabled
)
```


© 2023 Cadence Design Systems, Inc.

The figure below shows a sample `markNetOptions` file. Notice that there are separate sections in the file for via, connected, and stop layers. These sections can be present in any order. You can edit the `markNetOptions` file manually.

```
;Do not remove or change these header lines.
; markNetOptions file version 1.0
viaLayers(
Metal1 Via1 Metal2 t
Metal2 Via2 Metal3 t
Metal3 Via3 Metal4 t
Metal4 Via4 Metal5 t
Metal5 Via5 Metal6 t
)
connectedLayers(
("Metal1" "net") ("Metal2" "net") nil
("Metal1" "net") ("Metal2" "drawing") nil
)
stopLayers(
("Metal1" "net") ("Metal2" "net") nil
("Metal1" "net") ("Metal2" "drawing") nil
("Metal1" "drawing") ("Metal2" "net") nil
Metal1 Metal2 nil
("Metal2" "net") ("Metal1" "net") nil
("Metal1" "drawing") ("Metal1" "pin") nil
("Metal1" "drawing") ("Metal2" "pin") nil
)
```

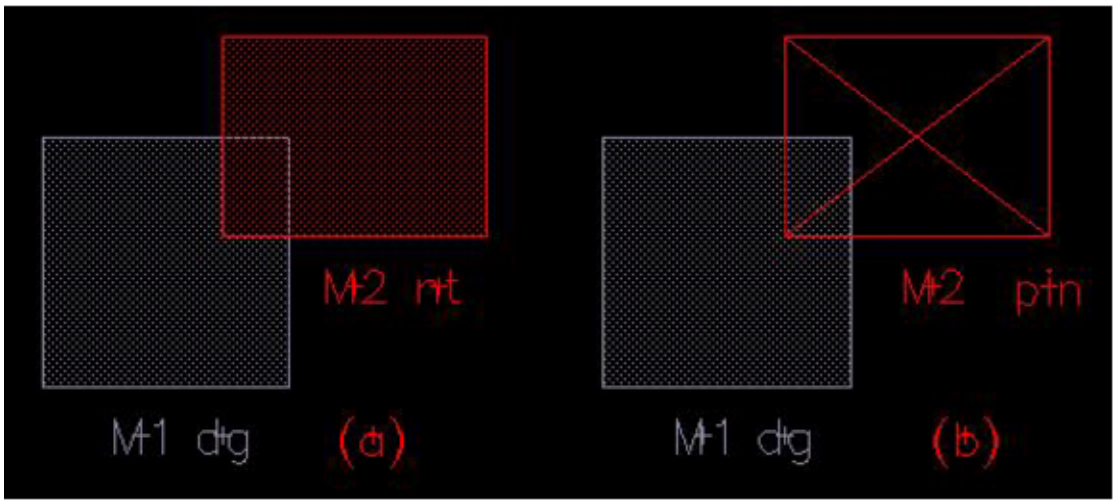
If the `markNetOptionsFile` environment variable has not been specified, the file specified in the technology file is preferred over the file specified at other locations such as `$CWD`.

You can also manually save the settings for any or all via, connected, and stop layers to a file and location of your choice by using the *Save* feature in the MarkNet Options form. By using the *Load* feature, you can later load the settings from the saved file.

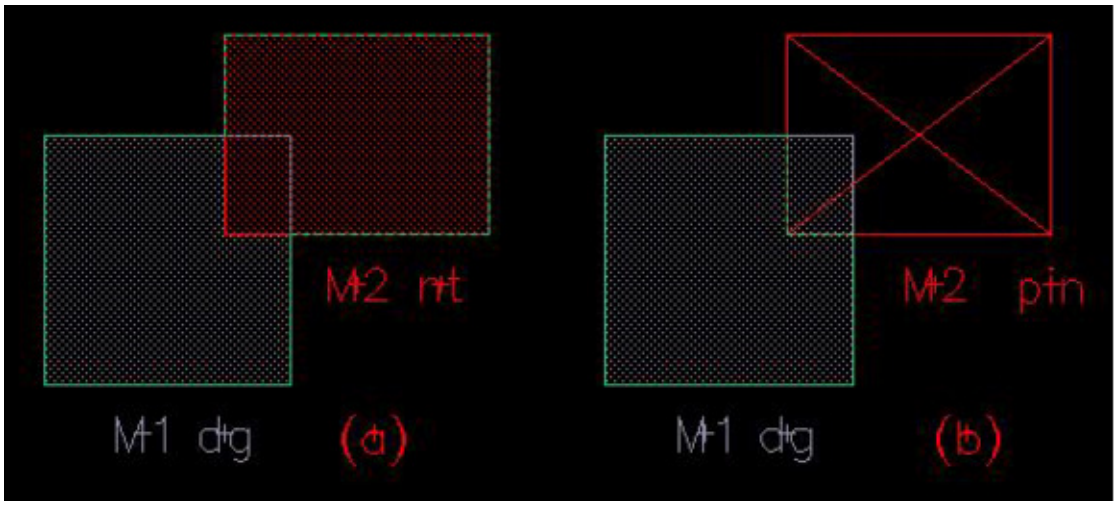
In the figure below, two sets of shapes are shown, labeled (a) and (b). In (a), `M1 dg` overlaps `M2 net`, and in (b), and `M1 dg` overlaps `M2 pin`. `M1 dg` and `M2 net` are defined as connected

© 2023 Cadence Design Systems, Inc.

layers, and `M2 pin` is defined as the stop layer for `M1 dg`. The *Mark Net* command will stop tracing the shape `M1 dg` when there is an overlapping `M2 pin` shape.



When you click `M1 dg` in (a), both `M1 dg` and `M2 net` are traced because they are defined as connected layers. Only `M2` purpose `pin` is defined as the stop layer for `M1 dg`, so tracing continues when the overlapping shape is `M2` purpose `net`. When you click `M1 dg` in (b), the tracing stops at the point the two shapes overlap because `M2 pin` is defined as the stop layer for `M1 dg`. The resultant tracing is shown in the figure below.



Related Topics

Via Layers Tab

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

Connected Layers Tab

MarkNet Options Form

Stop Layers Tab

MarkNet Options Form

Tracing Nets

To trace a net, you need to specify the tracing options, via layers, connected layers, and stop layers in the MarkNet Options form. You can then highlight the required nets on the canvas.

- Specifying Tracing Options
- Specifying Via Layers
- Specifying Connected Layers
- Specifying Stop Layers
- Highlighting the Required Nets

Specifying Tracing Options

To specify the options for tracing a net:

1. In the layout window, choose *Connectivity – Nets – Mark*.

© 2023 Cadence Design Systems, Inc.

2. Press `F3` to open the MarkNet Options form.



3. Specify the settings on the *Options* tab of the form.

   a. Set *Hierarchy Range*.

   If you choose *User defined* from the drop-down list, the *Start* and *Stop* fields become editable. Specify the start and stop levels for starting and stopping net tracing.

   b. In *Region*, specify the area to trace.

© 2023 Cadence Design Systems, Inc.

By default, the entire cellview is traced. You can also specify a rectangular or polygonal area.

c. Ensure that the *Auto Highlight Color* check box is selected if you want the color of highlighted nets to automatically cycle through Cadence-defined layer-purpose pairs (LPPs).

Deselect the *Auto Cycle* check box if you want to manually specify the color for highlighted nets. This activates the layer-purpose pair list. Select the required LPP from the list.

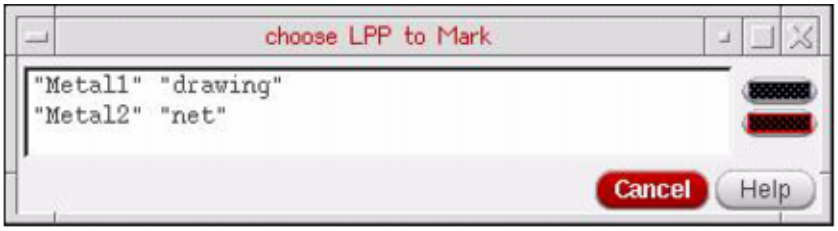
d. Select the *Highlight Incrementally* check box if you want to view a progressive highlighting of the connectivity.

Instead of displaying the connectivity incrementally, you can use `Ctrl+C` to stop the *Mark Net* command from further processing shapes and to display the marked shapes processed until then.

e. Select the *Highlight Individual Shapes* check box to display a separate highlight for each shape.

f. Select the *Select From Overlaps* check box.

If this check box is selected, the *Choose LPP to Mark* form appears when you click at a point where shapes overlap. You can select an LPP in this form to specify the shape from which to begin tracing.



g. Ensure that the *Customized Via Layers*, *Customized Connected Layers*, and *Enable Stop Layers* check boxes are selected to enable the settings specified on the *Via Layers*, *Connected Layers*, and *Stop Layers* tabs to be used for tracing nets. If any of these check boxes are not selected, the corresponding tab also gets disabled.

h. Select the *Retain Via Info for Save MarkNets* check box before running the *Mark Net* command if you later want to save via shapes in a cellview by using the *Connectivity – Nets – Save All MarkNets* command. By default, no via shapes are saved in the cellview used for storing the traced shapes.

© 2023 Cadence Design Systems, Inc.

i. In the *Save/Load custom settings for Via*, *Connected and Stop Layers* section, specify the file name to which you want to save settings or from which you want to load the settings.

To save the settings to a file, click *Browse* and specify the location and name of the file. By default, all the settings specified on the *Via Layers*, *Connected Layers*, and *Stop Layers* tabs are saved. If you do not want to save the settings for a layer type, deselect the corresponding check box. Then, click *Save To*.

To load the settings from a file, click *Browse* and specify the location and name of the file. By default, the *Via Layers*, *Connected Layers*, and *Stop Layers* tabs are all populated with the settings stored in the file. If you do not want to load the settings for a layer type, deselect the corresponding check box. Then, click *Load From*.

Specifying Via Layers

To specify via layers for tracing a net:

© 2023 Cadence Design Systems, Inc.

1. Click the *Via Layers* tab on the MarkNet Options form.



2. In the *layer1*, *viaLayer*, and *layer2* fields, specify the LPPs — the layer name in the *layer* column and the purpose name in the *purpose* column.

   If you leave the *purpose* field blank, the *Mark Net* command interprets it as all purposes of the specified layer.

3. Click *Add*.

© 2023 Cadence Design Systems, Inc.

The new via layer in the `(layer1) (viaLayer) (layer2)` format is displayed in the *Via Layers* list box. If any of the layers is invalid or the specified via layer combination already exists, clicking *Add* does not add the via layer to the list box. A warning message is displayed in the CIW.

By default, the *Enable* check box against the newly added via layer is selected and the via layer is used for tracing. You can deselect the *Enable* check box for any via layer that you do not want to be used for tracing. To select or deselect all the via layers in one go, select or deselect the *Enable* check box at the top of the *Enable* column.

4. Select the check box against a via layer in the *Delete* column and click *Delete* if you want to delete the via layer. If you want to delete all the via layers, select the *Delete* check box at the top of the *Delete* column and click *Delete*.
5. Select the *Reset to TechDB Defaults* check box if you want to use the default via layers defined in the technology file. All via layers defined in the technology file are displayed in the list box, and the *Enable* check box for each is selected by default.

   Any via layers added by you are deleted from the list box. If you want, you can add your own via layers after populating the ones from the technology file.

Specifying Connected Layers

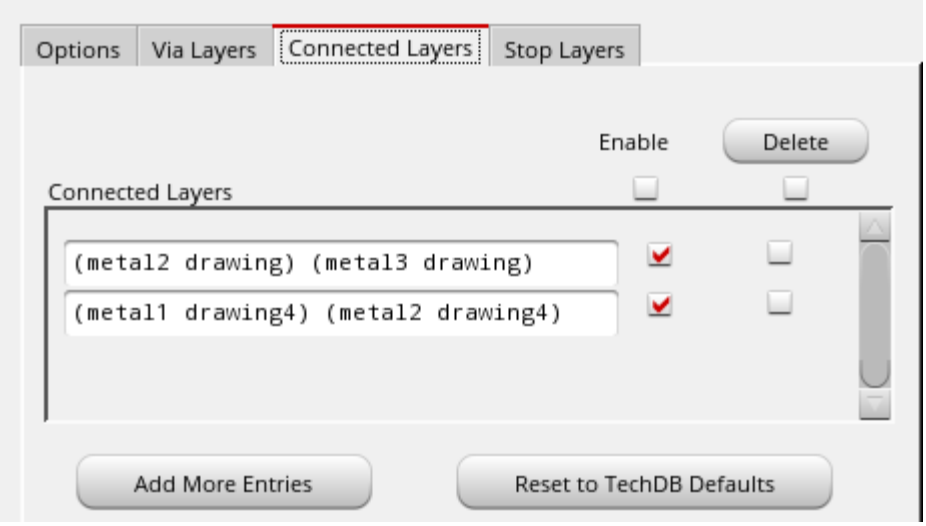
To specify connected layers for tracing a net:

1. Click the *Connected Layers* tab on the MarkNet Options form.
2. Click *Add More Entries*.

   A text field appears in the *Connected Layers* list box.
3. Specify the connected layer combination in the text field.

© 2023 Cadence Design Systems, Inc.

In the example shown in the figure below, the connected layers added imply that `metal2-drawing` is connected to `metal3-drawing` and `metal1-drawing4` is connected to `metal2-drawing4`.



You can specify more than two LPPs as connected layers. If you do not specify the purpose, all purposes of that metal layer are connected to the other specified LPP.

4. Click *Add More Entries* to add more entries for connected layers.

   By default, the *Enable* check box against the newly added connected layer is selected and the connected layer is used for tracing. You can deselect the *Enable* check box for any connected layer that you do not want to be used for tracing. To select or deselect all the connected layers in one go, select or deselect the *Enable* check box at the top of the *Enable* column.

5. Select the check box against a connected layer in the *Delete* column and click *Delete* if you want to delete the connected layer. Select the *Delete* check box at the top of the *Delete* column and click *Delete* if you want to delete all the connected layers.

6. Select the *Reset to TechDB Defaults* check box if you want to use the default connected layers defined in the technology file. All connected layers defined in the technology file are displayed in the list box, and the *Enable* check box for each is selected by default.

   Any connected layers added by you are deleted from the list box. If you want, you can add your own connected layers after populating the ones from the technology file.

Specifying Stop Layers

To specify stop layers while tracing a net:

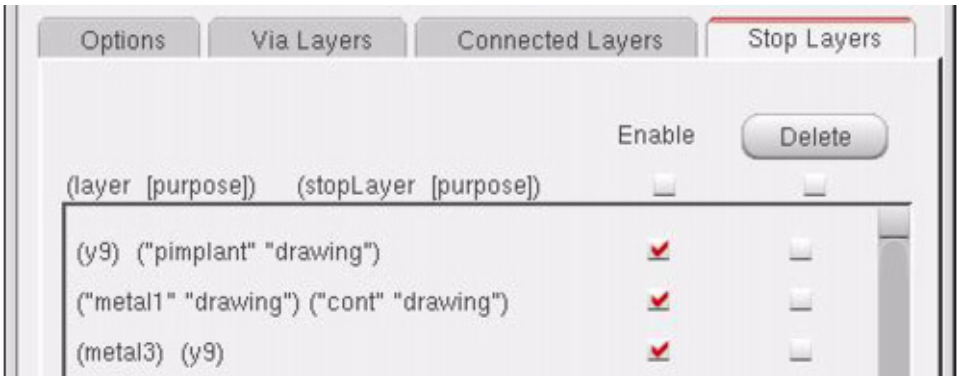
© 2023 Cadence Design Systems, Inc.

1. Click the *Stop Layers* tab on the MarkNet Options form.
2. In the *layer* and *stopLayer* fields, specify the LPPs — the layer name in the *layer* column and the purpose name in the *purpose* column.

   If you leave the *purpose* field blank, the *Mark Net* command interprets it as all purposes of the specified layer.
3. Click *Add*.

   The new stop layer in the `(layer) (stopLayer)` format is displayed in the list box.

   In the example shown in the figure below, the stop layers added imply that `pimplant-drawing` is a stop layer for all purposes on `y9`, `cont-drawing` is a stop layer for `metal1-drawing`, and all purposes on `y9` are stop layers for all purposes on `metal3`.



   By default, the *Enable* check box against the newly added stop layer is selected and the stop layer is used while tracing. You can deselect the *Enable* check box for any stop layer that you do not want to be used while tracing. To select or deselect all the stop layers in one go, select or deselect the *Enable* check box at the top of the *Enable* column.

   Select the check box against a stop layer in the *Delete* column and click *Delete* if you want to delete the stop layer. Select the *Delete* check box at the top of the *Delete* column and click *Delete* if you want to delete all the stop layers.

Highlighting the Required Nets

To highlight the required nets on the canvas, after specifying the settings in the MarkNet Options form:

1. In the design window, click the shape to highlight the required net.

   If you click to select an already highlighted net, the following warning message appears in the CIW:

   ```
   *WARNING* Selected object is on an existing marked net.
   ```


© 2023 Cadence Design Systems, Inc.

The net is highlighted on the specified levels of the hierarchy.

2. Click a point on the second net that you want to highlight. The second net is highlighted on the specified levels of the hierarchy.

Related Topics

MarkNet Options Form

Saving Marked Nets

To save a traced net:

1. In the layout window, trace a net.
2. Select *Connectivity – Nets – Save All MarkNets*.

   This opens the Save All Mark Nets form.
3. Specify the library, cell, and view names where you want to store the highlighted nets. Alternatively, click *Browse* to select the library, cell, and view names.

   If the specified cellview already exists, a warning message will be issued to confirm whether or not you want to overwrite the specified cellview.

If you delete some shapes or objects from a highlighted net before saving it, the deleted shapes or objects are not saved in the specified cellview and a warning message is issued.

Related Topics

Tracing Nets

Save All Mark Nets Form

Unmarking Nets

To unmark a traced net:

1. In the layout window, choose *Connectivity – Nets – Unmark* to delete the highlighting of one or all of the marked nets.

   The Unmark Net form appears.
2. To unmark one of the nets, click that net.

© 2023 Cadence Design Systems, Inc.

3. To unmark all the nets, click *Unmark All*.

To unmark all of the marked nets:

➡ Choose *Connectivity – Nets – Unmark All* to delete the highlighting of all the marked nets.

Related Topics

Unmark Net Form

Probing Nets, Instances, Pins, and Shapes

Probing allows you to find and highlight certain types of objects from among all the objects present in a design. It lets you highlight the logical connectivity between objects. These objects include nets, terminals, paths, and instances. Probing also lets you select objects to highlight their corresponding elements in a different representation of the same design (schematic and layout windows).

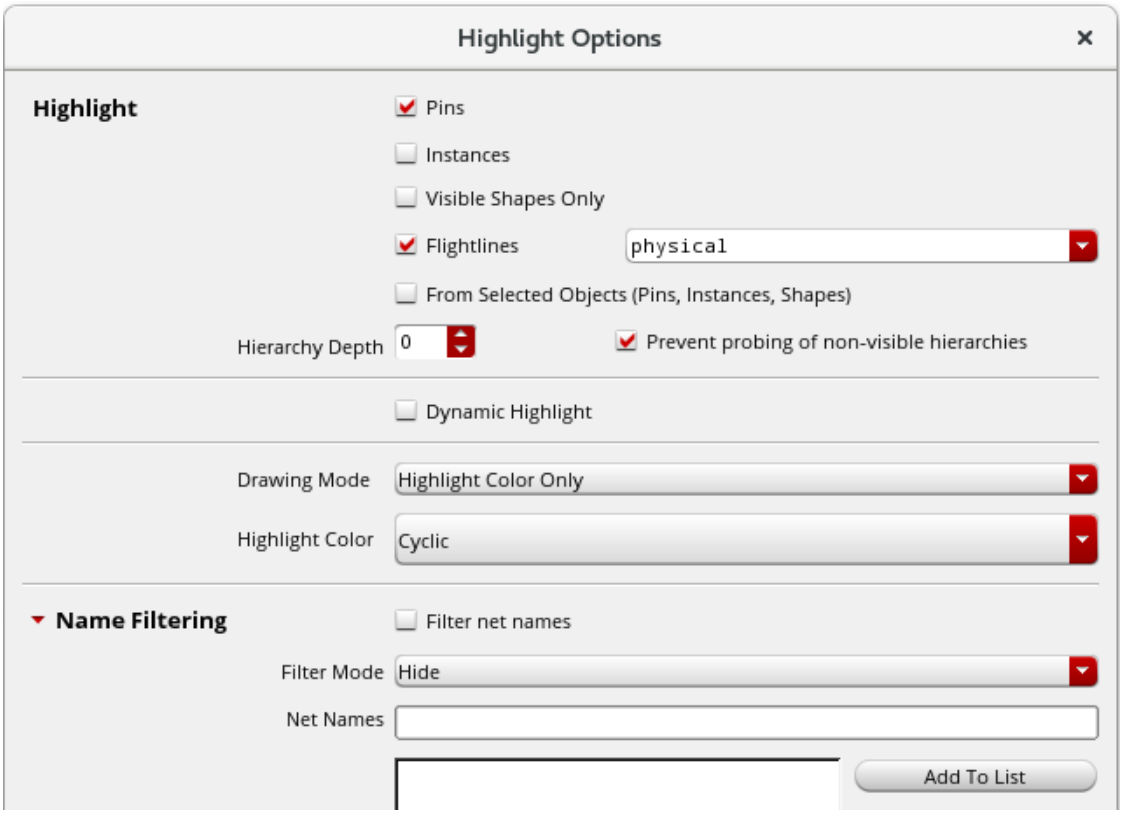
You can use the environment variables, trueColorProbe and probeHighlight to specify the true color highlight. These variables must be set before opening the Virtuoso session.

To select the objects to be highlighted when you probe or select a net, instance, pin, or shape:

1. In the layout window, choose *Options – Highlight*.

© 2023 Cadence Design Systems, Inc.

This displays the Highlight Options form.



2. In the *Highlight* section, set the following:

   a. Select the *Pins* and *Instances* check boxes if you want to highlight pins and instances on the probed/selected nets.

   b. Select *Flightlines* if you want all shapes and opens on the probed net to be highlighted.

   c. Select the *Visible Shapes Only* check box if you want to highlight only the shapes and pins on the visible layers.

   d. Set the *Hierarchy Depth* to specify the depth up to which you want to display net probing.

   e. Select *Dynamic Highlight* to highlight any object over which you hover the pointer.

   f. Select the drawing layer to be used for highlighting the nets.

   g. From *Highlight Color*, select the *Cyclic* option to use the next color for highlighting subsequent objects.

© 2023 Cadence Design Systems, Inc.

3. In the *Name Filtering* section, selectively display or hide the probes only for the specified list of nets.

   a. Use the *Filter Mode* to specify whether you want to show or hide the probes for specified nets.

   a. In *Net Name*, specify one or more net names, separated by space, for which you want to show or hide probes.

   b. Click *Add to List* to add the net names to the list box.

4. Click *Apply*.

After configuring the options in the *Highlight Options* form, you can probe nets, shapes, pins, and instances using either pre- or post-selection mode.

Related Topics

Highlight Options Form

Probe Functions

Probing Objects Using Pre-Selection Mode

Probing Objects Using Post-Selection Mode

Graphics Editor Functions

Probing Objects Using Pre-Selection Mode

To probe objects using pre-selection mode:

1. In the layout window, choose *Options – Highlight*.

   This displays the Highlight Options form.

2. Select the *Flightlines* check box.

3. Select the object that you want to probe. For example, select *Pins* to highlight pins on the probed/selected nets.

4. In the cellview, select one, a few, or all objects irrespective of their connectivity.

5. Enter the following SKILL function in the CIW:

```
geEnterAddNetProbe()
```

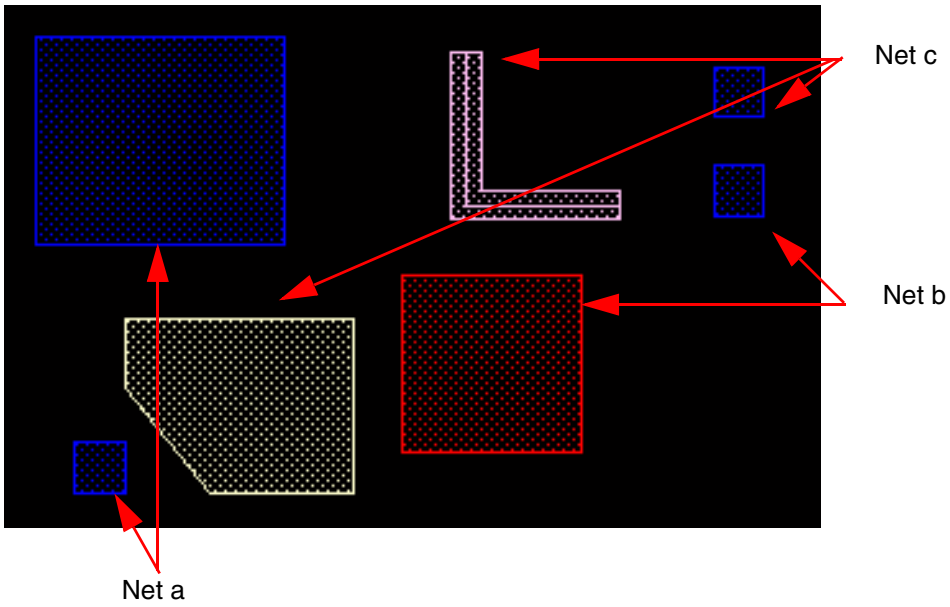

© 2023 Cadence Design Systems, Inc.

Objects with the same connectivity are probed with the same color.

6. Repeat steps 3 and 4 to probe more objects, if required.

   Use the command `geDeleteAllProbe` in the CIW to remove the probes.

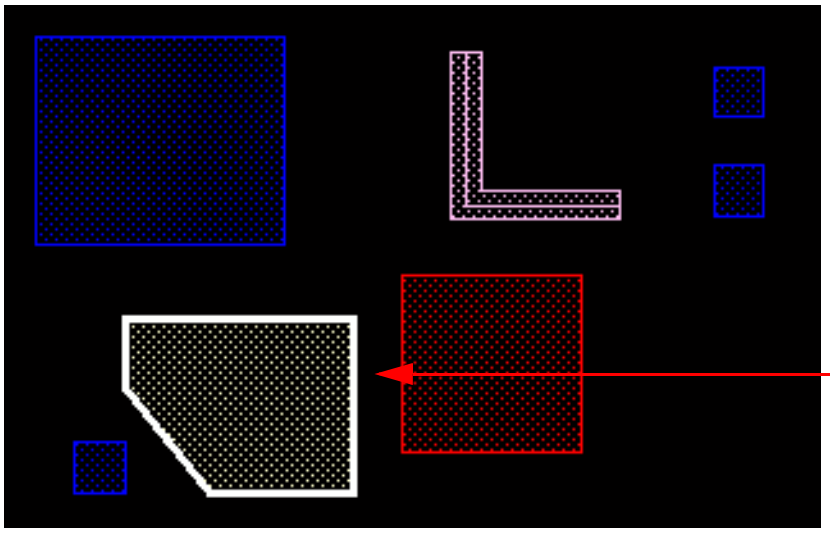
In the following design, different objects have different connectivities, as indicated:

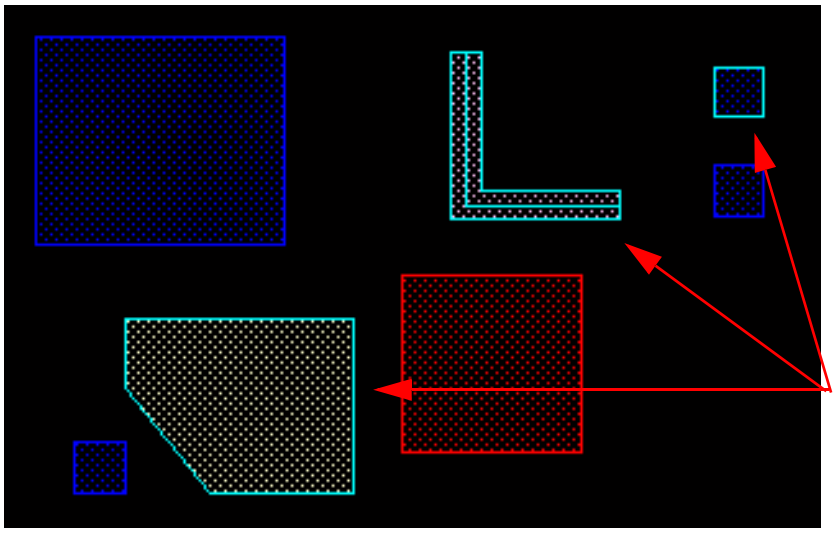


The following example illustrates the scenarios when the probe command is run after either one, two, or all of the objects are edited.

© 2023 Cadence Design Systems, Inc.

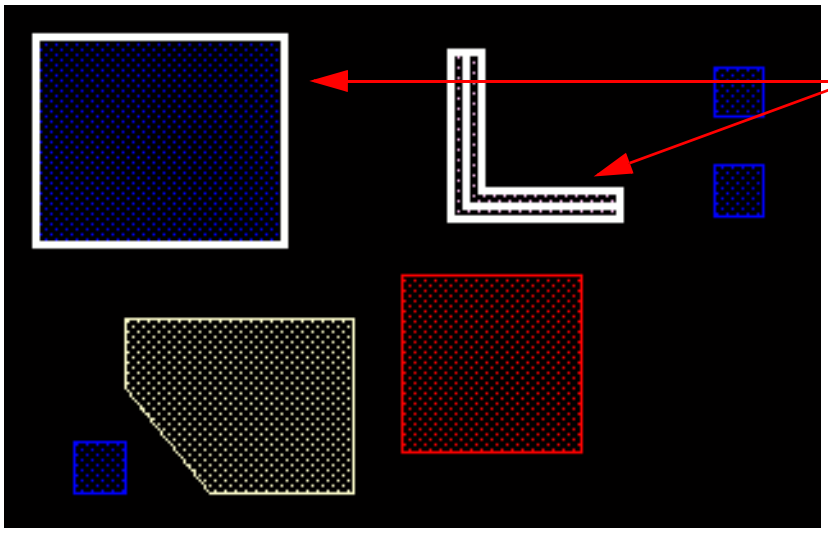
- When a single object is edited.

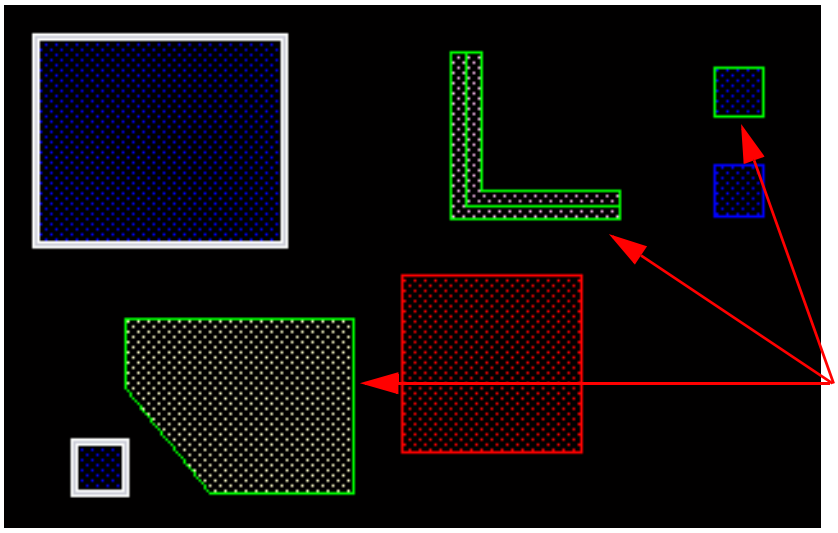




© 2023 Cadence Design Systems, Inc.

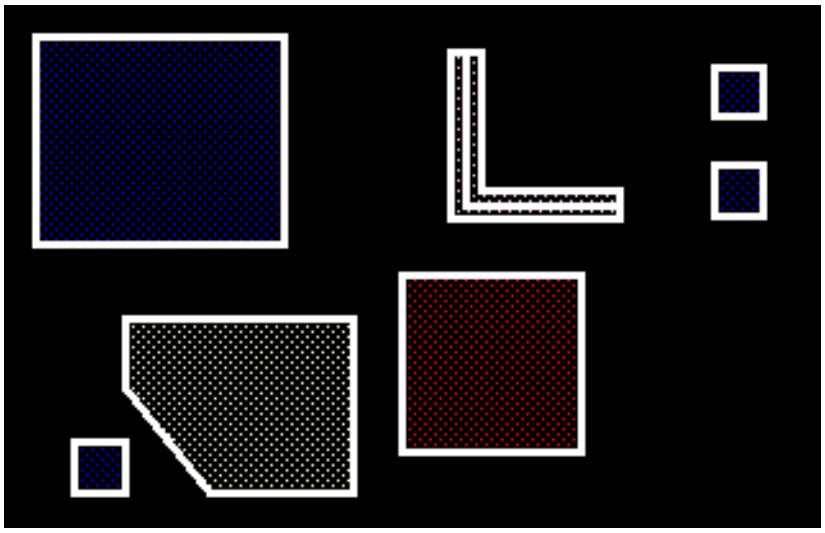
- When multiple objects are edited.



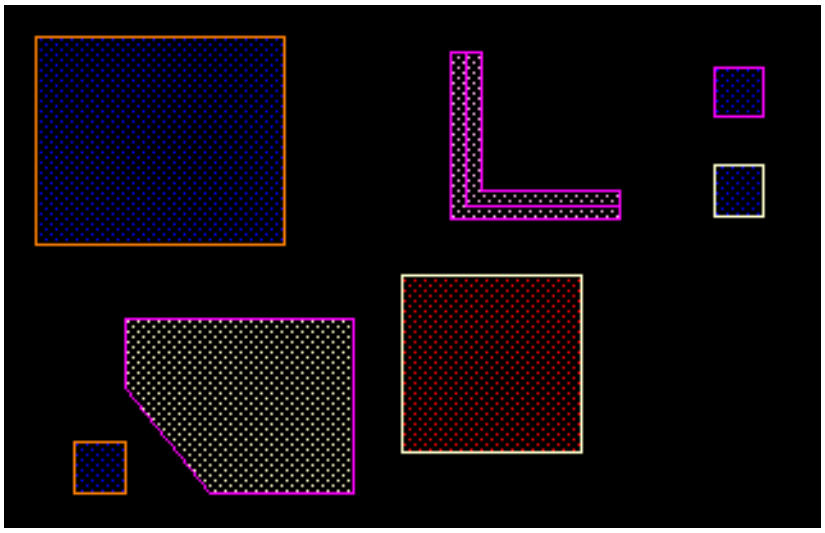


© 2023 Cadence Design Systems, Inc.

- When all the objects are edited.



All objects edited



Objects on the same net are probed using the same color.

Related Topics

Highlight Options Form

geEnterAddNetProbe

Probing Objects Using Post-Selection Mode

To probe objects using post-selection mode:

© 2023 Cadence Design Systems, Inc.

1. In the layout window, choose *Options – Highlight*.

   This displays the Highlight Options form.

2. Select the *Flightlines* check box.

3. Select the object that you want to probe. For example, select *Pins* to highlight pins on the probed/selected nets.

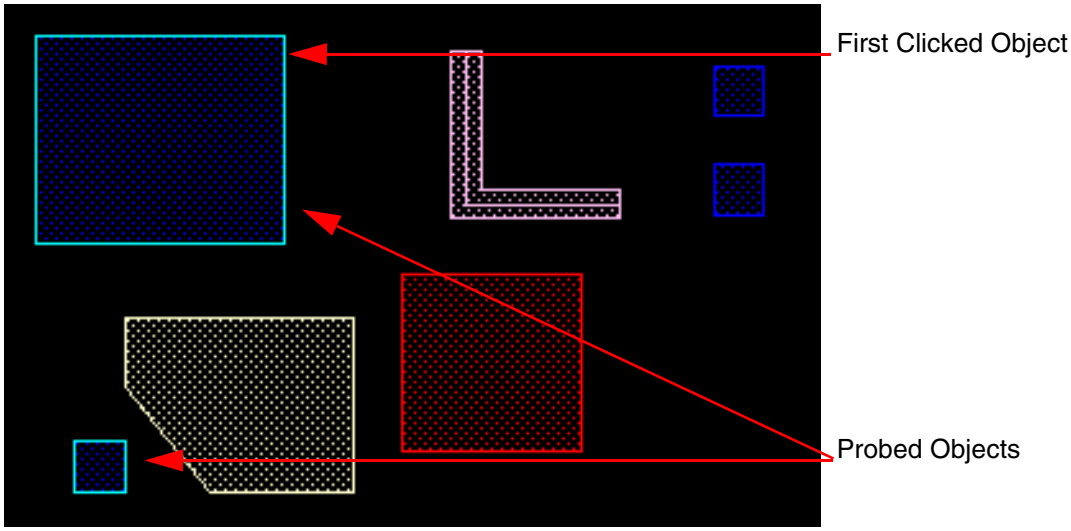
4. Enter the following SKILL function in the CIW:

   ```
   geEnterAddNetProbe()
   ```

5. In the cellview, click a single object.

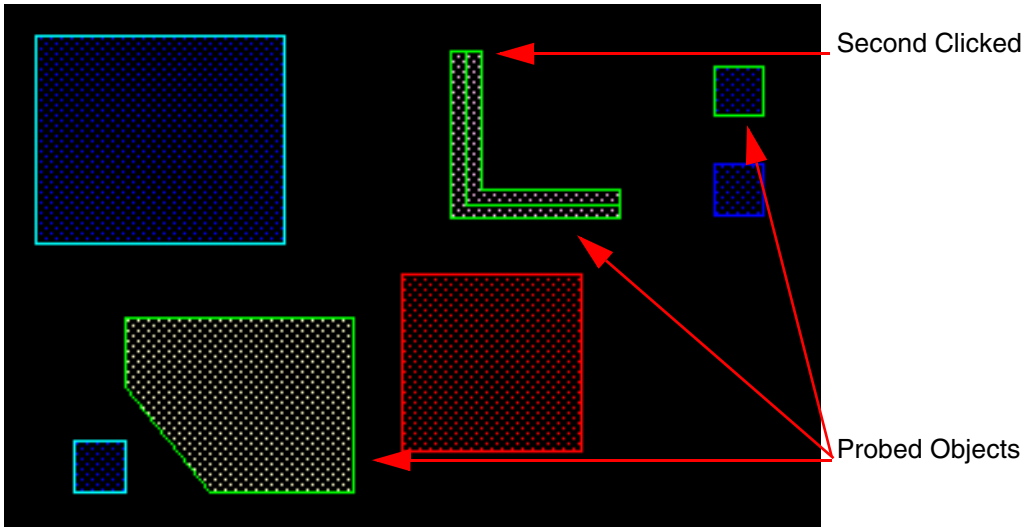
   Other objects with the same connectivity as the selected object are probed.

   In the post-selection mode, objects are probed only when you single-click an object. Objects are not probed if you select multiple objects or use the area selection method.



© 2023 Cadence Design Systems, Inc.

6. You can continue to select other objects, one at a time, with a single click. Objects with the same connectivity are probed with the same color. Earlier probed objects are retained.



7. Press `Esc` to end the probe command.

Related Topics

Highlight Options Form

geEnterAddNetProbe

© 2023 Cadence Design Systems, Inc.

11

Microwave Commands

Microwave layouts consist of placed component instances such as resistors, capacitors, inductors, and transistors, and the interconnects drawn to connect the components.

You can connect components in a microwave layout with regular layout geometries such as polygons, rectangles, and paths, as well as microwave-specific geometries: transmission lines, bends, and tapers.

You use the microwave commands to create transmission lines, bends, and tapers to connect your components. Microwave commands appear on the *Create* menu of layout editor.

You can edit microwave objects with the layout editor commands similar to the way you edit paths. You use these commands to change the angle of paths, stretch sections of paths, and reshape a path.

Related Topics

Creating Transmission Lines

Creating Bends

Creating Tapers

Transmission Line Bend Styles

Object Reshaping

Stretching Paths

Reshaping a Path

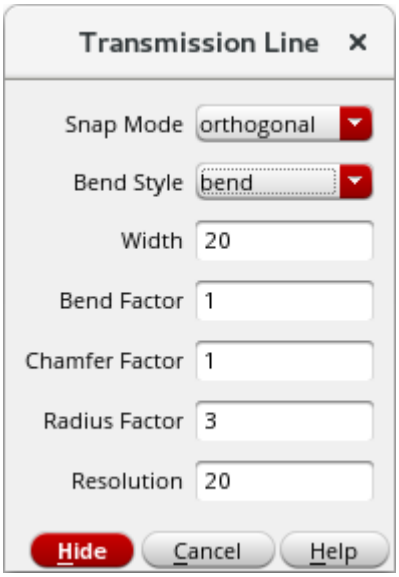
© 2023 Cadence Design Systems, Inc.

Creating Transmission Lines

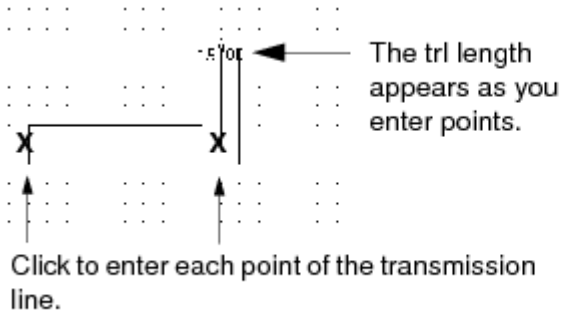
You can use the *Trl* command to create a transmission line (trl). A transmission line shows connections in microwave designs. You create a transmission line the same way you create paths.

To create a transmission line:

1. In the layout window, choose *Create – Microwave – Trl*.
2. Press `F3` to open the Transmission Line form.

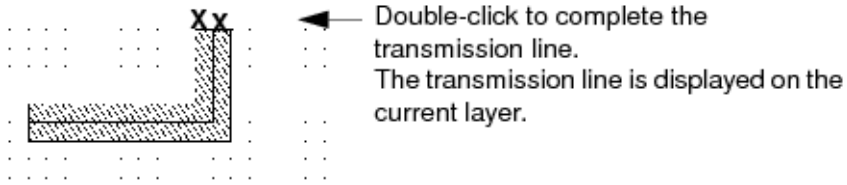


3. Set *Bend Style* to *bend*.
4. Set *Bend Factor* to `1`.
5. Set the `Width` to `2`.
6. Click on the canvas to create each point of the transmission line.



© 2023 Cadence Design Systems, Inc.

7. To complete the transmission line, double-click the last point.



Related Topics

Transmission Line Form

Creating a Path

Microwave Commands

Creating Bends

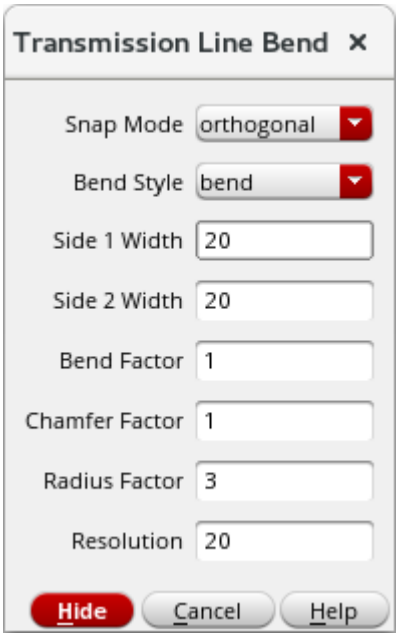
You can use the *Bend* command to create a two-segment transmission line with two different width settings. Bends connect the objects of different sizes, using a right-angle segment instead of a straight line. You create bends similar to the way you create paths.

To create a bend:

1. In the layout window, choose *Create – Microwave – Bend*.

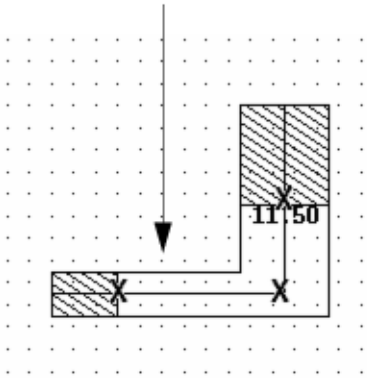
© 2023 Cadence Design Systems, Inc.

2. Press `F3` to open the Transmission Bend form.



3. Type the beginning and ending widths of the bend in the *Side 1 Width* and *Side 2 Width* fields.
4. Click on the canvas to enter each point of the bend.

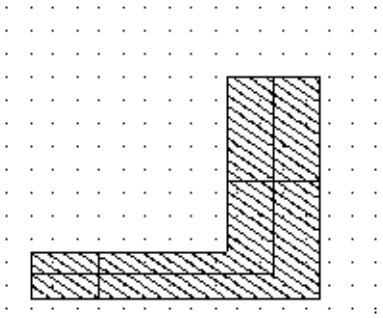
The length of the bend appears as you enter points.



Click on the first, second, and third points of the bend.

© 2023 Cadence Design Systems, Inc.

The following is a completed bend.



Related Topics

Transmission Line Bend Form

Creating a Path

Microwave Commands

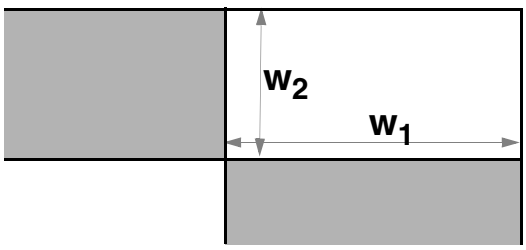
Transmission Line Bend Styles

Transmission line bend styles can be of three types: standard bend, chamfer, and radial.

- Standard Bends
- Chamfer Bends
- Radial Bends

Standard Bends

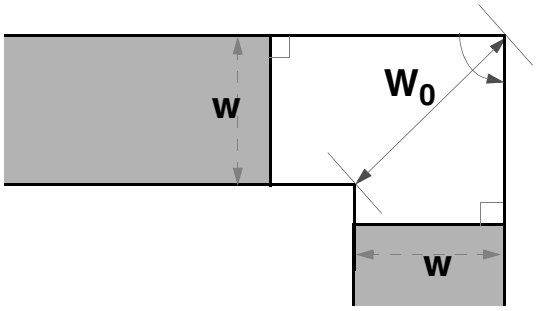
The following illustration shows a standard bend. All angles in a standard bend are 90 degrees.



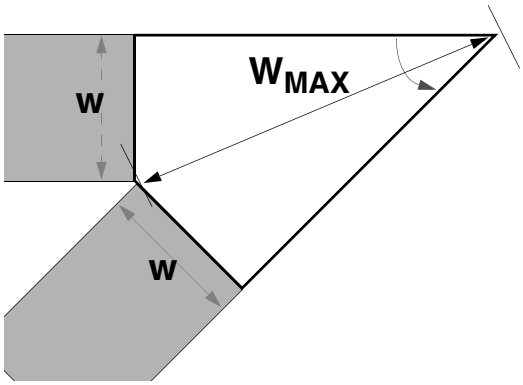
© 2023 Cadence Design Systems, Inc.

Bend Factor is the allowed ratio of the width (W_{MAX}) of the bend for a given angle to the width (W_0) of a 90-degree bend. If the width (W) of a bend is greater than W_{MAX}, the bend is chamfered so that the width is decreased to W_{MAX}. The effect of *Bend Factor* is shown in the following illustrations.

90-degree bend with W_0 width:

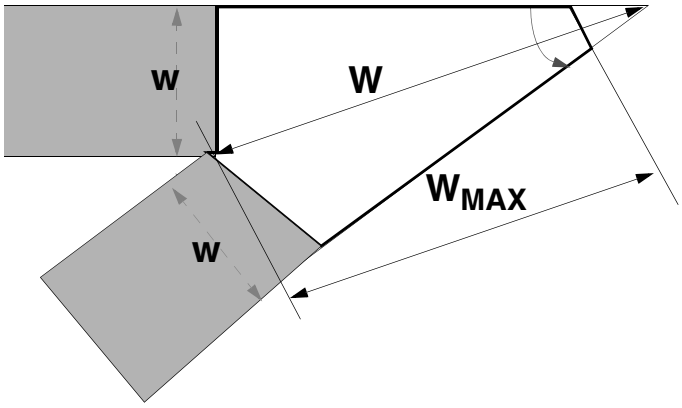


Specified value of bend factor defines W_{MAX} in the following figure:



$$\text{Bend factor} = \frac{W_{MAX}}{W_0}$$

For $W>W_{MAX}$, bend chamfered to decrease W to W_{MAX} in the following figure:

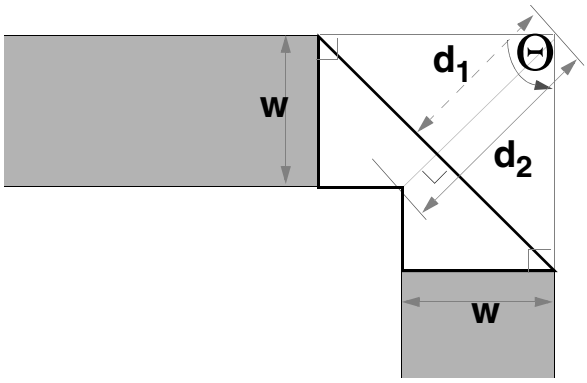


© 2023 Cadence Design Systems, Inc.

Chamfer Bends

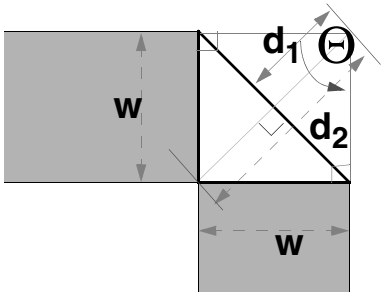
The types of chamfer bends and the effect of *Chamfer Factor* are shown in the following illustrations. The appearance of the corner is determined by the value of the chamfer factor, d_1/d_2.

Example 1



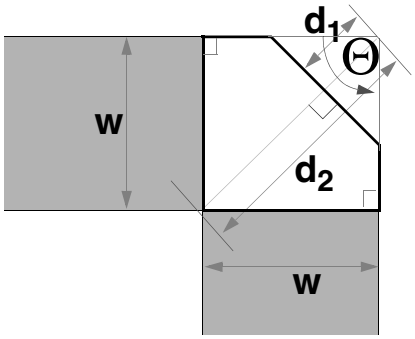
Chamfer factor = d1 / d2 = .7

Example 2



Chamfer factor = d1 / d2 = .5

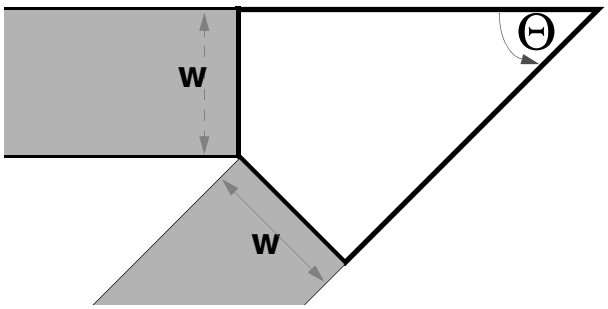
Example 3



Chamfer factor = d1 / d2 = .3

© 2023 Cadence Design Systems, Inc.

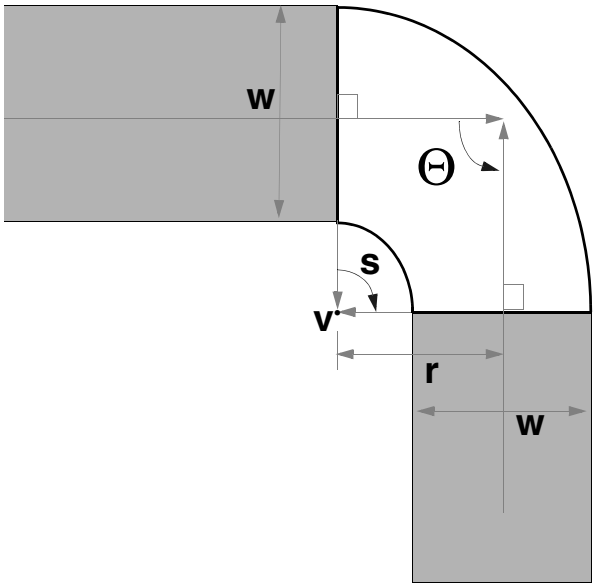
Example 4



Chamfer factor = 0.0

Radial Bends

A radial bend has a rounded corner determined by the *Radius Factor.* The *Resolution* is the number of segments in a 180-degree bend.



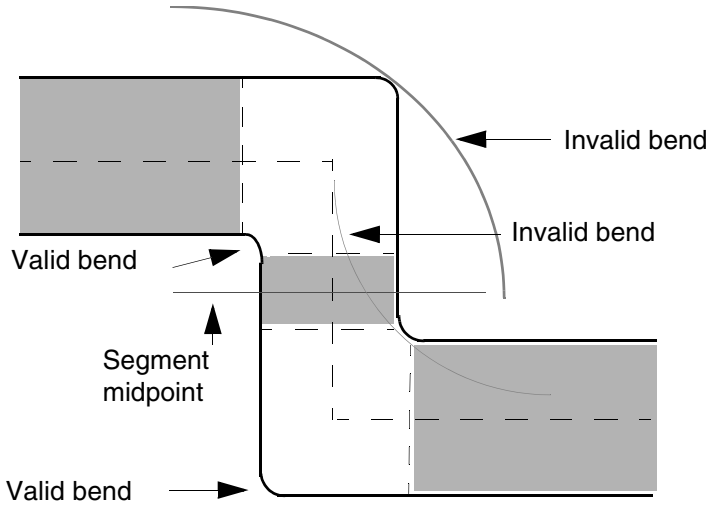
r = (Radius factor) * (Width)

Bends must be properly formed. For example, segments adjacent to a bend must not be too short with respect to the bend. To prevent this, the layout editor does not allow a bend on adjacent segments to extend past the midpoint of the shortest adjacent segment.

In other words, the bend and the segment must coincide before or at the midpoint of the segment so that segments approaching the midpoint from opposite directions match at the

© 2023 Cadence Design Systems, Inc.

midpoint. These segments must intersect the midpoint at 90 degrees. If necessary, the layout editor alters the bend to meet these conditions.



Related Topics

Creating Advanced Transmission Lines

Microwave Commands

Creating Advanced Transmission Lines

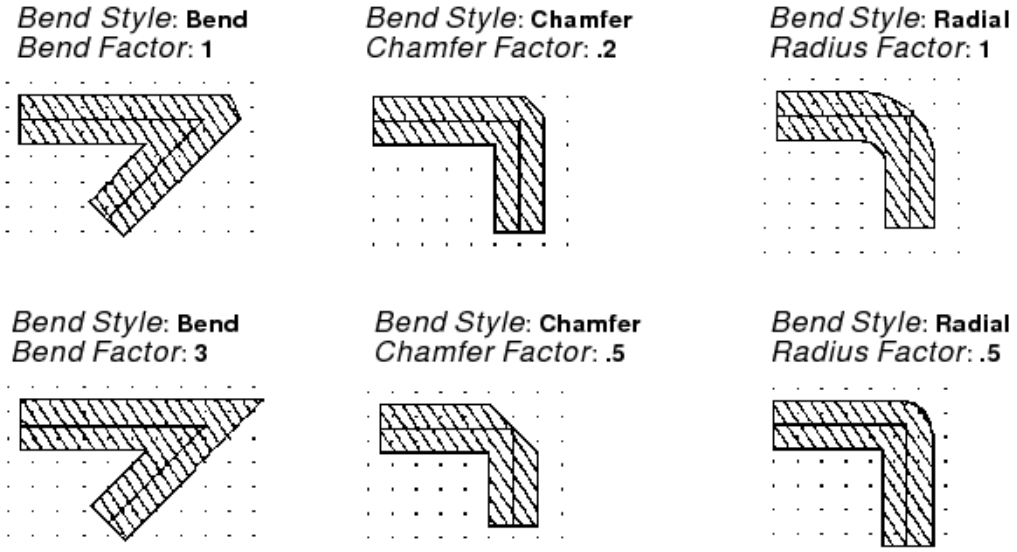
The *Bend Style* option lets you customize the corners of your transmission lines into standard bends, chamfers, and radial corners.

To create an advanced transmission line:

1. In the layout window, from *Create – Microwave*, choose *trl* or *Bend*.
2. Press `F3` to open the Transmission Line Bend form or the Transmission Line form, depending on the command you chose.
3. Choose a style from *Bend Style*. You can choose *bend*, *chamfer* or *radial*.

© 2023 Cadence Design Systems, Inc.

4. Type the corresponding factor to use, as shown in the following image.



5. On the canvas, click each of the three points of the bend.

Related Topics

Transmission Line Form

Transmission Line Bend Form

Transmission Line Bend Styles

Microwave Commands

Creating Tapers

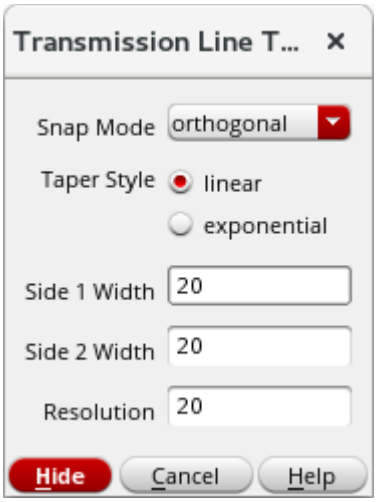
You can use the *Taper* command to create a one-segment transmission line that connects two transmission lines of different width settings. You create tapers similar to the way you create paths.

To create a taper:

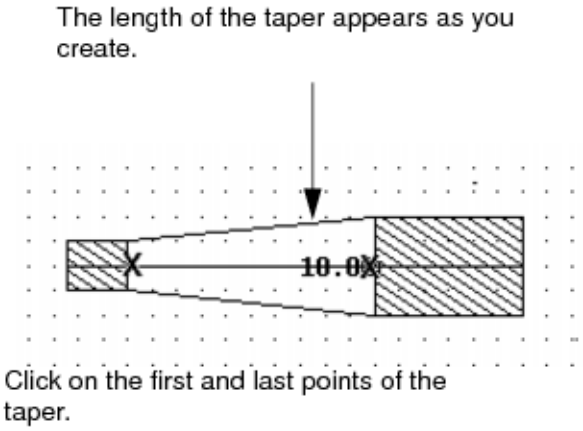
1. In the layout window, choose *Create – Microwave – Taper.*

   The Transmission Line Taper form appears.

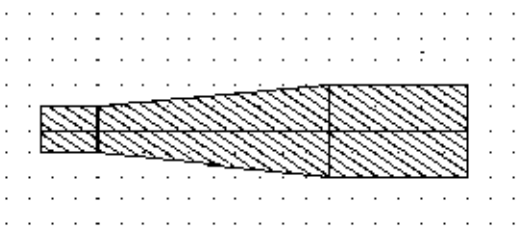
© 2023 Cadence Design Systems, Inc.



2. Type in the beginning and ending widths of the taper in the *Side 1 Width* and *Side 2 Width* fields.
3. Click on the canvas to create each point of the taper.



The following is a completed taper.



© 2023 Cadence Design Systems, Inc.

Related Topics

Transmission Line Taper Form

Microwave Commands

© 2023 Cadence Design Systems, Inc.

12

Troubleshoot Basic Editing Issues in Layout XL

While working on your designs using the layout editor, sometimes you might encounter issues such as those related to accessing your design, creating, editing, or selecting objects, or stretching parametrized cells (Pcells). You must resolve such issues quickly to work on your designs efficiently.

- Troubleshoot Design Access Issues
- Troubleshoot Entered-Points Issues
- Troubleshoot Mouse and Cursor Issues
- Troubleshoot Object Selection Issues
- Troubleshoot Pcell Stretching Issues
- Troubleshoot Basic Command Issues
- Basic Editing Frequently Asked Questions

Troubleshoot Design Access Issues

You might encounter the following common design access situations while using the layout editor.

- Cannot Find a Library
- Cannot Open a Cellview
- Cannot Write to a Cellview
- Cell Instances Are Missing
- ASCII MPP Template File Loading Returns a CIW Warning

© 2023 Cadence Design Systems, Inc.

Related Topics

Troubleshoot Basic Editing Issues in Layout XL

Cannot Find a Library

In the *Library Name* field of the Open File form, if you do not see the name of the library you want, one of the following might be true:

- The library path in the `cds.lib` file is incorrect.
- The library is not in the `cds.lib` file.

Related Topics

Open File Form

Troubleshoot Design Access Issues

Cannot Open a Cellview

If you cannot open a cellview in a library, you might not have the read access to the cellview files or the library might be empty. In these cases, the Open File form displays `<None>` in the *Cell Name* and *View Name* fields.

To gain read access to the cellviews in a library, do one of the following:

- Change the access permissions using the Access Permissions form. You can access the form by choosing *Tools – Library Manager – Edit – Access Permissions* in the CIW.
- Use the UNIX command `chmod` to change the access permissions in the UNIX directory containing the library.

Related Topics

Open File Form

Access Permissions Form

Troubleshoot Design Access Issues

© 2023 Cadence Design Systems, Inc.

Cannot Write to a Cellview

If you have write access to a library but cannot open a cellview for editing, one of the following might be true:

- You do not have write access for the cellview file.
- Another user is editing the cellview and has locked it.

If the owner of the cellview set privileges to read only, you see a message like this:

```
The cellView could not be opened for edit. Do you want to open it for read?
```

To gain write access to the cellviews in a library, do one of the following:

- Change the access permissions using the Access Permissions form. You can access the form by choosing *Tools – Library Manager – Edit – Access Permissions* in the CIW.
- Use the UNIX command `chmod` to change the access permissions in the UNIX directory containing the library.

Related Topics

Access Permissions Form

Troubleshoot Design Access Issues

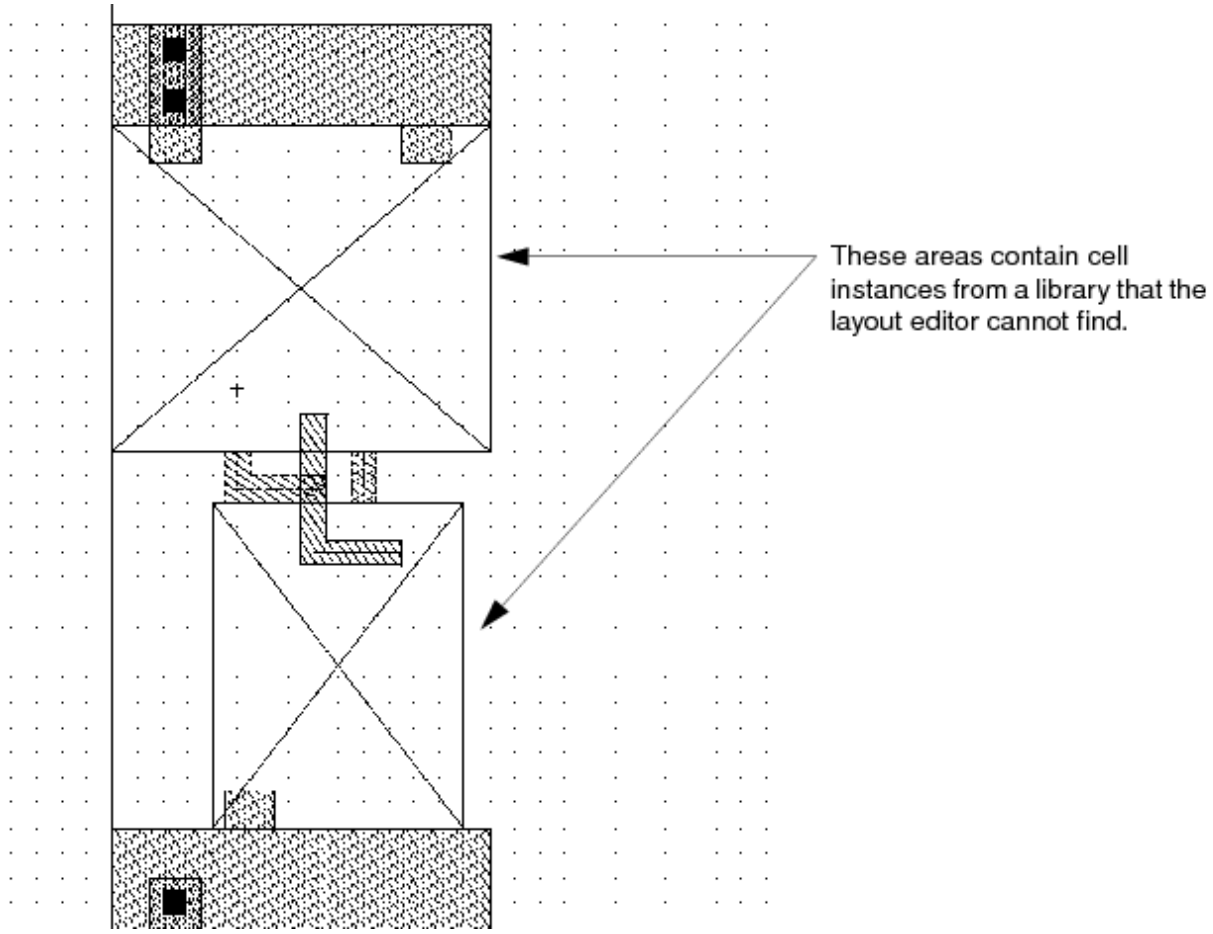
Cell Instances Are Missing

A cellview often contains instances of cells from other design libraries. If you open a cellview that contains instances of cells from a library that the layout editor cannot find, the following happens:

- When you try to open the cellview, you see a warning dialog box listing cells that the layout editor cannot find.

© 2023 Cadence Design Systems, Inc.

- When you close the dialog box, the cellview opens, but each area containing a missing cell displays a flashing box with an X.



To include the missing cells, add the path to the library containing the cell masters to the `cds.lib` file.

Related Topics

Using the Library Path Editor

Troubleshoot Design Access Issues

ASCII MPP Template File Loading Returns a CIW Warning

When loading an ASCII template file, the following warning message is displayed in the CIW:

```
MPP template name is replacing an existing MPP template
```


© 2023 Cadence Design Systems, Inc.

When you load an ASCII multipart path (MPP) template file, the system checks the names of the MPP templates in the ASCII file against the names of MPP templates that already exist in the technology file in the virtual memory. If there is a name conflict, the system displays a warning in the CIW stating that an MPP template in the ASCII file is replacing an existing MPP template.

The existing template is overwritten in virtual memory. If you do not want to overwrite the original MPP template in your binary technology library on the disk, then do not save changes to the technology file. You can also avoid overwriting existing templates by changing the duplicate names in your ASCII file.

If you want to change the names of the MPP templates in your ASCII file right away, or you want to be able to save other changes to your technology library on disk without overwriting the original template, do the following:

1. Exit the layout software without saving technology file changes.
2. Edit your ASCII file to change template names.
3. Restart the layout software.
4. Reload your ASCII MPP template file.

Related Topics

Loading an ASCII Multipart Path Template File

Troubleshoot Design Access Issues

Troubleshoot Basic Editing Issues in Layout XL

Troubleshoot Entered-Points Issues

While creating or editing objects by entering points, you might encounter the following situations.

- Points Entered Incorrectly
- Cannot Move Objects or Enter Points As Required
- Object Moves Instead of Stretching
- Object Changes Shape When Resizing
- Cannot Paste an Object

© 2023 Cadence Design Systems, Inc.

- Pcell Parameters Not Appearing
- Cannot Edit the Properties of Selected Instances

Related Topics

Troubleshoot Basic Editing Issues in Layout XL

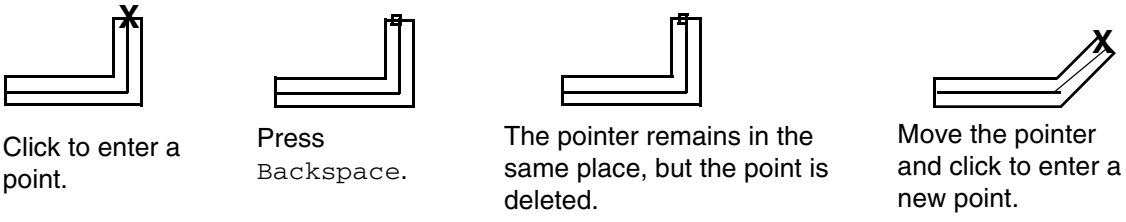
Points Entered Incorrectly

While creating an object if you entered a point incorrectly, you can delete the point.

To delete the last point entered:

1. Press `Backspace`.

   The last point you entered is deleted.
2. Click to enter a new point.



If you finish creating an object and then find it is incorrect, remove the object by undoing the last command by choosing *Edit – Undo*.

You create objects on layers that represent the layers of your physical design. If you notice you are creating objects on the wrong layer, then click the layer that you want in the Layers panel of the Palette assistant. The current entry layer changes to the one you choose.

If you finish creating an object and notice it is on the wrong layer, change the layer for that object by doing either of the following:

- To move the object to a new layer, use the *Edit – Move* command.
- To change the layer property of the object, use the *Edit – Basic –* Properties command.

© 2023 Cadence Design Systems, Inc.

Related Topics

Object Creation by Entering Points

Palette Panels

Object Editing in Layout XL

Object Moving

Troubleshoot Entered-Points Issues

Troubleshoot Basic Editing Issues in Layout XL

Cannot Move Objects or Enter Points As Required

You might need to change the snap mode if you encounter the following issues:

- Cannot enter points to create or re-create objects.
- Cannot move or stretch an object.

To change this behavior, change the snap mode by doing the following:

1. While a command is active, open the form by pressing `F3`.
2. Choose the snap mode you want.

To change the snap mode defaults for creating and editing objects:

1. Choose *Options – Display*.

   The Display Options form appears.
2. Choose a new value for the *Create* or *Edit Snap Modes*.
3. Click *OK*.

Related Topics

Object Creation by Entering Points

Display Options Form

Object Editing in Layout XL

© 2023 Cadence Design Systems, Inc.

Troubleshoot Entered-Points Issues

Troubleshoot Basic Editing Issues in Layout XL

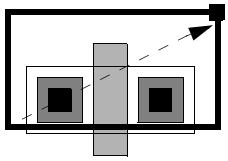
Object Moves Instead of Stretching

If an object moves when you expected it to stretch, you probably selected the whole object rather than its edge or corner. To stretch an object, you must select only the edge or corner that you want to stretch.

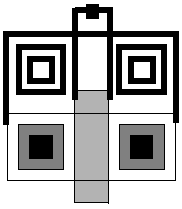
If you want to select edges or corners (vertices), you must first set partial selection on. Partial selection lets you select edges or corners with the selection box.

To toggle partial selection on or off, press `F4`.

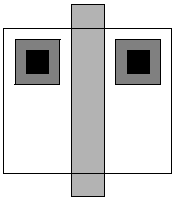
When partial selection is on, the Status Banner or the Status Toolbar show `P` in front of the number of selected objects; for example: `(P) Select: 1`.



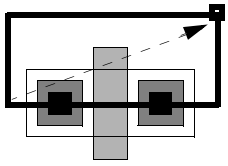
If you select whole objects along with edges...



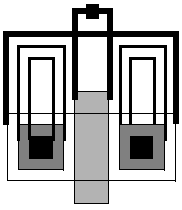
...the whole objects move.



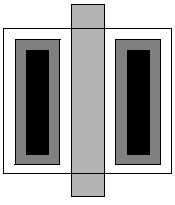
The stretched and moved objects



If you select only partial objects (edges or corners)...



...all objects stretch.



The stretched objects

Related Topics

Object Moving

Object Stretching

© 2023 Cadence Design Systems, Inc.

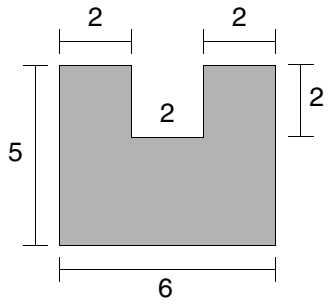
Troubleshoot Entered-Points Issues

Troubleshoot Basic Editing Issues in Layout XL

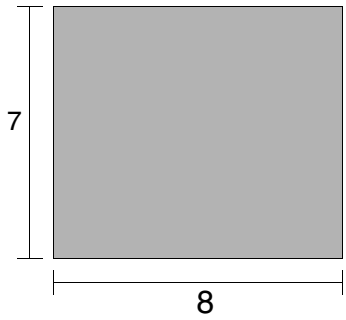
Object Changes Shape When Resizing

The *Size* command and the *Grow By* option for the *Layer Generation* command each let you resize an object by stretching its corners. Sometimes, enlarging or reducing the object causes it to change shape.

For example, if you use *Size* to enlarge the following polygon by 1 unit, each vertex stretches in both the X and Y direction by 1 unit, and the notch in the polygon disappears. The resulting shape is stored as a rectangle, rather than a polygon.



Original shape



After using *Size* to enlarge the shape by 1 unit

To cancel the changes you made by resizing a shape, immediately choose *Edit – Undo*.

Related Topics

Modifying Object Size

Troubleshoot Entered-Points Issues

Troubleshoot Basic Editing Issues in Layout XL

Cannot Paste an Object

If you use *Yank* and *Paste* commands to cut through the hierarchy and place parts of an instance into a different library, the new library must define all the layers you yanked. If it does not, you see the following message:

© 2023 Cadence Design Systems, Inc.

```
Shapes on invalid or invisible layers not pasted.
```

The *Yank* command lets you copy an area of the cellview, cutting through layers of hierarchy. You can select just part of an instance to copy with *Yank*, then use *Paste* to place the parts of the instance you copied into any cellview.

Related Topics

Object Yanking and Pasting

Troubleshoot Entered-Points Issues

Troubleshoot Basic Editing Issues in Layout XL

Pcell Parameters Not Appearing

The parameters for parameterized cells (Pcells) appear in the Create Instance form as you place an instance of the Pcell.

When you type the name of a cell into the Create Instance form, you must specify to the system that you are done before the Pcell parameters can appear.

- If Pcell parameters do not appear in the Create Instance form, press `Tab` to indicate that you are done typing the cell name.
- If the form is at the bottom of your screen, the parameter section can scroll off the screen. Move the form up to see the parameters.

If you use Component Description Format (CDF) C-level function parameters for your library or for any cells in the library, and your Pcell was compiled in version 4.2.1 or earlier, the CDF parameters suppress the Pcell parameter display.

- To recompile Pcells created in version 4.2.1 or earlier, choose *Pcell – Compile – To Pcell.*

Related Topics

Create Instance Form

Troubleshoot Entered-Points Issues

Troubleshoot Basic Editing Issues in Layout XL

© 2023 Cadence Design Systems, Inc.

Cannot Edit the Properties of Selected Instances

When multiple instances are selected in the property editor, some fields may be grayed out and uneditable in the property editor. This is because any instance parameters that have Component Description Format (CDF) SKILL function callbacks cannot be edited, unless all the instances are of the same component, in the case of Pcells the same super master, and have identical parameter values. Instances of different components without identical parameter values must be edited individually to ensure consistent data.

Related Topics

Editing Properties of an Object

Troubleshoot Entered-Points Issues

Troubleshoot Basic Editing Issues in Layout XL

Troubleshoot Mouse and Cursor Issues

While working with your designs, you might encounter the following mouse- and cursor-related issues.

- Pointer Changes Shape
- Cursor Sticks to Objects
- Cursor Snaps to the Wrong Object
- Cursor Does not Snap to a Point
- Right Mouse Button Does not Work

Pointer Changes Shape

The mouse pointer changes shape to show you how you can use it to select, move, or stretch objects. If you see the pointer change shape, this is not an error.

Cursor Sticks to Objects

If the mouse cursor seems to jump to or stick to objects in your cellview, the gravity setting is most likely on.

© 2023 Cadence Design Systems, Inc.

Gravity causes the mouse cursor to automatically snap to objects and grid points in the cellview. You typically use gravity when you want to snap the cursor to specific types of objects as you create them.

To turn gravity off:

➤ Move the mouse pointer to a layout window and press `G`.

If this does not work, your system administrator changed the default key bindings. Use the Layout Editor Options form to set gravity.

Cursor Snaps to the Wrong Object

If you want to use gravity but the cursor snaps to the wrong object, you might need to change the gravity settings. Use the Layout Editor Options form to change gravity settings.

Cursor Does not Snap to a Point

If gravity is disabled and the cursor does not snap to the grid point you want, the snap spacing might be set too wide. The X and Y snap spacing settings control the number of units at which the cursor snaps to the grid. Use the Display Options Form to change the snap spacing settings.

Right Mouse Button Does not Work

By default, the right mouse button works as follows:

- Displays the shortcut menu when you click the right mouse button once.
- Zooms in when you press and hold the right mouse button and create a box on the canvas.
- Zooms out when you press the `Shift` key and hold the right mouse button and create a box on the canvas.

If the right mouse button does not do any of these tasks, it is probably set to create strokes. A stroke is a pre-programmed figure you can create to start a command.

➤ To cancel the stroke-creation capability, you must exit and restart the software.

If strokes are still on when you restart, then the commands to load strokes are included in your `.cdsinit` file.

To remove the stroke commands from your `.cdsinit` file:

© 2023 Cadence Design Systems, Inc.

1. Use a text editor such as `vi` to open your `.cdsinit` file, which is usually located in your current or working directory.
2. Look for these lines in your `.cdsinit` file:

```
load(prependInstallPath( "etc/sted/stroke.il"))
load(prependInstallPath( "etc/sted/defstrokes.il"))
hiLoadStrokeFile("def.strokes" "Layout")
```

3. Type a semicolon (`;`) in front of each line to comment it out.
4. Save the edited `.cdsinit` file.
5. Exit and restart the software.

Related Topics

Layout Editor Options Form

Display Options Form

Troubleshoot Basic Editing Issues in Layout XL

Troubleshoot Object Selection Issues

While working with your designs, you might encounter the following common object selection-related issues and situations.

- Unable to Select Instances
- Unable to Select Objects on a Layer
- Unable to Select All Objects During Edit-In-Place
- Unable to Select Objects in a Dense Design
- Unable to Select Copied Objects
- Selection Box Zooms the Cellview Instead of Selecting Objects
- How to Select Invisible Objects

Related Topics

Troubleshoot Basic Editing Issues in Layout XL

© 2023 Cadence Design Systems, Inc.

Unable to Select Instances

If you cannot select an instance, check to see if selectability of *Instances* in the Objects panel of the Palette Assistant is turned off.

➤ If the *Selectability* check box for *Instances* is turned off, click to turn it on.

Related Topics

Palette Panels

Troubleshoot Object Selection Issues

Troubleshoot Basic Editing Issues in Layout XL

Unable to Select Objects on a Layer

The only objects that you can select in a cellview are on selectable layers in the Layers panel of the Palette assistant.

All layers are selectable if they are included and set to selectable and valid in the `techDisplays` subsection in the technology file, whether or not they appear in the Layers panel in the Palette Assistant. If they do appear in the Layers panel, you can turn on or off their selectability or validity.

In the Palette assistant, you can turn off selection for one or more layers. When you do, a layer that is not selectable appears with its name shaded in the Layers panel.

To make a layer selectable:

➤ Right-click the layer name in the Layers panel.

To display a layer on the Layers panel, choose *Edit – Set Valid Layers* from the shortcut menu in the panel. If you are using the `leLswLayers` section of the technology file, only the layers listed in that file are displayed.

Related Topics

Palette Panels

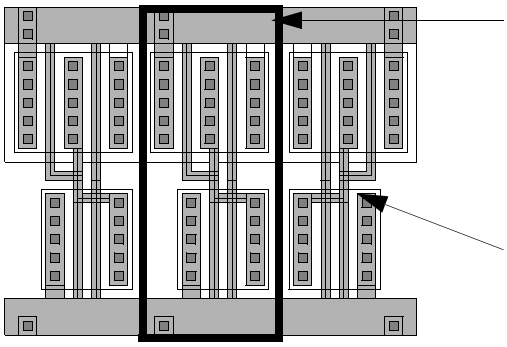
Troubleshoot Object Selection Issues

Troubleshoot Basic Editing Issues in Layout XL

© 2023 Cadence Design Systems, Inc.

Unable to Select All Objects During Edit-In-Place

If you are editing a cell instance in place, you can select and edit only the objects within the instance. Even though you can see objects around the instance, you cannot edit them.



- To check to see if you are editing in place, look at the cellview banner to see what instance you are editing.

  If the name in the banner does not match the name of the current cellview, then you are editing a cell in place.

- To return to editing the current cellview, choose *Edit – Hierarchy – Return*.

Related Topics

Hierarchical Designs

Troubleshoot Object Selection Issues

Troubleshoot Basic Editing Issues in Layout XL

Unable to Select Objects in a Dense Design

If you click objects in a dense design and the layout editor does not select the object you want, try any of the following:

- If objects share the edge you chose, click again in the same place.

  The layout editor selects another of the overlapping objects.

- If possible, move the cursor closer to another edge of the object and click it.

- If possible, zoom in on the edge you want to select.

© 2023 Cadence Design Systems, Inc.

Use the *View – Save/Restore – Previous View* command to zoom out again after selecting the object.

- Use the Palette assistant to turn off selectability for:
  - Layers that overlap the object
  - Pins or instances, if a pin or instance overlaps the object

Related Topics

Layers Panel

Troubleshoot Object Selection Issues

Troubleshoot Basic Editing Issues in Layout XL

Unable to Select Copied Objects

If you move or copy an object into a cellview of a different library and then are not able to select the object:

1. Click in the new cellview.
2. Look for the name of the layer for the object in the Layers panel of the Palette assistant.
   - If the layer name is unselectable (shaded), right-click the layer name to make it selectable.
   - If the layer name does not appear in the Layers panel, the layer either is not defined for this library or has been removed from the panel.

If the layer is not defined for the library, cancel the copy by doing either of the following:

- Choose *Edit – Undo* in the original cellview.
- Choose *File – Discard Edits* in the new cellview to delete all edits you made since you last saved the cellview.

If the layer has been removed from the Layers panel, replace it by choosing *Edit – Set Valid Layers*.

Related Topics

Layers Panel

© 2023 Cadence Design Systems, Inc.

Troubleshoot Object Selection Issues

Troubleshoot Basic Editing Issues in Layout XL

Selection Box Zooms the Cellview Instead of Selecting Objects

If you create a selection rectangle and the image in the cellview is enlarged or reduced, you drew the rectangle using the right mouse button instead of the left mouse button. Dragging by using the right mouse button starts *Zoom In* and *Zoom Out* commands.

- Creating a rectangle with the right mouse button enlarges the cellview image.
- Creating a rectangle by pressing `Shift` with the right mouse button reduces the cellview image.

If you inadvertently zoom in or out instead of selecting objects:

1. Choose *Previous* from the *View – Save/Restore – Previous View* menu to re-display the cellview as it appeared before you zoomed in or out.
2. Click and drag to create a rectangle to select objects.

Related Topics

Object Selection

Troubleshoot Object Selection Issues

Troubleshoot Basic Editing Issues in Layout XL

How to Select Invisible Objects

If *None Visible* (*NV*) outlines an area of your cellview that appears blank, you probably turned off the visibility of all layers that appear in an instance and selected that invisible instance.

For example, if you clicked *NV* in the *Layers* panel to set all layers invisible, but *Instances* in the *Objects* panel is still on, you can select invisible instances.

To select invisible objects, do one of the following:

- Set *Instances* off in the Objects panel so that instances are not selectable.
- Click *AV* in the Layers panel to make all layers visible.

© 2023 Cadence Design Systems, Inc.

Related Topics

Layers Panel

Troubleshoot Object Selection Issues

Troubleshoot Basic Editing Issues in Layout XL

Troubleshoot Pcell Stretching Issues

The following describe how you can handle issues related to Pcell stretch handles.

- Pcell Stretch Handles Are Not Visible
- Difficulty Selecting Pcell Stretch Handles

Pcell Stretch Handles Are Not Visible

If you cannot see handles on stretchable parametrized cells (Pcells), the *Stretch Handles* option in the Display Options form might be turned off or the stretch handle layer might not be visible.

You can check whether Pcell stretch handles are set to display using the Display Options form or using the graphic and layout editor environment variable displayStretchHandles. The default is to display stretch handles. You cannot stretch a handle unless it is displayed.

If the *Stretch Handles* option is turned on, and you still cannot see stretch handles on your Pcell, the layer containing them might not be visible.

Turning On Stretch Handle Display in the Display Options Form

To turn on *Stretch Handles* in the layout editor Display Options form:

1. In the layout window, choose *Options – Display* to open the Display Options form.
2. Turn on *Stretch Handles* and click *OK*.

Turning on Stretch Handle Display Using the CIW

To turn on `displayStretchHandles` environment variable:

1. Type the following in the CIW:

© 2023 Cadence Design Systems, Inc.

```
window = hiGetCurrentWindow()
```

2. To display stretch handles, type the following in the CIW:

```
window~>displayStretchHandles = t
```

Making the Stretch Handle Layer Visible

The default layer for stretch handles is `y0 drawing`. To make this layer visible:

1. In the Layers panel in the layout window, choose *Edit Valid Layers.*
2. In the Layers panel, click *AV* to make all layers visible. Alternatively, you can enable the visibility for `y0 drawing`.
3. Choose *View – Redraw* to redraw the window to see the results of your changes.

If you want to use a different layer for stretch handles, you can change the layer with the stretchHandlesLayer environment variable.

Difficulty Selecting Pcell Stretch Handles

If you have trouble selecting stretch handles, use area selection rather than point selection or try turning off the layout editor *Gravity On* option.

To turn off gravity:

1. In the layout window, choose *Options – Editor.*

   The Layout Editor Options form opens.
2. Deselect the *Gravity On* check box in the *Gravity Controls* section and click *OK*.
3. Try again to select a Pcell stretch handle.

Related Topics

Display Options Form

Layout Editor Options Form

Layers Panel

Troubleshoot Basic Editing Issues in Layout XL

© 2023 Cadence Design Systems, Inc.

Troubleshoot Basic Command Issues

The following list describes how you can handle unexpected results that arise while using layout editor commands.

- **Command starts before points are entered**

  If you press a bindkey to start a command and the command seems to start before you click a point, infix mode is probably on. When infix is on, any command you start with a bindkey uses the current pointer location as its first point.

- **Command does not stop**

  - If you cannot stop a command, one of the following might be happening:
    - Each time you finish the command, it restarts. This means you are using a command that automatically repeats.
    - A form appears on screen, and you want to close it without executing the command.

  In either case, cancel the command by pressing `Esc` or click *Cancel* on the form.

  It is not always necessary to cancel a repeating command. If you are running a repeating command and then start a non-repeating command, the repeating command pauses while the new command runs. After you are finished with the non-repeating command, the repeating command continues. Also, if you start another repeating command, the first repeating command stops.

- **Command does not repeat when it should**

  By default, editing commands repeat automatically only if you first choose the command, then select the object. Commands do not repeat if you first select the object, then choose the command.

  If you chose an editing command before selecting an object, and the command still does not repeat, one of the following might be the cause:

  - An object is selected that you cannot see.

    Look at the number next to *Select* in the Status Banner or the Status Toolbar to see if one or more objects are selected. For example, if one object is selected, the Status Banner or the Status Toolbar displays: `Select: 1`. Zoom out the cellview so you can see all objects.

  - The *Repeat Commands* setting for the editor has been turned off.

© 2023 Cadence Design Systems, Inc.

To turn the *Repeat Commands* setting on, choose *Options – Editor* and in the Layout Editor Options form, set Re*peat Commands* on.

Related Topics

Basic Command Functions in Layout XL

Layout Editor Options Form

Basic Editing Frequently Asked Questions

The following are some of the areas with frequently asked questions (FAQs) related to Virtuoso Layout Suite.

- Window Display FAQs
- Object Creation FAQs
- Object Editing FAQs
- Ruler-Related FAQs
- Layer-Related FAQs

Window Display FAQs

The following are the FAQs related to window display:

- How do I display Instance contents?
- How do I display instances being edited?
- How do I see what the search command finds?
- How do I find the cds environment variables corresponding to the fields in the Display Options and Layout Editor Options forms?
- How do I define the layout cellview selection box?

How do I display Instance contents?

If you see only an outline of each instance instead of the details of objects inside the instance, change the display level settings.

© 2023 Cadence Design Systems, Inc.

To turn on instance detail, press `Shift+F`.

`Shift+F` is the bindkey equivalent of using the Display Options form to set the display levels to 0 through 32.

How do I display instances being edited?

If you are editing a cell in place, you can set the window to display only the instance that you are editing.

To see the rest of the instances:

1. Choose *Options – Display*.

   The Display Options form appears.
2. Set *EIP Surround* on.
3. Click *OK*.

How do I see what the search command finds?

If you use the *Search* command to search for or select objects and your cellview is zoomed in, the command might find an object that is not visible in the current window.

To zoom the cellview so the objects searched appear in the window:

➤ In the Find/Replace form, set *Zoom To Figure* on.

Now, each time you click *Previous* or *Next*, the cellview is redrawn so that the object is centered in the window.

How do I find the cds environment variables corresponding to the fields in the Display Options and Layout Editor Options forms?

You can find the cds environment variables corresponding to the fields in the Display Options and Layout Editor Options forms by searching the field in the Search Assistant. You access the *Search* assistant by selecting *Window – Assistants – Search.*

How do I define the layout cellview selection box?

A cell is selectable when the cursor moves over an invisible layer near the cell boundary. You can change the invisible layer so that the selectability of the cell is very tiny. For example, the

© 2023 Cadence Design Systems, Inc.

selectability is only the lower-left corner of the cell. When the cursor moves over the cell, it will not be selectable, or dynamically highlighted till the cursor is on the lower-left corner of the cell. For this, you should enable the *Use True BBox* option on the Display Options form to create user-defined instance or drawing shape to display the outline of the instance instead of the cellview bounding box.

Related Topics

Display Options Form

Find/Replace Form

Search Assistant

Object Creation FAQs

The following are the FAQs related to creating objects:

- How do I set Auto mode in the Create Label command as the default label creation mode?
- When I create pins, why are the labels not being displayed?
- How do I create pins by reading data from a text file?
- How do I set the default via on the Create Via form?
- How do I map the parameters on the System tab while creating a custom via?

How do I set Auto mode in the Create Label command as the default label creation mode?

You can control default mode for the Create Label command by setting the `labelMode` option environment variable in the home directory or the `.cdsenv` file, as shown below:

```
layout   labelMode   cyclic  "auto"
```

You can also set it through the `.cdsinit` file:

```
envSetVal("layout"  "labelMode"  'cyclic  "auto")
```

Also, the leHiCreateLabel SKILL API supports a second argument through which you can pass on the mode in which you want to launch the Create Label command.

```
leHiCreateLabel(hiGetCurrentWindow()  "auto")
```


© 2023 Cadence Design Systems, Inc.

You can also permanently change the bindkey settings to set the auto mode as the default mode for the Create Label command, as shown below:

```
hiSetBindKey("Layout" "<Key>L"  "leHiCreateLabel(hiGetCurrentWindow()  \"auto\")")
```

In IC614 and above releases, you need to press F3 to open the Create Label form. As a workaround, you can define a SKILL function and a bindkey to display the form by default, as shown below:

```
;---------------
procedure(CCScreateLabel()
envSetVal("ui" "showOptionForms" 'boolean t)
leHiCreateLabel()
envSetVal("ui" "showOptionForms" 'boolean nil)
)
hiSetBindKey("Layout" "<Key>L" "CCScreateLabel")
 ;--------------
```

The `L` bindkey can be used to invoke the Create Label command.

When I create pins, why are the labels not being displayed?

You should enable the *Pin Names* option on the Display Options form to display the labels for pins. You can also enable the display of pin names by setting the `displayPinNames` environment variable in the `.cdsenv` file, as shown below:

```
layout displayPinNames boolean t
```

How do I create pins by reading data from a text file?

You can automate the pin creation process by reading an ASCII/text file as an input. The input file can contains details of pin coordinates, layer, pin names, and access direction, as shown in the sample input text file and SKILL code below.

Input text file (`indata`):

```
->layer purpose shape points termName termDirection accessDirection<-

"Poly" "drawing" "rectangle" 1.7 1.5 2.4 2.6 "my_pin1" "inputOutput"
"top"

"Metal1" "drawing" "rectangle" 3.2 3.3 4.3 4.8 "my_pin2" "input"
"bottom"
```


© 2023 Cadence Design Systems, Inc.

```
"Metal2" "drawing" "rectangle" 5.1 5.2 6.4 6.7 "my_pin3" "output"
"none"
```

The following is a sample SKILL code that can be used for pin creation:

```
;----------------CCSCreatePin.il-------------------------------------------------
---------------
procedure(CCSCreatePin(pinListfile libName cellName viewName)
 let((z inPort cvId nextline layer purpose lpp shape p1 p2 p3 p4 points termName
termDir accessDir)
  inPort = infile(pinListfile)
  cvId=dbOpenCellViewByType(libName cellName viewName "" "a")
  when( inPort
   while(z=gets(nextline inPort)
   layer=readstring(nth(0 parseString(z)))
   purpose=readstring(nth(1 parseString(z)))
   lpp=list(layer purpose)
   shape=readstring(nth(2 parseString(z)))
   p1=readstring(nth(3 parseString(z)))
   p2=readstring(nth(4 parseString(z)))
   p3=readstring(nth(5 parseString(z)))
   p4=readstring(nth(6 parseString(z)))
   points=list(p1:p2 p3:p4)
   termName=readstring(nth(7 parseString(z)))
   termDir=readstring(nth(8 parseString(z)))
   accessDir=list(readstring(nth(9 parseString(z))))
   leCreatePin(cvId lpp shape points termName termDir accessDir)
   )
  close(inPort)
  );when
 );let
);procedure
```

To create pins by reading data from a text file:

1. Load the SKILL script `CCSCreatePin.il` through the CIW or `.cdsinit` file. Ensure that the layout cellview exists.
2. Call the below mentioned SKILL procedure:

   ```
   CCSCreatePin("./indata" "library" "cell" "layout")
   ```

   You should specify the first argument as the input file name, followed the library name, cell name, and view name of the layout cellview where the pins need to be created.

© 2023 Cadence Design Systems, Inc.

How do I set the default via on the Create Via form?

The Create Via form reads the constraint group set in *Create Via* section of Layout Editor Options form. The list of valid vias for this constraint group are populated in *Via Definition* field. You can change the default value of *Via Definition* in the Create Via form. For example, you might want that whenever you open a new layout window, by default, the `M5_M4v` via should appear in the *Via Definition* field.

You can use the following sample SKILL code to change the default via.

```
procedure(CCSRegProc(args)
    hiRegTimer("CCSSetDefVia(args->window)" 5)
    leHiCreateVia()
) ;proc

procedure(CCSSetDefVia(window)
    let((cv cstG currCG techId validVias)
      cancelEnterFun('le0ViaForm)
        cv=geGetWindowCellView(window)
        if(envGetVal("layout" "viaConstraintGroup")=="Same as Wire" then
               currCG=envGetVal("layout" "wireConstraintGroup")
              else
              currCG=envGetVal("layout" "viaConstraintGroup")
          ) ;if
      techId=techGetTechFile(cv)
      cstG=cstFindConstraintGroupIn(techId currCG)
      validVias=car(setof(x cstG~>objects  x~>defName=="validVias"))~>value

       if(member("M5_M4v" validVias) then
          le0ViaForm->viaSingleModeScrollView->viaDefName->value="M5_M4v"
         ) ;if
) ;let
) ;procedure

deRegUserTriggers("maskLayout" nil nil 'CCSRegProc)
deRegUserTriggers("maskLayoutXL" nil nil 'CCSRegProc)

hiSetBindKey("Layout" "<Key>o" "CCSSetDefVia(hiGetCurrentWindow())
leHiCreateVia()")
```

To use the above sample SKILL code:

1. Save the above code in a file, for example, `CCSChangeDefaultVia.il`.

© 2023 Cadence Design Systems, Inc.

2. Add the `load` command to load this file to your `.cdsinit` file:

   `load"<path_to_file>"/CCSChangeDefaultVia.il"`

3. Start Virtuoso and open any layout cellview. If the constraint group set for the Create Via option in the Layout Editor Options form has `M5_M4v` in the list of valid vias, then it will become the default via in the *Via Definition* field.

4. Change the constraint group for Create Via in the Layout Editor Options form. Then, open the Create Via form using the bindkey `O`. After this, you may use either the bindkey or the menu option.

How do I map the parameters on the System tab while creating a custom via?

You can use the *Rows* and *Columns* values of the System tab, instead of the *User defined* tab, of the Create Via form if you specify the parameter names in the via Pcell code as `row` and `column`. For mapping the Pcell via, use the following names to map the system parameters: `row`, `column`, `w`, `l`, `xCutSpacing`, `yCutSpacing`, `xBias`, and `yBias`.

If the Pcell via has the enclosure information, it is displayed under the *User Defined* tab and you can modify the value only on this tab.

Related Topics

Object Creation in Layout XL

Create Label Form

Create Via Form

Display Options Form

Object Editing FAQs

The following are the FAQs related to editing objects:

- How do I change the number of rows and columns during a subsequent copy operation?
- How do I replace cellviews hierarchically in a design?
- How do I list the hierarchy from top to bottom?
- When I stretch a path that overlaps a shape, why does the path remain connected to the shape?

© 2023 Cadence Design Systems, Inc.

How do I change the number of rows and columns during a subsequent copy operation?

To change the number of rows and columns in the subsequent copy operation, you can set the copyResetRowCol environment variable in the home directory or the `.cdsenv` file.

```
layout copyResetRowCol boolean t
```

If `copyResetRowCol` is set to `nil`, the default value, which is the last value used for rows and columns, is used for the next copy operation. However, if `copyResetRowCol` is set to `t`, the values of rows and columns in the Enter Points form get reset.

You can also set `copyResetRowCol` in the `.cdsinit` file using the following command:

```
leSetEnv("copyResetRowCol" t)
```

How do I replace cellviews hierarchically in a design?

You can quickly replace cellviews hierarchically in a design. For example, there is a cell, `MYCAP`, with multiple layout views, `layout_A` and `layout_B`. The `layout_A` of `MYCAP` cell is instantiated in the cell, `MY_CAP_ARRAY`, in the library, `testlib`. To swap `layout_A` with `layout_B` in the cell `MY_CAP_ARRAY`, you can use the following sample SKILL API:

```
CCSTop("tstlib" "MY_CAP_ARRAY" "layout" "MYCAP" "layout_A" "layout_B" '("refLibA"
"refLibB") )
```

In the example, `refLibA` and `refLibB` are read-only reference libraries.

How do I list the hierarchy from top to bottom?

When you use the command *Edit – Hierarchy – Tree* with the *Top to bottom* option selected, the complete hierarchy is not listed. The *Display Level* settings in the Display Options form limits the number of levels listed by the `Tree` command. If the *Display Level* is set to 0 through 3, even if you select the *Top to bottom* option on the Tree form, the output will be for levels 0 through 3 only. So, before running the `Tree` command with the *Top to bottom* option selected, change the *Display Level* value to 32.

When I stretch a path that overlaps a shape, why does the path remain connected to the shape?

You should use the *Keep Wires Connected To* option on the Stretch form to specify the connectivity option. If you have not selected this option, the path will not remain connected to the rectangle. For example, you created a rectangle and then created a path that is on the same layer and overlaps the rectangle. If you try to stretch the path, the end of the path will

© 2023 Cadence Design Systems, Inc.

remain connected to the rectangle, even when the *Maintain Connections* option on the Layout Editor Options form is deselected.

Related Topics

Object Editing in Layout XL

Hierarchical Designs

Ruler-Related FAQs

The following are the ruler-related FAQs:

- Why does the Zoom In command take a long time to refresh the layout if the layout has multiple large sized rulers?
- Why does the ruler disappear from the screen when I save a design if the Save Rulers option is not selected?
- How do I cycle through the snapping targets in the Create Ruler command?

Why does the Zoom In command take a long time to refresh the layout if the layout has multiple large sized rulers?

In IC6.1.5 ISR13 and above releases, set the `useTheNewMeasurementImpl` environment variable to `t` to activate the new drawing code for the rulers. This will reduce the time to refresh.

```
graphic useTheNewMeasurementImpl boolean t
```

Or enable the variable through the `cdsinit` file or the CIW:

```
envSetVal("graphic" "useTheNewMeasurementImpl" 'boolean t)
```

In IC6.1.6 and above releases, the `useTheNewMeasurementImpl` graphic environment variable is not available and has been removed from the `cdsenv` file of the hierarchy. In these releases, the ruler behavior is controlled by the layout `rulerImpl` environment variable that is set to sting `new`, by default. Also, in IC6.1.6 and above releases, OA supports scratch cellviews and can creates transient rulers. A transient ruler is a ruler that cannot be saved. However, by default, persistent rulers are created in the layout editor. A persistent ruler can be saved. If you want to create transient rulers, deselect the *Create Ruler as a savable object* option on the *General* tab of the *Create Ruler* form.

© 2023 Cadence Design Systems, Inc.

Why does the ruler disappear from the screen when I save a design if the Save Rulers option is not selected?

You can retain rulers on the screen and not save them in the database with the help of the following SKILL code. You can load the SKILL code through the `.cdsinit` file or the CIW:

```
load "CCSrestoreRulers.il"
```

This SKILL code uses the global variable `rulerPtListGLOBAL`.

```
*/
/* Presave trigger for maskLayout views */
procedure(CCSgrabRulers(cv)
let((currentRulers)
if(cv~>cellViewType=="maskLayout"
     then
;; Get all the ruler ID's
    currentRulers=setof(sh cv~>markers sh~>objType=="ruler")
;; If rulers are in design then store ruler points in global list
         if(currentRulers then
              rulerPtListGLOBAL=foreach(mapcar rulerId currentRulers
                                                        rulerId~>points
                                                          )
;; Delete all the rulers
               foreach(rulerId currentRulers dbDeleteObject(rulerId))
;; If no rulers then make sure that global list is empty
                         else
                         rulerPtListGLOBAL=nil
                         )
               )
     )
) ;; Pre save proc ends
/* Postsave trigger for maskLayout views */
procedure(CCSrestoreRulers(cv)
if(cv~>cellViewType=="maskLayout" && rulerPtListGLOBAL
then
;; Stop the cellview modification counter otherwise cell will be inconsistent
dbDisableCellViewCounter(cv)
;; Recreate all the rulers from global list
foreach(rulerPt rulerPtListGLOBAL leCreateRuler(cv rulerPt))
               )
;; After creation empty the global list
rulerPtListGLOBAL=nil
```


© 2023 Cadence Design Systems, Inc.

```
;; Enable cellview modification counter for normal working
dbEnableCellViewCounter(cv)
)  ;; Post save proc ends
/* Procedure for truly saving rulers */
procedure(CCStrulySaveRulers()
dbUnregSaveTrigger('CCSgrabRulers)
)
```

The code ensures that rulers are not saved in the database but are displayed on the screen even if the design is saved. After loading this file, pre-save and post-save triggers are registered. These triggers work for layout views only. The SKILL code enables the `saveRulers` environment variable by default. This environment variable enables the *Save Rulers* option on the Layout Editor Options form. However, if you want to save the rulers, use the `F9` bindkey. Once the *Save Rulers* option is selected on the Layout Editor Options form, the pre-save trigger is disabled. If you have enabled the *Save Rulers* option on the Layout Editor Options form, to get back to the pre-save trigger mode, use the `F10` bindkey so that the pre-save trigger is registered once again.

- Press the *F9* key before the save operation for pre-save trigger mode:

  ```
  hiSetBindKey("Layout" "<Key>F9" "CCStrulySaveRulers()")
  ```

- Press the *F10* key to ensure that the rulers are not saved:

  ```
  hiSetBindKey("Layout" "<Key>F10" "dbRegSaveTrigger('CCSgrabRulers)")
  ```

- Ensure that the `saveRulers` environment variable is set to true:

  ```
  leSetEnv("saveRulers" t)
  ```

- Register the pre and post-save triggers:

  ```
  dbRegSaveTrigger('CCSgrabRulers)
  dbRegPostSaveTrigger('CCSrestoreRulers)
  ```

How do I cycle through the snapping targets in the Create Ruler command?

Press the spacebar to cycle through the snapping targets in the *Create Ruler* and *Quick Align* commands.

Related Topics

Rulers

© 2023 Cadence Design Systems, Inc.

Layer-Related FAQs

The following are the FAQs related to layers:

- How do I Make a Design Layer Visible?
- How do I enable the Used option in the Layers panel for every layout view that I open?
- How do I ensure a specific layer is used to trace the net using the Mark Net functionality?

How do I Make a Design Layer Visible?

If you cannot see one of your design layers, it is probably set to be invisible. A layer that is invisible appears with its layer color shaded in the Layers panel.

To make a layer visible:

1. In the Layers panel, click the middle mouse button on a shaded layer.
2. Choose *View – Redraw* to see the results of your changes.

How do I enable the Used option in the Layers panel for every layout view that I open?

In the `.cdsenv` file, set the `pteUsedOnlyScope` environment variable to `t`.

```
layout pteUsedOnlyScope boolean t
```

How do I ensure a specific layer is used to trace the net using the Mark Net functionality?

To make the Mark Net functionality use a specific layer to trace the nets:

1. Set the markNetAutoColorCycle and markNetThickline environment variables in the appropriate `.cdsenv` files as shown below:

   ```
   layout markNetAutoColorCycle boolean nil

   layout markNetThickline boolean t
   ```

   The `markNetAutoColorCycle` environment variable disables the *Cycle Through System Colors Automatically* option.

2. Load the SKILL code listed below in the CIW or through the `.cdsinit` file. It defines the bindkey `F10` that can be used to reset the MarkNet Options form so that it sets the default layer in the form.

© 2023 Cadence Design Systems, Inc.

```
load " CCSFixMarkNetColor.il"
procedure(CCSFixMarkNetColor()
let(()
;;Change the index of nth function to specify the specific layer to use in the
order of y0-y9, like for y0 nth(0..), for y1 nth(1…) etc and so on.
hiRegTimer("le0MarkNetForm->optionTabs->page1->markNetColor->value=nth(4
le0MarkNetForm->optionTabs->page1->markNetColor->choices)" 1)
leHiMarkNet()
t
) ;let
) ;procedure CCSFixMarkNetColor

hiSetBindKey("Layout" "F10" "CCSFixMarkNetColor()")
```

3. Open the layout window and press the `F10` bindkey to invoke the mark net and trace a net. The `y4 drawing` layer will be used to mark the nets.

   Note: In IC 6.1.6 and above releases, new SKILL functions `leMarkNet`, `leSaveMarkNet`, `leUnmarkNet` are available. In IC6.1.6.500.1 and above, and ICADV12.1.500.3 and above, the following function can also be used to mark the nets on a specific `y0 drawing - y9 drawing` layers.

```
leMarkNet('(5.7 103.0) ?startLevel 0 ?stopLevel 32 ?thickLine t ?markNetColor
"y2")
procedure(CCSMarkNet()
enterPoint(
?prompts list("Click on a point to Mark the Net")
?doneProc "CCSTraceNet"
);enterPoint
);procedure CCSMarkNet

procedure(CCSTraceNet(win done pt "wgl")
let(()
when(done
geGetWindowCellView(win)
pt=car(pt)
printf("Clicked point is %L\n" pt)
leMarkNet(pt ?startLevel 0 ?stopLevel 32 ?thickLine t ?markNetColor "y2")
printf("Marked the Net\n")
); when
);let
);procedure CCSTraceNet

hiSetBindKey("Layout" "<Key>F10" "CCSMarkNet()")
```


© 2023 Cadence Design Systems, Inc.

```
hiSetBindKey("Layout" "<Key>F11" "leUnmarkNet()")
```

Related Topics

Layers Panel

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

13

Snap Pattern Snapping for FinFET Devices in Layout Editor

Snap patterns help to ensure that placement constraints defining width and spacing rules are honored when FinFET devices are created or edited. You can define snap patterns for each layer-purpose pair on which FinFET devices are placed. You can display fin grids on the layout canvas and automatically snap shapes and design objects to a fin grid.

Snap pattern snapping is supported in the *Create Rectangle*, *Create Pin*, *Create Instance*, *Create Via*, *Create Wire*, *Copy*, *Move*, *Stretch*, *Flip*, *Rotate*, *Yank* and *Paste*, and *Repeat Copy* commands.

Snap pattern definitions (snapPatternDefs) can be local or global. A local snap pattern definition is drawn using a snap pattern shape in the layout. A global snap pattern definition can be activated through either a constraint or by drawing a corresponding snap pattern shape.

Related Topics

Turning On Snap Pattern Snapping

Controlling the Display of Snap Patterns

Object Snapping to Local Snap Pattern Shapes

Track Group Snapping

Object Snapping to Global Snap Pattern Grid

Snap Pattern Display Controls in the Palette Assistant

Instance Placement Modes

Object Snapping in singleTrackCenter Snapping Mode

Snap Pattern Display Packets

© 2023 Cadence Design Systems, Inc.

Turning On Snap Pattern Snapping

To turn on the snap pattern snapping in layout editor:

1. In the layout window, choose *Options – Editor*.

   The Layout Editor Options form opens.

2. Select the *Snap Pattern Snapping* check box and click *OK*.



To turn on the *Snap Pattern Snapping* option at startup, you can set the snapPatternSnapping environment variable in your `.cdsenv` file.

Video

For more information on snap pattern snapping, see the video *Snap Pattern Enabling*.

Related Topics

Layout Editor Options Form

Controlling the Display of Snap Patterns

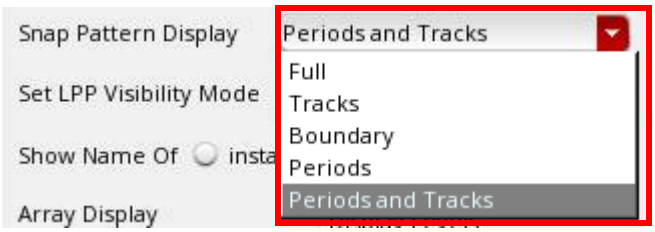
You can display a fin grid or the full fin shapes inside fin boundaries. You can switch between the full fin detail display, which is used to debug fin alignment-related DRC violations, and the high-level fin boundary display, which is used to focus on wiring and placement without being distracted by fin shapes. You can display the boundary, tracks, or both for snap patterns.

To control the display of snap patterns:

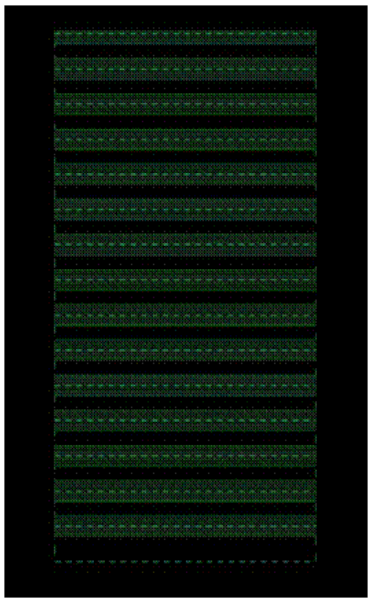
1. In the layout window, choose *Options – Display*.

© 2023 Cadence Design Systems, Inc.

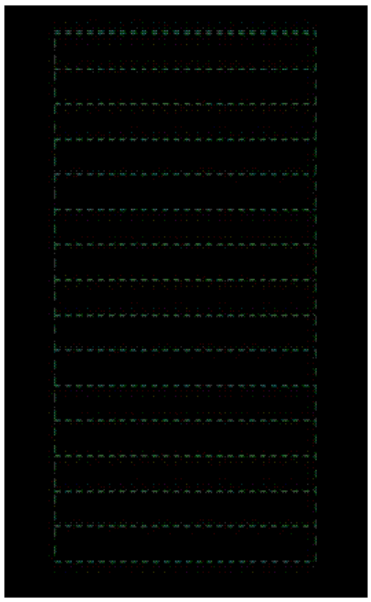
The Display Options form opens. The *Snap Pattern Display* list shows the snap pattern displays available.



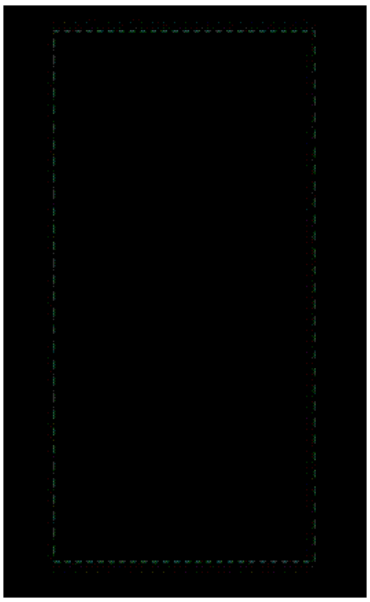
2. Select one of the following from the *Snap Pattern Display* list:
   - *Full*: Displays the snap pattern boundary, tracks, fins, and width spacing patterns periods.
   - *Tracks*: Displays the snap pattern boundary and tracks.
   - *Boundary*: Displays only the snap pattern boundary.
   - *Periods*: Displays only the width spacing pattern periods.
   - *Periods and Tracks*: Displays width spacing pattern periods and tracks.



Full



Tracks



Boundary

3. Click *OK*.

© 2023 Cadence Design Systems, Inc.

You can use the leCycleSnapPatternDisplay SKILL function to cycle through the different snap pattern display types. You can set the default value using the new snapPatternDrawingStyle environment variable.

Related Topics

Display Options Form

Object Snapping to Local Snap Pattern Shapes

A local snap pattern definition (snapPatternDefs) is drawn using a snap pattern shape in the layout.

To create a snap pattern definition for the fin boundary layer-purpose pairs and snap objects to snap patterns, you need to follow these high-level tasks:

- Create a snap pattern definition for the fin boundary in the ASCII technology file.
  - Specify the fin pitch (`step`), fin width (`trackWidth`), direction (`stepDirection`), distance of the first fin center line from the lower edge (`offset`), and type (`type`) of the fin as local or global.
  - Specify the name of the layers, typically the *Active* layer, whose objects will snap to the fin grid (`snappingLayers`) and the required fin enclosures from the center line (`enclosures`). You can specify multiple snapping layers (`layer`) in the snap pattern definition. Each layer can optionally have multiple purposes (`purposes`) and enclosure values (`enclosures`).
- Draw the fin boundary in the layout editor.
- Move a Pcell (including modgen Pcells), a via, an *Active* shape, or an instance over the fin boundary. The objects are automatically snapped to the fin grid.

Related Topics

snapPatternDefs

Object Placement Inside the Fin Boundary

Object Snapping

© 2023 Cadence Design Systems, Inc.

Object Placement Inside the Fin Boundary

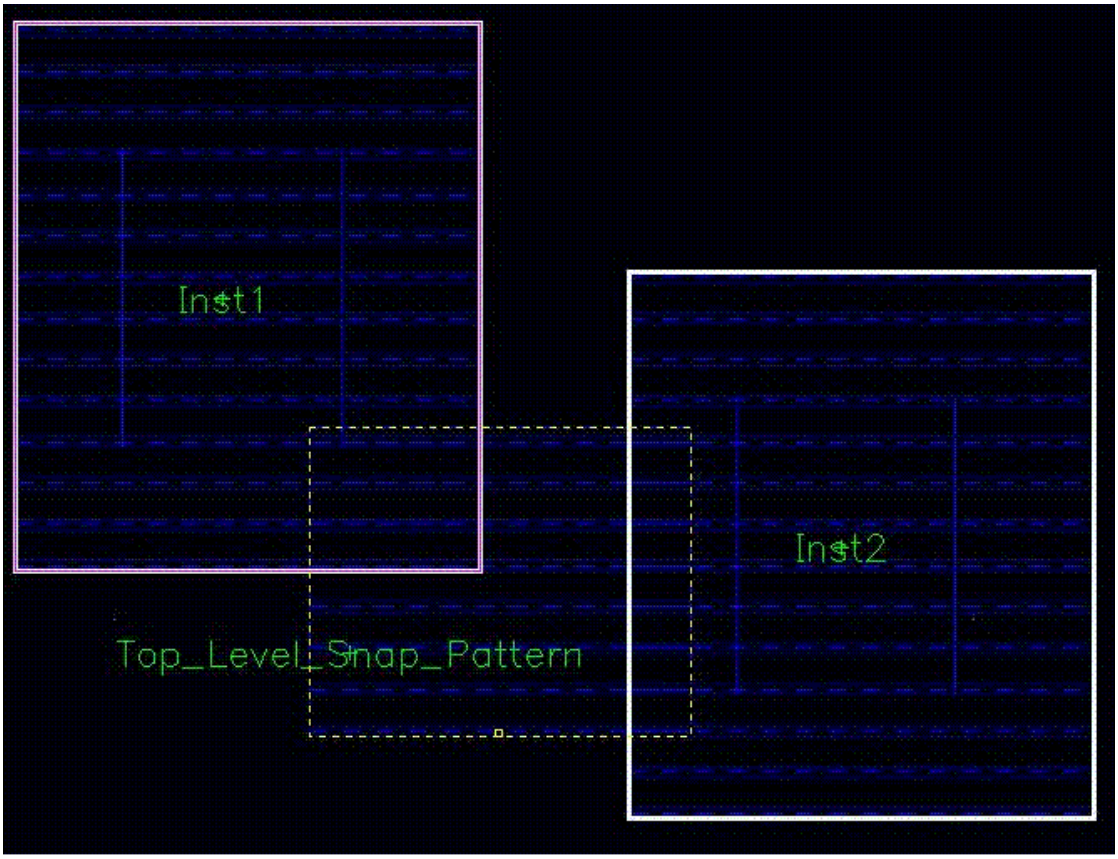
The layout editor provides rapid placement of shapes, Pcells, instances, and vias in a fin boundary by using a shape snapping algorithm. This ensures a correct-by-construction layout on the fin grid. Automatic alignment is achieved by the alignment engine that contains the alignment semantics.

The layout editor commands used for creating and moving Pcells, instances, and vias in a layout make use of the alignment engine to ensure placement correctness. The alignment engine accepts a set of shapes and based on their overlaps with fin boundaries in the design, places them appropriately. The following are the automatic placement requirements and the classification of their results:

- When a snap pattern shape or a Pcell, instance, via, or figGroup containing a snap pattern shape (the free figure) overlaps another snap pattern shape or a Pcell, instance, via, or figGroup containing a snap pattern shape (the fixed figures) on the same layer-purpose pair (LPP), then the alignment engine tries to align the snap pattern tracks. In this process, it determines the transformation that is applied to the free figure such that the closest tracks of the overlapping pair are aligned.

© 2023 Cadence Design Systems, Inc.

The following figure shows the result of snap pattern shape pair alignment. No modification of the Pcell code is necessary to enable snapping of Pcell instances to the fin grid since the snapping is done based on the existing fin boundary shape in the Pcell.



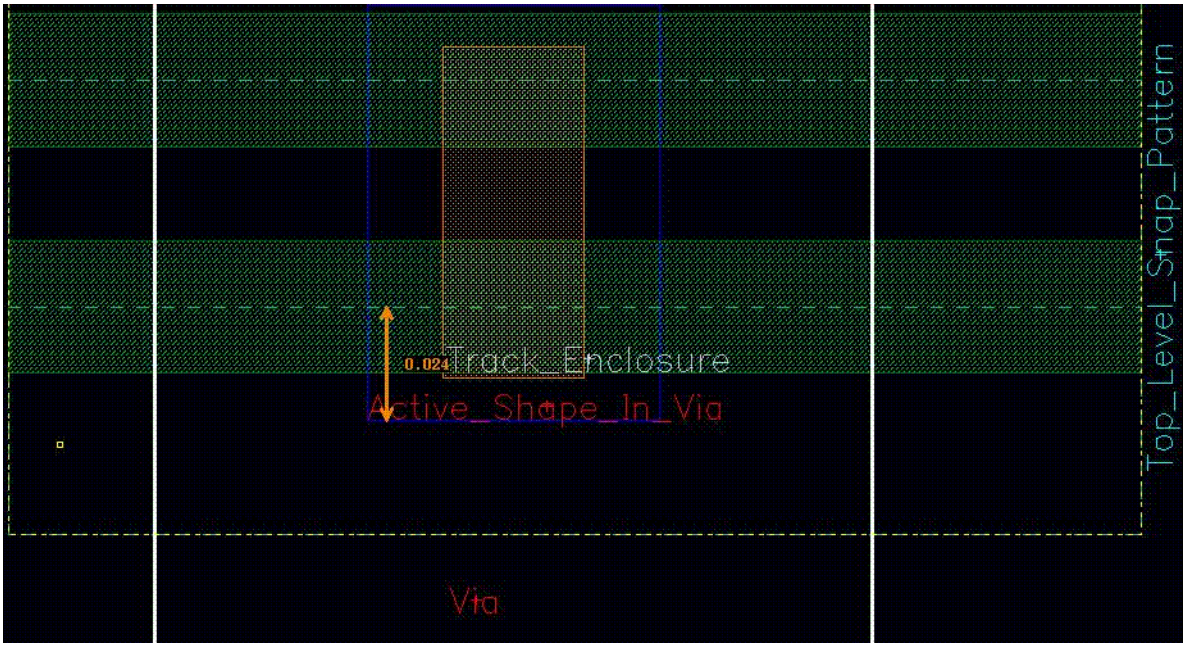
Snap-pattern-to-snap-pattern snapping: The alignment engine aligns the tracks.

- A snap pattern definition specifies a list of LPPs, and shapes on these LPPs snap to the snap pattern grid. When a shape on such a layer, or a Pcell, instance, via, or figGroup containing a shape overlaps a matching snap pattern shape, then the engine computes the transformation required to place the shape as defined by the applicable track enclosure.

  The following figure shows the result of shape-to-snap-pattern snapping for a via that contains this shape. No change to the via is needed to enable snapping of substrate

© 2023 Cadence Design Systems, Inc.

contacts to the fin grid because snapping is done based on the existing shape on the *Active* layer in the via master.



Shape-to-snap-pattern snapping: The alignment engine aligns the snap shape in a via to the grid based on the track enclosure

When you select or drag a snap pattern shape, the shape behaves in the same way as any other shape in the layout editor. You can change the visibility and selectability of the snap pattern shapes using the Palette assistant.

Related Topics

Palette Assistant

Object Snapping to Local Snap Pattern Shapes

Object Snapping

Object Snapping

The layout editor provides rapid placement of instances, shapes, vias, and groups in a fin boundary. Each of these objects are snapped to snap pattern shapes differently. For example, two snap pattern shapes drawn on the same layer-purpose pair snap to each other. Whereas, when an instance containing multiple snap pattern shapes overlaps a snap pattern shape, then the instance is snapped based on only one of the snap pattern shapes.

© 2023 Cadence Design Systems, Inc.

Object snapping includes the following:

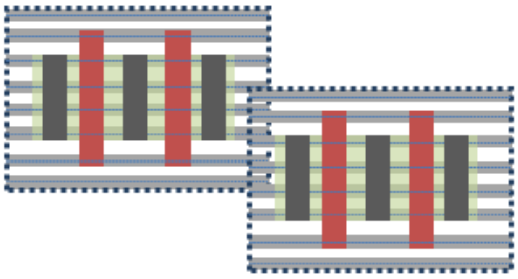
- PR Boundary Snapping
- Snap-Pattern-to-Snap-Pattern Snapping
- Instance Snapping
- Shape Snapping
- Via Snapping
- Group Snapping

PR Boundary Snapping

When you create a PR boundary with *Create Method* as *From Selected Set* and the allowedPRBoundaryDimensions constraint is set in the technology file, the PR boundary will be a multiple of the value specified in the `allowedPRBoundaryDimensions` constraint.

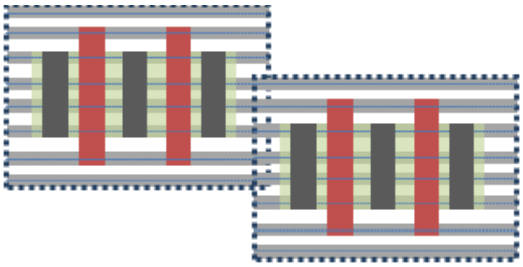
Snap-Pattern-to-Snap-Pattern Snapping

Two snap pattern shapes drawn on the same layer-purpose pair (LPP) snap to each other. Assume a FinFET process where snap pattern shapes are used to draw fin boundaries. If two Pcell instances containing fin boundaries are placed close to each other such that their fin boundaries overlap, then Virtuoso automatically aligns the fins of these snap pattern local regions, as shown in the figure below.



Fixed snap pattern local region

Move another snap pattern local region over the existing one



Virtuoso aligns the snap pattern local regions

When an instance containing multiple snap pattern shapes overlaps a snap pattern shape, then the instance is snapped based on only one of the snap pattern shapes. If the instance partially overlaps the snap pattern shape, one of the overlapping snap patterns is used for snapping.

© 2023 Cadence Design Systems, Inc.

Related Topics

Object Snapping to Local Snap Pattern Shapes

Object Snapping

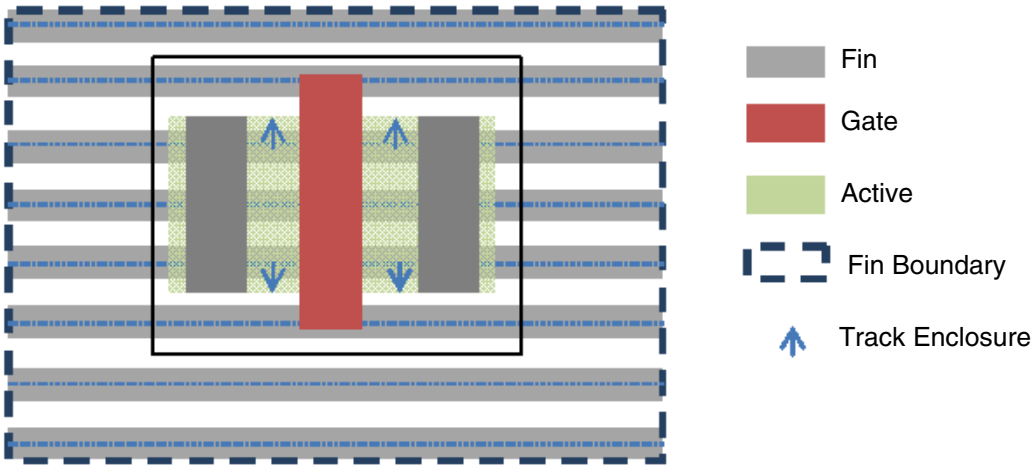
Instance Snapping

When an instance overlaps a snap pattern shape and the master does not contain a snap pattern shape on the same layer-purpose pair (LPP), then the instance is snapped based on its shapes, assuming it contains a shape on an LPP that snaps to the snap pattern shape.

For example, consider a Pcell instance that does not contain a snap pattern, but it contains a shape on the *Active* layer. If this Pcell instance is moved over a snap pattern shape that applies to the *Active* layer, then the instance is snapped to the grid based on the shape on the *Active* layer using the `enclosures` attribute of the snap pattern definition.



Pcell master without snap pattern shape



Pcell instance snapped to the grid after the `Move` command

© 2023 Cadence Design Systems, Inc.

If there are multiple matching shapes, snapping is done based on the first LPP combined bounding box that is found, where the relevant edge overlaps the snap pattern shape. The relevant edge is the lower edge for vertical snapping and the left edge for horizontal snapping.

If the snap pattern shape applies to shapes on multiple LPPs and the instance master contains shapes on several such LPPs, the snapping algorithm performs snapping based on any one of the LPPs.

If the instance overlaps multiple snap pattern shapes that are *Active*, then the algorithm performs snapping based on just one snap pattern shape.

The layout editor automatically snaps the Pcells in modgens to the snap pattern. Therefore, the entire modgen block is snapped to the top-level snap pattern shape. A modgen comprises multiple Pcell instances in the form of a complex, structured array.

Related Topics

Object Snapping

Object Snapping to Local Snap Pattern Shapes

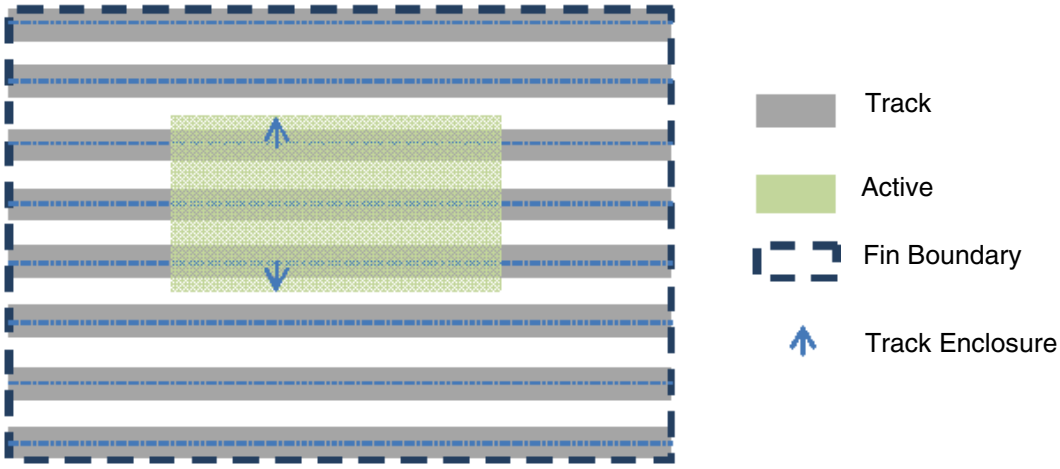
Shape Snapping

Shapes that overlap a snap pattern shape are snapped to the grid after a copy or move command. For example, consider a snap pattern shape that applies to the *Active* layer and

© 2023 Cadence Design Systems, Inc.

it is moved over a snap pattern shape. The shape is snapped to the grid based on the `enclosures` attribute of the snap pattern definition.

Single Shape



Shape snapped to the grid after the `Move` command

A shape may overlap multiple snap pattern shapes and more than one of those may apply to it. In this case, the snapping algorithm performs snapping based on one snap pattern shape. If a shape overlaps multiple snap pattern shapes, snapping is done only if all the snap pattern shapes are aligned.

When you are creating a rectangle and the rectOnSnapPatternGrid environment variable is enabled, a DRC-correct rectangle is created.

Related Topics

Object Snapping

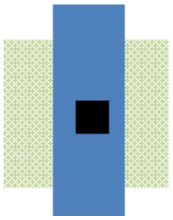
Object Snapping to Local Snap Pattern Shapes

© 2023 Cadence Design Systems, Inc.

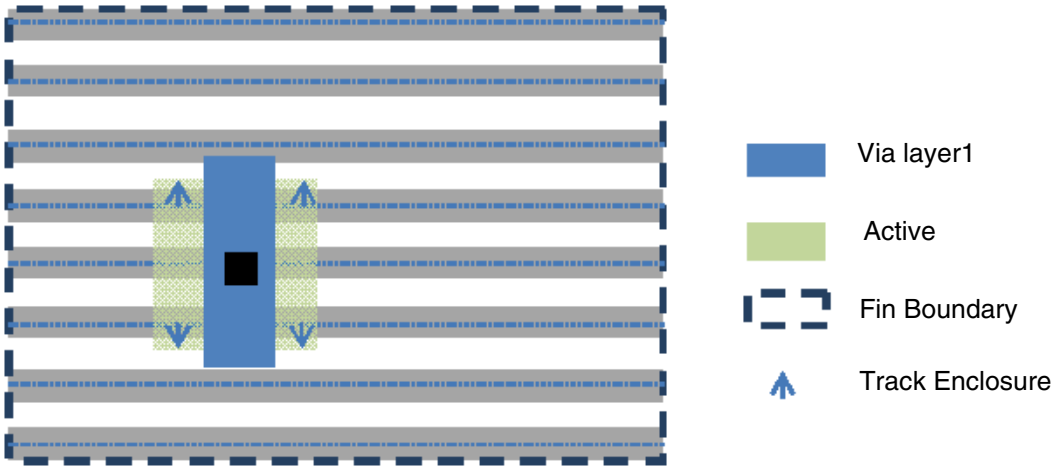
Via Snapping

Custom layout designs can contain stdVia, cdsViaDevice, cdsGenVia, and customVia types of vias. Except for customVias, none of the vias insert a snap pattern shape into the via subMaster. Therefore, vias are snapped based on the shapes they contain, instead of a snap pattern. Also, when a via in a via stack is snapped to a snap pattern shape, the entire via stack is snapped. A via stack is snapped as a group. Via and via stacks that overlap a matching snap pattern shape after a copy, move, or create operation are snapped to the grid.

For example, consider a FinFET process where snap pattern shapes are used for fin boundaries. When a via containing an *Active* shape is placed on a fin boundary, it is automatically aligned to the fin grid based on the `enclosures` of the snap pattern shape. The following figure contains a via with a layer on *Active/drawing*. When the via is moved over a snap pattern shape that applies to all *Active* shapes, the via is automatically snapped to the fin grid based on the `enclosures` attribute of the snap pattern.



Via with Active implant layer



Via is snapped to the snap pattern

When a `snapPatternDef` does not have any `snapPurposes`, it applies to all shapes on the `snapLayer`.

© 2023 Cadence Design Systems, Inc.

If the snap pattern applies to shapes on multiple LPPs and the via master contains shapes on several such LPPs, the snapping algorithm performs snapping based on one of the LPPs.

If the via overlaps multiple snap pattern shapes that are *Active*, then the snapping algorithm performs snapping based on one snap pattern shape.

Snap pattern shapes on the same LPP must snap to each other. Snapping should be performed after a copy, move, or stretch operation or creation of an instance or a via.

You can use the environment variables snapPatternSnappingDepth and snapPatternSnappingDepthMethod to specify the snap depth and the method used to define the snap depth for snap pattern snapping, respectively.

Related Topics

Object Snapping

Object Snapping to Local Snap Pattern Shapes

Group Snapping

The following objects in a cellview are snapped as a group:

- Members of a figGroup
- Vias in a via stack
- A chain of abutted MOS devices

  If the chain is moved in such a way that snap pattern snapping is triggered after the move, all the devices on it are snapped together. This ensures that the abutment chain remains aligned. For *Move* and *Copy* commands, if the *Chain Mode* is set to *All,* all the devices are snapped as a group even if one device is selected.
- Members of a selected set, only if the selected set has a shape

  For *Move* and *Copy* commands, if the selected set has a shape, then the entire selected set is snapped as a group. If the selected set has less than 25 figures, the representative figure is the nearest snappable shape in the hierarchy inside the selected set. However, if the selected set has more than 25 figures, the representative figure is the nearest top-level figure in the selected set; the hierarchy analysis is not done.

© 2023 Cadence Design Systems, Inc.

Related Topics

Object Snapping

Object Snapping to Local Snap Pattern Shapes

Track Group Snapping

The layout editor supports the track groups available in snapPatternDefs. When a `snapPatternDef` has track groups, tracks are combined in groups. Within each track group, the existing `step` is used as the track space. The `period` of the `snapPatternDef` is computed from the group spacings and the number of tracks in each group.

Related Topics

Shape Snapping for Track Groups

Instance Snapping for Track Groups in Detail Mode

Shape Snapping for Track Groups

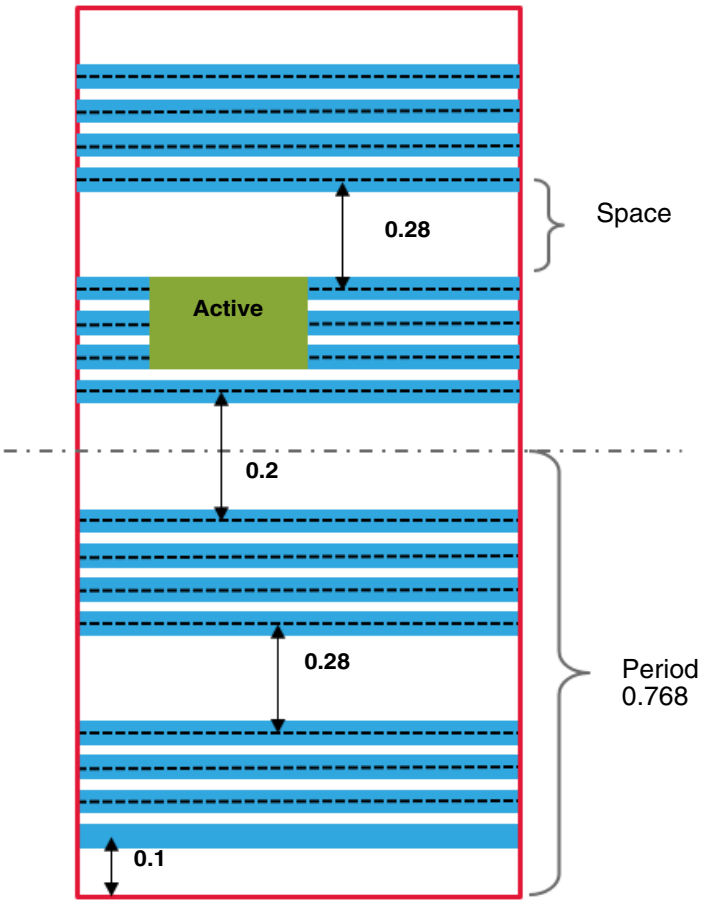
Snap pattern shapes of the `snapPatternDef` snap to their period lines. Shape snapping for track groups is supported both for global and local snap pattern definitions.

For shape-to-snap-pattern snapping, the shape is snapped by having the lower or left edge of the shape enclose a track by the `trackEnclosure` value. The shape is snapped to the nearest track. To move a snapped shape from one track group to another, you need to move the pointer by at least half the track group spacing.

© 2023 Cadence Design Systems, Inc.

The shape-to-snap-pattern snapping with track groups is depicted in the figure below.

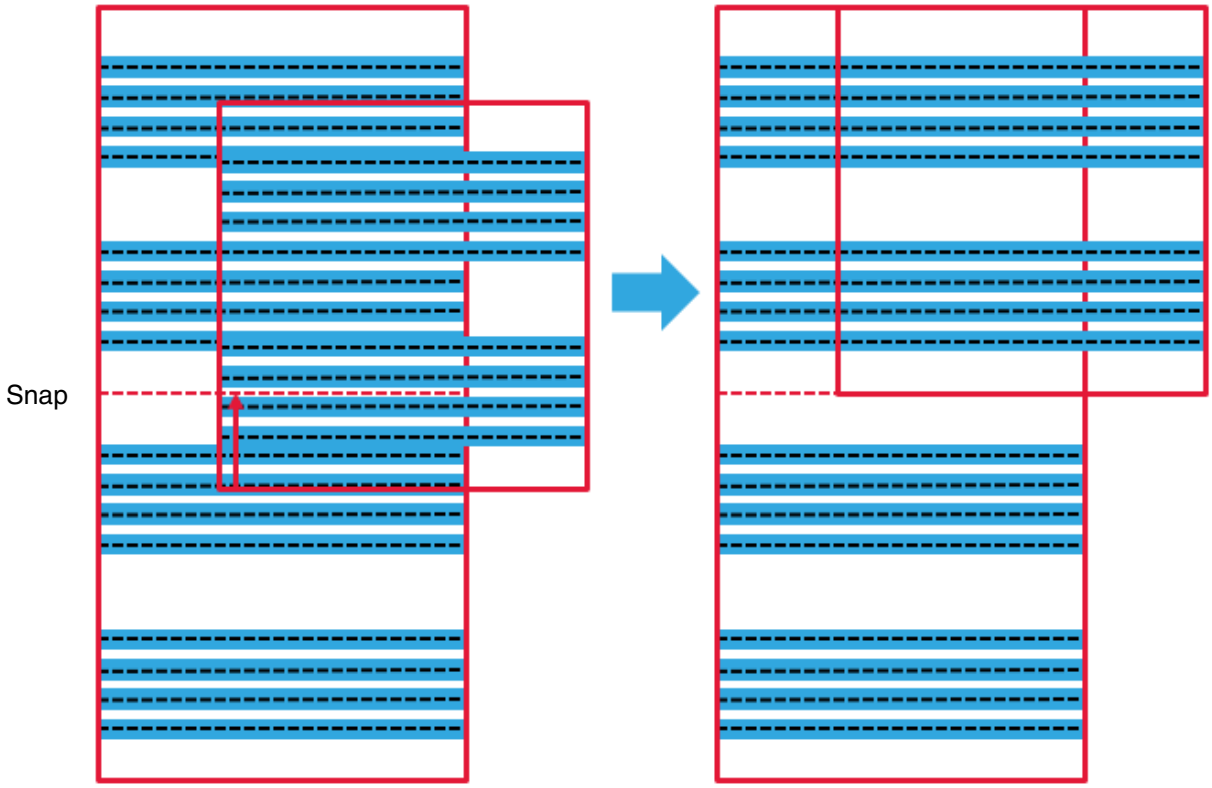
Shape-to-Snap-Pattern Alignment



© 2023 Cadence Design Systems, Inc.

For snap-pattern-to-snap-pattern snapping, the computed period is used. The lower-left edge of the moving snap pattern is snapped to the closest period line of the target snap pattern. The track groups in snap-pattern-to-snap-pattern snapping are depicted in the figure below.

Snap-Pattern-to-Snap-Pattern Alignment



Related Topics

snapPatternDefs

Track Group Snapping

Instance Snapping for Track Groups in Detail Mode

Instance Snapping for Track Groups in Detail Mode

When snapping instances containing snap patterns with track groups, non-uniform snap patterns have a lower precedence than the uniform snap patterns. The following snapping behavior is used:

© 2023 Cadence Design Systems, Inc.

- If an instance has both uniform and non-uniform snap patterns, only the uniform snap pattern is used for snapping.
- If an instance contains only non-uniform snap patterns, snapping is done to period lines.
- If there are no non-uniform snap patterns, there is no change in the snapping behavior.

Instance snapping based on snap patterns is usually more accurate than shape-based snapping. As a result, if a snap pattern shape is found, shape snapping is disabled for that direction. In this case, shape snapping is not performed even if snapping is not possible for the snap pattern shape.

Related Topics

snapPatternDefs

Track Group Snapping

Shape Snapping for Track Groups

Object Snapping to Global Snap Pattern Grid

The layout editor can display a uniform fin grid over the entire design window when you add a global snap pattern grid to the design.

You can create a global snap pattern grid by specifying the value `global` in the `type` constraint of the snap pattern definition, as shown below.

```
snapPatternDefs(
        (GFG             ("CellBoundary" "global")
        'offset           0.024
        'type             "global"
        'step             0.048
        'stepDirection    "vertical"
        ...
```

The `type` constraint can have the value `global` or `local`. A local snap pattern definition is drawn using a snap pattern shape in the layout. A global snap pattern definition can be activated through either a constraint or by drawing a corresponding snap pattern shape.

You can enable the display of global grid by using the `snapGridVertical` and `snapGridHorizontal` constraints. The constraint value is the name of the global `snapPatternDef` and the direction is same as that specified in `snapPatternDef`.

© 2023 Cadence Design Systems, Inc.

In the following example, the constraint value is `GFG`, which is the name of the global, vertical `snapPatternDef`.

```
spacings(
    snapGridVertical("GFG")
  );spacings
```

Related Topics

snapPatternDefs

snapGridVertical

snapGridHorizontal

Global Grid Anchoring

The global grid is anchored in the following order of priority:

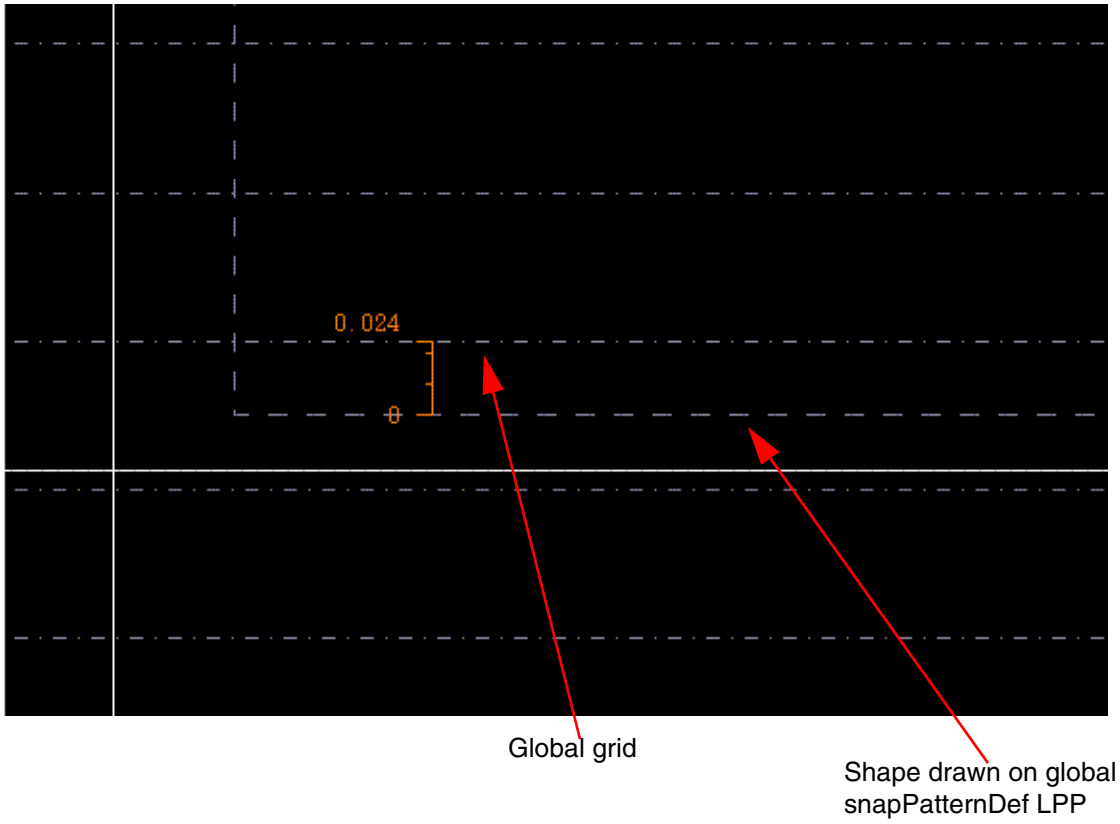
1. Shape on the global snapPatternDef layer-purpose pair (LPP)
2. Snap boundary
3. PR boundary
4. Origin axis

You can enable or disable the global grid anchoring using the snapPatternGlobalGridAnchoring environment variable. When the global grid anchoring is disabled, the global grid is anchored to the origin axis.

The following are a few examples of global grid anchoring:

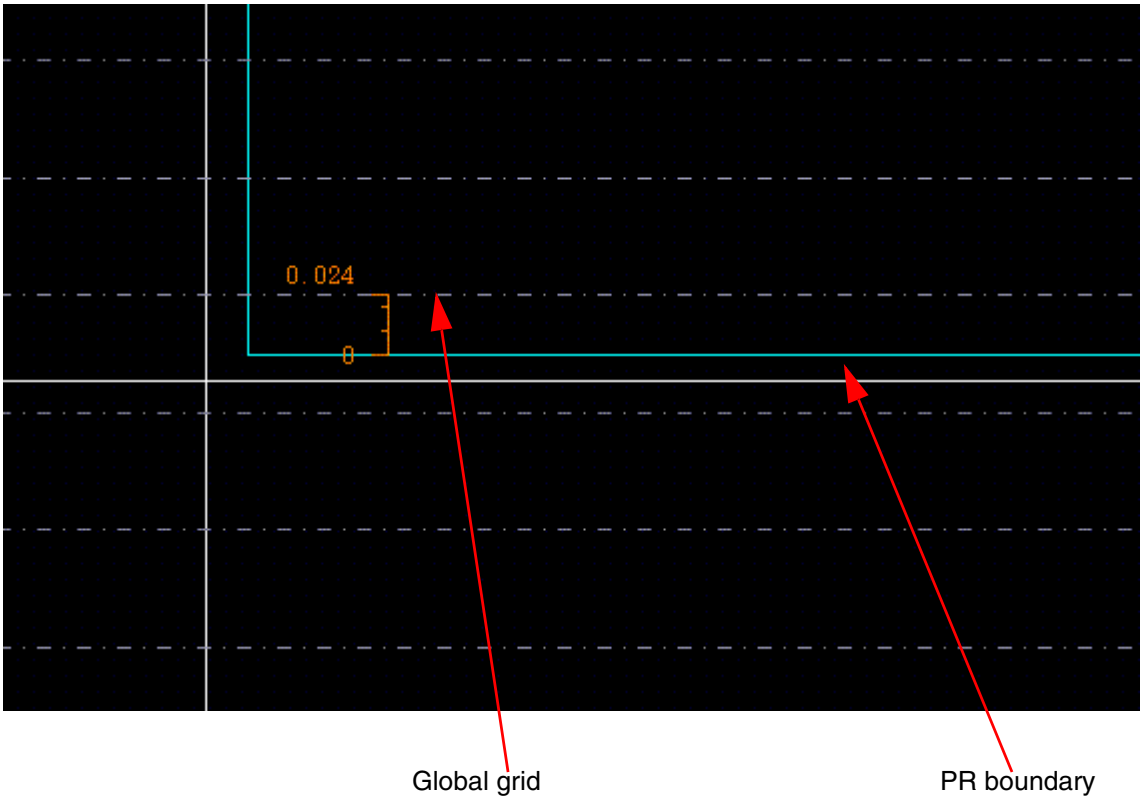
© 2023 Cadence Design Systems, Inc.

- Anchoring by shape on global snapPatternDef LPP: The vertical grid is anchored to the lower edge of the drawn shape. In the example below, an offset of 24nm is shown.



© 2023 Cadence Design Systems, Inc.

- Anchoring by PR boundary: If no shape is drawn on global grid LPP, Virtuoso looks for a PR boundary. If it is found, then the global grid is anchored to the lower edge of the PR boundary.



By default, the PR boundary snaps to the least common multiple of the grid values in the X and Y directions. You can use the boundarySnapXGrids and boundarySnapYGrids environment variables to override the default behavior and snap the PR boundary to the specified grids in X and Y directions, respectively.

When the creation method for PR boundary is *Rectangle*, both the edges on the X and Y axis are snapped. However, when the creation method for PR boundary is *Polygon*, all points are snapped. When the creation method for PR boundary is *From Selected Set* and the *Rectilinear* option is not selected, the bounding box is snapped.

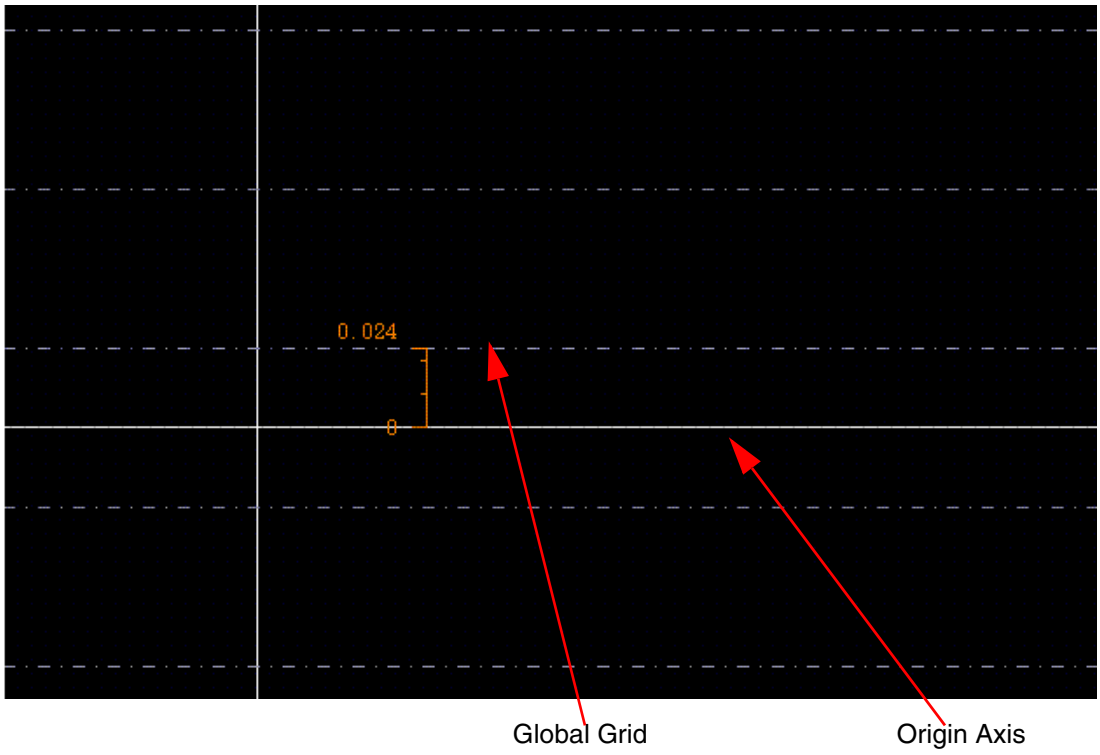
During creation, only the outer edges of the rectilinear PR boundary snap to the grids. During stretch, all the selected edges, inner or outer, of the rectilinear PR boundary snap to the grid.

For editing command, the PR boundary snaps with the snap boundary as reference, if dbGetGlobalGridOffsetReferenceType of the cellview is specified as `boundary`. If the drdEditSmartSnapAllowedWidthSnap environment variable is specified and the placement grid is present in the design, these settings are prioritized for snapping.

© 2023 Cadence Design Systems, Inc.

The PR boundary snaps to the global grid irrespective of global grid offset reference being set to boundary or origin. A virtual hierarchy that has an area boundary snaps similar to the PR boundary.

- Anchoring by Origin Axis: If a PR boundary, snap boundary, or a drawn shape on LPP is not found, then the global grid is anchored to the origin axis.



Global snapping is done similar to local snapping. When you move an object in a design, if the object overlaps the local snap pattern shape, then the object snaps to the local snap pattern shape. However, if no local snap pattern shape is found, then the object snaps to the global snap pattern grid.

Related Topics

snapPatternDefs

Object Snapping to Global Snap Pattern Grid

© 2023 Cadence Design Systems, Inc.

Detail Mode Snapping to Specific Tracks of a Global Grid

By default, devices can be snapped to any track of a global snap pattern definition grid. When snap period-based snapping is enabled, you can snap devices to specific tracks of a global grid by using detail mode snapping. For hierarchical blocks, auto abstract mode snapping is performed, in which a block is moved by the value of the snap period.

Detail Mode Snapping to Specific Tracks of a Global Grid

To enable detail mode snapping of devices to specific tracks:

- Ensure the snap period-based snapping is enabled, by setting the `snapPatternIndexSnappingLayers` environment variable to `t`.
- Define a snap period for the global grid by adding the `snapGridPeriod` constraint in the technology file.
- Define the snappable tracks of the global grid by adding the `leSnapPatternSnappingVerticalGridDetail` or `leSnapPatternSnappingHorizontalGridDetail` property to an instance or a master cellview.

For example, you want that devices can be snapped only to the third track of a period of a global vertical grid named `GFG`, where each period contains four tracks.

You can define the period value as `4` in the following way:

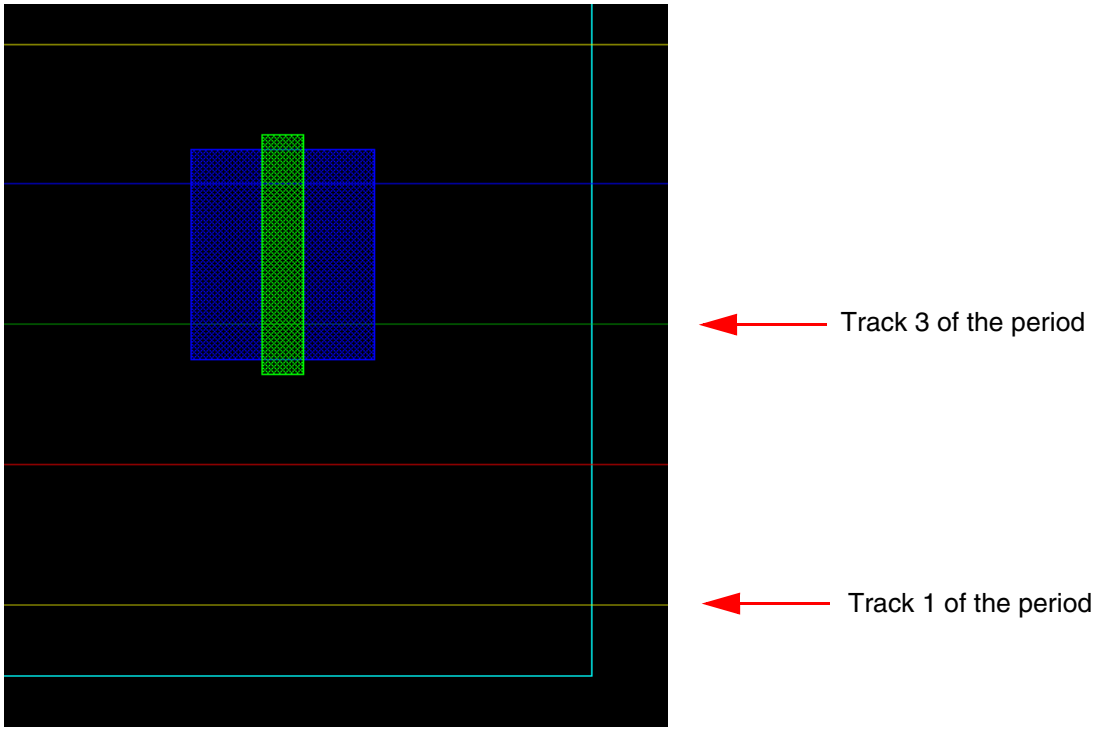
```
spacings(
    snapGridPeriod('snapPatternDef "GFG" 4)
) ;spacings
```

To specify the third track in each period of `GFG` as the valid track for snapping, define the `leSnapPatternSnappingVerticalGridDetail` property on an instance as follows:

```
leSnapPatternSnappingVerticalGridDetail "GFG 3"
```


© 2023 Cadence Design Systems, Inc.

The following image shows a global vertical grid with four tracks in a period, with yellow as the first track. The image shows a device snapped to the third track of the snap period.



When detail mode snapping of devices to specific tracks is on, the snap period constraint is considered during PR boundary creation and editing to prevent instances from being placed off the grid.

Auto Abstract Mode Snapping

For hierarchical blocks, auto abstract mode snapping is performed, in which a block is moved by the value of the snap period. For example, when the snap period for a grid is set to four and you move a block, the block is always moved by four tracks. This ensures that all instances within the block are snapped correctly to their respective valid grids.

A block is snapped in auto abstract mode when the following conditions are met:

- The `snapPatternIndexSnappingLayers` environment variable is set to `t`, which is the default value.
- The `snapGridPeriod` constraint is defined for the global snap pattern definition grid in the technology file.
- The block has a P&R or snap boundary.

© 2023 Cadence Design Systems, Inc.

- The block has at least one or more levels of hierarchy, implying that it contains instances within.
- The block has at least one shape or layer that can snap to the global snap pattern definition grid for which a period constraint is defined.

Auto abstract mode snapping takes higher precedence than the default width spacing pattern (WSP) abstract snapping.

Related Topics

snapGridPeriod

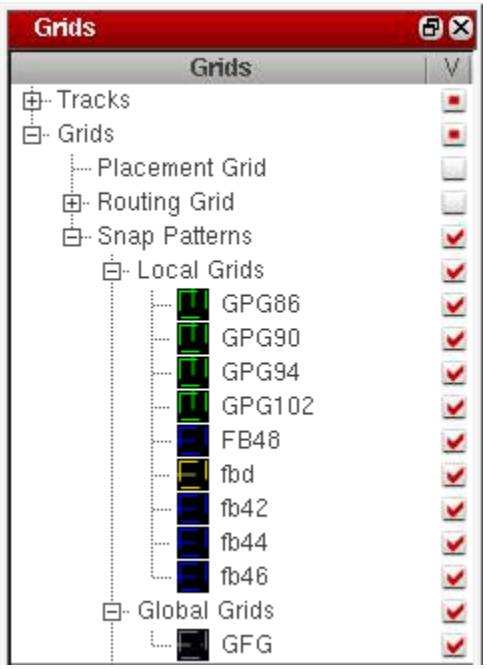
leSnapPatternSnappingVerticalGridDetail

leSnapPatternSnappingHorizontalGridDetail

Object Snapping to Global Snap Pattern Grid

Snap Pattern Display Controls in the Palette Assistant

You can control the visibility of snap patterns, both the local and global grids, through the Grids panel of the Palette assistant.



© 2023 Cadence Design Systems, Inc.

The grid drawing style is controlled through the *Snap Pattern Display* option in the Display Options form.

The selections that you make in the Layers, Objects, and Grids panels of the Palette assistant control the visibility of the shape boundary and grid on the canvas. When the grid visibility is on, the shape boundary and grid are displayed irrespective of the LPP and shape visibility settings.

The following table lists the options selected in the Palette assistant panels and the results displayed on the canvas.

LPP Visibility in the Layers Panel	Shape Visibility in the Object Panel	Grid Visibility in the Grids Panel	Result Displayed on the Canvas
t	t	t	Shape boundary and grid
t	nil	t	Shape boundary and grid
nil	nil	t	Shape boundary and grid
nil	t	t	Shape boundary and grid
t	t	nil	Shape boundary
t	nil	nil	Nothing
nil	nil	nil	Nothing
nil	t	nil	Nothing

Note: Starting with ICADV12.1 ISR10, the layerset file references snap pattern entries by the snap pattern definition name. In ICADV12.1 ISR9 and earlier releases, snap pattern entries were referenced by the layer-purpose pair (LPP) name, instead of the snap pattern definition name. As a result, older layerset files that contain LPPs corresponding to these grids are not read correctly in ICADV12.1 ISR10 and later releases.

Related Topics

Display Options Form

Controlling the Display of Snap Patterns

© 2023 Cadence Design Systems, Inc.

Palette assistant

Instance Placement Modes

In a FinFET-based chip, all instances must be aligned to a fin grid, either to several independent local grids or to a global grid.

For the placement of Pcells and small instances, placement is done by searching the instance master for fin grids. If no fin grids are found, snapping shapes are searched. This functionality is referred to as detail mode placement. Detail mode placement is suitable only for small blocks.

For the placement of large blocks, it is not feasible to search the instance master for fin grids or snapping shapes. It is recommended that you use abstract mode placement, in which an instance snaps to a global snap pattern grid using its snap boundary or PR boundary. If a snap boundary or PR boundary is not found, then the origin axis of the instance is used as the reference point for placement. Abstract mode placement allows fast placement of large layout blocks.

You can switch between detail and abstract modes of instance placement using the snapPatternSnappingMode environment variable. The following example shows how you can enable detail mode placement:

```
envSetVal("layout" "snapPatternSnappingMode" 'cyclic "detail")
```

There may be a situation when both detail mode and abstract mode placements are not appropriate. One such example is when the design has a third-party IP block laid out in a different methodology. For such cases, you can use manual mode placement override.

Related Topics

Instance Snapping

Abstract Mode Placement

Manual Mode Placement Override

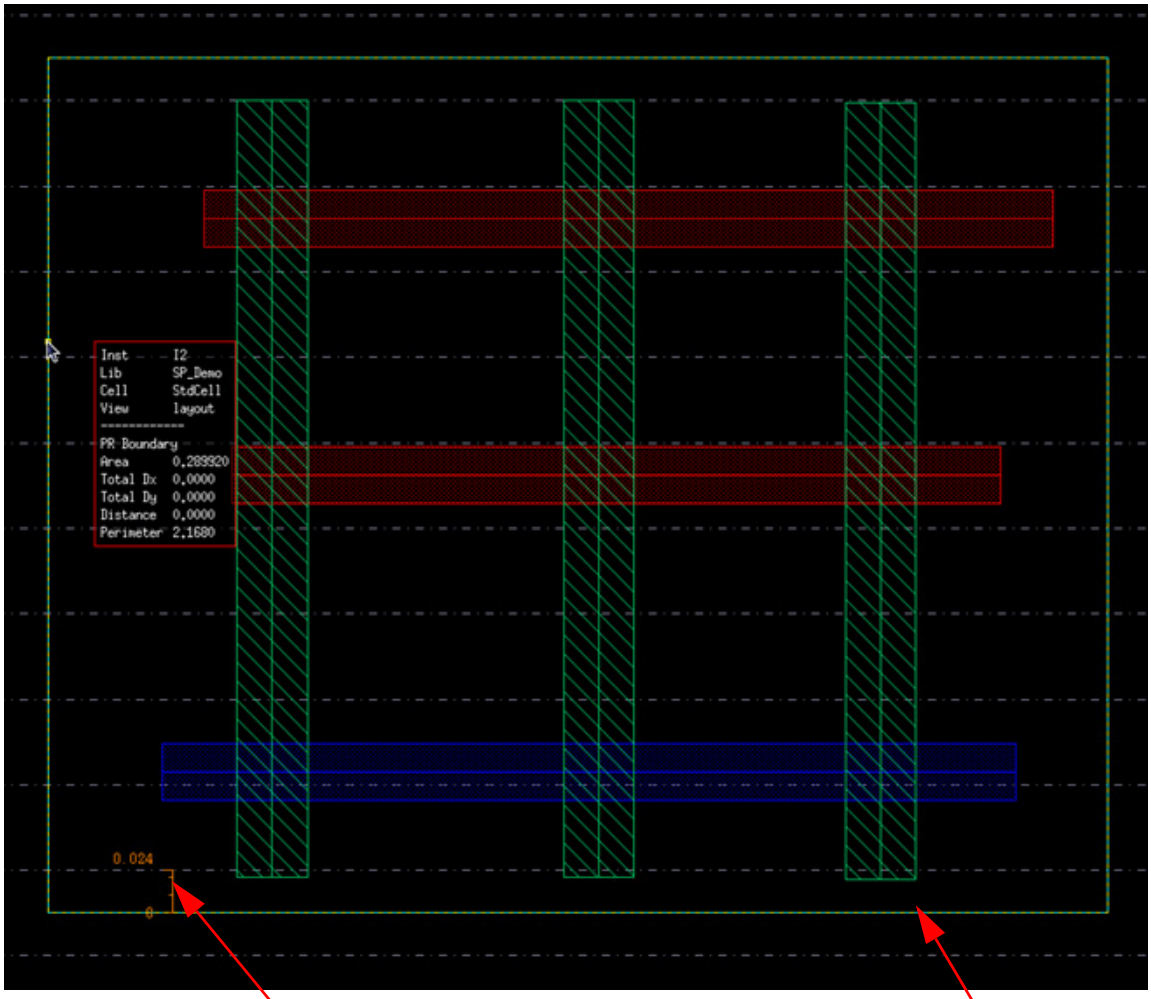
Abstract Mode Placement

When abstract mode placement is enabled and an instance is placed in the design window, Virtuoso does not search for the master for individual shapes. Instead, an instance snaps to

© 2023 Cadence Design Systems, Inc.

a global snap pattern grid using its snap boundary or PR boundary. If no global grid is enabled, then there is no instance snapping.

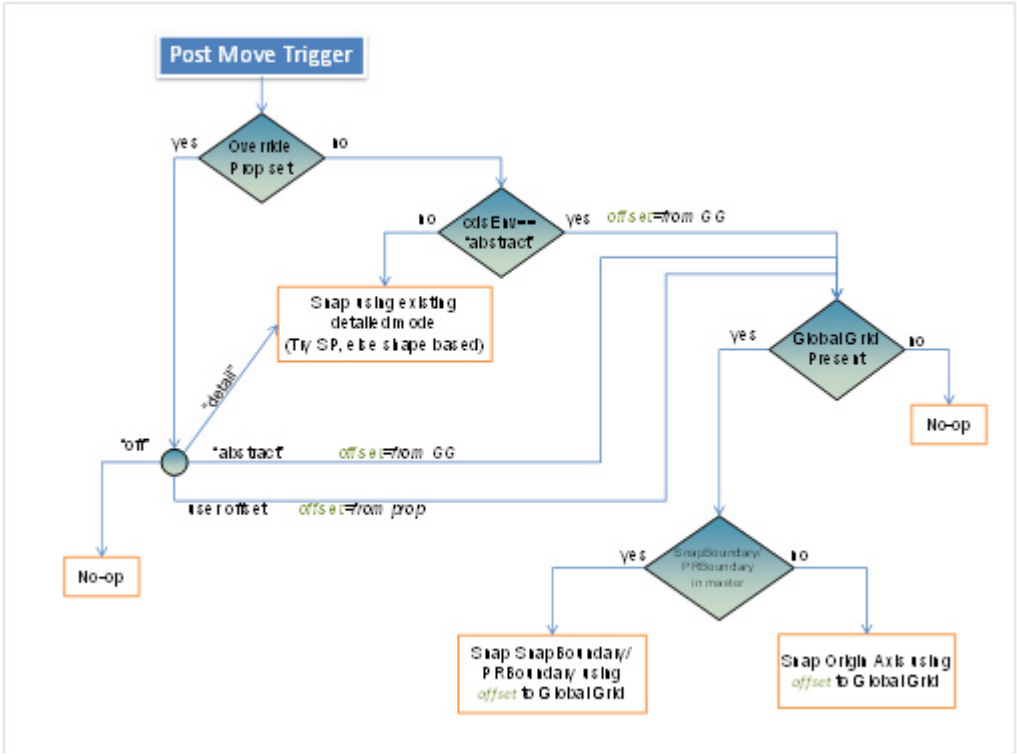
The following figure depicts PR boundary-based snapping to the global grid.



The placement offset is taken as the offset defined in the snap pattern definition of the global snap pattern. If the master contains a snap boundary, it is used for snapping. If no snap boundary is found, the PR boundary is used for placement. Otherwise, the origin axis of the instance is used as the reference point for instance placement.

© 2023 Cadence Design Systems, Inc.

The following figure shows the high-level flow diagram of instance snapping:



Related Topics

Instance Snapping

Manual Mode Placement Override

Manual Mode Placement Override

The manual mode placement override is used in cases where a third-party IP block is used in a design. This mode enables you to manually specify the placement mode or offset on a per-instance or master basis. This is done by setting the leSnapPatternSnapping property on either the instance or the master.

If the `leSnapPatternSnapping` property is set on both the instance and its master, the instance definition overrides the master definition.

© 2023 Cadence Design Systems, Inc.

Related Topics

Instance Snapping

Abstract Mode Placement

Object Snapping in singleTrackCenter Snapping Mode

You can specify the snapping mode for layers as `multiTrackCenter` or `singleTrackCenter`. Snapping modes are specified in the `snappingMode` argument of snapPatternDefs in the technology file.

In `multiTrackCenter`, a shape can be centered on any number of tracks. When the `singleTrackCenter` mode is used for a layer, a shape is snapped only if it can be centered on a single track while meeting the enclosure rule on both sides. This means that a shape would snap only if its dimension is twice the `trackEnclosure` value. In this case, the combined snapping layer bounding box is not considered. If the device contains any shape meeting the dimension rule, the shape is snapped even if the combined layer bounding box does not meet the enclosure on both sides. The `singleTrackCenter` snapping is supported for both local and global grids.

Related Topics

Rectangles Created on a Layer with singleTrackCenter Snapping Mode

Snapping Mode Behavior in Edit Commands

Layer Shapes Exempted from Snapping by Using exceptOverlap

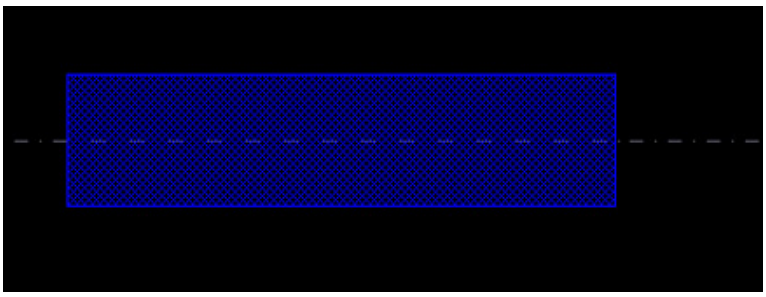
Rectangles Created on a Layer with singleTrackCenter Snapping Mode

In the *Create Rectangle* command, both the points are snapped to the grid by applying negative enclosure for the lower edge and positive enclosure for the upper edge of the rectangle. While drawing rectangles on layer-purpose pairs (LPPs) with the `singleTrackCenter` snapping mode, the points are snapped with respect to the same track, restricting the dimension to twice the enclosure value. You can disable snapping while drawing rectangles using a special mode enabled by using the `Shift` key.

To create a rectangle of single track height, start the *Create Rectangle* command without pressing the `Shift` key. The first point of the rectangle is snapped according to the enclosure value. For vertical grids, the height of the rectangle is fixed after the first point is entered and the mouse can only be moved in the X-direction. Similarly, for horizontal grids, the width of

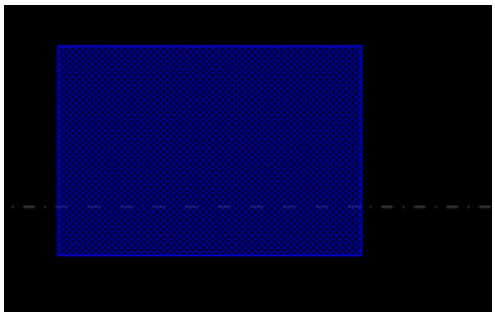
© 2023 Cadence Design Systems, Inc.

the rectangle is fixed after the first point is entered, and the mouse can only be moved in the Y-direction.



A rectangle of single track height

To create a rectangle of arbitrary height with the first point snapped, start the *Create Rectangle* command. The first point is snapped according to the enclosure value. Press the `Shift` key during mouse drag to disable snapping for the second point. A rectangle of arbitrary height is created, as shown below.



A rectangle of arbitrary height

To create a rectangle of arbitrary height without snapping, start the *Create Rectangle* command. Press the `Shift` key during mouse drag to disable snapping for the first point. The second point of the rectangle will also not be snapped.

If the `allowedWidthRanges` constraint is specified for the LPP, the height of the rectangle is a multiple of the discrete width specified. If this constraint is not specified, a rectangle of arbitrary height is created.

Related Topics

Object Snapping in singleTrackCenter Snapping Mode

Creating a Rectangle

© 2023 Cadence Design Systems, Inc.

Snapping Mode Behavior in Edit Commands

Shape Snapping

For edit commands, if a shape is on a snapping layer-purpose pair (LPP) with the `singleTrackCenter` snapping mode, depending on the grid directions, snapping is performed only if the height or width of the shape is equal to twice the enclosure value. For other cases, snapping is disabled.

For shapes on a snapping LPP with the `singleTrackCenter` snapping mode, the value specified in the checkAllSPGridsEnclosures environment variable is ignored and the shape dimensions are verified in all cases.

Instance Snapping

LPPs with the `singleTrackCenter` snapping mode in the instance master have a higher precedence than `multiTrackCenter` LPPs.

- If there is a shape on the `singleTrackCenter` LPP in the master, then the shapes on `multiTrackCenter` LPPs are not snapped.
- The `multiTrackCenter` LPPs are snapped only when there are no shapes on any of the `singleTrackCenter` LPPs.
- Snapping is not performed in a direction if shapes exist on the `singleTrackCenter` LPP, but no shape meets the dimension rule.
- On `singleTrackCenter` LPPs, individual shapes inside the master are snapped instead of the combined bounding box used for `multiTrackCenter` LPPs.
- A shape on `singleTrackCenter` LPP is snapped only if the dimension rule of twice the grid enclosure is met. In case of multiple `singleTrackCenter` LPPs, all the `singleTrackCenter` LPPs are examined until a shape meeting the dimension rule is found. If all the shapes on the `singleTrackCenter` LPP violate the dimension rule, snapping is not performed in that direction.

Related Topics

Object Snapping in singleTrackCenter Snapping Mode

Object Editing in Layout XL

© 2023 Cadence Design Systems, Inc.

Layer Shapes Exempted from Snapping by Using exceptOverlap

You can define the `exceptOverlap` attribute in snapPatternDefs for a snapping layer-purpose pair (LPP) to specify that the shapes on the LPP do not snap if they meet the dimension rule. You can specify this attribute for both `singleTrackCenter` and `multiTrackCenter` LPPs.

For a `singleTrackCenter` LPP, if a shape meeting the dimension rule is found:

- If the shape does not completely overlap any shape on the `exceptOverlap` LPP, it is snapped to the grid.
- If the shape completely overlaps any `exceptOverlap` LPP shape, the shape is ignored for snapping.

If all the shapes on the `singleTrackCenter` LPP violate the dimension rule or completely overlap the `exceptOverlap` LPP, snapping is not performed in this direction.

For a `multiTrackCenter` LPP, snapping is performed only if the combined bounding box of shapes does not completely overlap any shape on the `exceptOverlap` LPP.

You can control the except-overlap snap mode to be set to `full` or `partial` using the exceptOverlapSnapMode environment variable.

Related Topics

Object Snapping in singleTrackCenter Snapping Mode

Snap Pattern Display Packets

You can define display packets for snap patterns to control the display of tracks and fins, such as the stipple patterns, line styles, fill colors, outline colors, and fill styles of tracks and fins. To define a display packet, you can add the `drSetPacket` entry in the `drDefinePacket` section of the `display.drf` file. You can also use the Display Resource Editor to define a display packet.

You can define separate display packets for tracks and fins of a snap pattern.

- Display information for snap pattern tracks: You can define a display packet by using the *`snapPatternPacket`*`_snapPatternTracks` naming convention. Where, *`snapPatternPacket`* refers to the name of the packet used by the layer-purpose pair defined in the snap pattern definition in the technology file, and `snapPatternTracks` is the suffix you add to that name. For example, if the snap pattern packet is named

© 2023 Cadence Design Systems, Inc.

`M1_finPurpose`, you need to define the display packet as `M1_finPurpose_snapPatternTracks`, as shown in the following sample:

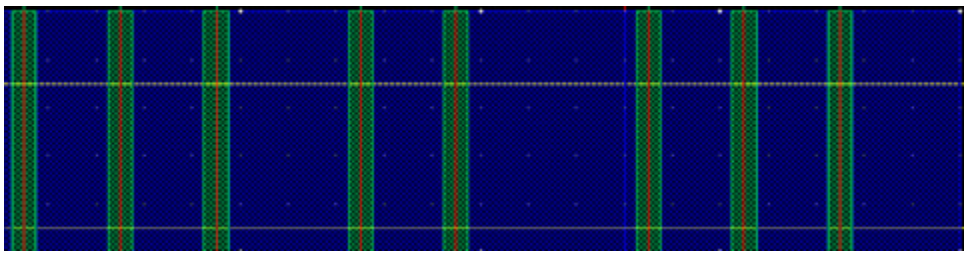
```
drSetPacket("display" "M1_finPurpose_snapPatternTracks" "x" "solid" "red"
"red" "outlineStipple")
```

A `snapPatternTracks` display packet defines the color of the centerline of tracks.

- Display information for snap pattern fins: You can define a display packet by using the *`snapPatternPacket`*`_snapPatternFull` naming convention. For example:

```
drSetPacket("display" "M1_finPurpose_snapPatternFull" "x" "solid" "green"
"green" "outlineStipple")
```

The following image shows the fins in green and the track centerline in red.



The display packets you define by using `snapPatternTracks` and `snapPatternFull` are applied to all tracks and fins in a snap pattern. However, you can create additional packets if you want to control the display of alternate tracks and fins in a snap pattern.

- To control the display of every alternate snap pattern track, define a display packet by using the *`snapPatternPacket`*`_snapPatternTracksAlt` naming convention. For example: `M1_finPurpose_snapPatternTracksAlt`

```
drSetPacket("display" "M1_finPurpose_snapPatternTracksAlt" "x" "solid" "green"
"green" "outlineStipple")
```

- To control the display of every alternate snap pattern fin, define a display packet by using the *`snapPatternPacket`*`_snapPatternFullAlt` naming convention. For example: `M1_finPurpose_snapPatternFullAlt`

```
drSetPacket("display" "M1_finPurpose_snapPatternFullAlt" "x" "solid" "red"
"red" "outlineStipple")
```

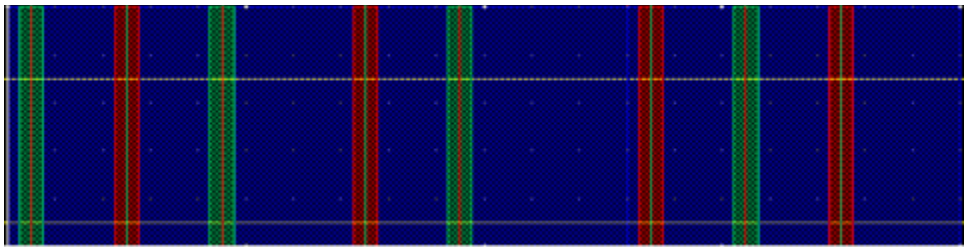
The image shows how the alternate snap pattern tracks and fins appear in different colors when you define the following display packets in `display.drf`:

```
drSetPacket("display" "M1_finPurpose_snapPatternTracks" "x" "solid" "red" "red"
"outlineStipple")
drSetPacket("display" "M1_finPurpose_snapPatternFull" "x" "solid" "green" "green"
"outlineStipple")
drSetPacket("display" "M1_finPurpose_snapPatternTracksAlt" "x" "solid" "green"
"green" "outlineStipple")
```


© 2023 Cadence Design Systems, Inc.

```
drSetPacket("display" "M1_finPurpose_snapPatternFullAlt" "x" "solid" "red" "red"
"outlineStipple")
```

The `snapPatternTracksAlt` and `snapPatternFullAlt` display packets are used for every other track, starting at the reference track.



Related Topics

drSetPacket

Display Resource File Development

Using Display Resource Editor

© 2023 Cadence Design Systems, Inc.

Snap Pattern Display Based on Track Index

You use display packets to control the display of track centerlines and fins in a snap pattern. You can control the display based on track index across periods in a snap pattern. For example, you can choose to display the centerline of every second track in a snap period in yellow and the fin of every third track in orange.

Display of Track Centerlines

To control the display of the centerline of specific tracks in a snap pattern, define a display packet by using the following naming convention:

*snapPatternPacket*_snapPatternTracks_trackIndex_*n*

Where:

- *snapPatternPacket* refers to the name of the display packet used by the layer-purpose pair defined in the snap pattern definition in the technology file.
- snapPatternTracks_trackIndex is the suffix you add to that name.
- *n* refers to the track number for which you want to define the display in a snap period.

The following example defines how the centerline of the second track in a snap period is displayed when the snap pattern packet named M1_finPurpose is used.

```
drSetPacket("display" "M1_finPurpose_snapPatternTracks_trackIndex_2" "x" "solid"
"yellow" " yellow " "outlineStipple")
```

The image in this topic shows the centerline of the second track in yellow. Assuming the snap period of the snap pattern is set to five, the centerline of the seventh track is also displayed in yellow.

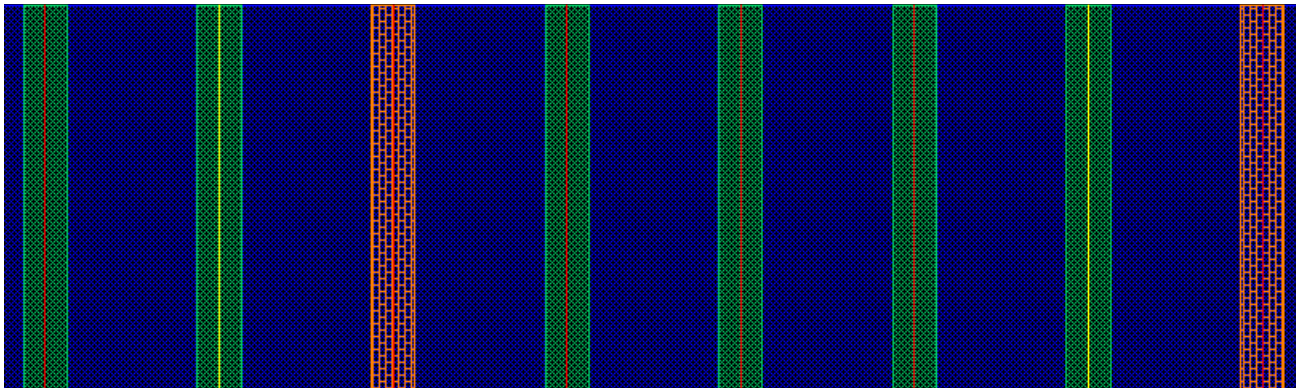
Display of Track Fins

To control the display of a specific track fin, define a display packet by using the *snapPatternPacket*_snapPatternFull_trackIndex_*n* naming convention. The following example defines the fin display of the third track in a snap period when the snap pattern packet named M1_finPurpose is used.

```
drSetPacket("display" "M1_finPurpose_snapPatternFull_trackIndex_3" "brick"
"solid" "orange" "orange" "outlineStipple")
```


© 2023 Cadence Design Systems, Inc.

The image shows that the fin of the third track is displayed in orange with brick stipple. Assuming the snap pattern period is set to five, the fin of the eighth track is also displayed in orange.

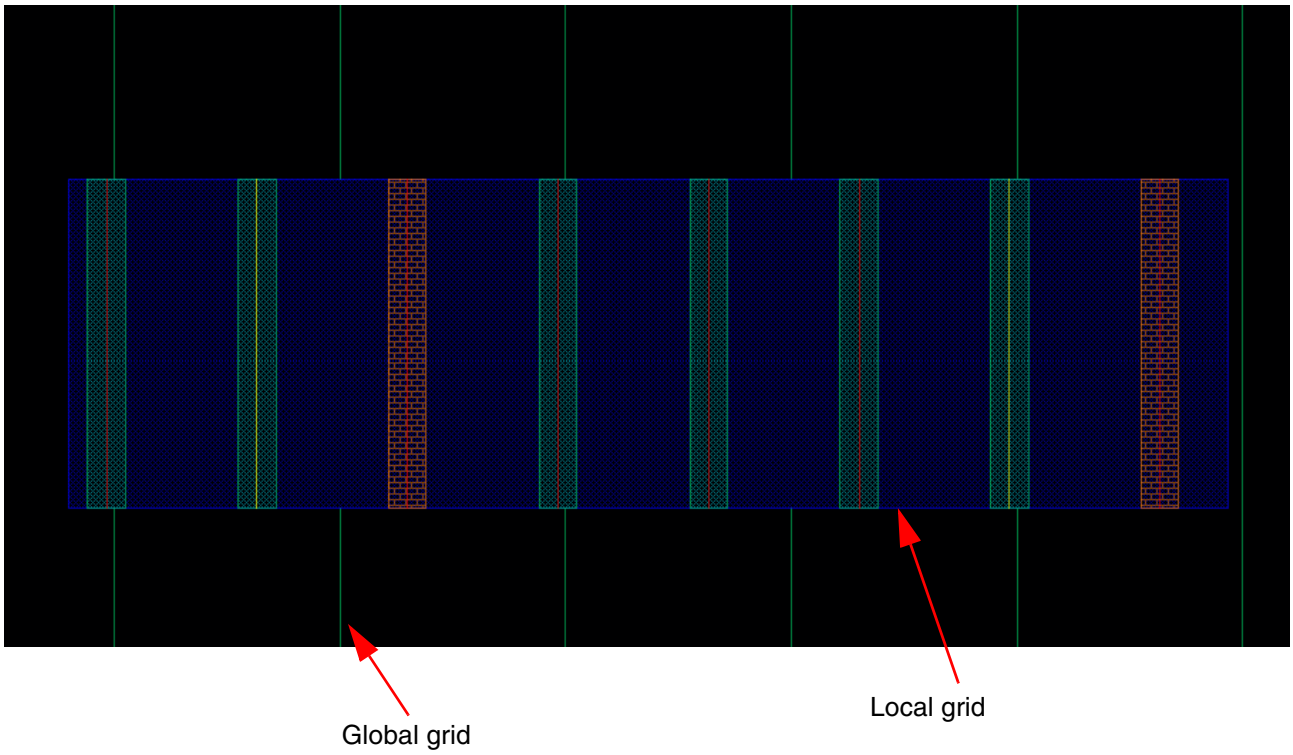


These display packets take effect when the snapPatternDisplayTrackIndex environment variable is enabled and the `snapGridPeriod` constraint defining the snap period is set for the snap pattern definition in the technology file.

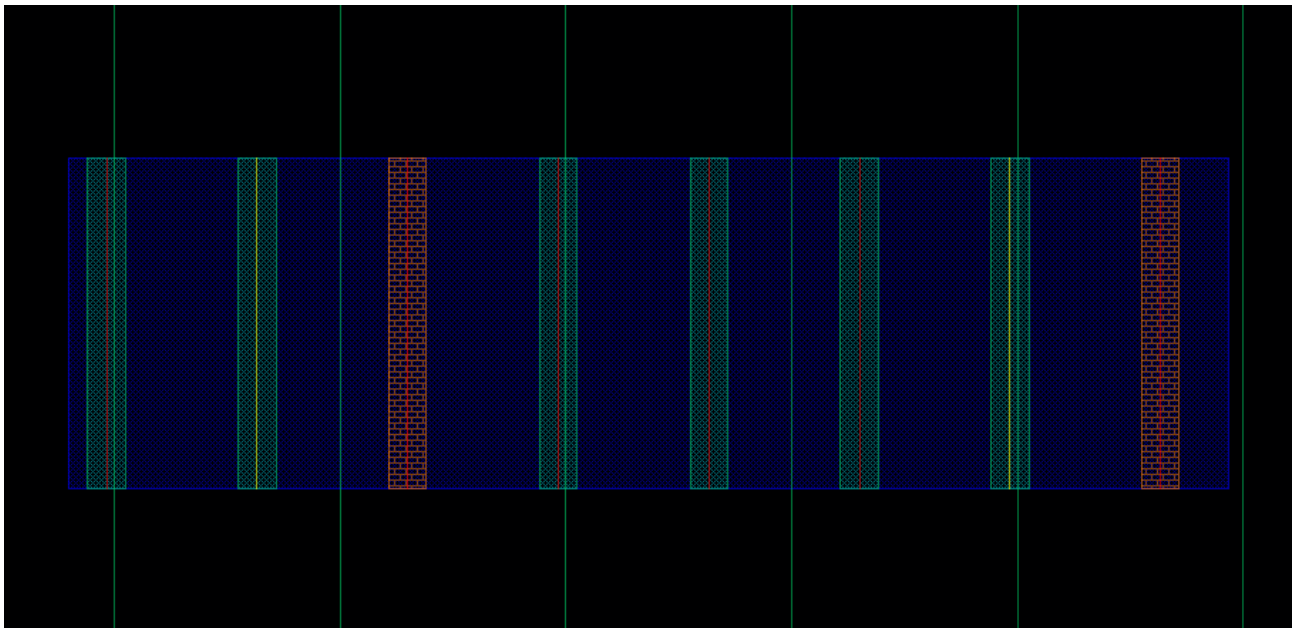
The `techLayerPurposePriorities` section in the technology file defines the layer-purpose pair (LPP) priority, from lowest to highest. By default, a local snap pattern grid having common snapping LPPs overlays a global snap pattern grid. However, the global grid overlays a local grid if the local grid LPP has a lower priority than the global grid LPP, when `snapPatternDisplayTrackIndex` is enabled.

© 2023 Cadence Design Systems, Inc.

The following image shows a local snap pattern grid overlaying a global snap pattern grid.



The following image shows a global snap pattern grid overlaying a local snap pattern grid.



© 2023 Cadence Design Systems, Inc.

Related Topics

snapGridPeriod

Snap Pattern Display Packets

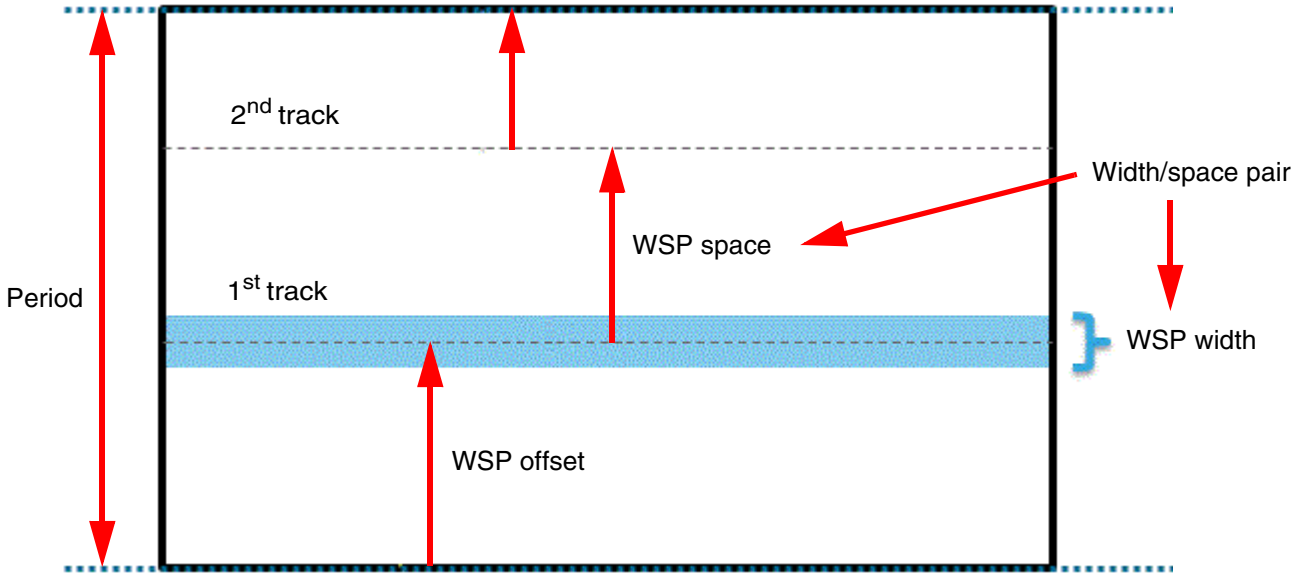
© 2023 Cadence Design Systems, Inc.

14

Width Spacing Patterns in Layout Editor

A width spacing pattern (WSP) definition is a sequence of width and spacing values that define the track patterns to which the shapes in a design are aligned. WSPs are an advanced form of snap pattern definitions and can have multiple width and space combinations. The width spacing pattern definitions used with widthSpacingSnapPatternDefs are stored either in the technology database or the design database.

The figure below illustrates that each width spacing pattern is a sequence of width-space pairs. Patterns are specified as width and space pairs. Groups of width and space pairs can be repeated. The first track of the pattern is anchored to the period grid using an offset.



Related Topics

Controlling the Display of Width Spacing Patterns

Object Snapping Behavior in Width Spacing Patterns

Object Snapping to WSPs During Editing

© 2023 Cadence Design Systems, Inc.

Pin Snapping to Width Spacing Patterns

Controlling Transparency of Tracks With Alpha Blending

Interactive Snapping

Controlling the Display of Width Spacing Patterns

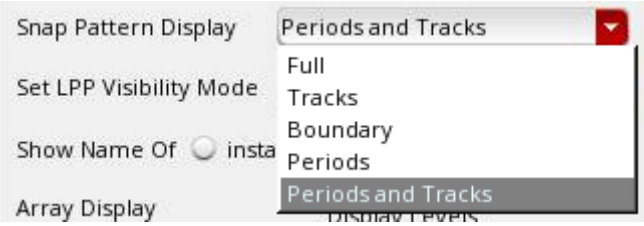
You can control the display of width spacing patterns (WSPs) using the *Snap Pattern Display* option in the Display Options form. In addition to boundary and tracks, you can display periods for WSPs.

To control the display of WSPs:

1. In the layout window, choose *Options – Display*.

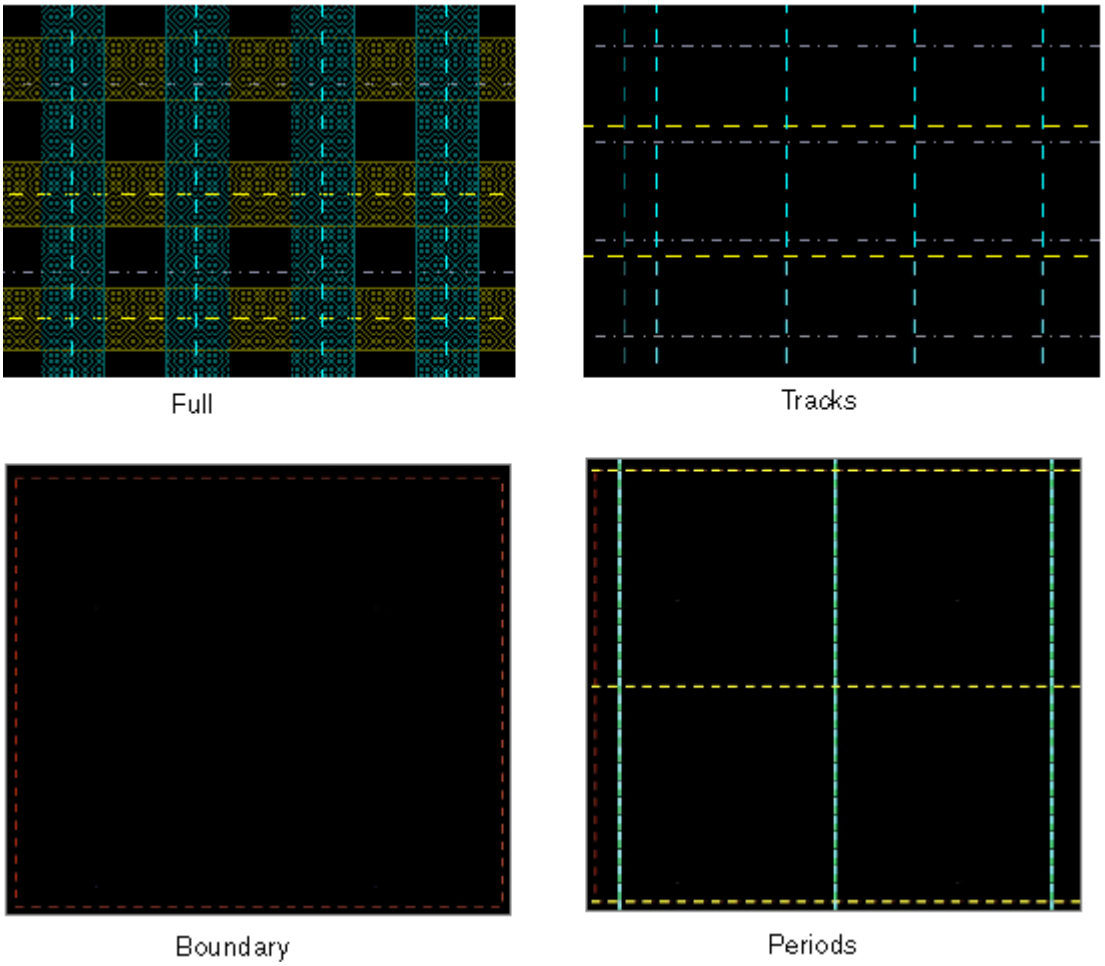
   The Display Options form opens. The *Snap Pattern Display* list shows the snap pattern displays available.

2. Specify whether you want to display tracks, boundary, periods, or all of these for WSPs. From *Snap Pattern Display* list, select any of the following: *Full*, *Tracks*, *Boundary*, *Periods*, or *Periods and Tracks.*



© 2023 Cadence Design Systems, Inc.

The following figure shows a few examples.



3. Click *OK*.

You can use the `leCycleSnapPatternDisplay` SKILL function to cycle through the different snap pattern display types. You set the default value using the new snapPatternDrawingStyle environment variable.

Related Topics

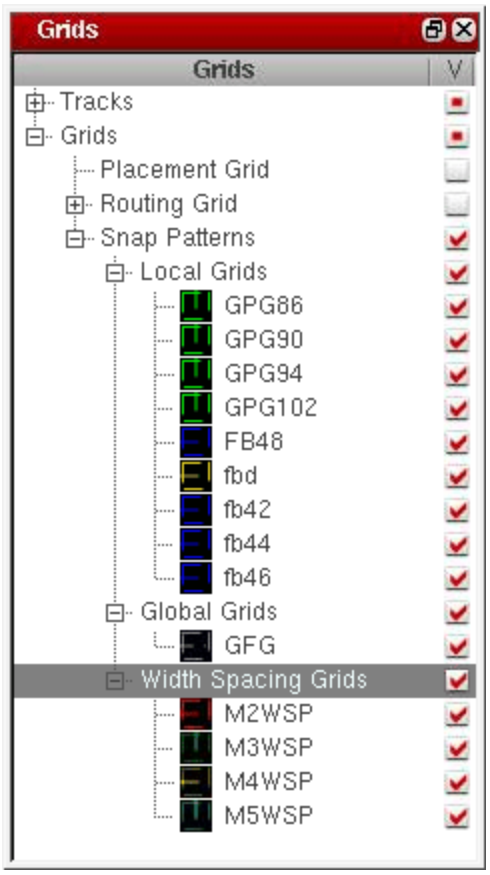
Width Spacing Patterns in Layout Editor

Display Options Form

© 2023 Cadence Design Systems, Inc.

Display Controls for Width Spacing Patterns in the Palette Assistant

You can control the visibility of snap patterns, which include local grids, global grids, and width spacing grids, through the Grids panel of the Palette assistant.



The layerset file references snap pattern entries by the snap pattern definition name. For example, the local grid named *fb46* and WSP grid named *M5WSP*, shown in the figure above, are listed as follows in the layerset file:

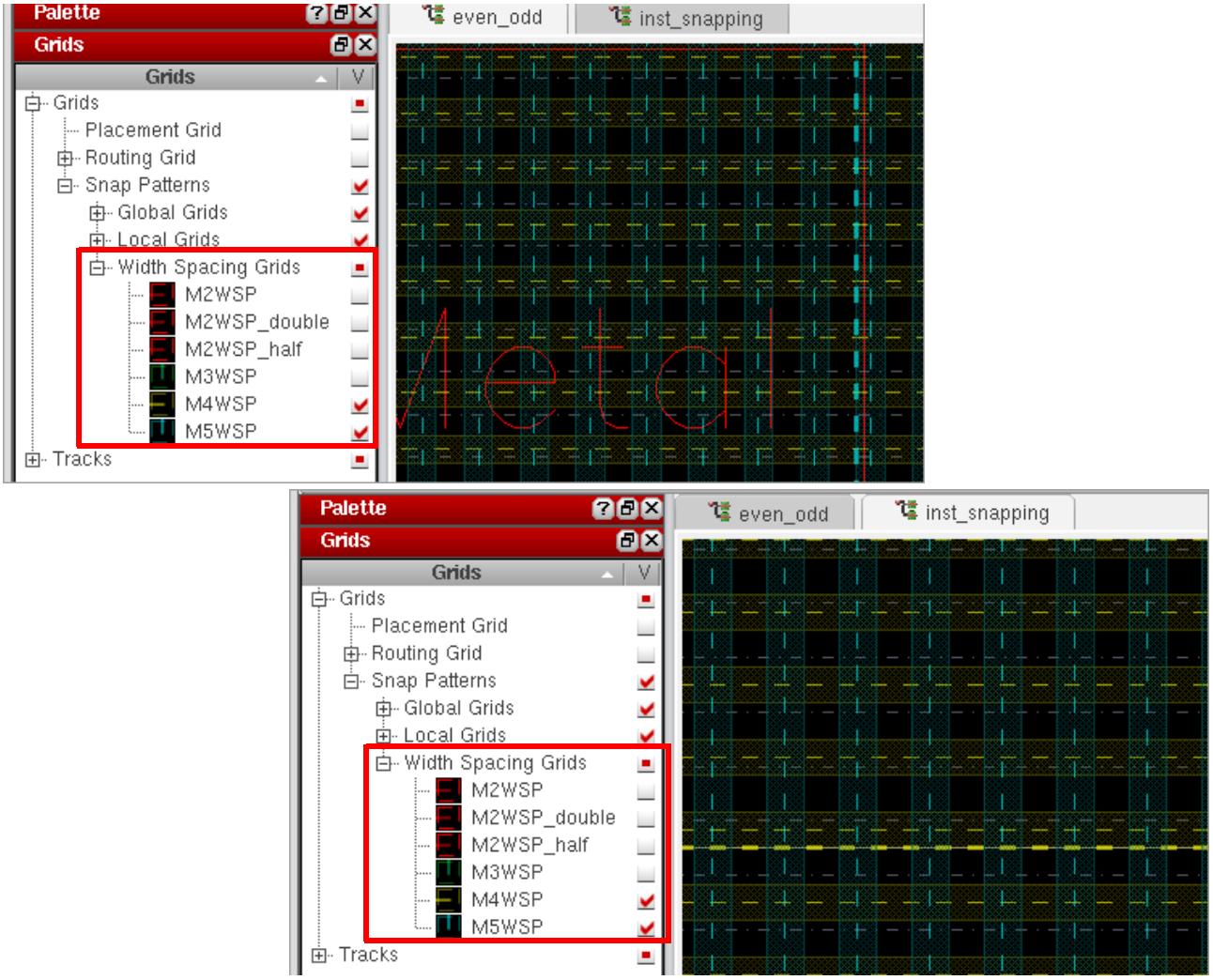
```
GridLayer Local_Grids;fb46 t NA

GridLayer Width_Spacing_Grids;M5WSP t NA
```

The visibility settings of WSP grids are cellview specific. Therefore, if you have two different cellviews open and you modify the visibility setting of a WSP grid in one cellview, the visibility setting of this WSP grid is not affected in the other cellview.

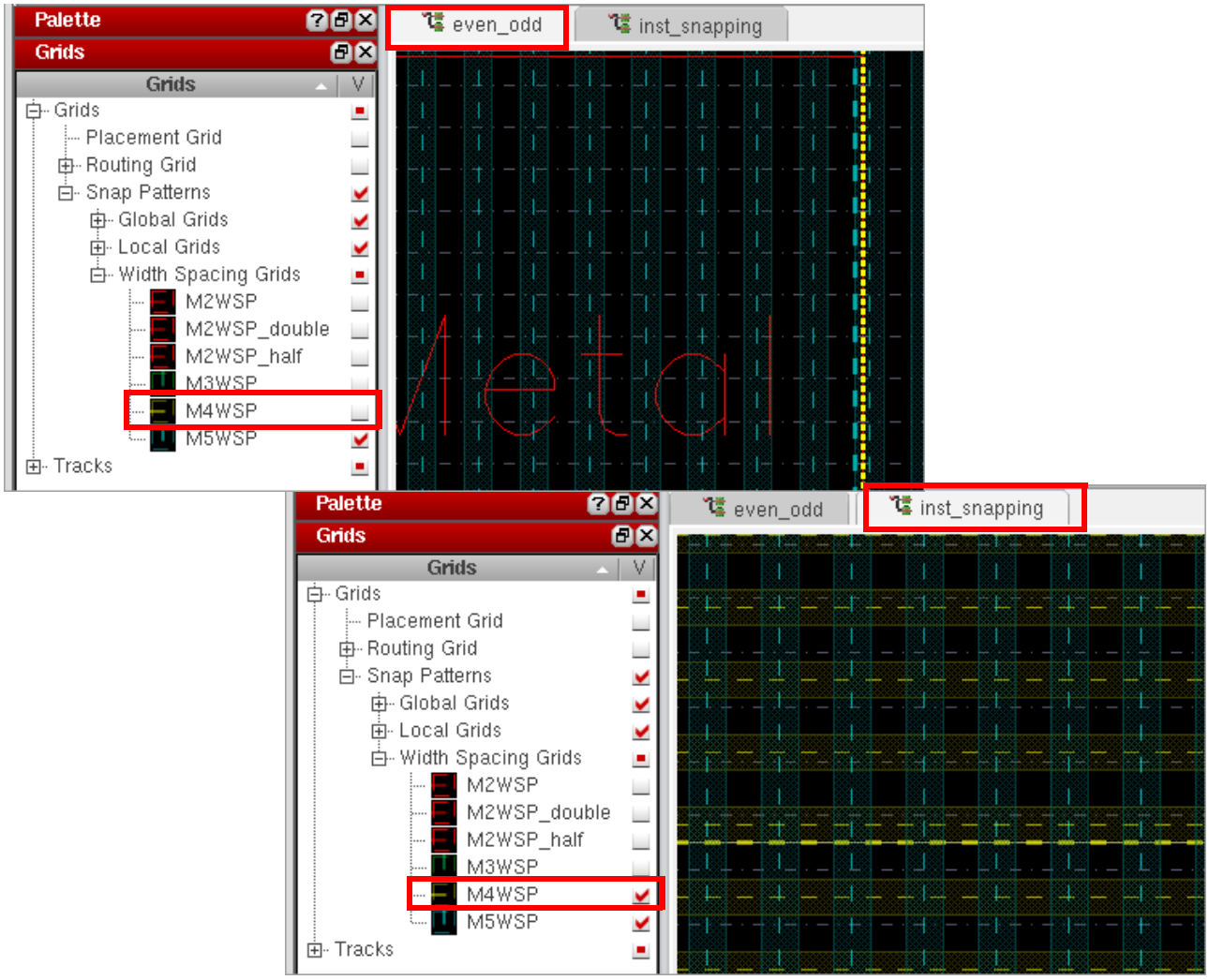
© 2023 Cadence Design Systems, Inc.

In the following example, the visibility settings of WSP grids in the two open cellviews are identical.



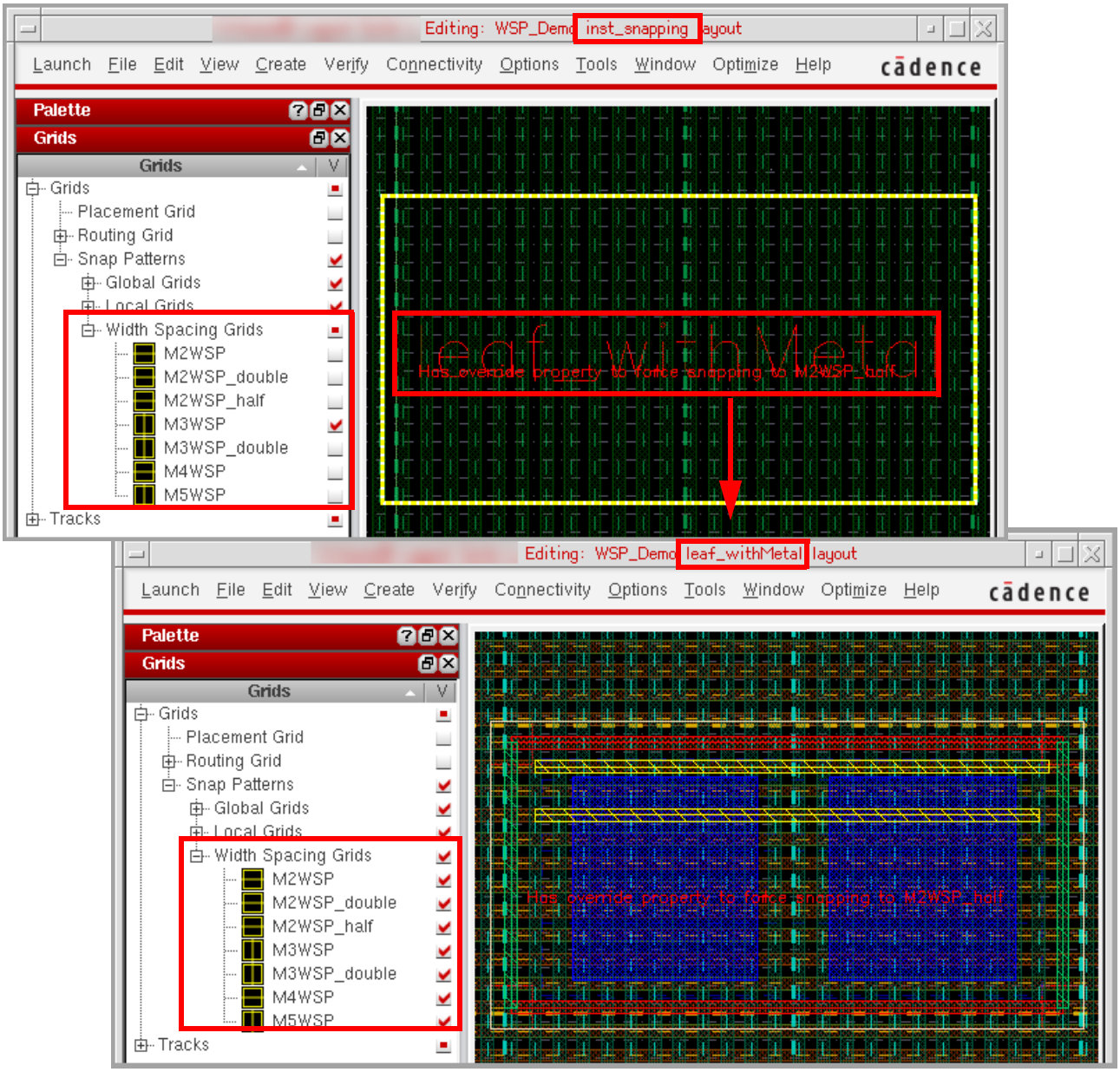
© 2023 Cadence Design Systems, Inc.

If you now turn off the visibility of WSP grid *M4WSP* in cellview `even_odd`, you will find that the visibility setting of WSP grid *M4WSP* in cellview `inst_snapping` remains unchanged.



© 2023 Cadence Design Systems, Inc.

However, while editing a cellview, if you run the *Edit In Place* command or descend into a different cellview, the visibility settings of WSP grids are refreshed to reflect the settings for the cellview being edited, as shown in the example below.



Related Topics

Width Spacing Patterns in Layout Editor

Palette assistant

© 2023 Cadence Design Systems, Inc.

Controlling the Display of Width Spacing Patterns

Object Snapping Behavior in Width Spacing Patterns

Objects are snapped to width spacing patterns (WSPs) depending on various conditions. For example, a pin that has the snapping layer-purpose pair that cannot be snapped to a snap pattern is snapped to a width spacing pattern. WSP snapping is only supported for rectangular pins and not polygon pins. Region snapping is supported only for rectangles and polygons. Instances are snapped to the WSP default abstract mode placement when certain properties have not been set.

- Pin Snapping
- Region Shape Snapping
- Region Group Snapping
- Instance Snapping in Abstract Mode Placement

Pin Snapping

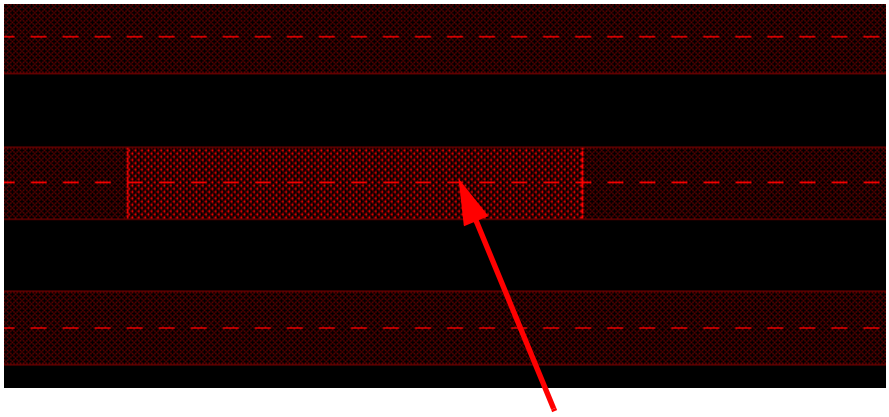
A pin that has the snapping LPP that cannot be snapped to a snap pattern will be snapped to a width spacing pattern (WSP). A pin will either snap to a snap pattern or a WSP, but not both.

If a pin shape can be snapped to a snap pattern, there will be no change in the behavior. Otherwise, it will try to snap to WSP. The center of the pin is snapped to the center of the closest track. WSP snapping is only supported for rectangular pins and not polygon pins.

For snapping a pin to a WSP region shape, the applied pattern is considered; and for a global WSP, the default active pattern is considered.

© 2023 Cadence Design Systems, Inc.

If the target track width is greater than the pin width, the pin width is updated to match the track width. However, if the track width is less than the pin width, the pin is updated based on the value of the `minWidth` constraint defined in the technology file.



Pin snapped to WSP

A pin inside an instance matches a WSP track if its width and mask color match that of the WSP track.

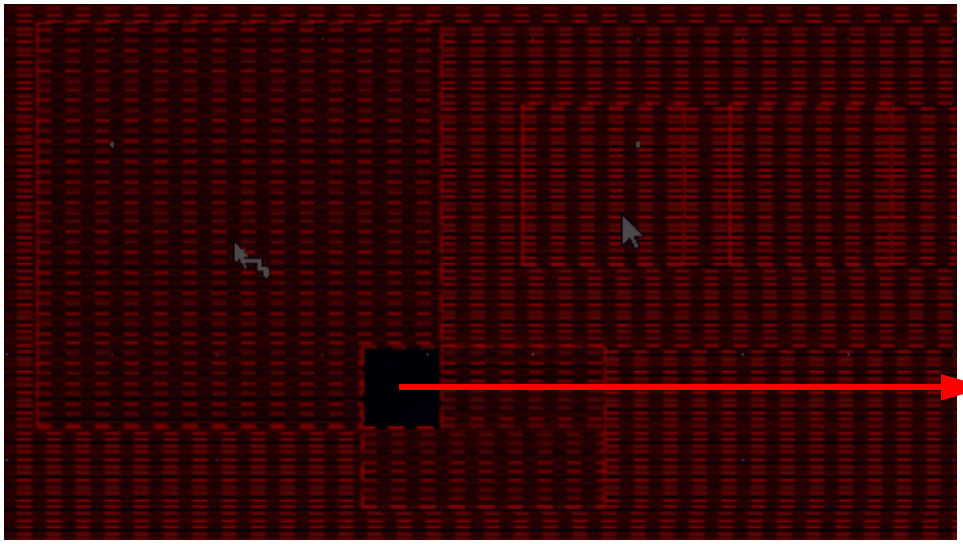
Pin Snapping for Overlapping Regions

There can be multiple overlapping regions. The snapping behavior for various scenarios is described below:

- **Region R1 partially overlaps another region R2**

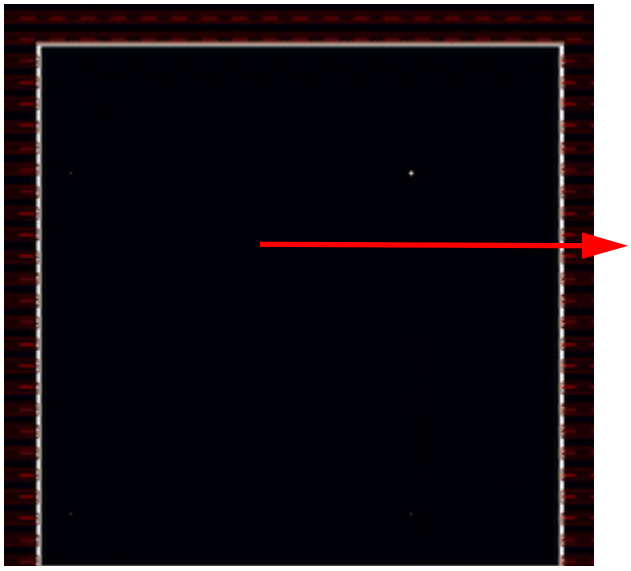
© 2023 Cadence Design Systems, Inc.

The region R1 has `WSP1` as the active pattern and the region R2 has `WSP2` as the active pattern. The overlap area of the two regions does not have any pattern available and displays a blank area. Also, a pin cannot be snapped inside the overlapping area.



- **Two regions with the same size overlap each other**

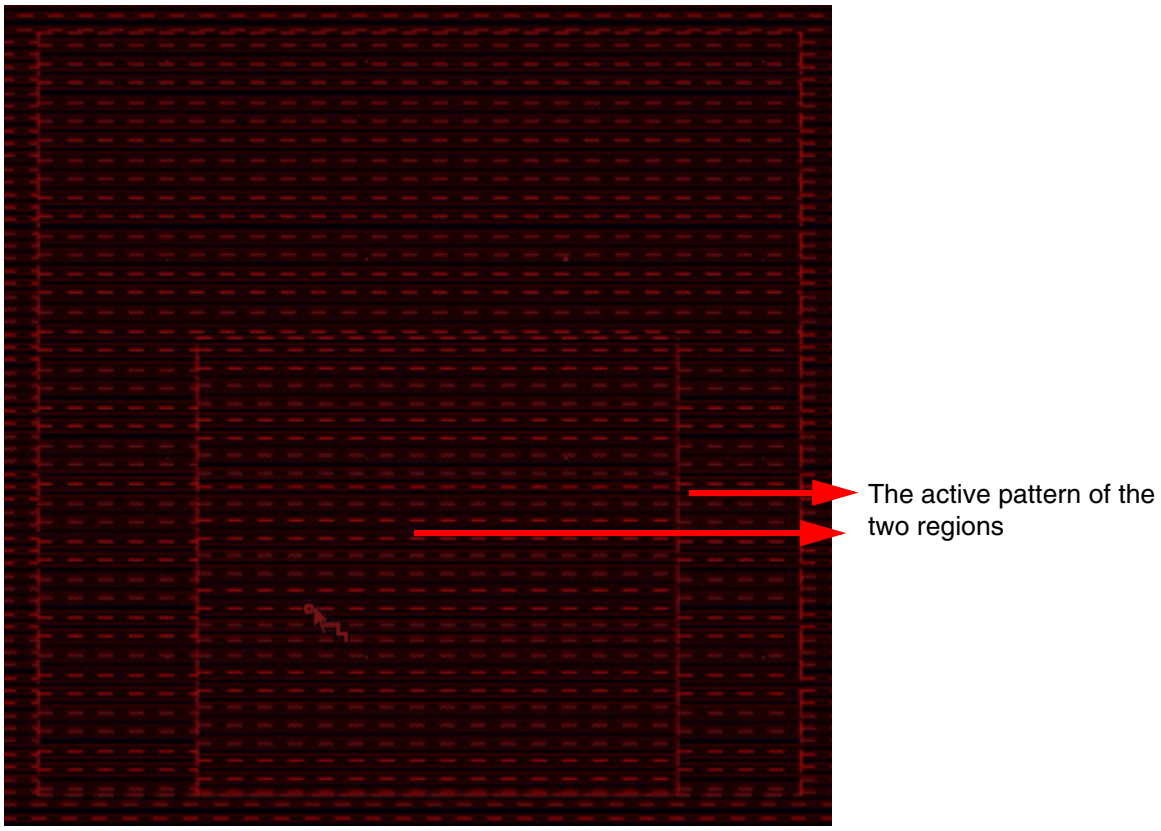
  The two regions R1 and R2 have the same size. The region R1 has `WSP1` as the active pattern and the region R2 has `WSP2` as the active pattern. In this case, no snapping is possible inside the overlapping region and pins cannot be created inside this region.



- **Region R1 is fully enclosed in region R2**

© 2023 Cadence Design Systems, Inc.

Region R1 has `2xWidth` as the active pattern and region R2 has `WSP1` as the active pattern. The tracks in the innermost region are effective for pin snapping. The tracks of R1 are used for snapping pins in the overlapping area and in the remaining area the tracks of R2 are used for snapping.



In WSP detail mode snapping, if your design has pins, they are used for snapping. However, if pins are not available in the design, rectangles, pathSegs, two-point paths, and rectangular polygons are used for snapping.

Also, in WSP detail mode snapping, all wspDefs are searched for in case of local region and non-active wspDefs are filtered in case of global grids.

If the snapRectToWSPGrid environment variable is set, rectangles snap to the WSP grid, similar to pins.

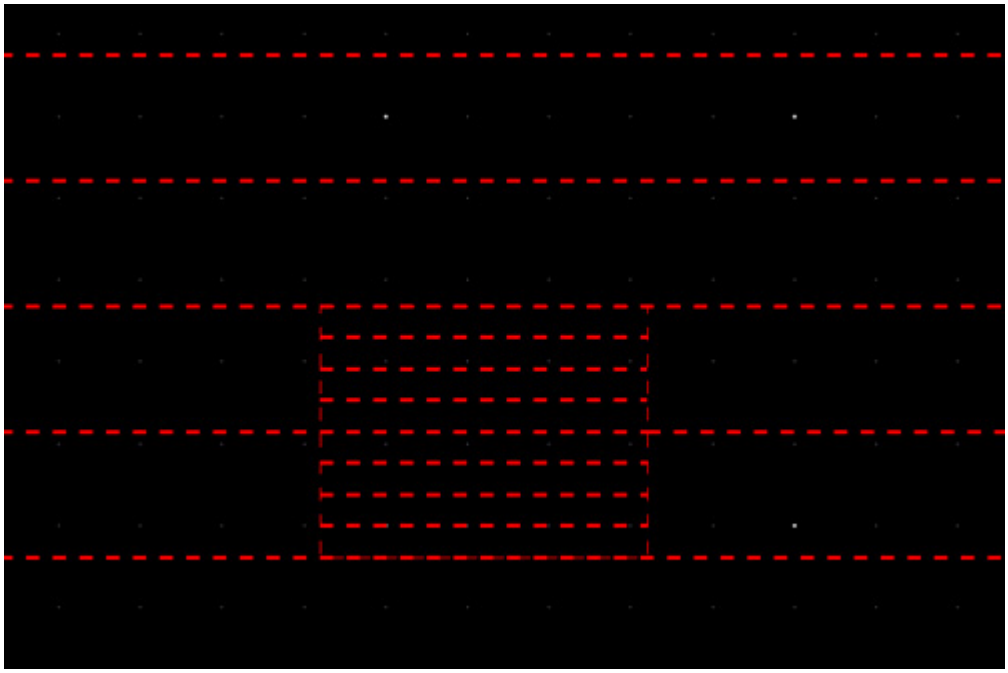
Related Topics

Object Snapping Behavior in Width Spacing Patterns

© 2023 Cadence Design Systems, Inc.

Region Shape Snapping

A region is any shape drawn on a width spacing pattern layer-purpose pair (WSP LPP). The `widthSpacingSnapPatternDef` of the region may or may not be defined as part of the global WSP grids. You can create a region shape on a non-global WSP grid with a period different than the period of the global grid.



WSP drawn on a non-global WSP grid

The snapping LPP specified for `widthSpacingSnapPatternDef` of the region is used to find the global WSP grid to which the region shape snaps. If there are multiple global grids to which the snapping layers of region `widthSpacingSnapPatternDef` can snap, the region is snapped to the first matching global grid.

For vertical WSPs, the region shape is snapped along the X-direction. For horizontal WSPs, the region shape is snapped along the Y-direction.

Region snapping is supported only for rectangles and polygons. If the WSP LPP is also the snapping LPP of a `snapPatternDef`, the region shape is snapped only to the WSP grid.

You can define the WSP grids in the technology file or you can draw them in the design.

Related Topics

Region Group Snapping

Object Snapping Behavior in Width Spacing Patterns

© 2023 Cadence Design Systems, Inc.

Region Group Snapping

A region group is an `oaFigGroup` that contains only snap pattern shapes or width spacing pattern (WSP) region shapes. The layer-purpose pair of each shape in the `oaFigGroup` has a `snapPatternDef` or `widthSpacingSnapPatternDef`.

The reference region shapes along each direction are determined based on the region with the largest period of the snapping `widthSpacingSnapPatternDef` and are snapped similar to region shape snapping. If there is no region along a direction in the group, snapping does not occur along that direction.

If a region group contains both WSPs and snap patterns, WSP snapping takes precedence over snap patterns.

Related Topics

Region Shape Snapping

Object Snapping Behavior in Width Spacing Patterns

Instance Snapping in Abstract Mode Placement

The abstract mode placement allows fast placement of large layout blocks. The width spacing pattern (WSP) default abstract mode placement is performed in the following cases:

- If the leSnapPatternSnappingVerticalGrid or leSnapPatternSnappingHorizontalGrid properties are not specified.
- If the snapPatternSnappingMode environment variable is not set to `abstract`.
- If the leSnapPatternSnapping property has not been specified.
- If the instance master has a PR boundary or a snap boundary and a shape that can snap to the WSP grid.

In the above cases, the following tasks are performed to snap to the default WSP grid in either direction:

- The largest period is identified.
- For the matching period, the definition with the highest mask number snapping layer is identified.
- In case there are multiple definitions on a snapping layer with the largest period, sorting is done based on the definition names.

© 2023 Cadence Design Systems, Inc.

- If no WSP is found along a given direction, the snap pattern global grid is used for abstract mode placement.

The following is an example of an advanced scenario when abstract mode placement is performed for WSPs if you specify the `leSnapPatternSnappingVerticalGrid` or `leSnapPatternSnappingHorizontalGrid` property.

The offset overrides can be specified for WSP abstract mode snapping using the `leSnapPatternSnapping` property only if one of the properties, `leSnapPatternSnappingVerticalGrid` or `leSnapPatternSnappingHorizontalGrid`, has been set. If you do not specify either of these properties and only specify the offset overrides using the `leSnapPatternSnapping` property, the largest grid in the constraint in either direction is considered for snapping. It is assumed that you want to specify an offset override only when you override the grid.

In this case, the grid that you have specified is accepted without any further check. If the `leSnapPatternSnapping` property has an offset override, it is applied based on the existing rules that apply to this property. If only one direction grid has been specified, the other grid is determined based on the default algorithm.

When leSnapGridHorizontal and leSnapGridVertical abstract override properties have been specified, the object closest to pointer is used as the reference object for snapping. However, in case of modgens, the object that has the abstract override properties specified is used as the reference object for snapping.

Related Topics

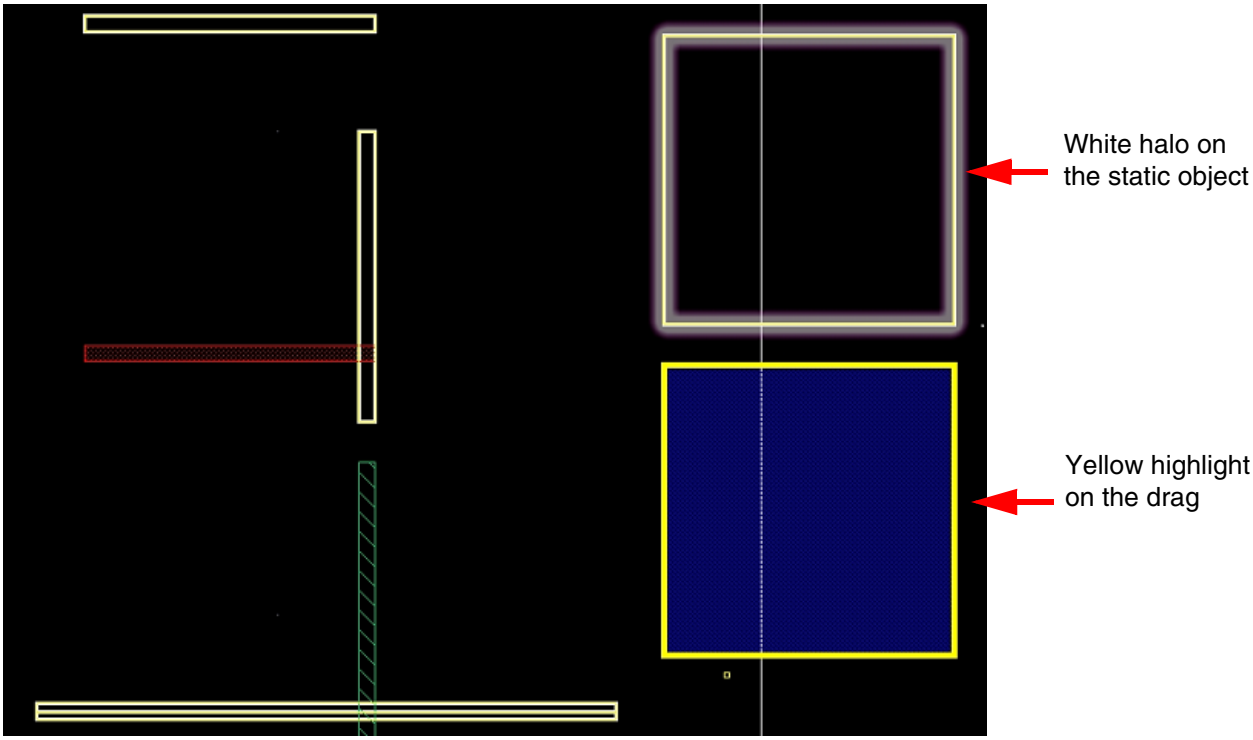
Object Snapping Behavior in Width Spacing Patterns

Object Snapping to WSPs During Editing

The `Move`, `Copy`, and `Stretch` commands support snapping for width spacing pattern layer-purpose pairs (WSP LPPs). If the selected set has multiple objects, the object closest to the reference point is snapped and all the objects in the selected set are moved by the same transform to maintain their relative spacing. The snapping object, closest to the pointer, in the selected set is the reference. In case there are multiple shapes closest to the pointer, the shape with the higher mask number takes precedence. If there is no shape closest to the pointer, the precedence that is followed is the closest instance, row, boundary, and then

© 2023 Cadence Design Systems, Inc.

blockage. As illustrated in the figure, in case multiple objects are selected, this reference object is highlighted on the canvas.



The object is snapped depending on its type: pin, region, or instance. The pointer is snapped to the snapping grid as you move the mouse. It gives a visual indication of how the objects will be placed after the editing command is complete.

- For *Move* and *Copy* commands, the selected pathSeg is resized if no other object is selected and in the Layout Editor Options form, the *Wires* option in the *Wire Snapping* section is set to *Snap Pattern*.
- For *Stretch* command for partially selected objects, the selected edge is snapped to the snapping grid. The edges that are not selected are not modified and these edges could be off-grid.
- For *Chop* and *Yank* commands, the WSP-specific information, like the definition name, present on the region shape is preserved.
- For *Move* and *Copy* commands, the reference pathSeg and vias are resized interactively according to the track width.

© 2023 Cadence Design Systems, Inc.

Related Topics

Object Snapping Behavior in Width Spacing Patterns

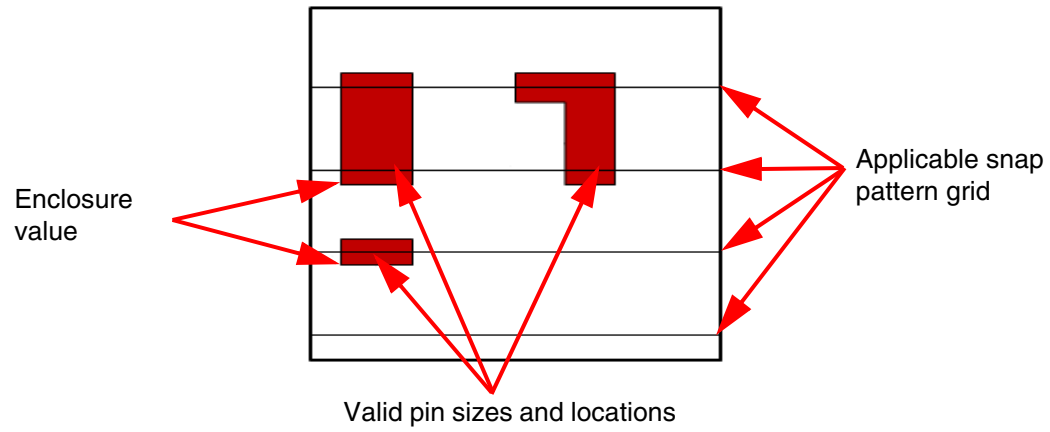
Object Editing in Layout XL

Pin Snapping to Width Spacing Patterns

A pin shape snaps to the snap patterns of the layer-purpose pair (LPP) on which the pin is being created or edited.

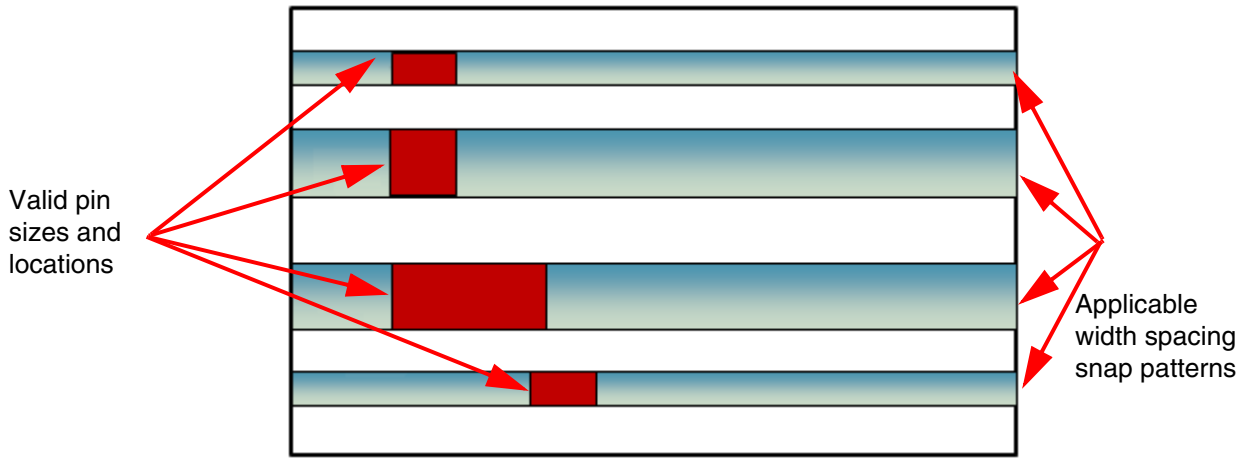
If the snap pattern is a snapPatternDef, the pin shape is snapped based on the width magnet parameters. If the snap pattern is a widthSpacingSnapPatternDef, the pin shape is snapped to the default active pattern available in the region. When you create a pin shape on a pattern, the height of the pin shape is always equal to the track pattern width.

By default, a rectangular pin is created where the height of the pin is equal to the track width. You can enable the square pin creation mode using the environment variable rectangularWSPPinAsSquare. In the square pin creation mode, the width of the pin is fixed by the track width. The length of the pin is also fixed by the track width. A square pin can be created on a WSP track with a single click. This mode is useful when you want to pre-fill net names in the Create Pin form. The square pins can be placed quickly, one pin per click. The figure below illustrates how a pin is snapped to a `snapPatternDef`.



© 2023 Cadence Design Systems, Inc.

The figure below illustrates how a pin is snapped to a `widthSpacingSnapPatternDef`.



The following precedence is used for pin shape snapping:

- Local grid (`snapPattern` followed by `widthSpacingSnapPattern`) available bound to shape: If the shape is bound to a WSP, the default active pattern applicable to corresponding WSSPD is applicable for pin shape snapping.
- Global grid (`snapPattern` followed by `widthSpacingSnapPattern`) available for pin LPP.
- If local or global grid is not found, the pin shape is snapped to the X or Y snap spacing grids.

The minWidth constraint defined in the technology file needs to be met for pin snapping. If the pin width is less than the `minWidth` constraint, the next track is searched.

When a track filter is active, pin editing is supported only for active tracks. Also, during pin creation, the pointer snaps to active tracks.

Related Topics

Dynamic Display During Pin Creation and Editing

Dynamic Display During Pin Creation and Editing

While running the pin creation and editing commands, the applicable WSP track is automatically displayed in the design so that you can visualize the available tracks and their width. This is important since WSP is so track centric.

© 2023 Cadence Design Systems, Inc.

Dynamic display is visible while running the pin creation and editing commands only when the *Snap Pattern Snapping* option is selected in the Layout Editor Options form and the reference object is snappable to the WSP grid.

For the *Create Pin* command, the entry LPP is used in the dynamic display. In case you are editing an existing pin, the LPP of the pin is used in the dynamic display.

When you are creating or editing a pin, the corresponding WSP grid is displayed even if the visibility of the grid is turned off in the Grids panel of the Palette assistant.

Related Topics

Create Pin Form

Layout Editor Options Form

Pin Creation

Object Editing in Layout XL

Controlling Transparency of Tracks With Alpha Blending

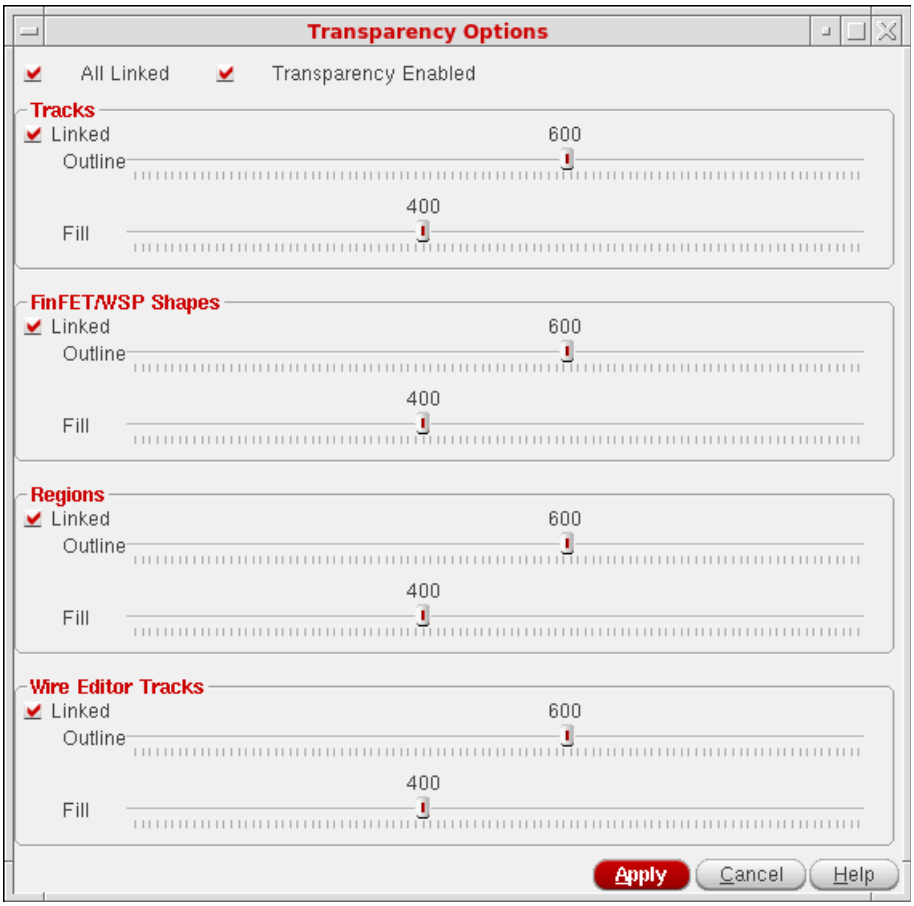
The alpha blending feature lets you control the transparency of tracks, FinFET and WSP shapes, regions, and Wire Editor tracks. You can control the transparency at different levels for the track fill or outline using the options available in the Transparency Options form.

To control the transparency of tracks:

1. In the layout window, choose *Options – Transparency*.

© 2023 Cadence Design Systems, Inc.

The Transparency Options form opens.



2. Select the *Transparency Enabled* check box to enable the transparency options in the form.
3. You can specify the outline and fill values for *Tracks*, *FinFET/WSP Shapes*, *Regions*, and *Wire Editor Tracks* using the sliders.

You can use the *Linked* check box to change the *Outline* and *Fill* values for *Tracks*, *FinFET/WSP Shapes*, *Regions*, or *Wire Editor Tracks* by tuning only one slider at a time. If you want to change the *Outline* and *Fill* values of *Tracks*, *FinFET/WSP Shapes*, *Regions*, and *Wire Editor Tracks* together by tuning only a single slider, select the *All Linked* check box.

You can also use the following environment variables to control the transparency at different levels for the track fill or outline.

alphaBlendingBoundaryFillValue	alphaBlendingBoundaryHaloValue
alphaBlendingBoundaryOutLineValue	alphaBlendingFinFillValue

© 2023 Cadence Design Systems, Inc.

alphaBlendingFinHaloValue	alphaBlendingFinOutLineValue
alphaBlendingTrackFillValue	alphaBlendingTrackHaloValue
alphaBlendingTrackOutLineValue	useAlphaBlending

Related Topics

Transparency Options Form

Interactive Snapping

Layout editor supports interactive snapping that enables you to visualize how an object will be placed after snapping. During interactive snapping, the object that is snappable and is closest to the reference point is used as the reference object. The snapping transform is calculated for the reference object at every mouse movement and the pointer is adjusted to ensure that the reference object is snapped to the grid.

For example, with the interactive snap pattern snapping feature, the first and the second points of the rectangle are snapped when you create a rectangle. The first point snaps to the upper enclosure and the second point snaps to the lower enclosure.

If discrete dimension is enabled, snapping does not update the second point. The dimension of the created rectangle is based on the value of the allowedWidthRanges constraint is specified in the technology file.

© 2023 Cadence Design Systems, Inc.

If discrete dimension is disabled, the second point discretely snaps by the step size of the snapping grid. The figure below depicts the snapping of the rectangle with the discrete dimension option enabled and disabled.



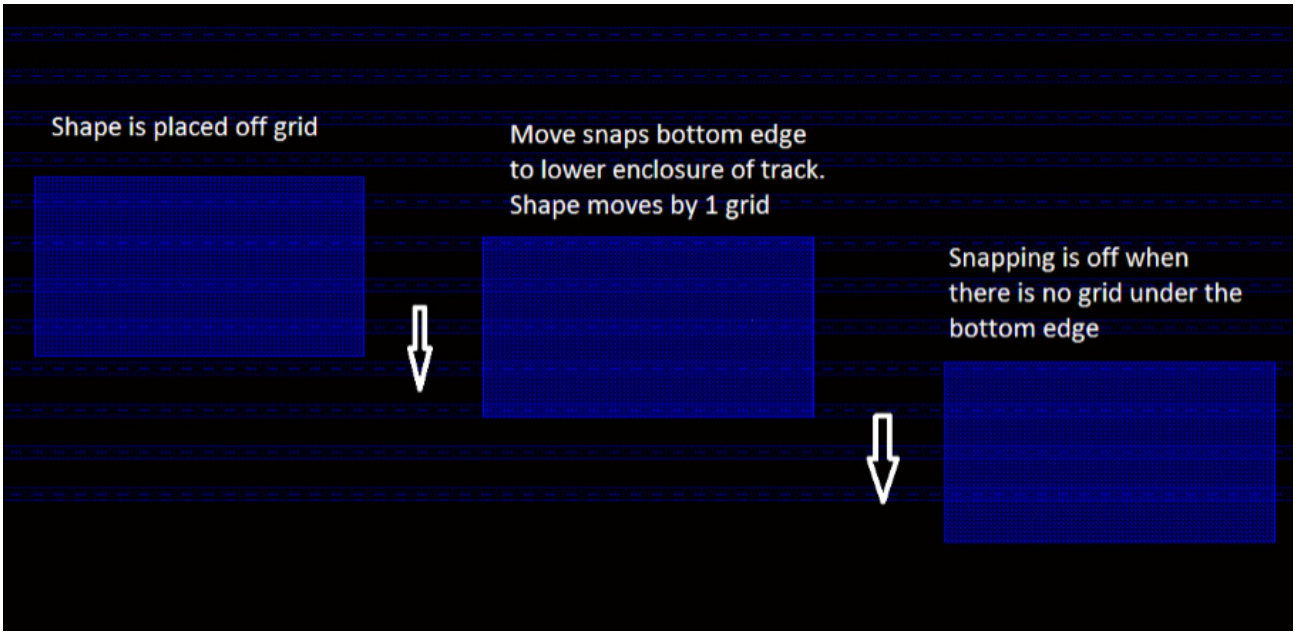
Interactive Snap Pattern Snapping When Creating a Rectangle

When you move objects, snapping ensures that the bottom edge (vertical snapping) and left edge (horizontal snapping) of the reference object is snapped to the grid. The reference object moves by the step size of the snapping grid. The reference object snaps if the bottom edge of the shape is inside a snapping region even if the reference point goes outside it.

If the reference object overlaps multiple snapping regions, it snaps to the region under the bottom edge of the shape. If the bottom edge of the reference object goes out of the snapping

© 2023 Cadence Design Systems, Inc.

region, then snapping is not performed even if some part of the shape overlaps the snapping region. The figure below depicts interactive snapping when moving objects.



Interactive Snap Pattern Snapping During Move Operation

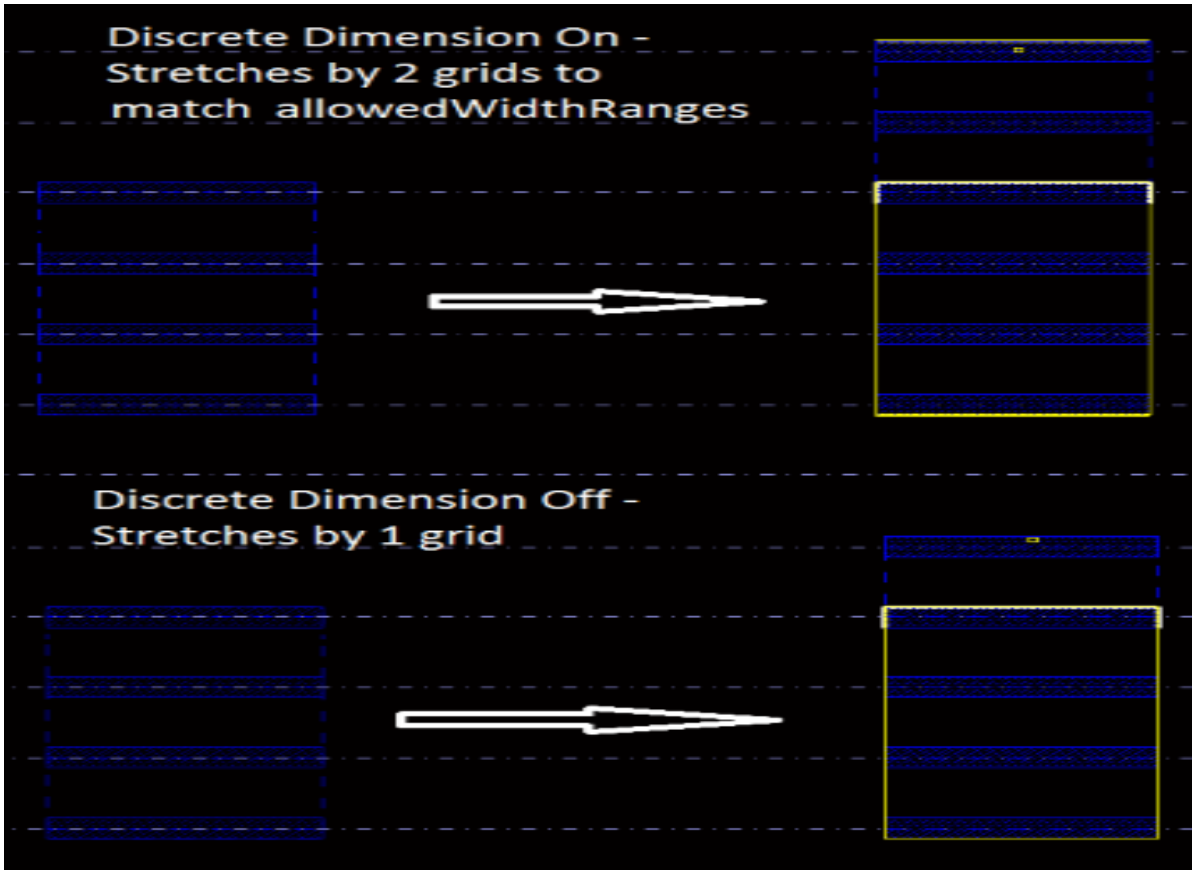
Interactive snapping for the *Copy* command works similar to the way it works for the *Move* command.

When you stretch objects, snapping ensures that the bottom edge (vertical snapping) and left edge (horizontal snapping) of the reference object is snapped to the grid. The reference object stretches by the step size of the snapping grid.

If discrete dimension is enabled and it adjusts the selected edge, snapping does not modify the edge. So, if the shape is off-grid before the *Stretch* command, it will remain off-grid after the *Stretch* command also. If the discrete dimension option is disabled, snapping ensures

© 2023 Cadence Design Systems, Inc.

that the edges of the shape are on grid after the *Stretch* command, as depicted in the figure below.



Interactive Snap Pattern Snapping During Stretch Operation

If you stretch a partially selected edge, the selected edge is snapped to the grid. In this case, the bottom edge snaps to the lower enclosure and the top edge snaps to the upper enclosure of the track.

When you stretch the top edge of the shape, it snaps to the grid till it is inside the snapping region. Snapping is not performed when the edge is out of the snapping region.

An instance is snapped interactively during the *Create Instance* command with every mouse drag.

Related Topics

Width Spacing Patterns in Layout Editor

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

A

Layout XL Basic Editing Forms

The following table lists the basic editing forms available in Virtuoso® Layout Suite XL.

A-D	E-Q	R-Y
Add Property Form	Edit Cellview Properties Form	Refresh Cellviews Form
Add Shape to Net Form	Edit Properties Form	Remaster Instances Form
Align Form	Edit Cut Pattern Form	Remove Shape From Net Form
Area and Density Calculator Form	Edit Synchronous Clone Form	Rename Command Name Form
Assign Net Form	Edit Via Array Pattern Form	Repeat Copy Form
Auto Via Assistant Options Form	Enter Points Form	Reshape Form
Background Cellview Form	Export Image Form	Restore View Form
Choose Object for Label Creation Form	Export Label Form	Return To Level Form
Chop Form	Find Marker Form	ROD Subpart Form
Copy Form	Find/Replace Form	Rotate Form
Create Area Boundary Form	Flatten Form	Rotate Left Form
Create Blockage Form	Flip Form	Rotate Right Form
Create Circle Form	Flip Vertical Form	Save a Copy Form
Create Cluster Boundary Form	Flip Horizontal Form	Save All Mark Nets Form
Create Custom Placement Area Form	Freeze Pcell Instance Form	Save Configuration Form

© 2023 Cadence Design Systems, Inc.

A-D	E-Q	R-Y
Create and Edit Clusters Form	Group Return Form	Save Multipart Path Templates Form
Create Encore Report Form	Guard Ring Form	Save Recording Form
Create Group Form	Halo Editor Form	Save/Restore Selection Set Form
Create Instance Form	Highlight Options Form	Save Via Variant Form
Create Label Form	Instance to Mosaic Form	Save View Form
Create Measurement Form	Layer Generation Form	Select Area Shape Type Form
Create Multipart Path Form	Layout Editor Options Form	Select Array Member Form
Create Path Form	Layout Integrity Check Form	Selection Options Form
Create Pin Form	Magnifier Options Form	Selection Protection Highlight Options Form
Create Pins From Labels Form	Make Cell Form	Set Area View Level Form
Create Polygon Form	Make Readonly/Editable Form	Set Default Application Form
Create P&R Boundary Form	MarkNet Options Form	Set Pin Label Text Style Form
Create Rectangle Form	Marker Text Window	Show Selected Set Window
Create Row Form	Modify Corner Form	Size Form
Create Slot Form	Modify Property Form	Split Form
Create Snap Boundary Form	Move Form	Stretch Form
Create Via Form	Open Form	Summary Window
Create Via Options Form	Pad Opening Info Form	Tap Form
Cross Section Viewer Form	Pad Opening Info: Filter Form	Track Pattern Editor Form

© 2023 Cadence Design Systems, Inc.

A-D	E-Q	R-Y
Custom Show Layer/ Custom Show Purpose Form	Palette Options Form	Transmission Line Bend Form
Define Net/Pin Connections Form	Paste Form	Transmission Line Form
Delete Form	Print Hierarchy Tree Form	Transmission Line Taper Form
Delete All Markers Form	Quick Align Form	Transparency Options Form
Delete Configuration Form	Quick Align Form	Tree Form
Delete Marker Form		Uniquify Form
Descend Form		Unmark Net Form
Descend Options Form		View By Form
Display Options Form		Yank Form
Dynamic Display Form		
Dynamic Selection Assistant Options Form		

© 2023 Cadence Design Systems, Inc.

Add Property Form

Use the Add Property form to add your own properties to the currently selected object.

Field	Description
Name	Specifies the name to assign to this property. This name will appear on the Edit Properties or the Edit Cellview Properties form.
Type	Specifies the value type for the property. This can be one of the following: ■ *String* (any text, displayed in a cyclic field) ■ *Int* (integer) ■ *Float* (floating-point number) ■ *Time* (date and time) ■ *Boolean* (on or off) ■ *ILList* (list of IL values)
Value	Shows either a text entry field, a Boolean button, or a cyclic field, depending on the *Type* setting.
Choices	Sets the possible values that appear in the cyclic field for a string property. One of the values listed here must match the default shown in *Value*.
Minimum and Maximum	Displays the minimum and maximum values for integer, floating-point number, and time properties. These ranges appear in parentheses after the property name in the Edit Properties form and the Edit Cellview Properties form.

Related Topics

Adding a New Cellview Property

Edit Properties Form

Edit Cut Pattern Form

© 2023 Cadence Design Systems, Inc.

Add Shape to Net Form

Use the Add Shape to Net form to add selected shapes to a specified net or based on the pins overlapping the selected shapes.

Field	Description
Net Name	Lets you enter a net name when the *Auto* button is off.
Auto	Determines whether you enter a net name or get the net name automatically: ■ *On*: Selects the net based on the pins overlapping the selected shapes. The overlapping pin and shape must have the same layer-purpose pair. Each selected shape is given the net of a pin that overlaps it, if the shape does not already have a net. ■ *Off*: Lets you specify the net name. The selected shape does not have to be overlapping the specified pin.

Related Topics

Adding Shapes to a Net

leHiAddShapeToNet

© 2023 Cadence Design Systems, Inc.

Align Form

Use the Align form to align objects to a point, an edge, a layer, or to a target axis.

The form comprises the following sections:

Section	Description
Reference	Allows you to choose a reference point or edge of an object, shape or layer to which you want to align other objects. Environment variable: alignReference
Align Using	Allows you to choose which point or edge of the objects will align to the reference point, object or layer selected through the Reference section of the form. Environment variable: alignUsing
Spacing and Sorting	Allows you to control the spacing and sorting functionality.

Reference

The following table describes the fields available on the *Reference* section.

Field	Description
Point	Aligns objects, shapes, or layers to a point you select. Select the point by using the pointer or by typing the numerical X and Y values of the point in the CIW.
Object	Aligns objects, shapes, or layers to an object. Select the type of object from the cyclic field, then use the pointer to select the object. Layout editor supports instances, rows, cluster boundaries, groups, and pins. The *Align* command does not support area halos, layer halos, PR boundaries, snap boundaries, area boundaries, and blockages. When you select the *Object* option, the *Use* field is activated. Environment variable: alignRefObjType

© 2023 Cadence Design Systems, Inc.

Field	Description
Layer	Aligns objects or shapes to a layer edge of a reference object, shape or layer. Select the layer from the cyclic field, then use the pointer to select the reference object. The cyclic field displays layers defined in the technology file. When you select the *Layer* option, the *Use* field is activated. You can search for specific layer-purpose pair using the search filter in the *Layer* list box. **Note:** If the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form, only the layers selected using the search filter in the *Layers* panel of the Palette are visible in the *Layer* list box. Environment variable: alignLayer
Align Direction	Sets the alignment direction. ■ *Horizontal*: Aligns objects in the X direction. ■ *Vertical*: Aligns objects in the Y direction. Environment variable: alignDirection

© 2023 Cadence Design Systems, Inc.

Field	Description
Use	Sets the edges that are aligned to the objects, shapes or layers. ■ If you select a *Point* to use as reference and select *Horizontal* or *Vertical* as the *Align Direction*, you can align the point to an object or layer by using the options in the Align Using section of the form. The *Use* cyclic field in the Reference section is disabled when you select *Point*. ■ If you select either the *Object* or *Layer* option to use as reference and *Horizontal* as the *Align Direction*, the valid values of *Use* are *Top Edge*, *Bottom Edge*, and *Center*. If you select *Vertical* as the align direction, valid values are Left Edge, Right Edge, and Center. If *Object* is set to *Instance*, the *Use* valid value also includes *Origin* for both align directions. You can align the object to an object or layer by using the options in the Align Using section of the form. You can either select the objects and then set the alignment criteria (ion mode) or set the alignment criteria and then select the objects (post-selection mode). Environment variable: alignRefMethod
Set New Reference	Clears any existing reference point and prompts you to specify a new reference point to which the selected objects are to be aligned. Click the *Set New Reference* button, then click in the cellview window at the point where you want the objects to begin alignment.

© 2023 Cadence Design Systems, Inc.

Align Using

The following table describes the fields available on the *Align Using* section.

Field	Description
Object	Sets the object type of objects that can align to the point, object or layer selected in the Reference section. The valid objects include *Instance*, *Any Shape*, *Row*, *Cluster Boundary*, *Group*, and *Pins*. The default value is *Instance*. When *Pins* are selected in the *Reference* section, this object list is restricted to *Pins* only. Environment variable: alignObjType
Layer	Sets the layer of the objects to align to the point, object or layer selected in the Reference section. The layers defined in the technology file display in the cyclic field. You can search for specific layer-purpose pair using the search filter in the *Layer* list box. **Note:** If the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form, only the layers selected using the search filter in the *Layers* panel of the Palette are visible in the *Layer* list box. Environment variable: alignLayer

© 2023 Cadence Design Systems, Inc.

Field	Description
Use	Sets the object or layer edge to align with the point, object ,or layer edge selected in the Reference section of the form. ■ If you select *Point* in the *Reference* section and *Object* in the *Align Using* section, the *Use* cyclic field choices are Upper Left, Upper Center, Upper Right, Center Left, Center Center, Center Right, Lower Left, Lower Center, Lower Right, and Origin. ■ If you select *Object* or *Layer* in the Reference section, *Horizontal* as the *Align Direction*, and *Object* in the Align Using section, the *Use* cyclic field choices are *Top Edge*, *Bottom Edge*, and *Center*. If you select *Vertical* as the align direction, the *Use* cyclic field choices are *Left Edge*, *Right Edge*, and *Center*. If *Object* is set to *Instance* in the Align Using section, the *Use* valid value also includes *Origin*. Environment variable: alignMethod

Spacing and Sorting

This following table describes the fields available in the *Spacing and Sorting* section.

Field	Description
Apply Spacing check box	Aligns objects with respect to the reference object as per the spacing value specified in the text field, as per alignment direction. If you do not select this option, the *Align* command aligns objects and leaves the original spacing between objects intact. Environment variable: alignApplySpacing
Apply Spacing field	Specifies the spacing value between the aligned objects as per alignment direction. The value must be a floating point number. The default value is `1.0`. Environment variable: alignSpacingValue

© 2023 Cadence Design Systems, Inc.

Field	Description
Orthogonal Spacing check box	Aligns objects with respect to the reference object as per the spacing value specified in the text field, in the direction perpendicular to *Align Direction*. This option is available only when the *Apply Spacing* option is selected.
Orthogonal Spacing field	Specifies the spacing value between the aligning edges of objects, in the direction perpendicular to *Align Direction*. The value must be a floating point number. The default value is `1.0`. If *Align Direction* is *Horizontal*, then the *Apply Spacing* field value is used to space objects along X-axis while *Orthogonal Spacing* field value is used to space objects along Y-axis. Similarly, if *Align Direction* is *Vertical*, the *Apply Spacing* field value will space objects along Y-axis while *Orthogonal Spacing* field value will be used to space objects along X-axis. You can deselect the *Orthogonal Spacing* option if you want to space objects only in *Align Direction.*
Between Objects/ Pitches	■ *Between Objects*: Specifies the distance, in user units, maintained between adjacent edges of objects to be aligned. The first object overlaps the selected reference point. The next object aligns at a distance specified in the Apply Spacing field from the adjacent edge of the previous object. If you specify a negative number, the aligned objects overlap. ■ *Pitches*: Specifies the distance, in user units, to maintain across objects between the specified points (*Use* in the Align Using section). The first object overlaps the selected reference point. The next object aligns at a distance specified in the *Apply Spacing* field from the aligned point of the previous object. The *Pitches* option is available only if you select *Point* in the *Reference* section. The spacing distance is specified in the *Apply Spacing* field. Environment variable: alignSpacingType

© 2023 Cadence Design Systems, Inc.

Field	Description
Sort Objects By	Allows you to control how the selected objects are sorted in relationship to the reference object. This option is valid only with preselected objects. ■ *Align Direction Order*: Sorts the objects by the horizontal or vertical alignment direction selected in the Reference section of the form. ■ *Horizontal Order*: Sorts the objects aligned horizontally, retaining their original horizontal order. ■ *Vertical Order:* Sorts the objects aligned vertically, retaining their original vertical order. ■ *Order Selected:* Sorts the objects in the order they were selected. However, objects are placed above or below, or to the right or left of the reference point according to their original position in the cellview, irrespective of the order in which they are selected. To have all objects above the reference point, they must originate above the reference point. ■ *Instance Name:* Sorts the objects alphanumerically by instance name. ■ *Net Name:* Sorts the objects alphanumerically by net name. Environment variable: alignSortMethod
Reverse Sort	Allows you to reverse the sorting of objects specified in *Sort Objects By*. Environment variable: alignReverseSort

© 2023 Cadence Design Systems, Inc.

Field	Description
Chain Mode	Enables you to align device chains to another point, object, or layer. Environment variable: chainMode ■ *All:* Aligns all the instances in the chain. ■ *Selected:* Aligns only the selected instances. ■ *Selected Plus Left:* Aligns the selected instance and all the instances to the left of the selected instance in the chain. ■ *Selected Plus Right:* Aligns the selected instance and all the instances to the right of the selected instance in the chain.
Match layer/purpose with the reference pin	Lets you change the target pin layer purpose of the pin to the reference pin layer purpose. Environment variable: alignMethod
Match size with the reference pin	Lets you want to change the target pin size to that of the reference pin. Environment variable: alignMatchRefSize
Update pin label layer	Lets you update the layer of labels to that of the pin. This option can be used when the layers of the pin and label are different. Environment variable: alignUpdateLabelLayer

Related Topics

Object Alignment

leHiAlign

© 2023 Cadence Design Systems, Inc.

Area and Density Calculator Form

Use the Area and Density Calculator form to compute the total area and the density of shapes with specified layers within a region.

Field	Description
Object	Lets you specify the objects for which you want to calculate the area and density. You can select *All* (all objects) or *Selected* (only selected objects will be considered for area calculation). If an instance, group, via, or guard ring is selected, the area of the shapes inside the selected instance is calculated.
Layer	Lets you select the layer for which you want to calculate the area and density. You can select the layer pursose pair from the drop-down list or you can specify a layer purpose pair. You can select a single layer or multiple layers for area calculation. The syntax that you should use to specify the layer is: ■ Single Layer: `Met1/drawing` ■ Multiple Layers: `Met1/drawing1 Met1/drawing2 Met2/drawing1` ■ All Purpose: `Met1` ■ Use () to merge different layers: `(Met1 M1) (Met1/drawing1 Met1/drawing2)`
Hierarchy Range	Sets the levels of hierarchy to be considered for area and density calculation. ■ *current cellView:* Performs the calculation in the current edit cellview in the layout window. ■ *current to bottom:* Performs the calculation from the current edit cellview level to the bottom of the hierarchy. ■ *current to stop level:* Performs the calculation from the current edit cellview level to the level specified in the *Display Levels Stop* field in the Display Options Form. ■ *current to user level:* Performs the calculation from the current edit cellview level to the level that you specify.

© 2023 Cadence Design Systems, Inc.

Field	Description
Region	■ *Entire CellView:* Considers the entire cellview for area and density calculation. This option is selected by default. ■ *Viewing Area:* Considers the portion of the cellview that is displayed in the canvas at that instant for area and density calculation. ■ *A Selected Boundary:* Considers a selected boundary for area and density calculation. ■ *Draw a Rectangle Region on the Canvas or Draw a Polygon Region on the Canvas:* Enables you to specify a rectangular or polygonal area for area and density calculation. Select the required option, and then click in the canvas to specify the vertices of the rectangle or polygon.

Related Topics

Display Options Form

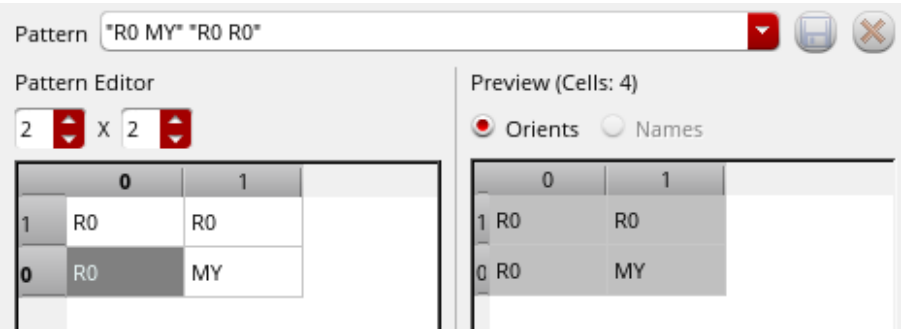
Calculating Area and Density

leComputeAreaDensity

© 2023 Cadence Design Systems, Inc.

Array Pattern Form

Use the Array Pattern form to specify the orientation of individual cells of a group array. You can select a predefined pattern or create your own pattern.

Field	Description
Pattern	Lets you select an orientation pattern for the group array.
Save orient pattern	Saves a custom orientation pattern with the name specified in the *Pattern* field.
Delete orient pattern	Deletes the custom orientation pattern selected in the *Pattern* field. Predefined orientation patterns cannot be deleted.
Pattern Editor	Lets you define the orientation of a pattern. You can use the spin boxes in this section to set the row and column count for the pattern. In the orientation table, you can define the orientation value for each cell in the pattern. When you modify the row and column count or the orientation values, the updated pattern is populated in this section. Also, the updated orientation values are listed in the *Pattern* field of the form and the *Preview* section displays the updated pattern applied to the group array. Right-clicking any cell provides you the option to clear the row to the end. 
Preview	Displays the preview of the selected pattern applied to your group array. It also lists the total number of cells in the group array.

Related Topics

Group Arrays

© 2023 Cadence Design Systems, Inc.

Assign Net Form

Use the Assign Net form to assign nets to shapes, instance pins, and routed objects like vias and pathsegs. You can access the form by pressing F3 when the `leHiAssignNet` function is active.

Field	Description
Net Name	Specifies the net name to be assigned. You can select a net name from the list or type a name. If the specified net does not already exist, a new net is created by that name. You can also use the icon displayed next to this field to pick an existing net name by clicking a shape in the canvas.
Pop up selection window	Displays a form in case of overlapping shapes. The form lists all the shapes on various layers, with details of the assigned net name. You can use this form for both picking and assigning a net.
Propagate connectivity	Propagates connectivity to physically connected shapes, vias, instance pins, and pathSegs during net assignment. This option follows Extractor settings specified for the constraint group and the *Extract connectivity to level* option in the Connectivity form. Environment variable: propagateAssignNet

Related Topics

Assigning and Unassigning Nets on Shapes

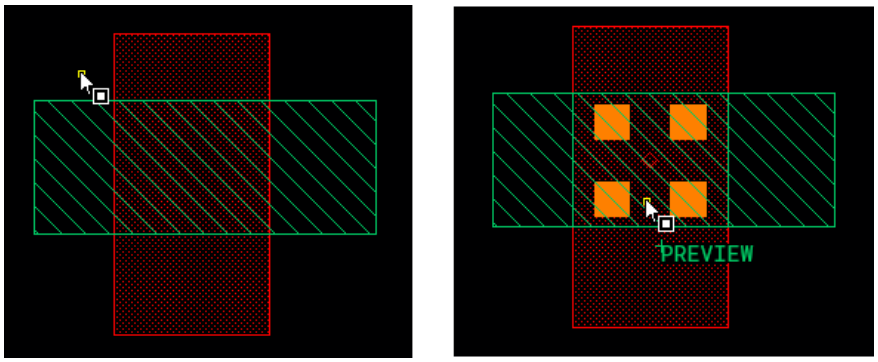
leHiAssignNet

Connectivity Form

© 2023 Cadence Design Systems, Inc.

Auto Via Assistant Options Form

In Virtuoso Layout Suite EXL and higher tiers use the Auto Via Assistant Options form to specify the options for analyzing or creating auto vias.

Field	Description
Show preview	Shows a preview of the via that will be created. This check box is selected by default.
	 When you move the pointer over the overlap, the preview of the via is displayed
Show report	Shows a report of the via that will be created. This check box is selected by default.
Create in route	Lets you to create a via in a new route. If this option is selected, every new via is created in a new route. This check box is selected by default.
Create vias using top 2 layers	Lets you to create vias using top two layers only.
Override variants	Lets you to update a via variant and overwrite the existing via parameter values. If this check box is selected, it allows the via engine to update a via variant.
Fill Method	Specifies the fill method to be used while creating the via. You can choose from `Fully`, `Partially`, `Horizontally`, or `Vertically`.

© 2023 Cadence Design Systems, Inc.

Field	Description
Align cuts	Specifies the alignment of cuts between different tiles of a polygonal overlap. When this option is enabled, cuts between different tiles of a polygonal overlap are aligned. When this option is disabled, the polygonal overlap is decomposed into a set of largest tiles and vias are created independently in each tile, which may result in cuts not being aligned between different tiles.
Overlaps Section	This section lets you specify the purposes and blockages to use for overlaps.
Valid Purposes	Lets you to specify a list of valid purposes for via creation. The via will not be created even if one purpose of the overlap shapes does not belong to the list of valid purposes specified in this field.
Ignore Purposes	Lets you specify the list of purposes that will be ignored for via creation.
Consider blockages	Does not create any via in case a blockage exists between the top and bottom metal of the current overlap.
Enclosures Section	This section lets you specify how the via enclosures should be extended.
Extend beyond overlap	Lets you to extend the via enclosures beyond the overlap region between two intersecting shapes. The shape layer that exists beyond the overlap region helps to accommodate the extended enclosures.
Fill overlap	Lets you to extend via enclosure such that it covers the complete overlap area between the two intersecting shapes, that is, the bounding box of the via enclosure coincides with the bounding box of the overlap region.
DRC Check Section	This section lets you specify the DRD checks to be performed while creating the via.

© 2023 Cadence Design Systems, Inc.

Field	Description
Hierarchy Range	Sets the levels of hierarchy to be considered for layout integrity check. *current to stop level:* Performs the check from the current edit cellview level to the stop level specified in the *Display Levels Stop* field in the Display Options form. *current to bottom*: Performs the check from the current edit cellview level to the bottom of the hierarchy. *current to user level:* Performs the check from the current edit cellview level to the level that you specify in the spin box.

Related Topics

Create Via Form

Auto Via Assistant

Analyzing a via using Auto Via Assistant

Creating a via using Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

Background Cellview Form

Use the Background Cellview form to display two superimposed cellviews in the same window.

Field	Description
Library, Cell, and *View*	Specifies the library, cell, and view names of the cell you want to open as the background cellview. You cannot open the top display cellview as the background cellview.
X and Y	Specifies the coordinates to place the background cellview in the window.
Rotation	Specifies the angle at which the background cellview is placed in relation to the top display cellview.
Display	Specifies the luminosity at which the background cellview is displayed. ■ *off*: Does not display the background cellview. ■ *on*: Displays the background cellview. ■ *dimmed*: Displays the background cellview at the luminosity determined by Dim Intensity.

Related Topics

Dim Intensity

Background Cellview

© 2023 Cadence Design Systems, Inc.

Choose Object for Label Creation Form

Use the Choose Object for Label Creation form to choose a layer when clicking on a coordinate with objects on overlapping layers, if there are multiple overlapping shapes in the manual and auto label creation modes. You can select one or more objects for label creation.

Field	Description
Label	Displays the label for the layer. This field is available only if the label creation mode is auto.
Layer	Displays the layer for the label.
Purpose	Displays the purpose for the label.
Object	Displays the object.

Related Topics

Label Creation

© 2023 Cadence Design Systems, Inc.

Chop Form

Use the Chop form to remove parts of objects or break objects into multiple objects.

Field	Description
Chop Shape	Controls the shape you want to use as the cutter. You can choose from *rectangle*, *polygon*, *line,* or *wire*. If *Chop Shape* is set to *line*, *Remove Chop* has no effect. Environment variable: chopShape ■ *Rectangle:* Specifies the width, height, and justification of the chop shape using the *Width*, *Height*, and *Justification* fields.Select the *Specify Size* option to specify the size of the chop shape. Environment variables: chopRectHeight, chopRectJustify, chopRectWidth ■ *Circle:* Selects the *Specify Size* option to specify the size of the chop shape. Specify the radius and justification of the chop shape using the *Radius* and *Justification* fields. Environment variables: chopCircleJustify, chopCircleRadius ■ *Line:* Selects the *Specify Spacing* option to specify the size of the chop shape. You specify the type, width, and justification of the chop shape using the *Type*, *Width,* and *Justification* fields. You can select *User Spacing* or *Min Spacing* from the *Type* drop-down list. *User Spacing* lets you specify width of spacing and *Min Spacing* calculates the spacing based on the minimum spacing specified in the technology file. Also, for path, pathSeg, and rectangle, when the *Min Spacing* option is selected, the begin point of the line is on the lower orthogonal grid and the end point of the line is on the upper orthogonal grid. Environment variables: chopLineJustify, chopLineType, chopLineUserWidth

© 2023 Cadence Design Systems, Inc.

Field	Description
	■ *Wire:* When you select the chop shape as wire and there is a selectable wire, path or pathSeg, at the first point that you click, the wire is chopped and the Chop command looks for next click. However, if there is no selectable wire at the first point that you click, this point becomes the first point of the line, the prompt changes and the command looks for other points for the line. You can only chop selectable wires, paths or pathsegs, for single-click chop or multiple-click chop.
	For all chop shapes, a pathSeg is not converted to a polygon.
	When there are no applicable rule in the technology file, the *Chop* command chops shapes without any gap.
Snap Mode	Controls the shape of polygon or line segments. *Snap Mode* applies only when *Chop Shape* is set to *polygon* or *line*. You can choose from *anyAngle*, *diagonal*, *orthogonal*, *L90XFirst*, and *L90YFirst*.
	When snap mode is *anyAngle* or *diagonal*, paths and pathSegs are chopped similar to the snap mode *orthogonal*.
Remove chop	Removes the part of the object enclosed by *Chop Shape*.
Chop array	Enables you to chop an array of instances (mosaic). This check box is selected by default. When this check box is selected and you define the chop area for an array, none of the points of the chop area (vertices of the rectangle and polygon and end points of a line) should be contained within the array. The vertices or end points should be either outside the array or on the boundary of the array.
	Environment variable: chopArray
Snap Section	This section lets you define snap settings for the chopped object.

© 2023 Cadence Design Systems, Inc.

Field	Description
Pins and boundary to grid	Snaps the chopped object to a grid depending on the block type. If it is an *analog* block, the place and route boundary and pin edges are snapped to the manufacturing grid. If it is a *digital* block, the place and route boundary is snapped to the placement grid and the pin centers are snapped to the routing grid. Pins with placement status unplaced, fixed, locked, or unknown are not snapped, even if they were touching the boundary before the chop. Environment variable: snapToGrid
Pins to boundary	Snaps pins to the place and route boundary automatically.
Soft Block Pins to	Snaps soft block pins automatically either to the *Modified Edge* or to the *Closest Edge* after the chop is complete.
All edges to manufacturing grid	Snaps all the edges of the chopped shape to the manufacturing grid. Environment variable: snapToMfgGrid

Related Topics

Setting Up the Snap Mode

Object Chopping

Soft Block Editing

leHiChop

© 2023 Cadence Design Systems, Inc.

Copy Form

Use the Copy form to copy selected objects in the specified window.

Field	Description
Keep Copying	Enables you to create multiple copies of an object. With this check box selected, a copy of the selected object is created every time you click in the design display area. You can customize this mode of copying by configuring the other settings in the form. The *Delta*, *Spacing*, and *Absolute* radio buttons in the *Spacing* section are not available when you select this check box.
Snap Mode	Controls the direction in which you can move the copied object.
Change To Layer	Enables you to copy an object to another layer. You can search for the specific layer-purpose pair using the search filter in the *Change To Layer* list box. Environment variable: copyToLayer. **Note:** If the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form, only the layers selected using the search filter in the *Layers* panel of the Palette are visible in the *Change To Layer* list box. The value selected in the *Change To Layer* field is reset after each copy operation.
Copy color	Specifies that the coloring information from source objects will be copied from source objects as-is to the corresponding destination objects when copying.
Synchronous copy	Creates a synchronized copy of the selected layout objects. This implies any edits made to the source object are also replicated in the copied object, and vice versa. The *Synchronous copy* field is reset after each copy operation. *Synchronous copy* is disabled when you select the *Create as Group Array* check box.

© 2023 Cadence Design Systems, Inc.

Field	Description
Display draglines	Controls whether draglines are displayed during the *Copy* command. By default, draglines indicate connections from the pins of the object you are moving to the pins of the nearest objects. For draglines to be displayed, both *Copy connectivity* and *Display Draglines* should be selected. *Display Draglines* gets disabled when you select the *Create as Group Array* check box.
Copies	*This section lets you specify the type of copy you want to create.*
Type	Specifies the type of copy: ■ *Single*: Lets you create a single copy of objects. ■ *Step*: Lets you create a line of copied objects. Use the *Number* field to specify the number of copies to create. ■ *Array*: Lets you create an array of copied objects. Use the *Rows* and *Columns* field to specify the dimensions of the array. The array of copied objects can be placed only in the original cellview; it cannot be copied to another cellview. Environment variable: copyOrder
Create as Group Array	Creates a group array. Environment variable: copyCreateGroupArray If the copyGroupArrayLockXY environment variable is set to `t`, you cannot specify spacing mode or values of the group array in the Copy form. Spacing mode is read from the copySpacingMode environment variable, whereas spacing values are read from the groupArrayXSpacing and groupArrayYSpacing environment variables.

© 2023 Cadence Design Systems, Inc.

Field	Description
Array Pattern	Lets you specify the orientation of individual cells of a group array. You can specify the orientation values directly in the text field by enclosing them in quotation marks. You can also select a predefined or custom pattern from the drop-down list. Alternatively, you can use the Array Pattern form, which is accessed by clicking the *Open Array Pattern editor* button  . The *Array Pattern* option is available when the *Create as Group Array* check box is selected.
Spacing	*This section lets you specify the spacing settings for the copied objects.*

© 2023 Cadence Design Systems, Inc.

Field	Description
Mode	Lets you select the spacing mode. ■ *Pitch*: Applies between objects the delta distance that you specify in the *Delta X, Y* field. The distance is measured between the same-edge corners of the objects in an array; for example, the distance between the left edges of objects. If the mode is *Pitch*, the *Reference LPP* check box is disabled. ■ *Spacing*: Applies between objects the distance that you specify in the *Spacing X, Y* fields. The distance is measured between the adjacent-edge corners of the objects in the array; for example the distance between the bottom edge of an object and the top edge of another object. ■ *Absolute*: Enables you to copy an object at the exact location that you specify in the *X, Y* fields. The *Absolute* mode is not available when the *Create as Group Array* check box is selected. Environment variables: copySpacingMode, copyXField, and copyYField When the *Create as Group Array* check box is selected, the spacing mode for a group array is read from the `copySpacingMode` environment variable and the spacing values are read from the groupArrayXSpacing and groupArrayYSpacing environment variables.
Apply	Copies the selected objects based on the settings specified in the form.

© 2023 Cadence Design Systems, Inc.

Field	Description
Reference LPP	Specifies the layer-purpose pair used for computing the bounding box of the cells in a group array. The combined bounding box of the geometries on the specified reference LPP is used for calculating the spacing between cells. When the value is set to *None*, the combined bounding box of all the geometries in a cell is used for determining the spacing between cells. *Reference LPP* is available only in *Spacing* mode when the *Create as Group Array* check box is selected.
Connections	This section enables you to specify the connectivity settings.
Copy connectivity	Copies the connectivity information on path, pathSeg, fluid guard ring, and via to the new object. When the *Display Draglines* option is also selected, draglines are displayed during the Copy command to indicate the connections from the pins of the object you are moving to the pins of the nearest objects. Environment variable: copyConnectivity
Increment net name	Increments the bus bit on the copied object by one if the selected object has a net name with a bus bit. This option is available only when the *Copy connectivity* option is selected. Environment variable: copyIncrementNetName
Chain Mode	Enables you to copy device chains. ■ *All*: Copies all instances in a chain. ■ *Selected*: Copies only the selected instances. ■ *Selected Plus Left*: Copies the selected instance and all the instances to the left of it in the chain. ■ *Selected Plus Right*: Copies the selected instance and all the instances to the right of it in the chain.

© 2023 Cadence Design Systems, Inc.

Field	Description
Snap	*Pins and boundary to grid:* Snaps the pin edges to the manufacturing or the routing grid, depending on the selected block type. ■ If the block type is custom, the pin edges are snapped to the manufacturing grid. ■ If the block type is digital, the pin edges are snapped to the routing grid. Environment variable: snapToGrid
Rotate/Flip	Rotates or flips objects on the canvas. ■ *Rotate Left:* Rotates objects by 90 degrees in the counterclockwise direction. ■ *Rotate Right*: Rotates objects by 90 degrees in the clockwise direction. ■ *Flip Vertical:* Mirrors objects along the Y-axis. ■ *Flip Horizontal*: Mirrors objects along the X-axis.

Related Topics

Object Copying

Copying an Object

Creating a Group Array by Using the Copy Form

Repeat Copy Form

Generating a Synchronous Copy

leHiCopy

© 2023 Cadence Design Systems, Inc.

Create Area Boundary Form

Use the Create Area Boundary form to create an area boundary.

Field	Description
Name	Assigns a name unique within the cellview to the area boundary.
Create Method	Chooses which method is used to create the area boundary object. ■ *From Selected Set:* Creates the area boundary using the rectangular bounding box of the selected set of objects. ■ *Rectangle:* Creates the area boundary using two points. ■ *Polygon:* Creates the area boundary using multiple points.
Apply	Creates the PR boundary using the selected objects.
Snap Mode	Selects snap mode to use as you enter points when *Polygon* create method is selected.
Create as ROD Object	Creates the area boundary as a ROD object. *Name:* Assigns a unique name to the new area boundary in the current cellview if the *Create as ROD Object* check box is selected. This unique name consists of the prefix boundary followed by an integer. For example, the first area boundary is named boundary0, the second one boundary1, and so on. You can edit the default name, if required, but ensure that the new name is unique in the cellview.
Soft Fence	Creates the soft fence. Objects enclosed by the area boundary with a soft fence are compacted relative to each other as well as compacted as an entity relative to objects outside the boundary. Selecting this option adds the property `ctuBoundaryType` to the area boundary, with the property value `prSoftFence`.

Related Topics

Creating an Area Boundary

leHiCreateAreaBoundary

© 2023 Cadence Design Systems, Inc.

Create Blockage Form

Use the Create Blockage form to create blockages.

Field	Description
Blockage Type	Selects the type of blockage to create. The choices are: placement, routing, pin, feedthru, or screen. Environment variable: blockageType
Material	Selects the material on which the blockage is enforced. The choices are: all, other, nwell, pwell, nplus, pplus, ndiff, pdiff, poly, metal, diffusion, li, cu, or passivationCut. Not all materials are legal for all types of blockages. When Blockage Type is placement, the Material field is grayed out. Other denotes undefined materials. Environment variable: blockageMaterial
Layer	Selects the layer associated with the selected material. The layers displayed are those defined in the technology file as having one of the above materials. The default material is `"All"`. The layers listed in this field with `"All"` material include those layers that belong to an LPP with the purpose name `"blockage"`. When Blockage Type is placement, the *Layer* field is grayed out. Environment variable: blockageLayer
Color	Lets you select the color in which the blockage is created. The default value is *grayColor*. The options available in the *Color* field are based on the layer selected in the *Layer* field. If you select a layer in the *Layer* field that support *mask1Color* and *mask2Color*, the *Color* field will have the options, *grayColor*, *mask1Color*, *mask2Color*, *blackColor*, and *multiColor*. If you select multiple layers in the *Layer* field, the options in the *Color* field are a combination of all the colors supported by the selected layers.

© 2023 Cadence Design Systems, Inc.

Field	Description
Compute spacing using	Lets you select method used to compute spacing. You can select *Blockage geometry*, *Effective width*, or *Specify* a spacing value. The *Blockage geometry* option computes spacing using the blockage geometry specified. The *Effective width* option is a blockage attribute that you can assign to any blockage type, such as routing, pin, feedthru, defined by a layer or layers. You can also specify a spacing using the *Specify* option. When *Blockage Type* is *placement*, the *Compute spacing using* field is grayed out. Environment variables: blockageSpacingType, blockageSpacingValue
Max Density	Sets the maximum percentage of the total design area that the metal in the specified layer may occupy. Max Density applies to blockage types placement and screen. Max Density does not apply to blockage types routing, pin, and feedthru. Environment variable: blockageMaxDensity
Allow PGNet	Allows shapes with signal types power and ground to be placed within the blockage area. When creating a routing blockage with *Allow PGNet* selected, the devices obey the setting and allows power and ground nets to pass through. Environment variable: blockageAllowPGNet
Attach to Owner	Assigns one instance or one cluster as the blockage's owner, depending upon the type of blockage. Environment variable: blockageAttachOwner
Keep Owner	Retains the owner information. When you create multiple blockages, the owner is automatically assigned to the additional blockages. Keep Owner remains grayed out until you select the cluster radio button in the Attach to Owner field. Environment variable: blockageKeepOwner
Pushed Down	Indicates if the blockage has been pushed down from a higher-level block. Environment variable: blockageIsPushedDown

© 2023 Cadence Design Systems, Inc.

Field	Description
Create as ROD Object	Creates the blockage as a ROD object. *Name:* Assigns a unique name to the new blockage in the current cellview if the *Create as ROD Object* check box is selected. This unique name consists of the prefix blockage followed by an integer. For example, the first blockage is named blockage0, the second blockage1, and so on. You can edit the default name, if required, but ensure that the new name is unique in the cellview.
Create Method	Environment variable: blockageCreateMethod ■ *Manual:* Requires that you enter points to define the blockage. ❑ *Snap Mode:* Selects mode to use when you enter points. Valid snap modes are anyAngle, diagonal, orthogonal, L90XFirst, L90YFirst. ■ *Automatic:* Derives points from objects you select, then creates the blockage from these points. ❑ *Derive From:* Sets the type of object or objects from which you can derive the points to create a blockage.Select one of the following choices from the cyclic Field Selected Set bBox, Instance Halo, Instance prBoundary, or Instance bBox, then select the corresponding object or objects. The software derives the points from the objects you select. Environment variable: blockageDerivePoints ❑ *Apply:* Creates the blockage using the derived points.

Related Topics

Blockages

leHiCreateBlockage

© 2023 Cadence Design Systems, Inc.

Create Circle Form

Use the Create Circle form to create a circle in the specified window. You are prompted to point to a location for the center of the circle and the outer edge of the circle.

Field	Description
Shape Type	Enables you to select the shape of the circle that you want to create (full circle, half circle, or quarter circle). Environment variable: circleShapeMode
Conic Sides	Sets the number of segments to use for creating the circle. The value specified in the Conic Sides field of the Layout Editor Options Form is displayed in this field.
Specify radius	Enables the *Radius* field.
Radius	Sets the radius of the circle. Environment variable: circleRadius

Related Topics

Creating a Circle

leHiCreateCircle

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

Create Cluster Boundary Form

Use the Create Cluster Boundary form to create a cluster boundary.

Field	Description
Cluster Name	Lists the clusters in the design.
Cluster Boundary Name	Assigns by default a unique name of `CB_integer`, where `integer` is incremented each time a cluster boundary is created within the cellview.
Points	Lists the points of the cluster boundary.
Overwrite Cluster Boundary	Lets you overwrite the existing cluster boundary. By default, this field is selected. If this field is not selected, the other fields on the form are not enables and you cannot edit the existing cluster boundary. If there is no cluster boundary, this field is not enabled.
Fit Member Instances Into Cluster Boundary	Places all the components in a cluster inside the specified cluster boundary. Components are placed according to their size from left to right and bottom to top. That means that the largest object is placed in the bottom left corner and the smallest object in the top right corner. Abutting instances may result in shorts. A placer such as VCP can be used for connectivity aware placement.
Cross-Highlight Boundaries	Highlights in the layout window the cluster boundary associated with the cluster selected in the form.

© 2023 Cadence Design Systems, Inc.

Field	Description
Create Method	■ *Manual:* Creates a boundary when you enter points in the cellview window. Environment variable: createBoundaryMethod ❑ *Rectangle:* Creates the cluster boundary using two points. ❑ *Polygon:* Creates the cluster boundary using multiple points. Use *Snap Mode* to select a snap mode. Valid snap modes are *anyAngle*, *diagonal*, *orthogonal*, *L90XFirst*, *L90YFirst*. ■ *Automatic:* Creates the boundary using the aspect ratio, origin, and target area. ❑ *Aspect Ratio (W/H):* Sets the aspect ratio. Aspect Ratio W/H is the width-to-length ratio of the cluster boundary. Environment variable: clusterBoundaryAspectRatio ❑ *Origin X and Origin Y:* Sets the origin point.
Create as ROD Object	Creates the boundary as a ROD object. *Name:* Lets you assign a name to the new boundary in the current layout cellview. The name must be unique in the cellview. If you do not edit this field, the system assigns a unique name consisting of the prefix `boundary`. The prefix is followed by a number. For example, the first ROD boundary would be named `boundary0`, the second, `boundary1`, and so on.

© 2023 Cadence Design Systems, Inc.

Field	Description
Area Information	Specifies the area required to create the cluster boundary automatically when you select *Create Method* as *Automatic*. ■ *Area Estimator:* Estimates the area required for the cluster boundary when you click *Estimate*. The options are *PRBoundary Based* and *BBox based* to estimate the area. ■ *Estimated Area of Cluster*: Specifies by default the estimated area of cluster. Estimated area is the sum of area of all the members of the cluster. Either PRBoundary or bbox of the member is used to calculate the estimated area of cluster depending upon the option selected in the *Area Estimator* field. ■ *Utilization*: Specifies the area utilized to create the cluster boundary. Changing the utilization changes the target area and vice-versa. ■ *Target Area:* Specifies the float area value in user units. `Target area = (Estimated area * 100) / Utilization`
Apply	Creates the blockage using the points derived from the aspect ratio, the x and y origin, and the target area.

Related Topics

Creating a Cluster Boundary

leHiCreateClusterBoundary

Create and Edit Clusters Form

© 2023 Cadence Design Systems, Inc.

Create Custom Placement Area Form

Use the Create Custom Placement Area form to create a custom placement area. This form contains the *Dressing Template Editor* button that brings up the Dressing Template Editor sub form.

Field	Description
Create Method	Lets you create an area in the following modes: ■ *Manual*: Lets you create a custom placement area by creating a rectangle using two points with an area size greater than zero. In this mode, the coordinates are not necessarily on placement grid. ■ *Assisted*: Lets you use all the options except the Number of Placement Areas field. If the values are not compatible with the placement grid coordinates the values are adjusted to multiples of the placement grid. ■ *Auto*: Lets you use all the options. If the values are not compatible with the placement grid coordinates the values are adjusted to multiples of the placement grid. Environment variable: customPlacementAreaCreateMethod
Definition Section	This section lets you define the attributes of the custom placement area.
Base Name	Assigns by default a unique name of CPA_integer, where the integer is incremented each time a custom placement area is created within the cellview. You can overwrite this name with a user assigned name, as long as the name is unique to the cellview.
Height	Sets the height of the placement area. Environment variable: customPlacementAreaHeight
Width	Sets the width of the placement area. Environment variable: customPlacementAreaWidth
Number of Placement Areas	Sets the number of custom placement areas to create. Environment variable: customPlacementAreaNumberOfAreas
Channel Width	Sets the width between the placement areas you create. Environment variable: customPlacementAreaChannel

© 2023 Cadence Design Systems, Inc.

Field	Description
Orientation	Sets the orientation of the placement areas. Environment variable: customPlacementFirstUnitOrient
Flip Every Other Area	Lets you flip every other placement area. You can put grounds and power together when you flip the rows Environment variable: customPlacementAreaFlipUnit
Flip First Area	Lets you flip the first custom placement area. Environment variable: customPlacementAreaFlipFirstUnit
Abut boolean button	Lets you abut custom placement areas. Environment variable: customPlacementAreaAbutUnit
Areas	Lets you enter a positive integer that sets the number of custom placement areas that will abut together. Environment variable: customPlacementAreaAbutNumberOfUnits
Abut First Area	Lets you abut the first placed custom placement area. Environment variable: customPlacementFirstUnitOrient
Templates Section	This section lets you specify the rails and allowed component types automatically placed in the custom placement area.
Type	Lets you select *Rail*, *Allowed Component*, or *Both*. If pre-defined rails or components exist, they are listed in the *Pre-Defined* box. Select the ones you want to use for the custom placement area and move them to the *Selected* box by using the arrow icons. Once they are in the *Selected* box, they are available to be placed in the cellview. You can then add the components individually selected in the *Selected* box.

Related Links

leHiCreatePlacementArea

Dressing Template Editor

© 2023 Cadence Design Systems, Inc.

Create and Edit Clusters Form

Use the Create and Edit Clusters form to create and edit clusters and sub-clusters.

Field	Description
Clusters	Lists all the current clusters defined in the cellview. Use this field to change the information displayed in the current cluster editing area, which is the area immediately below the *Clusters* list.
Type	Displays the types of clusters that are assigned to the cluster selected in the *Clusters* list. Choose the clusters from the *Clusters* list and then use the *Type* list to change what type they are. ■ *inclusive*: Specifies that instances other than instances in this cluster may be placed in its associated cluster boundaries. ■ *exclusive*: Specifies that only instances in this cluster may be placed in its associated cluster boundaries. ■ *suggested*: Specifies that instances in this cluster may be placed outside of its associated cluster boundaries.
Current Cluster Editing Area	Displays clusters and instances in tree table format, and allows you to edit the cluster and sub cluster. This is the area that is displayed immediately below the *Clusters* list.
Name	Creates a cluster of the name you enter in the field.
Type	Assigns a type to the newly created cluster. The options are *inclusive*, *exclusive* and *suggested*.
Create	Creates a new cluster once you have entered the name and the type in the *Name* and *Type* fields, respectively.
Create Sub Cluster	Creates a new sub cluster once you have entered the name and type in the *Name* and *Type* fields, respectively.
Create Cluster Boundary	Opens the Create Cluster Boundary form, which allows you to enter a cluster boundary in the cellview. The Create Cluster Boundary form is seeded with the current cluster information in the cellview.
Remove From Cluster	Removes instances or sub clusters selected in the Current Cluster Area from the cluster.

© 2023 Cadence Design Systems, Inc.

Field	Description
Flatten	Removes instances or sub clusters you have selected in the Current Cluster Area from the currently displayed cluster by moving the selected elements to the parent one level up. If no parent exists, the objects will be part of the design pool. You can also control whether to move associated blockages one level up or not. If you move the blockages to the design pool, the blockages will exist without any owner.
Delete	Removes instances or sub clusters you select in the Current Cluster Area from the currently displayed cluster.
Cross-Highlight Instances	Highlights instances in the cellview window that are included in the currently displayed cluster, when you select the check box.
Cross-Highlight Boundaries	Highlights cluster boundaries in the cellview window that are included in the currently displayed cluster, when you select the check box.
Filter Instances and Clusters	Displays the name, type, library name, cell name, and view name of instances, clusters, and sub clusters in the cellview.
Filter Instances and Clusters	This section displays the name, type, library name, cell name, and view name of instances, clusters, and sub clusters in the cellview. You need to specify the filtering criteria to list the required objects.
Use	Determines which objects from the cellview are displayed in the Filter Instances and Clusters area. The choices are *Clusters*, *Instances*, and *Instances in selected set*.
Name	Specifies a filtering mechanism using a combination of a == or =! cyclic field and a filter field which accepts any regular expression.
Cluster Type	Sets the second filtering mechanism to cluster type when *Clusters* is selected in the *Use* list. The choices are All, inclusive, exclusive, and suggested.
Cell Type	Sets the second filtering mechanism to cell type when *Instances* or I*nstances in selected set* is selected in the *Use* list. The cell type choices are All, none, block, blockRing, cover, coverBump, pad, padSpacer, padAreaIO, core, coreSpacer, coreAntenna, corner, softMacro, blockBlackBox, and via.
Filter By	Displays objects in the Filter Instances and Clusters section as per the specified filtering criteria.

© 2023 Cadence Design Systems, Inc.

Field	Description
Add to Cluster	Moves instances or clusters that you have selected in the Filter Instances and Clusters section of the form to the Current Cluster Editing Area of the form. The selected instances and clusters are added to the trunk, or if you select the branch in the Current Cluster Editing Area, the instances and clusters are added to the correct branch.

Related Links

leHiCreateClusters

Create Cluster Boundary Form

© 2023 Cadence Design Systems, Inc.

Create Encore Report Form

Use the Create Encore Report form to customize the report format to meet your requirements.

Field	Description
Headers	Lets you customize the header information for your report. ■ *>>* button: Lets you move the selected elements from the left pane to the right pane. ■ *Move Up* and *Move Down*: Let you change the order of the elements in the right pane. ■ *<<* button: Removes the selected elements from the right pane. The available elements are: ■ *Empty Line:* Inserts a blank line. ■ *Separator:* Inserts a line of asterisks. ■ *Title:* Inserts the title “Pad Opening Locator”. ■ *Created:* Inserts the date and time the report was generated. ■ *Libname:* Inserts the name of the library.
Columns	Lets you specify what information is included for each pad shape in the report. The use model is the same as that described above for the *Headers* section. The available columns are: ■ *Pad Opening Name:* Specifies the label name (if any), otherwise NO_NAME. ■ *X:* Specifies the X coordinate of the center point of the pad shape. ■ *Y:* Specifies the Y coordinate of the center point of the pad shape. ■ *Area:* Specifies the area of the pad shape. ■ *Width/Height:* Specifies the width and height of the pad shape.

© 2023 Cadence Design Systems, Inc.

Field	Description
File Path	Lets you specify the location and file name for the generated report. Type the full path in the field provided, or use the *Browse* button to navigate to the required location on disk.
Filter	Lets you specify filters to limit the data shown in the report. Click *Set Filter* to open the Pad Opening Info: Filter Form where you can specify the filters to be applied.

Related Topics

Generating a Customized Report for Pad Shapes

Pad Opening Info: Filter Form

Pad Opening Info Form

© 2023 Cadence Design Systems, Inc.

Create Group Form

Use the Create Group form to select figures to form a new group.

Field	Description
Name	Specifies the group name.
Orientation	Sets whether the group is rotated or mirrored. See Orientation Valid Values.

Related Topics

Creating a Group

leHiCreateGroup

© 2023 Cadence Design Systems, Inc.

Create Instance Form

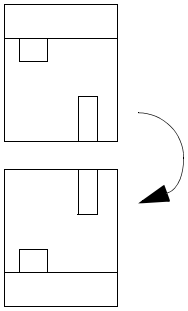
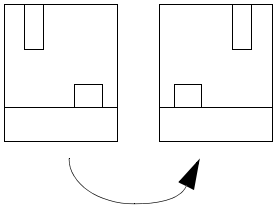
Use the Create Instance form to create an instance in the specified window.

Field	Description
Browse	Lets you select the library, cell, and view names in the library browser.
Library, Cell, and *View*	Sets the library, cell, and view names of the master cell you want to place as an instance in this cellview. A thumbnail of the library, cell, and view that you have selected will be displayed on the Create Instance form. The *Library*, *Cell*, and *View* text fields, which can be auto-completed. When you type the characters, the drop-down list shows the names containing the matching characters at any position in the name (beginning, middle, or end of the name). Select from the drop-down list and specify the name in the text field. The drop-down list is retrieved from cds.lib. The fields are highlighted if there are no matching library or cell or view names for the characters mentioned in the field.
Names	Sets the name assigned to this instance. You can type any name unique to the cellview here or let the layout editor automatically assign instance names that begin with the letter `I`, followed by a number. You can enter multiple names (separated by a space) to place several instances of the same cell. If the design has EMH, you can specify a hierarchical path name for creating the instance at the desired location in the EMH.
Halo	Turns on or off the ability to define halo blockages for the instance.
Define Halo	Opens the Halo Editor Form.
Physical Only	Creates an instance in such a way such that it exists only in the block domain and not in the module domain. This option is enabled only for instances and is disabled for mosaics. Once you create an instance as a physical only instance, you cannot change its domain visibility. This attribute is used for filler (pad and core) and corner cells. Environment variable: instPhysicalOnly

© 2023 Cadence Design Systems, Inc.

Field	Description
Mosaic	■ *Create as Mosaic:* Lets you create an instance array when you select the check box. When this option is selected, the *Physical Only* option is disabled. Environment variable: instMosaicCreation ■ *Rows, Columns:* Sets the number of rows and columns in an array of instances. These fields are enabled when the mosaic check box is selected. ■ *Delta Y, X:* Sets the spacing between rows and columns in an array of instances. The value of *Delta X* and *Delta Y* fields is calculated based on the value (*BBox*, *P&R Boundary*, or *Snap Boundary*) specified in the *Instance Editing Mode* field of the Layout Editor Options form. These fields are enabled when the mosaic check box is selected.
Parameters	Lists the instances in the Parameters section, if they have associated parameters. The orientation buttons at the bottom of the form let you change the orientation of the selected instance.
Cluster	Adds the instance being created to an existing cluster in the design. Cluster drop-down list specifies the name of the cluster to which the instance is added. Choose an existing cluster from the drop-down list or type the name of a new cluster into the field.
Type	Specifies the type of cluster as *inclusive*, *exclusive*, or *suggested*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Rotate and Flip icons	■ *Rotate Left* and *Rotate Right* buttons: Rotate an instance or array 90 degrees counterclockwise and clockwise, respectively. Alternatively, you can use the middle mouse button to perform rotation. Rotation of a mosaic is applied to the complete array. **Note: (Virtuoso RF Option)** For SiP object types, the Rotate button is disabled. In addition, rotation using the middle mouse button during Create Instance is disabled for SiP object types. You can rotate SiP objects by changing the value of the angle parameter in the Parameters section of the Create Instance form. ■ *Flip Vertical* and *Flip Horizontal* buttons: Mirror an instance vertically or horizontally. As you place the instance, you can alternatively press `Ctrl` and click the middle mouse button to mirror the instance sideways/horizontally. Press `Shift` and click the middle mouse button to mirror the instance upside down/vertically. At the first click, the instance mirrors along the X axis.  At the second click, the instance mirrors along the Y axis. 

© 2023 Cadence Design Systems, Inc.

Related Topics

Halo Editor Form

Layout Editor Options Form

Creating Instances

leHiCreateInst

© 2023 Cadence Design Systems, Inc.

Create Label Form

Use the Create Label form to generate labels. The form includes two modes, *Manual* and *Auto*. The following table describes the two modes and the common options available in the form.

Mode	Description
Manual	Derives label name from the text string specified in the *Label (Pattern)* field and places the label at the point of click. *Manual* is the default mode of creating labels. You can place multiple same or different labels on a single object. You can label any type of shape and object except the DB object types 'textDisplay' and 'label'. You can also add labels in empty spaces.
Auto	Generates labels automatically from the connectivity information on objects and places the label at the point of click. *Auto* also automatically generates labels for instances from the instance name or instance master cell and places the label at the center of the instance. Auto mode contains two tabs, From Interconnect and From Instance. There is a check box above each tab. You can select one or both the check boxes to enable label creation from connectivity information or instances or both.
Snap Mode	Locks the direction in which you can draw the scan line. You can set it to *anyAngle*, *diagonal*, *orthogonal* (default), *horizontal*, or *vertical*. The default selection is *orthogonal*. The setting in this list is applicable while using the *Scan Line* and *Auto Step* (*Custom*) methods for generating labels.
Rotate	Turns the label 90 degrees counterclockwise.
Sideways	Mirrors the label along the X axis. Drafting must be off for this button to work.
Upside Down	Mirrors the label along the Y axis. Drafting must be off for this button to work

© 2023 Cadence Design Systems, Inc.

Manual

The following table describes the fields available in *Manual* mode of the Create Label form.

Field	Description
Label Pattern	Specifies the text or indexed pattern to use for assigning label names. You specify the pattern inside the < and > symbols. These symbols are retained in the generated labels and represent bus bits. You can also use the [and] symbols to enclose the pattern, in which case, the symbols are not part of the labels generated. If you use an escape character (\) preceding a sequential pattern, the expansion is overridden. You can include special characters in the label name. Spaces in a pattern are used to distinguish the different label names in the sequence. After you have placed all the labels of a sequence and if *Keep Label* is selected, the labels in the sequence are re-generated and are available to be placed. If, however, *Keep Label* is off, you are prompted to specify the label text again in the *Label (Pattern)* field. With *Keep Label* selected, labels are re-generated only if the *Create Label* command is in repeat mode.
Label Layer/Purpose	This section enables you to specify the layer and purpose for the label.
Use current entry layer	Places the label on the active layer displayed in the Layers panel. This option is selected by default.
Use same layer as shape, select purpose	Places the label on the same layer on which the shape is drawn. You select the purpose from the list. A warning is displayed in CIW if the combination of the shape layer and the selected purpose is not valid. You cannot use this option to add labels to empty spaces in the cellview. This option does not support labeling of instances, vias, groups, and mosaics.

© 2023 Cadence Design Systems, Inc.

Field	Description
Auto (Using stampLabelLayers section from technology file)	Derives the layer and purpose on which to create the label from the *stampLabelLayers* section of the technology file. If the *stampLabelLayers* section does not exist in the technology file, a warning is displayed in CIW. You cannot use this option to add labels to empty spaces in the cellview. With this option selected, you cannot label instances, vias, groups, and mosaics.
Select layer	Places the label on the layer purpose selected from the drop-down list. You can search for specific layer-purpose pair using the search filter in the *Select layer* list box. The layer selection using the search filter is available only if the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form.
Label Options	This section enables you to specify the font, style, and formatting of the label.
Font	Sets the font type of the label text. Choices include *eurostyle*, *gothic*, *math*, *roman*, *script*, *stick*, *fixed*, *swedish*, and *milSpec*. The default font is *stick*.
Height	Sets the height of the label in user units, usually microns. The default value is `1`.
Justification	Sets the location of the label origin relative to the position of the label text. The origin appears as a small square on the label when you place or select it. Choices include *lowerLeft*, *centerLeft*, *upperLeft*, *lowerCenter*, *centerCenter*, *upperCenter*, *lowerRight*, *centerRight*, and *upperRight*. The default is *centerCenter*.
Drafting	Prevents a label from being rotated by more than 90 degrees. This check box is selected by default.
Overbar	Determines how text strings containing underscore characters are displayed. If you select this check box, any underscore character in the text string or the specified pattern displays as an overbar. This check box is off by default.

© 2023 Cadence Design Systems, Inc.

Field	Description
Keep Label	Retains the label text or pattern specified in the *Label (Pattern)* field after you have placed all the generated labels. If this check box is off, labels are consumed from the *Label (Pattern)* field as you place them and the field becomes empty after you have placed all the generated labels. You are then prompted to specify a label text in the *Label (Pattern)* field. This check box is on by default in the *Manual* mode. This check box is not available in the *Auto* mode because in this mode, labels are generated from the connectivity information or from instances.
Attach	Attaches the label to the object you click in the cellview. The object to which you attach the label becomes the parent of the label. With the *Attach* check box selected, the label is automatically attached to the clicked shape. However, if you place the label in empty space, you are prompted to click a shape to which the label is to be attached. The *Attach* check box is on by default in the *Manual* mode. This check box is not available in the *Auto* mode because in this mode, labels are always attached to the shape.
Expand	Generates each label name of the sequence based on the pattern specified in the *Label (Pattern)* field. Consider the pattern `A<0:2> GND VDD`. If the *Expand* check box is selected, the generated labels will be `A<0>`, `A<1>`, `A<2>`, `GND`, and `VDD`. These labels will be assigned one by one to the shapes clicked. However, if the *Expand* check box is not selected, the single label `A<0:2> GND VDD` will be generated and the same label will be assigned to any shape you click, provided *Keep Label* is also selected. The *Expand* check box is on by default in the *Manual* mode. This check box is not available in the *Auto* mode because in this mode, labels are generated from the connectivity information or from instances. **Note:** You can override the sequence expansion even if the Expand check box is selected by preceding the sequence pattern with the backspace character (\).

© 2023 Cadence Design Systems, Inc.

Field	Description
Scan Line	This section enables you to specify label controls pertaining the scan line.
Label Selected Objects	Enables only the pre- or post-selected objects to be labeled when the scan line intersects multiple objects.
Auto Zoom Out	Zooms out the design display area automatically and pan the design if you extend the scan line beyond the visible design area. The design returns to the current zoom level after you release the middle mouse button.
Auto Step	This section enables you to create labels by specifying the distance between consecutive labels in the X and Y directions. This section is available only in the *Manual* mode.
Step	Enables you to specify the distance between consecutive labels by using the following methods: ■ *Custom*: Enables you to indicate the distance between labels and the direction of placing the labels by clicking in the design display area. ■ *Input X&Y*: Enables you to specify the distance between labels along the X and Y axes.
X and Y	Enables you to specify the distance between the labels in the X and Y directions. These fields are grayed out if *Custom* is selected in the *Step* list.

Auto

The *Auto* mode of the Create Label form includes the *From Interconnect* and *From Instance tabs*. There is a check box above each tab. You can select one or both the check boxes to enable label creation from connectivity information or instances or both.

- From Interconnect
- From Instance

From Interconnect

In the *Auto – From Interconnect* mode, if the shape to be labeled is a pin, the label name is derived from the pin's terminal name. For other shapes, the label name is derived from the

© 2023 Cadence Design Systems, Inc.

net name associated with the shape. You can place multiple labels on a single object, but all the labels on an object will have the same label name, derived from the object's connectivity information.

The following table describes the sections available in the *From Interconnect* tab of the Auto mode of the Create Label form. Many of the options available are same as those in Manual mode. For a detailed description of common options, refer to Manual Mode.

Section	Description
Label Layer/Purpose	Enables you to specify the layer and purpose for the label. All options available in the Manual mode are also available here.
Label Options	Enables you to specify the font, style, and formatting of the label. It includes the following options: *Font*, *Height*, *Justification*, *Drafting*, and *Overbar*. These options are same as those available in Manual mode. Additionally, the *Create As Label* option is available. It determines whether the object is created as a label or text display. If this option is selected, a label is created, else a text display is created. When a shape with a textDisplay is removed from a net, the textDisplay is automatically removed as well. If the shape is added back to a net, the textDisplay is also automatically recreated. Environment variable: createAsLabel
Scan Line	Enables you to specify label controls related to the scan line. It includes the options *Label Selected Objects* and *Auto Zoom Out*, which are same as those available in Manual mode.

© 2023 Cadence Design Systems, Inc.

From Instance

The following table describes the sections available in the *From Instance* tab of the Auto mode of the Create Label form. Many of the options available are same as in Manual mode. For a detailed description of common options, refer to Manual Mode.

Section	Description
Label Layer/Purpose	Enables you to specify the layer and purpose for the label. ■ *Use current entry layer*: Places the label on the active layer displayed in the Layers panel. This is the default option. ■ *Select layer*: Places the label on the selected layer purpose. You can search for specific layer-purpose pair using the search filter in the *Select layer* list. The layer selection using the search filter is available only if the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form.
Label Options	Enables you to specify the font, style, and formatting of the label. It includes the following options: *Font*, *Height*, *Justification*, *Drafting*, and *Overbar*. These options are same as the ones in Manual mode. Additionally, the following options are available in the Label Options section: ■ *Create As Label*: Determines whether the object is created as a label or text display. If this option is selected, a label is created, else a text display is created. When a shape with a textDisplay is removed from a net, the textDisplay is automatically removed as well. If the shape is added back to a net, the textDisplay is also automatically recreated. Environment variable: createAsLabel ■ *Show Name Of:* Enables you to specify whether to use the instance name or the instance master cell name for generating the labels.
Scan Line	Enables you to specify label controls related to the scan line. It includes the options *Label Selected Objects* and *Auto Zoom Out*, which are same as those available in Manual mode.

© 2023 Cadence Design Systems, Inc.

Related Topics

stampLabelLayers

Label Creation

leHiCreateLabel

Layers Panel

© 2023 Cadence Design Systems, Inc.

Create Measurement Form

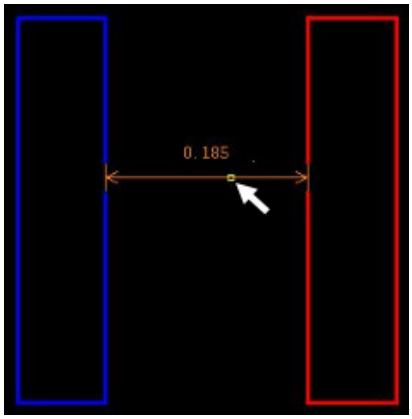
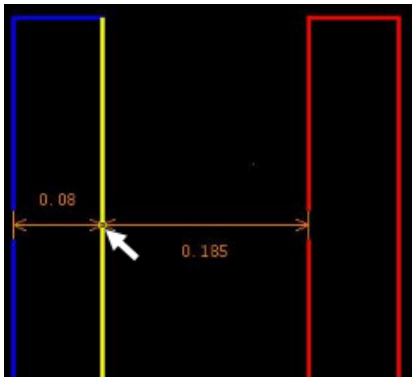
Use the Create Measurement form to create measurement and enter the function in the edited cellview of the specified window.

Field	Description
Type	Lets you specify the display type of the ruler. ■ *Ruler:* Displays the ruler in the design window. ■ *Distance:* Displays the distance between two points on the canvas. ■ *Both:* Displays both the ruler and the distance between two points on the canvas. Environment variable: rulerDisplayType
Segment Mode	Lets you select the type of the ruler created by the measurement. ■ *Single:* Lets you create a single-segment ruler. This is the measurement between points on the canvas. ■ *Multiple:* Lets you create a multiple-segment ruler. In this mode, you can view the measurement between consecutive points specified by the user. ■ *Auto:* Creates a multiple-segment ruler. In this mode, points are automatically calculated. There are two ways in which the ruler is created in the auto mode. The first way is an auto ruler is created between two specified points on the canvas. The second way is when you press the `Shift` key in the auto mode. In this case, an orthogonal ruler is created with all edges visible on the canvas in vertical or horizontal direction. In this case, the direction of the ruler is based on the edge closest to the pointer. ■ The *Mid Point of Distance*, *Dynamic Measurement*, and *Edge Measurement* options are not available when the segment mode is auto. Environment variable: rulerSegmentModeType

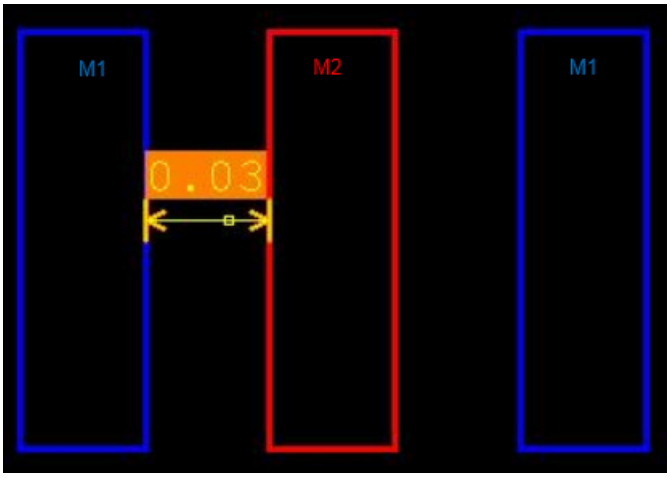
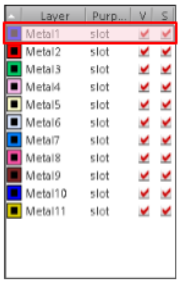
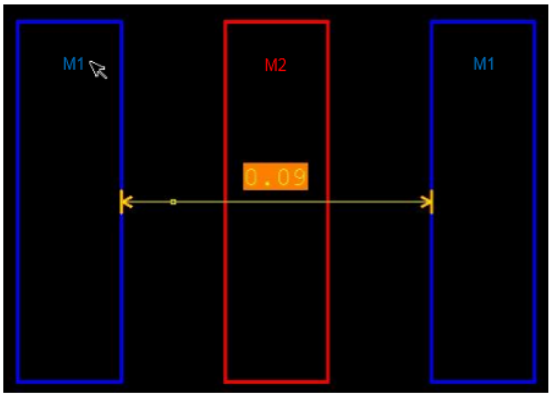
© 2023 Cadence Design Systems, Inc.

Field	Description
Mid Point of Distance	Lets you select the ruler point as a mid-point of the two points that you specify. ■ *Snap Mode:* Controls how the ruler snaps to the grid. ■ *anyAngle:* Creates the ruler segment at any angle to the X or Y axis. ■ *diagonal:* Creates the ruler parallel to the X or Y axis or at a 45-degree angle to the axes. ■ *orthogonal:* Creates the ruler parallel to the X or Y axis. ■ *L90XFirst:* Creates orthogonal two-segment rulers and creates the first segment in the X direction. ■ *L90YFirst:* Creates orthogonal two-segment rulers and creates the first segment in the Y direction. ■ *L90YFirst:* Creates orthogonal two-segment rulers and creates the first segment in the Y direction.
Snap Target	Lets you select the snap target. ■ *None:* Does not snap to the target. ■ *Edges:* Snaps the ruler to only the edges. ■ *Points:* Snaps the ruler to vertices, midpoint of edges, nexus, end points, and origin point of objects. ■ *Edges and Points:* Snaps the ruler to edges, centerlines, vertices, nexus, and end points of objects.

© 2023 Cadence Design Systems, Inc.

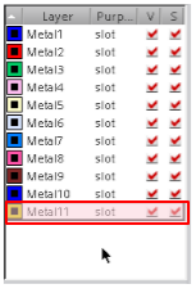
Field	Description
Dynamic Measurement	Enables you to dynamically display measurement between the pointer and the closest edges. If you move the pointer between two shapes, the distance between the two edges is displayed, as shown in the figure below.  If you move the pointer on an edge of a shape, the edge is highlighted in yellow. Also, the distances between the highlighted edge and the two closest edges are displayed. 

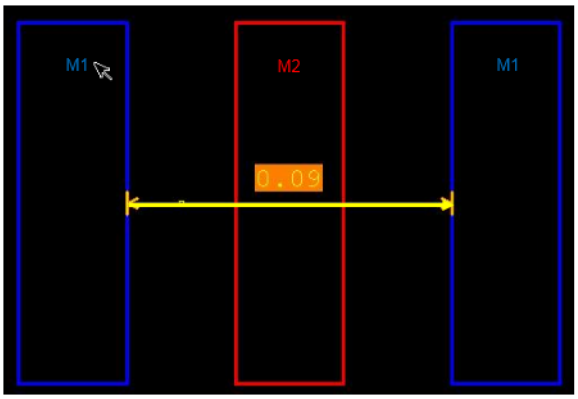
© 2023 Cadence Design Systems, Inc.

Field	Description
Layer Scope	Lets you select the layers to be considered for dynamic measurement. You can select *All*, *Active*, *Edge*, *Palette*, or *Selectable* layers. ■ If you select *All*, all the layers are considered for dynamic measurement. M1and M2 are considered for Dynamic measurement, as shown in the figure below.  ■ If you select *Active*, only M1 is considered for dynamic measurement because M1 is the active layer. M2 is ignored.  

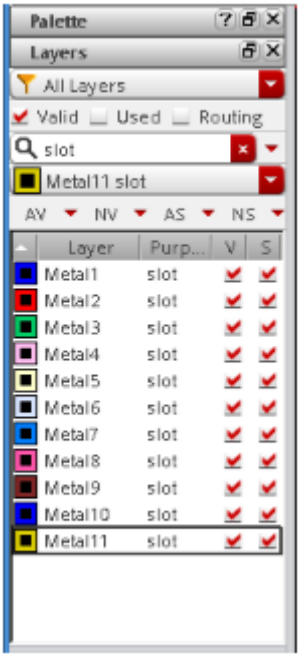
© 2023 Cadence Design Systems, Inc.

Field	Description
	■ If you select *Edge*, M1 is considered for dynamic measurement because the pointer is on an edge of M1, even though it is not the active layer.



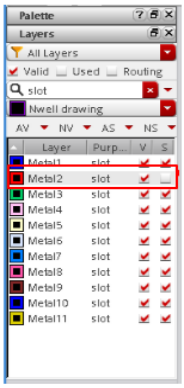


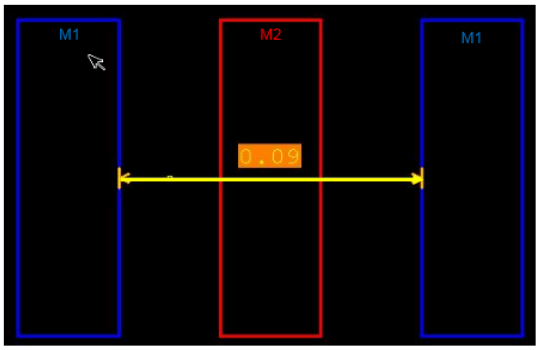
■ If you select *Palette*, all the layers on the Palette are considered for dynamic measurement (M1 to M11/slot). This option is available only if the Use Layer Palette LPPs Only option is selected in the Layout Editor Options form.
.



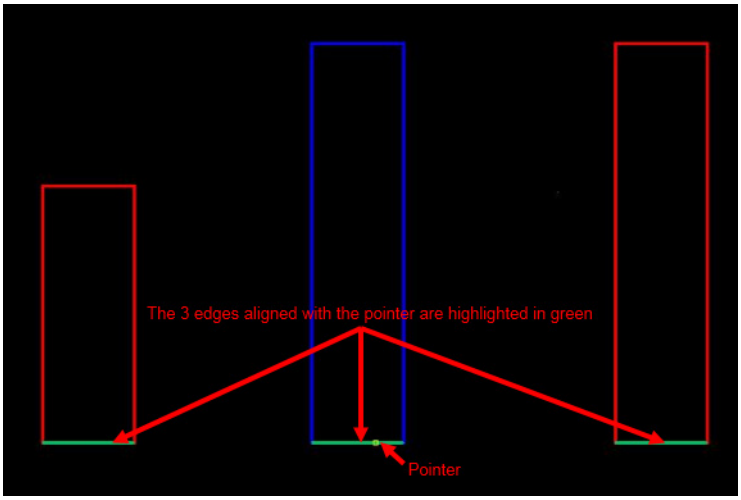
© 2023 Cadence Design Systems, Inc.

Field	Description
	■ If you select *Selectable*, only the selectable layers on the Palette are considered for dynamic measurement. In the figure below, M2 is not considered for dynamic measurement because it is not selectable.





■ If you select the *Highlight Aligned Edges On* option on the Dynamic Display form, the edges of the shapes aligned to the pointer are highlighted, as shown in the figure below.



© 2023 Cadence Design Systems, Inc.

Field	Description
Edge Measurement	Lets you enable the edge measurement mode for rulers. Edge measurement mode allows you to click the edge of an object and a measurement is drawn with the two vertices as the two ends. The first click lets you select the edge and the second click lets you place the measurement on the canvas. Environment variable: rulerEdgeMode The Edge Measurement check box is disabled when the *Snap Mode* is none or when the *Segment Mode* is *Multiple* or *Auto*.
Savable Measurement	Saves the persistent rulers in a cellview when you save the cellview. It is on by default. If this option is off, all the rulers existing in the cellview are deleted when you save the cellview. Environment variable: saveRulerssaveRulers
Display Options	This section lets you define the ruler and snap target visualization settings.
Font Size	Lets you select the font size of the ruler labels. You can choose from *Small*, *Medium*, or *Large*. Environment variable: rulerFontSize
Color	Lets you select the font color of the ruler labels.
Background Color	Lets you select the background color of the ruler labels. Environment variable: rulerLabelBackgroundColor
Bold	Displays the ruler labels in bold face. Environment variable: rulerFontBold
Snap Target Width	Enables you to specify the width of the snap target highlight. The default is `0`. Environment variable: targetWidth
Snap Target Color	Enables you to select the color of the snap target highlight. The default is `yellow`. Environment variable: targetColor
Snap Target Line Style	Enables you to select the type of snap target highlight. The default is `thickline`. Environment variable: targetLineStyle

© 2023 Cadence Design Systems, Inc.

Related Topics

Rulers

leHiCreateMeasurement

How to use Dynamic Measurement feature to measure distance

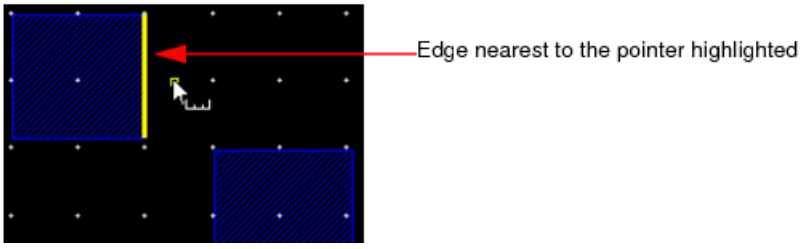
Dymanic Measurement in Create/Edit commands

Ruler Shortcut Menu

© 2023 Cadence Design Systems, Inc.

Ruler Shortcut Menu

Right-clicking in the design display area when the *Create Measurement* command is active displays the *Ruler* shortcut menu. The shortcut menu offers the following options:

Field	Description
Smart snapping off	Identifies and snaps ruler to objects in the vicinity of the pointer, based on the settings in the *Gravity Controls* section of the Layout Editor Options Form. The settings in the *Gravity Controls* section are applied only if *Gravity On* is selected. If *Gravity On* is not selected, ruler snapping to objects is disabled. In that case, you define the start and end points of a ruler by clicking at exact points in the design display area.
Snap to Edges	Highlights the nearest edge within the aperture distance from the pointer. Edges are highlighted depending on the option (*All*, *Edge, Centerline*) selected in the *Edges* section of the *Snapping Options* tab of the *Create Measurement* form. Up to five target edges in proximity of the pointer can be highlighted in succession by pressing the `Spacebar` key. Click to snap the pointer to the highlighted edge. 
Snap to Points	Highlights the points if they are within the aperture distance from the pointer. Points are highlighted depending on the option (*All*, *Vertex*, *Nexus*, *Edge Midpoint*) selected in the *Points* section of the *Snapping Options* tab of the *Create Measurement* form. Up to five target points in proximity of the pointer can be highlighted in succession by pressing the `Spacebar` key. Click to snap the pointer to the highlighted point.

© 2023 Cadence Design Systems, Inc.

Field	Description
Snap to Edges and Points	Highlights all the types of edges and points if they are within the aperture distance from the pointer. The edges and points that are highlighted include: ■ Point Types: *All*, *Vertex*, *Nexus*, *Edge Midpoint* ■ Edge Types: *All*, *Edge, Centerline* Up to eight targets in proximity of the pointer can be highlighted in succession by pressing the `Spacebar` key. Click to snap the pointer to the highlighted target. If a point and an edge are both potential targets, then the point is highlighted before edges.
Snap to Mid Point of distance between 2 targets	Highlights the nearest edge within the aperture distance from the pointer. After you click (twice) to snap the pointer to two such edges, the pointer snaps to the mid point of the distance between the points snapped at the two edges. To identify the target edges, you can press the `Spacebar` key to highlight in succession up to five target edges in proximity of the pointer.
Edge measurement	Lets you enable the edge measurement mode for rulers. Edge measurement mode allows you to click an edge of an object and a measurement is drawn with the two vertices as the two ends. The first click lets you select the edge and the second click lets you place the measurement on the canvas.
Label bold font	Enables you to display the ruler label in bold face.
Label background color	Enables you to select the background color of the ruler label.
Options	Opens the Create Slot Form to configure global ruler settings.

Related Topics

Create Slot Form

Rulers

© 2023 Cadence Design Systems, Inc.

Create Multipart Path Form

Use the Create Multipart Path form to create named paths that have relative object design (ROD) information associated with them. You can create simple one-part paths or complex paths containing several parts, such as a guard ring, transistor, bus, or shielded path.

Field	Description
Template Name	Shows the name of the currently selected template, or shows *New* if no template is selected.
MPP Template	Lets you select a template for a multipart path (MPP) or choose to create a new MPP. Environment variable: mppTemplate
Name	Lets you assign a name for a new MPP in the current layout window. The name must be unique in the cellview. If you leave this field blank, the system assigns a unique name consisting of the prefix `path`, followed by a number. For example, for the first path in the cellview for which you do not specify a name, the system assigns the name `path0`, if unique; for the second, `path1`, and so on.
Technology Library cyclic field	Lets you choose any templates that are defined in the "effective" technology library. Once you have chosen a technology library only the multipart path templates in the chosen library are displayed.
Choppable	Indicates whether or not the master path can be chopped. This field is valid only for a path that has at least one subpart. When a master path is choppable, all its subparts must be choppable. When a master path is not choppable, each of its subparts can be choppable or not. The system ignores this field for single-part paths. The default is choppable.

© 2023 Cadence Design Systems, Inc.

Field	Description
Width	Specifies the width of the master path. It must be a positive integer or floating-point number. If you do not specify the width, the system uses the `minWidth` rule for the master path layer from the technology file. If the `minWidth` rule is not defined in the technology file, and ■ This is the first time you have used this form in your current editing session and you do not specify a width, the system supplies a default and displays an warning saying that the default width is being used. ■ You have used this form earlier in your current editing session, the previous value for this field is displayed and the system displays a warning.
End Type	Defines how the ends of the master path are created. The default value is *flush*. If you want to specify a beginning or ending extension, you must choose *variable*. Subparts inherit the end type of the master path. ■ *truncate:* Creates ends that end at the first and last points of the path. ■ *extend:* Creates ends that extend beyond the first and last points of the path by one half the path width. ■ *round:* Creates octagonal-shaped ends that extend beyond the first and last points of the path by one half the path width. ■ *variable:* Creates path ends that extend beyond or retract from first and last points of the path by the value you enter in the *Begin Extension* and *End Extension* fields.
Offset	Specifies the distance by which the master path is offset from the points you enter (the point list), in relation to the direction of the master path. The direction of the master path is determined by the order in which you enter the first and last point of each segment. The *Offset* value must be a signed integer or floating-point number; the default is `0`. A positive value defines the master path to the left of the point list; a negative value defines the master path to the right of the point list.

© 2023 Cadence Design Systems, Inc.

Field	Description
Begin Extension	Specifies the distance by which the starting edge of the master path extends beyond its first point. The value must be a zero or a positive integer or floating-point number; the default is `0`. This field is ignored unless *End Type* is *variable*.
Justification	Defines the part of the master path (the centerline, left edge, or right edge) that is offset from the points you enter (the point list). *Justification* is relative to the direction of the master path. The direction of the master path is determined by the order in which you enter the first and last point of each segment. The default is *center*. Together, the *Offset* and *Justification* fields determine the location of the master path in relation to the point list. ■ *center:* Specifies that the centerline of the master path is offset from the point list by the value you enter for the *Offset* field. ■ *left:* Specifies that the left edge of the master path is offset from the point list by the value you enter for the *Offset* field. ■ *right:* Specifies that the right edge of the master path is offset from the point list by the value you enter for the *Offset* field.
End Extension	Specifies the distance by which the ending edge of the master path extends beyond its last point. The value must be a zero or a positive integer or floating-point number; the default is `0`. This field is ignored unless *End Type* is *variable*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Connectivity	Determines whether to associate the master path with a net and whether to make it into a pin. The default is *None*. When you choose *Net*, the Create Multipart Path form expands to show the net connectivity fields. When you choose *Pin*, the form expands to show both the net and pin connectivity fields. ■ *None:* Indicates that you do not want to define connectivity. ■ *Net:* Indicates that you want to associate the master path with a specific net. When the *Connectivity* is set to *Net*, the Connectivity section adds to the form where you are required to specify the net name. ■ *Pin:* Indicates that you want the master path to be a pin. When the *Connectivity* is set to *Net*, the Connectivity section adds to the form where you are required to specify the net name.
Load Template	Opens the Load Multipart Path Template form to let you enter the path and file name for an ASCII file containing one or more MPP template definitions. When you click *OK*, the system adds the templates in the file to the MPP templates currently in virtual memory and adds the template names to the list of template names in the *MPP Template* field on the Create Multipart Path form. If the filename you entered does not exist, the system displays a dialog box.
Save Template	Opens the Save Multipart Path Templates form to let you save any one template named in the *Template List* field or save all templates named in the *Template List* field either to an ASCII file in your local hierarchy, or, if you have write permission, to your technology library. In both cases, the system saves the template in the temporary technology file in virtual memory. When you save to an ASCII file, and a file with the same name already exists, the existing file will be overwritten.
Subpart	Opens the ROD Subpart form to let you define, change, or delete subparts for the associated master path. For more information, see ROD Subpart Form.
Connectivity Section	This section lets you select the following fields when the *Connectivity* field is set to *Net*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Net Name	Specifies the net with which you want to associate the master path or subpart. This field is required. If the net does not exist, the system creates it, using the name you type in this field. MPP templates do not contain net names, so when want to associate the MPP with a net, display the form and enter a name. When the *Connectivity* field is set to *Pin*, the net and pin connectivity fields are displayed in the Connectivity section. When you choose *Pin*, you must type a net name before you create the MPP in the layout cellview window; if you do not, the system beeps when you click to begin creating the MPP.
I/O Type	Assigns a property used by routers to identify the direction of the signal into or out of the terminal. The default is *inputOutput*. If the terminal specified in the *Net Name* field already exists, it must have the same direction type as defined in the *I/O Type* field. If the direction type is not the same, the system displays an error in the CIW and no MPP is created. ■ *input:* Allows a signal to come in only. ■ *output:* Allows a signal to go out only. ■ *inputOutput:* Allows signals to go in and out simultaneously (bidirectional). ■ *switch:* Allows a signal to go either in or out, but not both simultaneously. ■ *jumper:* Allows a signal to pass through the net.
Access Direction	Defines the access directions for the pin. The default is *Top*, *Bottom*, *Left*, and *Right*. When you choose *Any*, the *Top*, *Bottom*, *Left*, and *Right* options remain on.
Display Terminal Name	Defines whether to create a text-display object using the net name and associate it with the pin you are creating. The default is to not create a text-display object.The system uses the values of four fields to calculate the position of the pin name: *Reference Handle*, *Offset X*, *Offset Y*, and the *Justification* field on the Pin Name Display form.
Display Terminal Name Option	Opens the *Set Pin Label Text Style* form to let you set values for how the pin name is shown, such as the font, height, layer, rotation, and justification.

© 2023 Cadence Design Systems, Inc.

Field	Description
Reference Handle	Specifies the name of the handle on the master path or subpart to which you want to align the pin name. In this context, a handle is a point associated with the object you are defining (the master path or a subpart), such as the starting point or ending point of the object, or a point on the bounding box of the object. The default is *centerCenter*.
Offset X	Specifies a signed floating-point number specifying the distance of the origin of the pin name along the X axis from the point specified by *Reference Handle*. The default is `0`.
Offset Y	Specifies a signed floating-point number specifying the distance of the origin of the pin name along the Y axis from the point specified by *Reference Handle*. The default is `0`.

Related Topics

Setting Pin Label Text Style Options

Accessing Handles on ROD Objects

About Offsetting the Master Path

Multipart Paths

leHiCreateMPP

© 2023 Cadence Design Systems, Inc.

Create Path Form

The Create Path form lets you create a path in the specified window. The form contains the following tabs.

Tab	Description
Path Options	Enables you to define the settings for creating a path.
Slot Options	Enables you to define the settings for creating a path with slots. This tab is available when the *Enable Metal Slotting* option is selected on the Path Options tab. See the Create Slot Form for a description of fields and options. **(Virtuoso RF Option)** The fields in the *Slot Options* tab are not supported for curved shapes.

Path Options

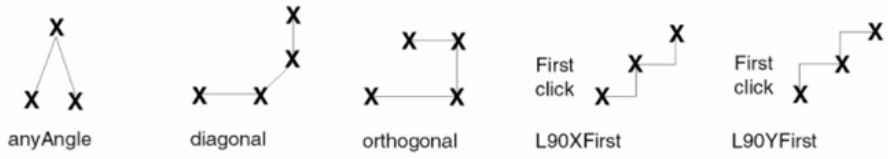
The following table describes the fields available on the *Path Options* tab of the Create Path form:

Field	Description
Width	Enables you to specify the path width. The path width that you specify must be greater than the minimum width (`minWidth`) defined in the technology file for the layer on which you are creating the path. If you do not specify the path width or if you specify a value less than the minimum width of the layer, the path width is set to the minimum width defined for that layer. You can click the *Defaults* button to set the default minimum width for the path anytime during the path creation process.
Fixed Width	Uses for all segments of the path the width that you specify in the *Width* field, even when you change layers. If the *Fixed Width* check box is not selected, the path width is reset to the minimum width (minWidth) defined in the technology file for the layer when you change layers. If you click the Defaults button, it deselects the *Fixed Width* check box and resets the *Width* field to the minimum width (minWidth) defined in the technology file for the current layer.

© 2023 Cadence Design Systems, Inc.

Field	Description
Justification	Determines the edge of the path that you create: left, center, or right. If you set a path offset, *Justification* determines the edge of the path that is offset from the points you enter.
Offset	Offsets the edge that you use to create the path from the entered points. Select the path edge to offset from the *Justification* list.
End Type	Controls how the path ends are created. ■ *truncate:* The path ends and path points end at the same points. ■ *extend:* The path ends extend from the path points by one half the path width. ■ *round:* The path ends extend from the path points by one half the path width, symmetrically in four points at each end. ■ *variable:* The path ends extend from the path points by the value that you specify in the *Begin Extension* and *End Extension* fields.
Begin Extension and End Extension	Enables you to specify a beginning and an ending extension when *End Type* is set to *variable*.
Net Name	Assigns the path to a specific net.
Create as ROD Object	Creates the path as a relative object design (ROD) object. ■ *Name:* Assigns a unique name to the new path in the current cellview if the *Create as ROD Object* check box is selected. This unique name consists of the prefix path followed by an integer. For example, the first ROD path is named path0, the second path1, and so on. You can edit the default name, if required, but ensure that the new name is unique in the cellview. **Note:** When the *Create as ROD Object* option is selected, the path with slots is created as a ROD object with subrectangles as subparts. When the *Diagonal* option is selected in the ROD Subpart form, it specifies whether to create subrectangles on diagonal portions of the path. This option works only when diagonal path segments are 45 degrees and all subrectangles on the path are square.

© 2023 Cadence Design Systems, Inc.

Field	Description
Enable Metal Slotting	Makes available the Slot Options tab. On this tab, you specify the settings for creating a path with slots. Environment variable: slotEnablePath
Snap Mode	Controls how path segments snap to the grid. ■ *anyAngle*: Creates lines at any angle. ■ *diagonal*: Creates lines parallel to the X or Y axis or at a 45-degree angle to the axes. ■ *orthogonal*: Creates lines parallel to the X or Y axis. ■ *L90XFirst*: Creates orthogonal two-segment lines and creates the first line in the X direction. ■ *L90YFirst*: Creates orthogonal two-segment lines and creates the first line in the Y direction.NNTo toggle between *L90XFirst* and *L90YFirst* while you are creating a path, click right. Following are some *Snap Mode* examples:  To switch between the *L90XFirst* and *L90YFirst* snap modes while creating a path, click the middle mouse button.
Acute Angle	Enables you to create paths that enclose acute angles. This check box is available only when *Snap Mode* is set to *anyAngle*. When this check box is not selected, you can create paths with angles ranging from 90 degrees through 180 degrees.
Connectivity	This section provides an option to highlight nets.
Probe Nets	Highlights nets while creating interconnect.

© 2023 Cadence Design Systems, Inc.

Related Topics

Creating Slots

Creating a Path

leHiCreatePath

© 2023 Cadence Design Systems, Inc.

Create Pin Form

Use the Create Pin form to create and place pins. The form supports the following modes of pin creation:

Mode	Description
Manual Mode	Enables you to create and place pins manually.
Auto Mode	Enables automatic pin placement.

Environment variable: pinMode

Manual Mode

The following table describes the fields available on the *Manual* mode of the Create Pin form.

Field	Description
Connectivity	■ *Strong:* Lets you create multiple pins as strongly connected. You can add multiple pin figures to the selected pin, to define all of them as strongly connected. This option is selected by default. ■ *Weak:* Lets you create multiple pins as weakly connected. Environment variable: defCreatePinConnModel
Must Connect All Pins	Lets you apply the `LEF58_MUSTJOINALLPORTS` property to all newly created pin terminals in the layout, which implies that the router must connect all pins of the selected terminal. Environment variable: mustConnectAllPins
Terminal Names	Assigns a name to this pin. Terminal names are used to identify the net to which this pin connects. You can create multiple pins by specifying a space separated list of terminal names, one for each pin.

© 2023 Cadence Design Systems, Inc.

Field	Description
Physical Only	Creates a terminal in such a way so that it exists only in the block domain and not in the module domain. Once you create a terminal as a physical only pin, you cannot change its domain visibility. This attribute applies only to terminals and not to pins. If the terminal name specified in the *Terminal Names* field already exists in the design (either in block or module domain or in both), switching the *Physical Only* option on or off will not be effective. This is because no new terminal will be created, and the option will not change the existing terminal's attribute.
Keep First Name	Specifies that multiple pins or pin figures can be created with the same terminal name. When selected, this option does not remove the first name from the *Terminal Names* field. In the case of strongly connected pins, this option adds pin figures to the same pin shape on the same terminal. In the case of weakly connected pins, this option creates multiple pins on the same terminal.
X Pitch	Sets the horizontal distance, in user units, between the center points of the pins. This field is editable when the terminal name represents a bus. For example, `data <0:7>`.
Y Pitch	Sets the vertical distance, in user units, between the center points of the pins. This field is editable when the terminal name represents a bus.
Create Label	Attaches a label, showing the terminal name, to the pin. ■ *manual:* Lets you place the label manually on the pin. ■ *auto:* Places the label automatically at the center of the pin. For polygonal pins, the label is placed at the boundary of the longest edge of the pin. Environment variable: pinLabelPlacementMode The label is displayed for the pin only if the *Pin Names* option in the Display Options form is enabled.
Options	Opens the *Set Pin Label Text Style* form, which lets you set the font, height, layer name and purpose, justification, and orientation of the pin name.

© 2023 Cadence Design Systems, Inc.

Field	Description
Create as ROD Object	Creates the pin as a ROD object. *Name:* Lets you assign a name to the new shape pin in the current layout cellview. The name must be unique in the cellview. If you do not edit this field, the system assigns a unique name consisting of the prefix rect, dot, or polygon, depending on the type of shape pin you are creating. The prefix is followed by a number. For example, the first ROD rectangle pin would be named rect0, the second, rect1, and so on.
Pin Figure	Specifies the shape of a manually created pin. ■ *rectangle:* Lets you create a rectangular pin. ■ *dot:* Lets you create a dot pin. You drag a rectangle to define the extent of the dot. Click twice on the same point for a zero-area dot pin. ■ *polygon:* Lets you create a polygon pin. ■ *circle:* Lets you create a circular pin.
Attach Pin to Overlapping Shape on Same Layer	Attaches the pin shape to an overlapping shape present on the same layer, the purpose can be different. This option is supported only when a single overlapping shape is available.
Schematic Aware	Specifies whether pin creation must be schematic aware. When the option is selected, the tool uses the I/O type of terminal and signal type of net present in the corresponding schematic for the new pins. If the option is not selected or a corresponding schematic reference is not found, the values specified in the Create Pin form are used.

© 2023 Cadence Design Systems, Inc.

Field	Description
Create Stacked Pin	Creates a stack of pins on the layers selected in the *From* and *To* fields. ■ *From* and *To*: Specify the range of metal layers that contain the pins to be combined into a pin stack. For example, if you set *From* layer field as `metal1 drw` and *To* layer field as `metal4 track`, the tool creates a pin stack between `metal1` and `metal4` layer with the purpose name as `drw`. Stacked pin supports only `metal` and `poly` layers for the *From* and *To* layer fields. ■ *create vias:* Creates a via between the *From* and *To* metal layers based on the geometry of the pin. Vias can be viewed in the layout by selecting the *Via Stacks* field in the Edit Via Stack Properties form.
I/O Type	Assigns a property used by routers to identify the direction of the signal into or out of this cellview. The signal can be input, output, inputOutput (bidirectional), switch (carries data either in or out, but not simultaneously), or jumper (passes data through this cellview), unused, tristate.
Snap Mode	Specifies how the pointer snaps when you create a polygon pin. This option is disabled when you are creating strongly connected pins. ■ *anyAngle:* Creates lines at any angle. ■ *diagonal:* Creates lines parallel to the X or Y axis or at a 45-degree angle to the axes. ■ *orthogonal:* Creates lines parallel to the X or Y axis. This is the default setting. ■ *L90XFirst:* Creates orthogonal two-segment lines and creates the first line in the X direction. ■ *L90YFirst*: Creates orthogonal two-segment lines and creates the first line in the Y direction.
Access Direction	Assigns a property used to identify the part of the pin to which routers can connect routing.

© 2023 Cadence Design Systems, Inc.

Field	Description
Signal Type	Lets you set the signal type of the net that is used for creating the pin. The options available for signal type are *signal*, *ground*, *power*, *clock*, *analog*, *tieOff*, *tieHi*, *tieLo*, *scan*, *reset*, *optical*, *singleModeOptical*, or *multiModeOptical*. The default is *signal*.

Pin Creation in Virtuoso Photonics Option

In *Manual* mode of pin creation, additional options are displayed when the layer function of the entry layer is waveguide or the pin is created on a net whose signal type is optical, that is, the *Signal Type* option is set to *optical*, *singleModeOptical*, or *multiModeOptical*.

The following table describes the properties associated with a photonic port.

Field	Description
Photonic Width	Enables you to specify the width of the facet of the waveguide that connects to this pin. Environment variables: photonicPinWidth
Photonic Angle	Enables you to specify the angle of the facet of the waveguide at the intersection of the pin. Environment variables: photonicPinAngle
Photonic Radius	Enables you to specify the radius of the curve of the waveguide center at the point it meets the facet. Environment variables: photonicPinRadius

Auto Mode

When mode is set to auto, a pin is placed at the end of the wire or the path nearest to the point you clicked. The pin width matches the shape and width of the wire or path and it is placed on the same layer as the shape. This feature works only on wire or path ends parallel to the X or Y axis. The access direction is automatically set in this mode. You can create auto pin on rectangle, polygon, path, pathseg, and ellipse.

Note: X and Y Pitch, Pin Shape, and Access Direction options are not available in auto mode.

© 2023 Cadence Design Systems, Inc.

The following table describes the fields available in Auto mode in addition to the fields available in Manual mode.

Field	Description
Pin Layer/Purpose	Lets you choose LPP to be used for auto pin creation. You can select the *Use current entry layer* option if you want to use the entry layer for auto pin creation. You can use the *Use same layer as shape, select purpose* to use the layer of the selected shape and select a purpose from the list of purposes. Environment variables: pinLPP
Pin Size	Enables you to create a pin of type rectangle and of width and height you specify at the point of click or where the scan line crosses the shape. When you use this option, there can be multiple overlapping shapes. The LPP of the chosen shape will be of the shape with dynamic highlight and it will be shown in drag mode. In case of scan line, instead of creating a rectangular pin at the either ends of the path and pathseg, a rectangular pin of width and height where the scan line crosses the shape will be created. Environment variables: autoModePinSize, autoPinRectHeight, autoPinRectWidth
Justification	Enables you to set the placement of the rectangular pin. You can choose *from lowerLeft*, *upperLeft*, *centerLeft*, *lowerCenter*, *centerCenter*, *upperCenter*, *lowerRight*, *upperRight*, and *centerRight*. The default value is *centerCenter*.
Snap to Wire End	Lets you snap the pin created on pathseg and rectangle to the edge. If this option is not selected, the pin is created at the point of click on the pathseg. The pin width is aligned to the pathseg width or the shorter edge of the rectangle.

Related Topics

Pin Creation

Stacked Pins and Vias

© 2023 Cadence Design Systems, Inc.

Create Pins From Labels Form

Use the Create Pins From Labels form to create pins from the labels and text displays in your layout cellview.

Field	Description
Creation Options	Lets you control the scope of pin generation, and which objects the system uses to create pins. ■ *Objects Within* ❑ *Cellview:* Creates pins only at the top-level cellview. ❑ *Selected Instances:* Creates pins within the currently selected instances. ■ *Object Type:* Specifies which objects are used to create pins. ❑ *Label:* Creates pins from labels. ❑ *Text Display:* Creates pins from text displays. ❑ *Both:* Creates pins from both labels and text displays. ■ *Objects* ❑ *All:* Creates pins for all the objects of the specified type. ❑ *Selected:* Creates pins only for the currently selected objects of the specified type. ■ *Attach Label to Pin:* Attaches a label to the pin. ■ *Define With* ❑ *Layer:* Shows layers in the *Pin Layer* field, and the default width and length in the *Width* and *Length* fields. ❑ *Text:* Shows all pin names in the *Objects Text* field, their default layers in the *Pin Layer* field, and the default width and length in the *Width* and *Length* fields.

© 2023 Cadence Design Systems, Inc.

Field	Description
Objects Layer	Shows the layers on which text labels were found when *Define With* is set to *Layer* and the *Object Type* is set to *Labels* (or *Both*). There is a single entry for each layer of text labels selected. This field is not editable.
Pin Layer	Shows the available layers for pins to be created on. Choose the layer you want from the drop down list.You can search for specific layer-purpose pair using the search filter. If the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form, only the layers selected using the search filter in the *Layers* panel of the Palette are visible in the *Pin Layer* list box.
Width	Sets the width of the pins to be created. The default width is selected in this order: **1.** The net width set in the Constraint Manager (if the pins are created). **2.** The `minWidth` spacing rule for the layer if no constraint exists, **3.** The `pinWidth` environment variable, if it is set.
Length	Sets the length of the pins to be created. By default, the length is the same as the default width.
Ignore	Specifies that no pin is to be created for that entry.

Related Topics

Creating Pins from Labels

leHiCreatePinsFromLabels

© 2023 Cadence Design Systems, Inc.

Create Polygon Form

Use the Create Polygon form to create a polygon in the specified window.

Field	Description
Snap Mode	Specifies how the pointer snaps when you create a polygon. ■ *anyAngle:* Creates lines at any angle. ■ *diagonal:* Creates lines parallel to the X or Y axis or at a 45-degree angle to the axes. ■ *orthogonal:* Creates lines parallel to the X or Y axis. ■ *L90XFirst:* Creates orthogonal two-segment lines and creates the first line in the X direction. ■ *L90YFirst:* Creates orthogonal two-segment lines and creates the first line in the Y direction.
Add an arc segment	Lets you create an arc in the polygon. **(Virtuoso RF Option)** If the *Add an arc segment* option is selected, the curved polygon and the arc segment are created using the LPP selected in the palette.
Net Name	Assigns the polygon to a specific net.
Probe Nets	Highlights nets while you are creating interconnect.
ROD	■ *Create as ROD object:* Creates the polygon as a ROD object. ■ *Name:* Assigns a unique name to the new polygon in the current cellview if the *Create as ROD Object* check box is selected. This unique name consists of the prefix polygon followed by an integer. For example, the first ROD polygon is named polygon0, the second polygon1, and so on. You can edit the default name, if required, but ensure that the new name is unique in the cellview.

Related Topics

Creating a Polygon

leHiCreatePolygon

© 2023 Cadence Design Systems, Inc.

Create P&R Boundary Form

Use the Create P&R Boundary form to create a PR boundary in the specified window.

Field	Description
Create Method	Chooses which method is used to create the PR boundary object. Environment variable: createBoundaryMethod ■ *From Selected Set:* Creates the PR boundary using the rectangular bounding box of the selected set of objects. ■ *Rectangle:* Creates the PR boundary using two points. ■ *Polygon:* Creates the PR boundary using multiple points.
Apply	Creates the PR boundary using the selected objects.
Snap Mode	Selects the snap mode to be used as you enter points, when *Polygon* create method is selected. Valid snap modes are *anyAngle*, *diagonal*, *orthogonal*, *L90XFirst*, and *L90YFirst*.
Create as ROD Object	Creates the boundary as a ROD object. ■ *Name:* Lets you assign a name to the new boundary in the current layout cellview. The name must be unique in the cellview. If you do not edit this field, the system assigns a unique name consisting of the prefix `boundary`. The prefix is followed by a number. For example, the first ROD boundary would be named `boundary0`, the second, `boundary1`, and so on.
Define Halo	Opens the Halo Editor form where you can define a halo blockage for the PR boundary.

Related Topics

Creating a PR Boundary

leHiCreatePRBoundary

Halo Editor Form

© 2023 Cadence Design Systems, Inc.

Create Rectangle Form

Use the Create Rectangle form to create a rectangle in the specified window. You are prompted to point to two coordinates for the opposite corners of the rectangle.

Field	Description
Net Name	Assigns the rectangle to a specific net.
Probe Nets	Highlights nets while you are creating an interconnect.
Size	■ *Enable smart snapping:* Enables smart snapping (identifies objects in vicinity of the pointer intelligently and attaches to them) when the rectangle is created. Environment variable: rectSmartSnapEnable ■ *Active layer only:* Enables smart snapping of the rectangle on only the current active layer only. Environment variable: rectSmartSnapOnActiveLayer ■ *Specify size:* Enables you to click at the desired location on the canvas and place the rectangle. ■ *Width:* Enables you to specify the width of the rectangle. Environment variable: rectWidth ■ *Height:* Enables you to specify the height of the rectangle. Environment variable: rectHeight ■ *Justification:* Determines the base creation point of rectangle that you create: Bottom left, Bottom right, Top left, Top right, Center. Environment variable: rectBaseMode
ROD	■ *Create as ROD object:* Creates the rectangle as a ROD object. ■ *Name:* Enables you to specify the name of the rectangle.

© 2023 Cadence Design Systems, Inc.

Field	Description
Slotting	■ *Enable:* Makes available the slotting options. With these options, you can specify the settings for creating a rectangle with slots. Environment variable: slotEnableRect **(Virtuoso RF Option)** The *Slotting* options are not supported for curved shapes. ■ *Layer:* Enables you to specify an LPP where you want slots to be created in ROD objects. Because no real holes are created in ROD objects in the design, the *Slot Layer* information is used later by manufacturers to punch holes in the metal layer where the slot LPP occurs. You do not need to specify a slot LPP for polygons because real holes are created in the polygon metal layer in the design. ■ *Specify width, area:* Enables you to specify the width and area parameters for a slot. The length of the slot is automatically calculated. Environment variables: slotWidth and slotSpecifyArea
Dimensions	■ *Width:* The spacing between slots is taken into consideration to derive the `minSpacing` value if a one-dimensional spacing table exists in the technology file. Both interslot spacing and the length of the slot are taken into consideration to derive the `minSpacing` value if a two-dimensional spacing table exists in the technology file. ■ *Length:* The spacing between rows is taken into consideration to derive the `minSpacing` value if a 1-D spacing table exists in the technology file. Both row spacing and the width of the slot are taken into consideration to derive the `minSpacing` value if a two-dimensional spacing table exists in the technology file. ■ *Area:* Displays the area of the slot.

© 2023 Cadence Design Systems, Inc.

Field	Description
Spacing	■ *Column:* Represents the width of the metal between slots. The metal spacing between the slots is governed by the `minWidth` and `maxWidth` rules. ■ *Row:* Represents the width of the metal between slots, which is governed by the `minWidth` and `maxWidth` rules. ■ *Edge:* Represents the width of the metal between a slot and the shape edge. Here too the metal spacing is governed by the `minWidth` and `maxWidth` rules. For information about the process rules associated with these fields, see Calculating Dimensions Associated with Slots.

Related Topics

Creating a Rectangle

leHiCreateRect

© 2023 Cadence Design Systems, Inc.

Create Row Form

Use the Create Row form to create a row in the specified window.

The Create Row form has following sections:

Section	Description
Definition	Lets you input basic information for rows, including the site definition name, user specified or software generated names, and orientations.
Templates	Allows you to place rails and/or components automatically into the rows.
Create Method	There are two create methods. You use the *Assisted* mode to create rows that will not take into account any areas or boundaries. You use the *Area Fill* mode to create rows that use an area or boundary to determine the number and placement of the rows.

Definition

The following table describes the fields available on the *Definition* section of the *Create Row* form.

Field	Description
Base Name	Allows you to enter a unique name for each row object you create. Environment variable: rowBaseName
Row Orientation	Allows you to set the row orientation using radio buttons for Horizontal and Vertical. Environment variable: rowFirstUnitOrient
Starting Site Orientation	Allows you to set the orientation of the site and the first row created. Environment variable: rowUnitOrient

© 2023 Cadence Design Systems, Inc.

Field	Description
Options	■ *Flip Every Other Row:* Allows you to flip every other row in the site. Environment variable: rowFlipUnit ■ *Flip The First Row:* Allows you to flip the first row in the site. Environment variable: rowFlipFirstUnit ■ *Abut:* Allows you to abut a specified number of rows. Abut overrides Channel Width. Environment variable: rowAbutUnit ■ *rows:* Allows you to specify how many rows you want to abut. The integer must be positive. Environment variable: rowAbutNumberOfUnits ■ *Abut The First Row:* Allows you to abut the first and second rows. Environment variable: rowAbutFirstUnit ■ *Channel Width:* Allows you to specify a floating-point number that is the space between the rows you create. If you specify the Abut rows, the Channel Width is overridden. Environment variable: rowChannelWidth
Filter By Site Definition	Allows you to filter for core site definitions or pad site definitions. You can see both by retaining *ALL* in the cyclic field. If you select *ALL*, all site definitions in the technology library are displayed. The form opens with the last value retained in the field.
Site Definition Name	Displays all site definition names in the list box. When you have multiple site definitions use the arrows to scroll up and down through the list. You can select one site name at a time.
Technology Library	Specifies a read-only field displaying the library that you have selected in the *Site Definition Name* field.

© 2023 Cadence Design Systems, Inc.

Field	Description
Site Information	Specifies a read-only field displaying the type of site definition (core or pad) that is the current site in the *Site Definition Name* field. This information contains the type (core or pad), the width and height of the site definition, and three types of symmetry. This information is specified in the technology library site definitions. The symmetry can be *SymmetricInX*, *SymmetricInY*, and *SymmetricInR90*.

Templates

The following table describes the field available in the *Templates* section of the *Create Row* form.

Field	Description
Type	Allows you to select *Rail*, *Allowed Component*, or *Both* in the *Pre-Defined* pane of the form. You move them to the *Selected* pane using the arrows. Once they are in the *Selected* pane, they are available to place in the cellview. You can then add the components individually selected in the *Selected* pane.

Create Method

The following table describes the field available in the *Create Method* section of the *Create Row* form.

Field	Description
Assisted	This section provides options to specify the type of toggle button method to create the specified number of rows and sites.
Number of Rows	Sets the number of rows you create, when you use the assisted create method. The number should be greater than or equal to one. When you place your pointer in the cellview window, the wire frame of the row objects appears in the cellview. The total sites of each row should be multiple of the Number of Sites you enter. If you want the maximum number of rows the number entered in this field should be one. Environment variable: rowNumberOfRows

© 2023 Cadence Design Systems, Inc.

Field	Description
Number of Sites	Lets you set the number of sites in a row. The minimum value is an integer >=1. The default is one. Environment variable: rowNumberOfSites
Area Fill	This section provides options to specify the type of toggle button method to use for an area or boundary to determine the number and placement of the rows.
Rectangular	Allows you to create an area in which as many sites possible will fit.
Rectilinear	Allows you to create a polygonal shape with rows that follow the edges of the polygon.
Partition	Allows you to create an area consisting of pre-defined areas. The choices are prBoundary, snapBoundary or clusterBoundary. The field can display more than one clusterBoundary. The snapBoundary will always snap to a region that allows numeric multiple row(s) with the current settings.
Offset	Allows you to place asymmetrical rows in the boundary areas. The offsets are similar to blockages in that they do not allow the rows to come closer to the boundary or cluster edge than the user units defined in the Offset field. ■ *Top:* Sets the distance from the top of the boundary that the rows will not enter. Environment variable: rowPartitionTopOffset ■ *Bottom:* Sets the distance from the bottom of the boundary that the rows will not enter. Environment variable: rowPartitionBottomOffset ■ *Left:* Sets the distance from the left of the boundary that the rows will not enter. Environment variable: rowPartitionLeftOffset ■ *Right:* Sets the distance from the right of the boundary that the rows will not enter. Environment variable: rowPartitionRightOffset

© 2023 Cadence Design Systems, Inc.

Related Topics

Rows

leHiCreateRow

© 2023 Cadence Design Systems, Inc.

Create Slot Form

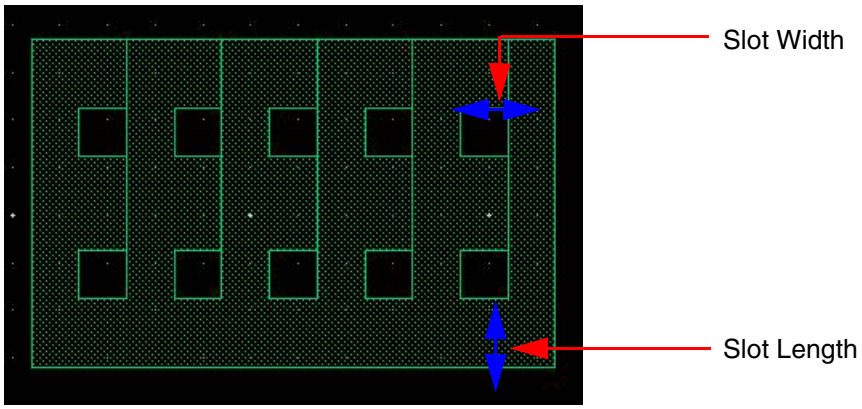
Use the Create Slot form to select a rectangle or path to which you want to add slots. To edit a slot, use the Edit Slot form.

The *Create Slot* command is available only for rectangles and paths. For information about creating slots while creating rectangles and paths, see Creating a Rectangle and Creating a Path.

The following table describes the options available in the Create Slot form and the Edit Slot form.

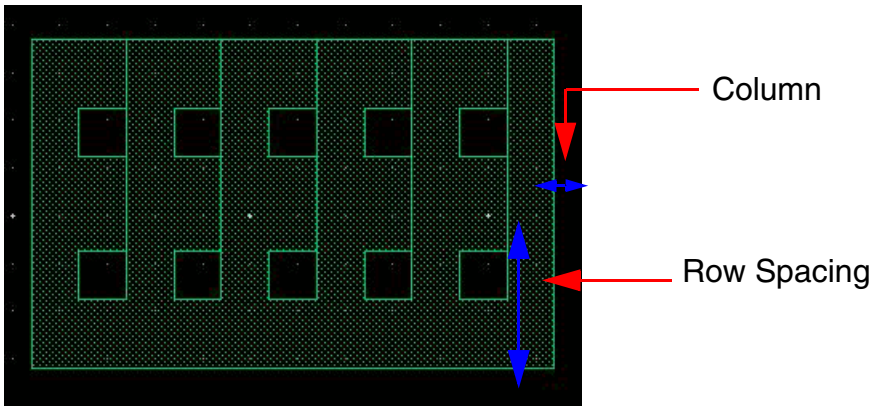
Field	Description
Layer	Enables you to specify an LPP where you want slots to be created in ROD objects. Because no real holes are created in ROD objects in the design, the *Slot Layer* information is used later by manufacturers to punch holes in the metal layer where the slot LPP occurs. You do not need to specify a slot LPP for polygons because real holes are created in the polygon metal layer in the design.
Shape Type	Indicates the type of the resultant shape that is created after slots are added to it. Environment variable: slotShapeMode ■ *Preserve:* Retains the original shape type after slots are added to the shape. A polygon shape remains a polygon after slots are added to it. Similarly, a ROD object remains a ROD object after slots are added to it. ■ *Convert to Polygon:* Converts a shape to a polygon after slots are added to the shape. ■ *Convert to ROD:* Converts a shape to a ROD object after slots are added to the shape.
Specify width, area	Enables you to specify the width and area parameters for a slot in the *Dimensions* field. The length of the slot is automatically calculated. Environment variables: slotWidth and slotSpecifyArea

© 2023 Cadence Design Systems, Inc.

Field	Description
Dimensions	*Width/Length:* Enables you to specify the width and length parameters for a slot. The area of the slot is automatically calculated. For information about the process rules associated with these fields, see Calculating Dimensions Associated with Slots. The following figure illustrates the width and length of slots. 

© 2023 Cadence Design Systems, Inc.

Field	Description
Spacing	The following figure illustrates the types of spacings between slots.



- *Column:* Specifies the distance between the slots in a row.

  Environment variable: slotInterSlotSpacing

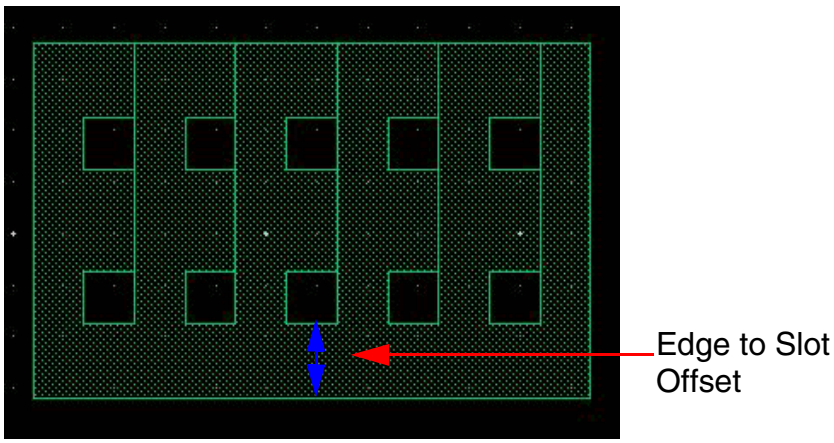
- *Row:* Specifies the distance between two slot rows.

  Environment variable: slotRowSpacing

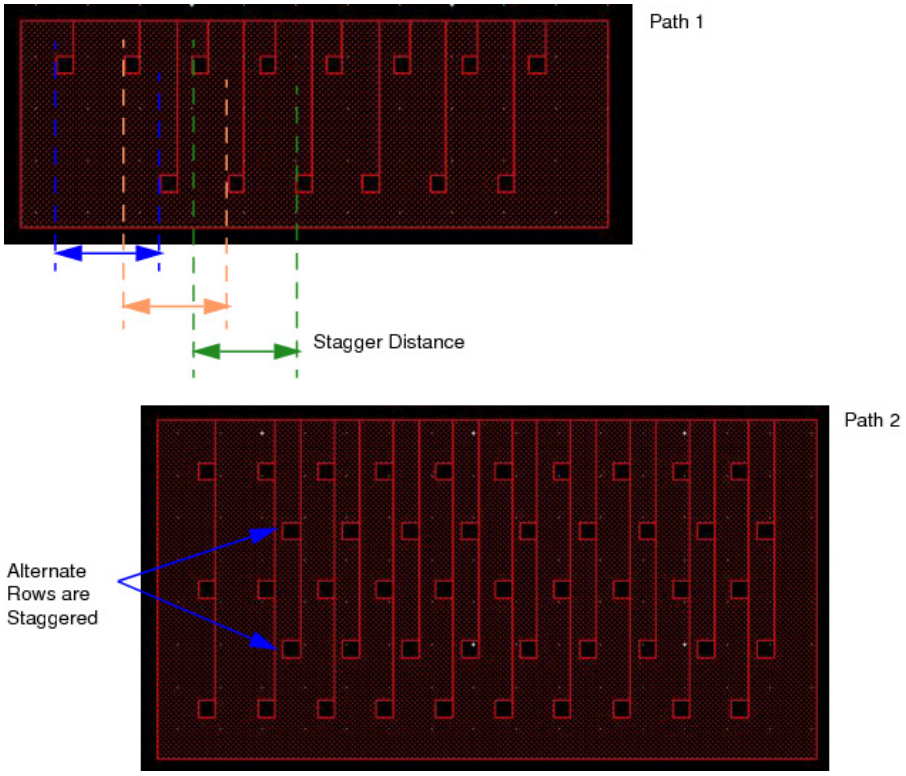
- *Edge:* Specifies the distance between the slot edge and the shape edge.

  Environment variable: slotEdgeToSlotOffset

The following figure illustrates the edge-to-slot distance.



© 2023 Cadence Design Systems, Inc.

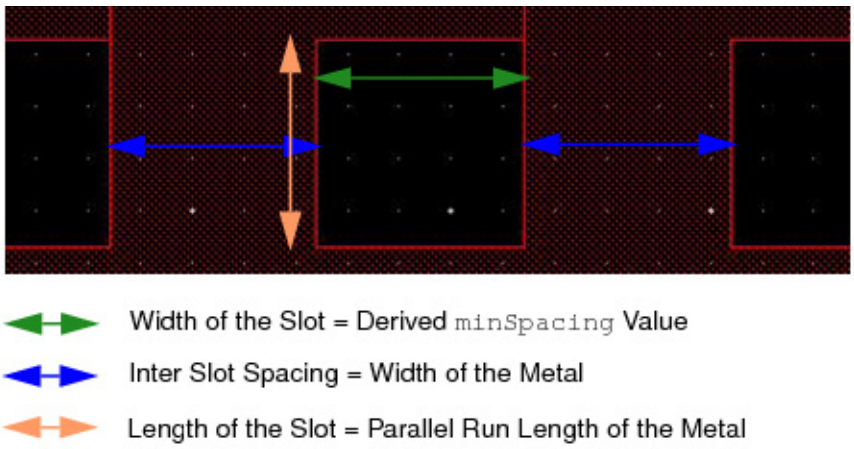
Field	Description
Stagger	Specifies the spacing between the edges of slots in consecutive rows. Only alternate rows are staggered.Staggering of slots is not supported in rectangles. Environment variable: slotStaggerOffset The following figures illustrate staggering of slots in paths. 

© 2023 Cadence Design Systems, Inc.

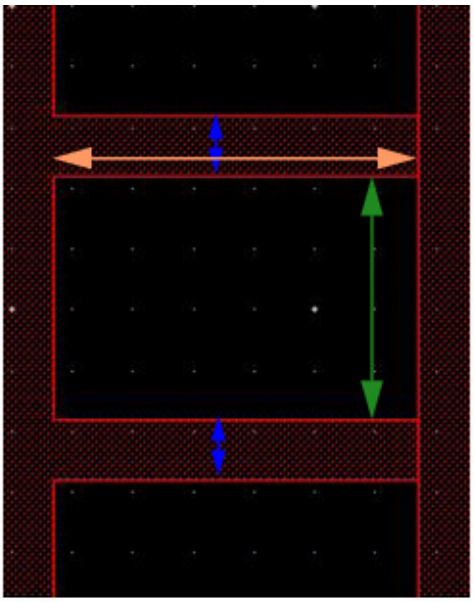
Calculating Dimensions Associated with Slots

The dimensions associated with slots are calculated based on the rules specified below:

Field	Description
Spacing (Column)	Represents the width of the metal between slots. The metal spacing between the slots is governed by the `minWidth` and `maxWidth` rules.
Spacing (Row)	Represents the width of the metal between slots, which is governed by the `minWidth` and `maxWidth` rules.
Spacing (Edge)	Represents the width of the metal between a slot and the shape edge. Here too the metal spacing is governed by the `minWidth` and `maxWidth` rules.
Stagger	Represents the metal spacing around slots. The metal spacing for this optional field is governed only by the `maxWidth` rule.
Width	Based on the values you specify in the fields listed above, the values in the *Width* and *Length* fields are calculated automatically. The *Width* and *Length* values comply with the `minSpacing` rule, the values are calculated. The spacing between slots is taken into consideration to derive the `minSpacing` value if a 1-D spacing table exists in the technology file. Both interslot spacing and the length of the slot are taken into consideration to derive the `minSpacing` value if a 2-D spacing table exists in the technology file.



© 2023 Cadence Design Systems, Inc.

Field	Description
Length	Considers the spacing between rows to derive the `minSpacing` value if a 1-D spacing table exists in the technology file. Both row spacing and the width of the slot are taken into consideration to derive the `minSpacing` value if a 2-D spacing table exists in the technology file.   Length of the Slot = Derived `minSpacing` Value  Row Spacing: Width of the Metal  Width of the Slot = Parallel Run Length of the Metal

Related Topics

Creating Slots

Editing Slots

leHiCreateSlot

© 2023 Cadence Design Systems, Inc.

Create Snap Boundary Form

Field	Description
Create Method	Chooses which method is used to create the snap boundary object. ■ *From Selected Set:* Creates the snap boundary using the rectangular bounding box of the selected set of objects. ■ *Rectangle:* Creates the Snap boundary using two points.
Create as ROD Object	Creates the boundary as a ROD object. ■ *Name:* Lets you assign a name to the new boundary in the current layout cellview. The name must be unique in the cellview. If you do not edit this field, the system assigns a unique name consisting of the prefix `boundary`. The prefix is followed by a number. For example, the first ROD boundary would be named `boundary0`, the second, `boundary1`, and so on.

Related Topics

Creating a Snap Boundary

leHiCreateSnapBoundary

© 2023 Cadence Design Systems, Inc.

Create Via Form

Use the Create Via form to specify the settings based on which you want to create vias. The form contains the following common options.

Field	Description
Mode	Lets you select the mode in which you want to create the via: *Single*, *Stack*, *Auto*, or *Fast Edit*. Environment variable: viaMode In *Auto* and *Fast Edit* modes, the Create Via form does not open automatically. You need to press the `F3` key to open the form.
Single Via Mode	Creates a single via or an array of vias
Stack Via Mode	Places the vias belonging to adjacent layers on top of each other.
Auto Via Mode	Places a via automatically at the intersection of two paths.
Fast Edit Via Mode	Lets you perform interactive via editing.
Options	Opens the Create Via Options Form where you can specify the option for via creation such as the cut spacing method and via definition.

Single Via Mode

The following table describes the tabs and common options available in the *Single* via mode on the Create Via form. The availability of a tab depends on the type of via definition that you select, standard or custom.

Field	Description
System Tab	Lets you define values for standard vias.
User Defined Tab	Contains all user-defined via parameters and CDF parameters, which are not part of Cadence-predefined via parameters.
Cut pattern Tab	Lets you create cut patterns on via arrays.
Array Pattern Tab	Lets you create array patterns on vias. This tab is enabled only if the selected via definition is a cdsGenVia.

© 2023 Cadence Design Systems, Inc.

Field	Description
Compute From	In *Single* and *Stack* via modes, you can choose any of the following: ■ *Row/Col:* Vias are created according to the number of rows and columns. ■ *Width*/*Height*: Vias are created according to the width and height of the layer. ■ *Shape(s):* Vias are created inside the overlap of the shapes on the canvas. ■ *Drawn Area*: Vias are created inside the area, rectangular or polygonal, that you draw on the canvas. In the *Auto* mode, when you click on the canvas, the via is placed in the overlap of two shapes. By enabling *From Area ,* rectangular or polygonal, you can define a region in which the layout editor will place the via in all overlapping regions. Environment variable: viaAutoViaCutArrayDirection All the remaining form field values in both the *Single* and *Stack* modes are calculated automatically by the *Create Via* command when you click the overlapping region of two intersecting shapes. In the *Single* mode, the command calculates the rows, columns, and enclosures, by using the information specified in the form. In the *Stack* mode, the command calculates the rows and columns of the top via and the spacing, sizes, and enclosures of all the vias in the stack, such that all the vias in the stack fit in the area of overlap between the two intersecting shapes. Environment variable: viaComputeFromMode
Constraint Group	Enables you to select the constraint group.
Use All Vias	Enables all vias to be listed in the *Via Definition* list irrespective of the constraint group selected.

© 2023 Cadence Design Systems, Inc.

Field	Description
Via Definition	Displays the valid viaDefs and valid via variants in a tree-like structure from the active constraint group (set in the *Via* list in the Layout Editor Options Form). By default, the viaDefs are listed in the same order as specified in the `validVias` constraint. If the `validVias` constraint does not exist in the active constraint group, the viaDefs are ordered by mask number. You can further update the order by using the appropriate *Sort* option in the Via Definition Selection widget. If any filtering options are set up by using the *Filter* widget, the *Via Definition* list displays only the viaDefs or via variants that satisfy the filtering options. Each node in the tree represents a standard or custom viaDef. A node with a plus sign (+) represents a parent standard or custom viaDef with child viaVariants. If a parent viaDef is not a valid via in the active constraint group, the viaDef is grayed out and cannot be selected. Choosing a viaDef or via variant from the list selects the via from the effective technology file.

© 2023 Cadence Design Systems, Inc.

Field	Description
Filter icon	Displays the via definition selection widget that lets you define how to filter and sort via definitions. The widget contains the following options: ■ *Filter:* Specifies an alphanumerical string to search for matching viaDefs from the `validVias` list. The wildcard characters such as `*`, `$`, and `^` can be used in the search string. The search is case-sensitive. Only viaDefs matching the search string are displayed in the *Via Definition* list on the form. ■ *Sort By:* Sorts the viaDefs listed in the widget based on either viaDef names in alphanumerical order or the mask numbers of the bottom layer of the viaDef. Sorting is done in ascending order. ■ *From* and *To:* Specifies the layer limits; viaDefs existing between the specified layers are listed in the widget. Searches for viaDefs by viaDef name or by the layers between which they are available are exclusive of each other. You can leave the *Filter* field blank and still search for viaDefs between two specific layers. ■ *List box:* Displays the filtered and sorted viaDefs in a tree-like structure. The viaVariants are grouped under the respective parent standard or custom viaDef. The viaDef or viaVariant selected from the list box updates the *Via Definition* field on the form as well as closes the via definition selection widget. The search criteria are not retained when you close or hide the Create Via form.
Via Type / Technology Library	Displays the via type and technology library of the viaDef selected in the *Via Definition* field. The via type can be `Standard` or `Custom`. If you select a via variant, the via type displays `Standard Via Variant` or `Custom Via Variant`, respectively. The technology library indicates the library to which the selected viaDef belongs.

© 2023 Cadence Design Systems, Inc.

Field	Description
Save Via Variant icon	Opens the Save Via Variant form, where you can change some parameter values of a standard or custom viaDef and then save it as a via variant. You can update the via even after it has been created to use the parent viaDef by using the button on the *Edit Via Properties* form or the *Property Editor* assistant. In the *Property Editor* assistant, the button is visible when you click the *ViaDef* value column.
More Options	Lets you associate a net with the via. ■ *Net Name:* Applies a net name to the via you are placing in the cellview. In case of a via stack, it associates the specified net with the entire via stack. If the net constraint exists, the constraint is applied to the via stack. If the specified net does not exist, a new net is created. If no net is specified, there is no association with the via or via stack. ■ *Create as ROD Object:* Creates the via as a ROD object. When the *Create as ROD Object* check box is selected, the *Name* field is enabled where you can specify a unique name to be assigned to the new via in the current cellview. This unique name consists of the prefix via followed by an integer. For example, the first ROD via is named via0, the second via1, and so on. You can edit the default name, if required, but ensure that the new name is unique in the cellview.

© 2023 Cadence Design Systems, Inc.

Field	Description
Reset Parameters to	Enables you to reset the values in the form to the technology database values if you want to revert from the manually updated parameter values to the defaults. When you first open the *Create Via* form, the via parameter values are calculated based on the *Via Parameters Calculation Mode* setting in the *Create Via* section in the Layout Editor Options Form. ■ *Minimum Rules then ViaDef*: All the via parameter fields, such as the number of cut rows and columns, cut sizes, spacing, and enclosures, are reset to the minimum rule or the viaDef default values, whichever is greater. The minimum rules are picked from the constraint group that you select from the *Via* list in the *Create Via* section in the Layout Editor Options Form. The calculated values are never less than the viaDef defaults. If minimum rules are not defined, viaDef defaults are used. ■ *Minimum Rules*: All the via parameter fields are reset to the minimum rule values. ViaDef defaults are not checked. Therefore, the calculated values can be less than the viaDef defaults. For example, if the `minExtensionDistance` rule is `0.05` and the default enclosure in viaDef is `0.1`, `0.05` is used as the enclosure value in the *Create Via* form. The viaDef default enclosure value is ignored. ■ *Minimum Rule Enclosures*: Minimum rules are used to calculate only the enclosure field values. Values in the other fields are not reset. ■ *ViaDef Defaults*: All the via parameter fields are reset to the viaDef defaults. Minimum rules are not checked.

© 2023 Cadence Design Systems, Inc.

Field	Description
	■ *ViaDef Defaults Enclosures*: ViaDef defaults are used to calculate only the enclosure field values. Values in the other fields are not reset. *ViaDef Defaults* and *ViaDef Default Enclosures* reset options are available only when *Compute Enclosures* is off. When *Compute Enclosures* is off, you can specify enclosure values less than the minimum rule values. **Note:** If you select a via variant from the *Via Definition* list, the *Reset Parameters to* list is not available. Environment variable: viaParamCalcMode

The following table explains how the parameter values are reset if one of the *Reset Parameters to* option is selected. The table illustrates scenarios when *Compute Enclosures* is either on or off.

Compute Enclosures	Parameter Values	Reset Parameters to				
		Minimum Rules then ViaDef	**Minimum Rules**	**Minimum Rule Enclosures**	**ViaDef Defaults**	**ViaDef Default Enclosures**
On	Enclosure Values	minimum rule and viaDef default	minimum rule	minimum rule	N/A	N/A
	Other Values	minimum rule and viaDef default	minimum rule	As is	N/A	N/A
Off	Enclosure Values	viaDef default	viaDef default	viaDef default	viaDef default	viaDef default
	Other Values	minimum rule and viaDef default	minimum rule	As is	viaDef default	As is

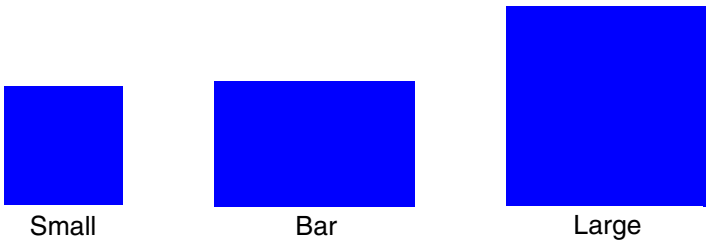
© 2023 Cadence Design Systems, Inc.

System Tab

The following table describes the fields available on the *System* tab in the *Single* via mode of the Create Via form. This tab contains values for standard vias.

Field	Description
Justification	Sets the origin of a via. Cyclic field choices are: `lowerLeft`, `centerLeft`, `upperLeft`, `lowerCenter`, `centerCenter`, `upperCenter`, `lowerRight`, `centerRight`, `upperRight`, and `offset`.
X	Sets the origin point of the via in the X direction when the cyclic field is set to offset. When columns are set to more than one and the cyclic field is set to a direction, X is grayed out and it displays the amount of the directional offset in the X direction.
Y	Sets the origin point of the via in the Y direction when the cyclic field is set to offset. When rows are set to more than one and the cyclic field is set to a direction, Y is grayed out and it displays the amount of the directional offset in the Y direction. For 1x1 via (one row and one column), all *Justification* values except for offset map to the same point. Therefore, the *Justification* field on the *Edit Via Properties* form displays only centerCenter and offset. For a via with one row and more than one columns, `lowerLeft, upperLeft,` and `centerLeft` map to the same point as do `lowerRight, upperRight,` and `centerRight`. Therefore, for such a via, the *Justification* field on the *Edit Via Properties* form displays `centerLeft, centerRight, and centerCenter.`
Cut	This section provides options to specify cut layer settings of the via.

© 2023 Cadence Design Systems, Inc.

Field	Description
Cut Class	Displays cut class choices only if a `cutClasses` constraint has been defined for the cut layer of the selected viaDef in the `layerRules` section of the technology file. Cut class names defined in the technology file must be unique. The Create Via command does not support identical cut class names. The following are examples of cut classes: ■ *Small* is a single cut via where both dimensions are short. ■ *Bar* is a large via that has one long and one short dimension. The shorter dimension is equal to the width or length of `Small`. ■ *Large* is a large via that has two long dimensions, equal to the longer dimension of `Bar`. If you update the *Cut Class* selection, the values in the fields, such as *Rows*, *Columns*, *Width*, *Length*, *Row Spacing*, and *Column Spacing*, are dynamically updated. 
X and Y	Sets the width and length of the via cut, in user units (typically microns). If a *Cut Class* is selected, the cut *Width* and *Length* are automatically updated to the cut size values defined for that cut class in the `layerRules` section of the technology file.

© 2023 Cadence Design Systems, Inc.

Field	Description
Size X and Y	Sets the number of rows or the number of columns of via cuts in a via array. You can specify a value up to `10000` in these fields. These fields do not seed the *Row* and *Columns* fields on the *Cut Pattern* tab. If you change the values in the *Rows* and *Columns* fields, the values in the other fields, such as *Width*, *Length*, *Row Spacing*, and *Column Spacing*, are dynamically updated. You can use the swap button to swap the values in the *Rows* and *Columns* fields. If the swapped values violate the minimum rules, the spacing and enclosure values are re-calculated based on the *Reset Parameters to* setting.
Spacing X and Y	*X* sets the distance between the horizontal edge to the edge cuts in a via. *Y* sets the distance between the vertical edge to the edge cuts in a via. You can use the swap button to swap the values in the *Row Spacing* and *Column Spacing* fields.
Enclosures	This section provides options to specify enclosure settings.
Compute	Calculates enclosures automatically. This is on by default. You can deselect this option to manually specify the enclosure values. Environment variable: viaCompuEnclosures
Show Enclosures/Hide Enclosures button	Toggles the display of the section in the form that sets the directional enclosure distances on Layer 1, the bottom routing layer, and Layer 2, the top routing layer. The swap button swaps the enclosure values between the first and second layer fields. If the swapped values violate the minimum rules, the enclosure values are re-calculated based on the *Reset Parameters to* setting.
Left	■ *Layer 1:*Sets the horizontal distance on the left side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the horizontal distance on the left side of the via from the cut edge to the enclosure edge on the layer designated layer 2 (top routing layer).

© 2023 Cadence Design Systems, Inc.

Field	Description
Right	■ *Layer 1:*Sets the horizontal distance on the right side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:*Sets the horizontal distance on the right side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (top routing layer).
Top	■ *Layer 1:*Sets the vertical distance on the top side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the vertical distance on the top side of the via from the cut edge to the enclosure edge on the layer designated layer 2 (top routing layer).
Bottom	■ *Layer 1:*Sets the vertical distance on the bottom side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the vertical distance on the bottom side of the via from the cut edge to the enclosure edge on the layer designated layer 2 (top routing layer). ■ The *Swap* button swaps the values in the *Left/Right* and *Top/Bottom* fields of layer 1 or 2. If the swapped values violate the minimum rules, the enclosure values are re-calculated based on the *Reset Parameters to* setting.
Implant X	■ *Layer 1:*Sets the horizontal distance from the left and right cut edge to the implant enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the horizontal distance from the left and right cut edge to the implant enclosure edge on the layer designated layer 2 (top routing layer). Implant layer enclosures cannot be specified for custom vias.

© 2023 Cadence Design Systems, Inc.

Field	Description
Implant Y	■ *Layer 1:*Sets the vertical distance from the top and bottom cut edge to the implant enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the vertical distance from the top and bottom cut edge to the implant enclosure edge on the layer designated layer 2 (top routing layer). If you manually specify the enclosure values, they are cached as long as they obey the minimum rules. When you update other via parameters in the form, the manually specified enclosure values are not re-calculated unless they violate the minimum rules. Note that implant layer enclosures cannot be specified for custom vias.
Purpose	This section appears in the *Create Via* form if you select a cdsVia from the *Via Definition* list.
Primary	Enables you to select the purpose for the metal layers. The purposes in this list are populated from the layer purposes defined in the `techLayerPurposePriortities` section of the technology file for the selected cdsVia metal layer.
Cut	Enables you to select the via layer purpose. By default, it displays the *As primary* value to indicate the same purpose as selected in the *Primary* list.
Other	Enables you to select the implant layer purpose. This list appears only if any implant layers are defined for the selected `cdsVia` device.

User Defined Tab

This tab contains all user-defined via parameters and CDF parameters, which are not in the set of Cadence predefined via parameters.

In the case of cdsVia, this tab displays the Pcell parameters that exist on the via. You can change the parameters as required, but the *Create Via* form does not perform any checks for them.

© 2023 Cadence Design Systems, Inc.

Cut pattern Tab

The following table describes the fields available on the *Cut pattern* tab of the *Single* via mode on the Create Via form. Use this tab to create cut patterns on via arrays.

Field	Description
Via Rows	Displays the number of rows in a via array. This is a read-only field that displays the number of rows specified in the *Rows* field on the *System* tab. Environment variables: viaRows
Via Columns	Displays the number of columns in a via array. This is a read-only field that displays the number of columns specified in the *Columns* field on the *System* tab. Environment variables: viaColumns
Pattern	Lets you select a predetermined pattern, such as Custom, Via Def, All On, All Off, Chess Board, H-V Stripes, H Stripes, V Stripes, Top, Bottom, Left, Right, U-L Corner, U-R Corner, L-L Corner, L-R Corner, and Cross. You can now define the default cut pattern for each standard via in the *standardViaDef* section of the DFII ASCII techfile. This default cut pattern is reflected on the *Cut pattern* tab of the *Create Via* form.
Repeat Pattern	Repeats the pattern specified in the *Pattern* field for the number of rows and columns specified in the *Rows* and *Columns* fields. If *Repeat Pattern* is not selected, the values in the *Rows* and *Columns* fields are the same as the values in the *Via Rows* and *Via Columns* fields, respectively.
Rows	Sets the number of rows of the repeated pattern. If *Repeat Pattern* is not selected, this field is disabled.

© 2023 Cadence Design Systems, Inc.

Field	Description
Columns	Sets the number of columns of the repeated pattern. If *Repeat Pattern* is not selected, the field is disabled. Depending on the number of rows and columns you specify in the *Rows* and *Columns* fields the row x column matrix is depicted by check boxes. You can select or deselect the check boxes to define the cut pattern that you want. **Note:** If the number of rows and columns exceeds 10, then the cut pattern is not displayed on the *Cut pattern* tab of the *Create Via* form. 

© 2023 Cadence Design Systems, Inc.

Array Pattern Tab

The following table describes the fields available on the *Array pattern* tab of the *Single* via mode on the Create Via form. Use this tab to create array patterns on vias. The tab is enabled only if the via definition selected in the *Via Definition* field is a cdsGenVia.

Field	Description
Array Spacing	■ *Spacing X:* Specifies the spacing to be applied between the arrays of cuts horizontally. ■ *Spacing Y:* Specifies the spacing to be applied between the arrays of cuts vertically. The swap button enables you to swap the values in the *Spacing X* and *Spacing Y* fields. ■ *Pattern X:* Specifies the number of cuts in the array horizontally. ■ *Pattern Y:* Specifies the number of cuts in the array vertically. For *Pattern X* and *Pattern Y* fields, you can provide multiple numbers separated by spaces. For example, if you specify 3 2 4 in the *Pattern X* field, then the via engine horizontally creates 3 cuts (separated by a “small” cut spacing), then a “large” *Spacing X*, then 2 cuts (separated by a “small” cut spacing), then a “large” spacing X, then 4 cuts (separate by a “small” cut spacing). This pattern is repeated or truncated depending on the number of cut columns specified on the *System* tab. For a 15x15 via, with 3 2 4 specified in the *Pattern X* field, the via engine will create a 3 2 4 3 2 1 via. The swap button enables you to swap the values in the *Pattern X* and *Pattern Y* fields.
Rotate	Turns the via 90 degrees counterclockwise. You can also click the middle mouse button to rotate the via.
Sideways	Mirrors the via or via stack along the X axis. You can also press `Ctrl` and click the middle mouse button to mirror the via or via stack.
Upside Down	Mirrors the via or via stack along the Y axis. You can also press `Shift` and click the middle mouse button to mirror the via or via stack.

© 2023 Cadence Design Systems, Inc.

Stack Via Mode

The following table describes the options available in the *Stack* via mode on the Create Via form.

Field	Description
Start Layer	Lets you select one of the two layers between which you want to place the via stack. The layer selected in the *End Layer* list does not appear in the *Start Layer* list.
End Layer	Lets you select one of the two layers between which you want to place the via stack. The layer selected in the *Start Layer* list does not appear in the *End Layer* list.
Justification	Sets the origin of a via. Cyclic field choices are: `lowerLeft`, `centerLeft`, `upperLeft`, `lowerCenter`, `centerCenter`, `upperCenter`, `lowerRight`, `centerRight`, `upperRight`, and `offset`.
Top Rows and *Top Columns*	Sets the number of rows and columns for the top via in the via stack. The default number of rows and columns is 1. Rows and columns cannot be specified for the individual vias in the via stack; they apply to the complete via stack. The command calculates the rows and columns of the vias below the top via in such a way so that all vias in the stack coincide. Environment variables: viaStackRows and viaStackColumns If the via stack contains custom vias, the default rows and columns of the custom vias are used for creating the via stack. You cannot modify the rows and columns of custom vias by using this form.
Via Definitions	Lets you select viaDefs for each layer pair between the specified *Start Layer* and *End Layer*. For each layer pair, the drop-down list displays the available viaDefs. The viaDefs are populated from the `validVias` constraint in the current constraint group in the technology file.
Cut Class	Displays cut class choices only if a `cutClasses` constraint has been defined for the cut layer of the selected viaDef in the `layerRules` section of the technology file.
Rotate	Turns the via 90 degrees counterclockwise. You can also click the middle mouse button to rotate the via.

© 2023 Cadence Design Systems, Inc.

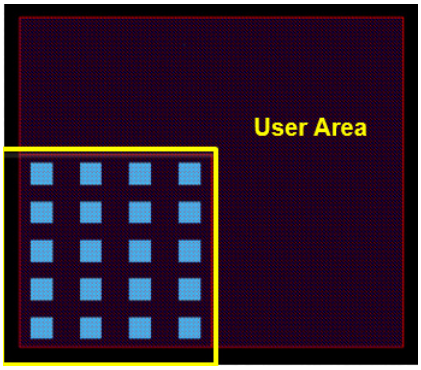
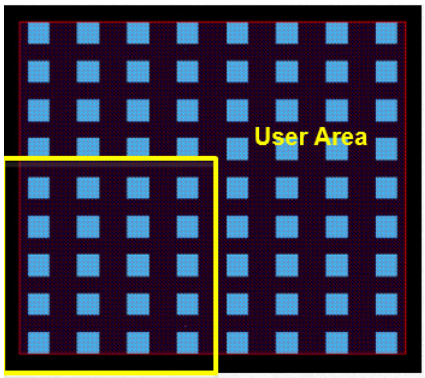
Field	Description
Sideways	Mirrors the via or via stack along the X axis. You can also press `Ctrl` and click the middle mouse button to mirror the via or via stack.
Upside Down	Mirrors the via or via stack along the Y axis. You can also press `Shift` and click the middle mouse button to mirror the via or via stack.

For descriptions of the common options *Compute From*, *Constraint Group*, and *More Options*, refer to Single Via Mode.

© 2023 Cadence Design Systems, Inc.

Auto Via Mode

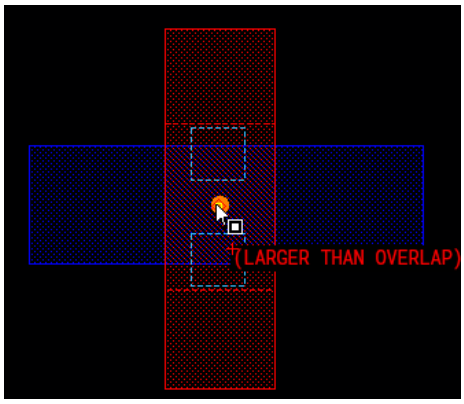
The following table describes the options available in the *Auto* via mode on the Create Via form. In the auto via mode, vias are automatically placed at the intersection of two paths.

Field	Description
Use Entire Overlap	Uses the entire overlap to create the via. If this option is not selected, an intersection between the user area and the overlap box is used to create the via. Environment variable: viaUseEntireOverlap ■ The following figure shows the effect when this option is deselected.  ■ The following figure shows the effect when this option is selected. 

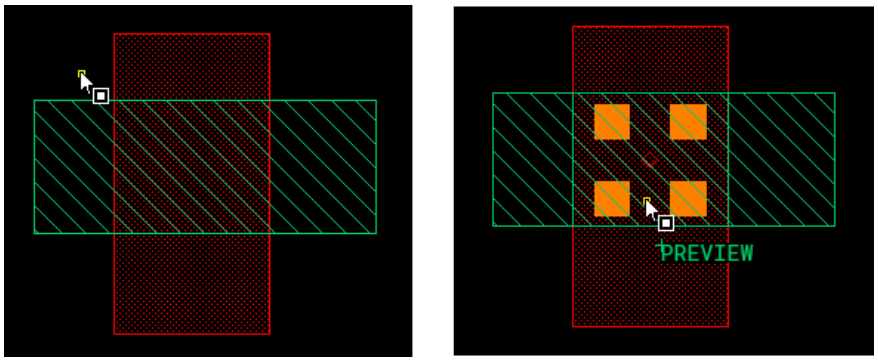
© 2023 Cadence Design Systems, Inc.

Field	Description
Cut Class	Displays cut class choices only if a `cutClasses` constraint has been defined for the cut layer of the selected viaDef in the `layerRules` section of the technology file. Environment variable: viaCutClassInAuto
Via Alignment	Specifies the alignment of the via using the alignment choices: *auto*, *top-left*, *top-center*, *top-right*, *center-left*, *center-center*, *center-right*, *bottom-left*, *bottom-center*, *bottom-right*, *edge-align*, *two-edge-align*, *and two-edge-align-center*. The default is *auto*. Environment variable: viaAutoViaAlignment
Cut Class Orientation	Specifies the preferred orientation for the cut class as one of the following: *auto*, *horizontal*, *vertical,* or *uniform in stack*. Environment variable: viaCutClassOrientation
Cut BBox Orientation	Specifies the preferred direction for the cut box as one of the following: *auto, horizontal*, or *vertical*. Environment variable: viaAutoViaCutArrayDirection
Min Number of Cuts	Specifies the minimum number of cuts to be automatically created for any overlapping area between two shapes. Environment variable: viaAutoViaMinNumCuts
Max Stack Depth	Searches the overlap with the same number of transitions or less than the specified number. You can specify the value of the maximum stack depth. If you specify `1` as the value in the `Max Stack Depth` field, and the layers under `Metal11` are `Metal2` and `Metal5`, only the overlap `Metal1`-`Metal2` is retained. The other overlaps, `Metal1`-`Metal5` (stack depth=4) and `Metal2`-`Metal5` (stack depth=3) are dropped. Environment variables: viaAutoViaEnableMaxStackDepth and viaAutoViaMaxStackDepth
More Options	This section lets you define the via settings like whether to prevent DRC violations and show previews and hints.
Connect Shapes on The Same Net	Connects the shapes that are on the same nets.

© 2023 Cadence Design Systems, Inc.

Field	Description
Fill Polygonal Overlap	Lets you define the fill style when the overlap is polygonal. You can choose to either fill in all the tiles of the polygon overlap, *Fully*, or only the tile under the pointer, *Partially*.
Align Cuts	Specifies the alignment of cuts between different tiles of a polygonal overlap. When this option is enabled, cuts between different tiles of a polygonal overlap are aligned. When this option is disabled, the polygonal overlap is decomposed into a set of largest tiles and vias are created independently in each tile, which may result in cuts not being aligned between different tiles. Environment variable: viaAlignCuts
Prevent DRC Violations On The Overlap	Prevents the creation of a via if it introduces DRC violations on the overlap. If this option is not selected, the via is created even if it introduces DRC violations on the overlap. Environment variable: viaPreventDRCOnOverlap
Prevent DRC Violations With Neighbors	Prevents the creation of a via if it introduces DRC violations with the neighboring shapes. If this option is not selected, the via is created even if it introduces DRC violations with the neighboring shapes. Environment variable: viaPreventDRCWithNeighbors
Prevent Via Larger Than Overlap	Prevents the creation of a via if it is larger than the overlap. 

© 2023 Cadence Design Systems, Inc.

Field	Description
Show Preview	Shows a preview of the via that will be created.  When you move the pointer over the overlap, the preview of the via is displayed Environment variable: viaAutoViaPreviewShow **Note:** In Auto and Fast Edit via creation modes, a label *PREVIEW* is displayed close to the pointer. This ensures that you are aware that the displayed via is a preview and not a real via existing in the layout design. When you click the overlap area, the via is created and the *PREVIEW* label disappears. If the layout editor is not able to display the preview of the via in a reasonable time, the label PREVIEW TOO COMPLEX is displayed. In the Fast Edit mode, the vias are updated instead of being re-created, wherever possible. Video For details on auto via preview and fast edit via creation mode, see the video *Auto Via Preview & Smart Auto Via*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Show Hints	Displays the *Auto Via Preview Bindkeys* hint box on the canvas. Environment variable: viaAutoViaHints ■ *Rows +/-*: Use `Ctrl` + mouse wheel up/wheel down to add or remove via rows. ■ *Column +/-*: Use `Shift` + mouse wheel up/wheel down to add or remove via columns. ■ *Swap Rows and Columns*: Use mouse wheel up/wheel down to swap rows and columns. ■ *Lock/Unlock Via*: Use the spacebar to lock or unlock the via displayed in the preview. When no via is displayed in the preview and the pointer is over an existing via, this feature locks/unlocks the via under the pointer.

Fast Edit Via Mode

Use the Fast Edit via mode if you want to perform interactive via editing. The following table describes the option available in the *Fast Edit* via mode on the Create Via form.

Field	Description
From Selected Vias	Lets you use fast edit command from a set of selected vias. Environment variable: viaFastEditFromSelectedVias

The other options available in the *Fast Edit* via mode are same as the Auto via mode options. For a description of the common options, refer to Auto Via Mode.

Related Topics

Layout Editor Options Form

Save Via Variant Form

Compute Enclosures

Technology File Via Definitions and Via Specifications

© 2023 Cadence Design Systems, Inc.

Via Creation

Create Via Options Form

leHiCreateVia

© 2023 Cadence Design Systems, Inc.

Create Via Options Form

Use the Create Via Options form to specify via creation options during the *Create Via* command.

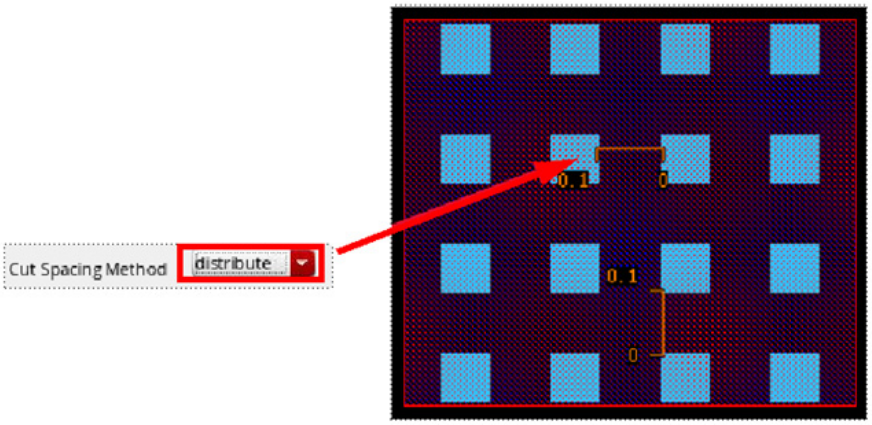
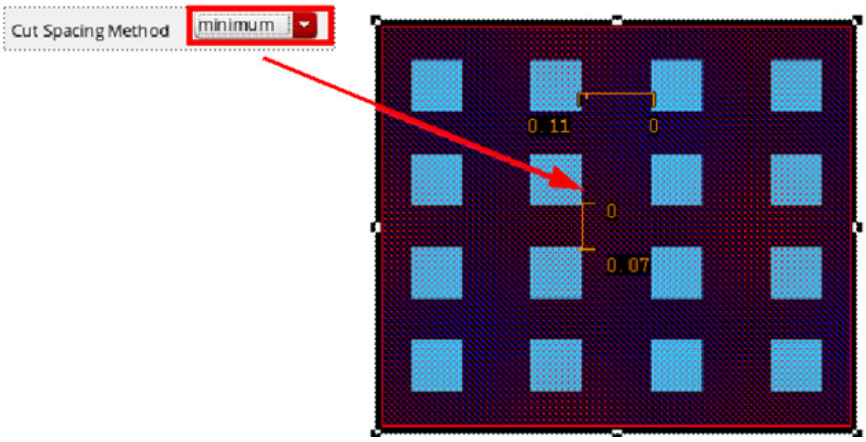
You can also use the form to specify via definitions during the *Create Wire* command. In that case, the form excludes the *Override Variant Parameters*, *Extend Enclosure Beyond Overlap*, and *Fill Overlap With Enclosure* options.

Field	Description
Override Variant Parameters	Lets you to update a via variant and overwrite the existing via parameter values. If this check box is selected, it allows the via engine to update a via variant. Environment variable: autoViaOverrideVariantParameters
Extend Enclosure Beyond Overlap	Lets you to extend the via enclosures beyond the overlap region between two intersecting shapes. The shape layer that exists beyond the overlap region helps to accommodate the extended enclosures. Environment variable: viaExtendEnclosureBeyondOverlap
Create In Route	Lets you to create a via in a new route. If this option is selected, every new via is created in a new route. This check box is selected by default. Environment variable: viaCreateInRoute
Fill Overlap With Enclosure	Lets you to extend via enclosure such that it covers the complete overlap area between the two intersecting shapes, that is, the bounding box of the via enclosure coincides with the bounding box of the overlap region. Environment variable: viaFillEnclosureOverlap
Create Vias Using Top Two Layers Only	Enables you to create vias using top two layers only. Environment variable: viaAutoOnlyOnTopLayers

© 2023 Cadence Design Systems, Inc.

Field	Description
Consider Blockages	Does not create any via in case a blockage exists between the top and bottom metal of the current overlap. Only via and cut layer blockages are considered, metal layer blockages are not considered. Environment variable: viaCheckBlockageTypes
Valid Purposes	Lets you to specify a list of valid purposes for via creation in automated modes: auto, fast edit, compute from shape: single, compute from shape: stack. The via will not be created even if one purpose of the overlap shapes does not belong to the list of valid purposes specified in this field. Environment variables: viaEnableValidPurposes, viaValidPurposes
Ignore Purposes	Lets you specify the list of purposes that will be ignored for via creation in automated modes: auto, fast edit, compute from shape: single, compute from shape: stack. Environment variable: viaIgnorePurposes

© 2023 Cadence Design Systems, Inc.

Field	Description
Cut Spacing Method	Lets you to select whether the cut spacing is spread or is using the minimum spacing rule constraints. You can select *distribute* or *minimum* from the drop-down list. Environment variable: viaCutSpacingMethod  
Transitions	Lets you to select the transitions when creating a via. You can select all the transitions by selecting the *All* check box. You can also select the transitions for which you want the via to be created.
Via Definitions	Lets you to define the default viaDefs/via variants to be used in the *Single*, *Stack*, and *Auto* via creation modes. The selected values appear as the default values in the *Single* and *Stack* modes when you open the Create Via Form.

© 2023 Cadence Design Systems, Inc.

Field	Description
Cut Class	Lets you to set up the default cut class for each viaDef/via variant to be used in the *Single*, *Stack*, and *Auto* via creation modes. The list displays the cut class choices if a `cutClasses` constraint is defined for the cut layer of the selected viaDef in the `layerRules` section of the technology file. Cut class names defined in the technology file must be unique. The Create Via command does not support identical cut class names. ■ The *Cut Class* value selected in the Create Via Options form appears as the default in the *Cut Class* list on the Create Via Form for the respective viaDef/via variant selected from the *Via Definitions* list in the Create Via Form. ■ Some examples of the cut class values include *Small*, *Bar*, and *Large*. If you set the *Cut Class* value as *Auto*, the *Cut Class* value is computed automatically by the via engine; in the *Single* mode, you can explicitly specify the *Cut Class*. For automatic *Cut Class* computation, the via engine chooses the *Cut Class* that maximizes the number of effective cuts in the overlap region.
User Min Cut Spacing	Lets you to specify the X and Y cut spacing for a transition. Select the check box to enable the X and Y fields for the transition for which you want to specify the min cut spacing values.

Related Topics

Create Via Form

Creating or Updating a Via Variant

Setting Up Via Creation Options

Changing Layers and Selecting Vias

© 2023 Cadence Design Systems, Inc.

Cross Section Viewer Form

Use the Cross Section Viewer form to customize the settings for displaying the cross section of a layout.

Field	Description
View mode	Specifies the mode for displaying the cross section of a layout. ■ *Assistant*: Displays the cross section in the Cross Section assistant. ■ *Floating*: Displays the cross section in a floating window that follows the mouse pointer. You can toggle between view modes by using the spacebar or the *Assistant/Floating* option on the shortcut menu. Environment variable: crossSectionViewModeAssistant
Orientation	Specifies the orientation of the cutline as horizontal or vertical. The area underneath a cutline is displayed in the cross section. You can toggle between orientations by using the middle mouse button or *Rotate* on the shortcut menu. Environment variable: crossSectionOrientationHorizontal
Cutline color	Specifies the color of the cutline. Environment variable: crossSectionCutLineColor
Size	Specifies the length of the cutline as *Small*, *Medium*, or *Large*. To modify the size of the cutline, you can also use the bindkey `Control` + `Shift` + wheel button. The length of the cutline is increased or decreased depending on whether you rotate the wheel button up or down. Environment variable: crossSectionSizeCutLine

© 2023 Cadence Design Systems, Inc.

Field	Description
Display	Specifies the details to be displayed for the cross section in the assistant or the floating window: ■ *Layer Name*: Lists the names of the layers that the cutline is cutting through in the layout. Environment variable: crossSectionDisplayLayerName ■ *Unused Layer*: Lists all the layers defined in the technology file but not used in the cross section of the layout. Environment variable: crossSectionDisplayUnusedLayer ■ *Sync with Palette Visibility*: Synchronizes the display of shapes and short markers, in the cross section viewer, with the palette visibility settings. If this option is inactive, shapes and short markers are displayed in the cross section viewer even if their visibility is inactivated in the palette. Environment variable: crossSectionSyncWithPaletteVisibility ■ *Net Name*: Displays the net names of objects that are under the cutline. Environment variable: crossSectionDisplayNetName ■ *Forward Cut*: Displays the vias present between two adjacent layers even if those vias are not directly under the cutline. This helps you to know that a via is present closer to the cutline. If this option is disabled, a via is displayed only when you move the pointer over the via. Environment variable: crossSectionDisplayForwardCut
Config File	Specifies the configuration file used for displaying the cross section of a layout. The file defines settings such as the thickness of layers, layers to be skipped, and layers to be merged on the same line in the cross section viewer.
Select config file on disk 	Lets you select the configuration file that you want to use for displaying the cross section of a layout.

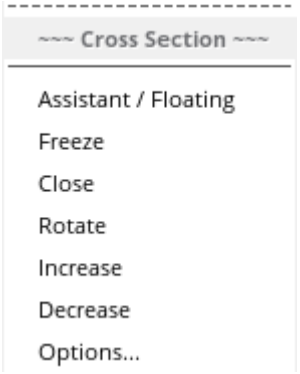
© 2023 Cadence Design Systems, Inc.

Field	Description
Create or edit config file 	Lets you edit the existing configuration fil specified in the *Config file* field. If a configuration file does not exist, clicking  generates a new file named `generatedConfig.il`. It contains certain default entries that you can edit as required. If you specify a new file name in the *Config file* field and click  , a new file is generated by that name and you can edit the settings. For more information, see Cross Section Viewer Configuration File.
Clear	Unloads the configuration file from the memory so that it is not considered for displaying a cross section.
Load	Loads the configuration file specified in the *Config File* field for displaying a cross section. By default, all layers are displayed using the same thickness.

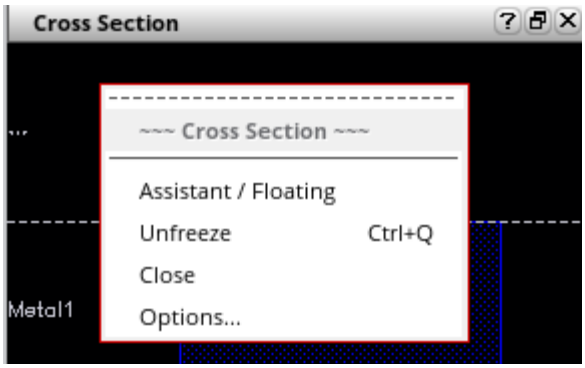
© 2023 Cadence Design Systems, Inc.

Cross Section Shortcut Menu

Use the Cross Section shortcut menu to quickly select certain settings for viewing a cross section.

Field	Description
Assistant/Floating	Lets you toggle between the mode for viewing the cross section of a layout. ■ *Assistant*: Displays the cross section in the Cross Section assistant. ■ *Floating*: Displays the cross section in a floating window that follows the pointer. 

© 2023 Cadence Design Systems, Inc.

Field	Description
Freeze	Freezes the cross section displayed and exits the Cross Section command. Alternatively, you can freeze a cross section by clicking on the canvas. When you freeze a cross section, the cutline is fixed and the command is exited. You can enable the command again by selecting `Unfreeze` from the Cross Section shortcut menu or by pressing `Control+Q`. For a frozen cross section, the shortcut menu is displayed when you right-click within the assistant or the floating window viewer. 
Close	Exits the Cross Section command.
Rotate	Toggles the orientation of the cutline between horizontal and vertical.
Increase	Increases the length of the cutline. The entire area under the cutline is displayed in the cross section viewer.
Decrease	Decreases the length of the cutline.
Options	Displays the Cross Section Viewer form, where you can customize the settings for displaying the cross section of a layout.

Related Topics

Cross Section Viewer

Viewing the Cross Section of a Layout

Specifying Cross Section Viewer Settings

© 2023 Cadence Design Systems, Inc.

Cross Section Viewer Configuration File

© 2023 Cadence Design Systems, Inc.

Custom Show Layer/Custom Show Purpose Form

Use the Custom Show Layer or Custom Show Purpose form to define the criteria for filtering layers in the Layers panel. In the Custom Show Layer form, you can filter on the basis of layers, and in the Custom Show Purpose form, you can filter on the basis of purposes.

Field	Description
Match any of the following	Displays in the Layers panel only those layers for which at least one of the specified criteria match.
Match all of the following	Displays in the Layers panel only those layers for which all the specified criteria match.
Add criteria	Adds a row where you can define a new criterion for displaying layers in the Layers panel. To define a criterion, you need to select an operator and a value for the column. In the Custom Show Layer form, you can select a layer name, and in the Custom Show Purpose form, you can select a purpose name.

Related Topics

Layers Panel

© 2023 Cadence Design Systems, Inc.

Define Net/Pin Connections Form

Use the Define Net/Pin Connections form to define a set of pins on a net to be connected externally. Depending on the pin connectivity command selected, the name of the form might be Define Must Connected Pins, Define Strongly Connected Pins, Define Weakly Connected Pins, or Define Pseudo Parallel Connected Net.

Field	Description
Net	Lets you enter the name of a net you want to select. You can type the following in the *Net* field. ■ A net name. The list box in the form highlights the net and scrolls the list to make the net name visible. ■ A part of a net name (*net*, for example). The list box highlights all the nets with *net* as an element in the name, for example, *net10, net12, net23, net30, net31*, and scrolls the list to make the first example visible. ■ A regular SKILL expression using part of a net name, for example **bias*. The list box highlights all the nets with the string *bias, for example Nbias, NMbias, Pbias*, and scrolls the list to make the first example visible.
Search	Finds and highlights in the list box the names of nets you type in the *Net* field. The list box displays the names of all the nets in the design. To select a net from which to define externally connected pins, you can click the net you want to select or click and hold down the `Ctrl` key to select more than one net. When you select a net in the list box, the software selects the net on the canvas and displays the flight lines for the net.

Related Topics

Defining Must-Connect Pins

Defining Pseudo Parallel Connected Nets

Defining Strongly Connected Pins

Defining Weakly Connected Pins

© 2023 Cadence Design Systems, Inc.

Delete Form

Use the Delete form to delete net interconnect, path segments, or device chains.

Field	Description
Net Interconnect	Deletes any selected paths, instances that are vias, and all non-pin shapes on the net. Pins, component connections (instTerms), and the net is not deleted.
Selected	Deletes any selected objects. ■ *Segment of a Path:* Deletes any selected objects in partial select mode only, and can be used to delete sections of a path while retaining path connectivity.
Preserve Non-Physical-Only Instances	Deletes the physical-only instances (exist only in the block domain) among the instances selected. Instances and parents of instances that exist in both block and module domains are not deleted. A message appears in the CIW to convey that the non-physical-only instance have not been deleted. For more information about creating physical-only instances, see the Create Instance Form. Environment variable: preventDeletionOfLogicalInstance
Chain Mode	Enables you to delete device chains. ■ *All:* Deletes all instances in a chain. ■ *Selected:* Deletes only selected instances. ■ *Selected Plus Left:* Deletes the instance selected and all instances to the left of the selected instance in the chain. ■ *Selected Plus Right:* Deletes the instance selected and all instances to the right of the selected instance in the chain.

Related Topics

Object Deletion

leHiDelete

Create Instance Form

© 2023 Cadence Design Systems, Inc.

Delete All Markers Form

Use the Delete All Markers form to delete all markers in the design window.

Field	Description
Severity	Defines which level of markers the system deletes in the list box and in your design. ■ *all:* Selects all marker types for deletion. ■ *none:* De-selects all marker types. ■ *fatal error:* Deletes all markers of the type fatal error. ■ *critical error:* Deletes all markers of the type critical error. ■ *signed off critical error:* Deletes all markers of the type signed off critical error. ■ *error:* Deletes all markers with error messages. ■ *signed off error:* Deletes all markers of the type signed off error. ■ *warning:* Deletes all markers with warning messages. ■ *acknowledged warning:* Deletes all markers of the type acknowledged warning. ■ *info:* Deletes all markers of the type info. ■ *annotation:* Deletes all markers of the type annotation. Environment variable: markerDeleteSeverities
Search Scope	Defines the scope of the deletion. ■ *cellview:* Deletes markers from the current cellview only. ■ *hierarchy starting from top cellview:* Deletes markers throughout the entire hierarchy. ■ *hierarchy starting from current cellview:* Deletes markers throughout the hierarchy starting with the current cellview. Environment variable: markerDeleteScope

© 2023 Cadence Design Systems, Inc.

Field	Description
Options	Enables you to specify whether the cellview is to be opened in edit or read mode. ■ *Cellview Access Mode:* specifies the mode for opening a cellview. Environment variable: markerDeleteAccessModeIsEdit ❑ *read:* Deletes all the markers and re-opens the cellviews in edit mode with no dialog box. ❑ *edit:* Deletes all the markers after opening the cellview in edit mode. In this mode, you can choose to select the *Confirm Access Mode Change* check box and specify the cellviews to be opened in read mode. ■ *Confirm Access Mode Change:* Opens the *Delete Marker* dialog box for you to confirm whether you want to open the selected cellview in edit mode. Markers are not deleted in cellviews that stay in read mode. This check box is enabled only if *Cellview Access Mode* to set to *edit*. Environment variable: markerDeleteConfirmModeChange
Sources	Specifies the category of markers that are to be deleted in terms of the program that generated them. ■ *all:* Deletes all markers that match the severity and scope criteria. ■ *select:* Signifies that a list of marker owners will be computed and only the sources that have been selected will be deleted. Environment variables: markerDeleteSourcesMode, markerDeleteSources

Related Topics

Deleting All Markers

© 2023 Cadence Design Systems, Inc.

Delete Configuration Form

Use the Delete Configuration form to delete the selected configuration.

Field	Description
Configuration	Displays a list of configuration from which to choose the configuration to be deleted.

Related Topics

Managing Object Information Configuration

© 2023 Cadence Design Systems, Inc.

Delete Marker Form

Use the Delete Marker form to delete individual markers in the design window.

Field	Description
Options	This section enables you to specify whether the cellview is to be opened in edit or read mode.
Cellview Access Mode	Specifies the mode for opening a cellview. ■ *read:* Deletes the selected markers and re-opens the cellviews in edit mode with no dialog box. ■ *write:* Deletes all the markers after opening the cellview in edit mode. In this mode, you can choose to select the *Confirm Access Mode Change* check box and specify the cellviews to be opened in read mode. Environment variable: markerDeleteAccessModeIsEdit
Confirm Access Mode Change	Opens the *Delete Marker* dialog box for you to confirm whether you want to open the selected cellview in edit mode. Markers are not deleted in cellviews that stay in read mode. This check box is enabled only if *Cellview Access Mode* to set to *edit*. Environment variable: markerDeleteConfirmModeChange

Related Topics

Deleting Specific Markers

© 2023 Cadence Design Systems, Inc.

Descend Form

Use the Descend form to select the cellview to descend into.

Field	Description
View Name	Lets you choose a different view of the master cell to open with the *Descend* command.
Open in	Lets you select to open the cellview in a *new tab*, the *current tab*, or a *new window*.

Related Topics

Descending into a Cellview

leDescend

© 2023 Cadence Design Systems, Inc.

Descend Options Form

Use the Descend Options form to enable the option to be able to choose the view opened when descending into instances.

Field	Description
Prompt For View Name	Enables the *Descend* form to appear when you click an instance. You can then choose a different view to open. If there is only one view of the master cell, *Descend* does not prompt you for a view name. By default, *Descend* descends into the same view name as the current cellview.
Open in	Lets you select to open the cellview in a *new tab*, the *current tab*, or a *new window*.

Related Topics

Descending into a Cellview

© 2023 Cadence Design Systems, Inc.

Display Options Form

Use the Display Options form to control the appearance of objects and the behavior of commands in this cellview.

The form comprises the following sections:

Tab	Description
Display Controls	Enables you to specify which design objects or attributes are displayed.
Zoom/Pan Controls	Enables you to activate the dynamic zoom feature and define the zoom mode and scale.
Grid Controls	Enables you to define grid controls and the snap grid spacing.
Resolution	Enables you to change display resolution, which controls the level of design details displayed in a cellview. The resolution determines how much detail of a design is displayed in the cellview.
Snap Modes	Enables you to specify how line segments snap to the grid as you create or edit objects.
Display Abstraction Controls	Enables you to define controls for merge, trim, and slot shapes.
Dimming	Enables you to control luminosity of objects in a cellview.

Display Controls

The following table describes the fields available on the *Display Controls* tab of the Display Options form:

Field	Description
Open to Stop Level	Sets the display level to the Display Levels Stop level. Environment variable: openToStopLevel

© 2023 Cadence Design Systems, Inc.

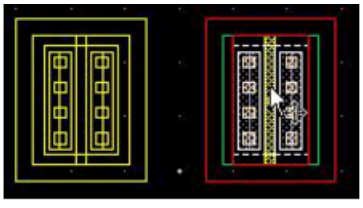
Field	Description
Nets	Shows flight lines between objects on the same net. It also shows outlines over pins. If your design contains many nets, your screen may turn white, causing the `instTerms` on top of the instance to not be seen. To see the `instTerms`, turn on *Instance Pins*. Environment variable: netsOn
Axes	Displays the cellview X and Y axes. Environment variable: dragDisplayNetName
Access Edges	Shows routing edges of pins. Environment variable: accessEdgesOn
Instance Origins	Shows origin of the instances. Environment variable: originMarkersOn
Instance Pins	Shows pins in instances. When the *Instance Pins* option is selected, even if the virtual hierarchy is opaque you can view the virtual pins. Environment variable: drawInstancePins
Instance Pin Names	Shows pin names in instances. Environment variable: drawInstancePinNames
Instance Orientation	Shows orientation of the instances. Environment variable: displayInstanceOrientation
Array Icons	Shows outlines of array cells when *Display Levels* suppresses cell details. Environment variable: iconsOn
EIP Surround (edit in place)	Displays the surrounding design when you edit a cell in place. Environment variable: drawSurroundingOn
Label Origins	Marks the origins of labels with diamond markers. Environment variable: textJustificationOn
Pin Names	Shows terminal names of pins that have pin name text displays or labels. Environment variable: displayPinNames

© 2023 Cadence Design Systems, Inc.

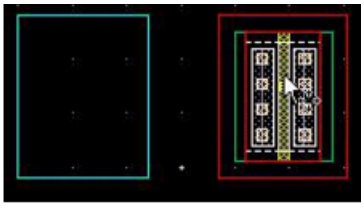
Field	Description
Use True BBox	Displays the instance master bounding box when turned on.When off, displays the cellview bounding box, which can cause large designs to open faster because masters are opened down to the display stop level only. Environment variable: useTrueBBox
Dot Pins	Displays via origins with diamond markers when you set *Display Levels* to show instance content. Environment variable: dotsOn
Cross Cursor	Displays the cursor as dot shaped or as dotted-line cross shaped. Environment variable: cursorShape
Net Expressions	Displays the net expression instead of the terminal name of a pin. When there are net expressions in instances, the terminal name is displayed, not the net expression, even when *Net Expressions* is set on. Environment variable: leDispNetExpr
Row Name	Displays the names of placed rows, which are displayed on the rows in the horizontal or vertical direction depending upon the rotation of the rows. Environment variable: displayRowName
Stretch Handles	Displays the handles on a Relative Object Design parameterized cell (Pcell) that indicate that the Pcell can be stretched. A *stretch handle* is a named set of coordinates assigned to a specific parameter of the Pcell. Stretch handles look like small diamonds. For more information, see Stretchable Parameterized Cells Environment variable: displayStretchHandles
Row Site	Displays the position of row sites, which are represented by a cross marker on the row. Environment variable: displayRowSite

© 2023 Cadence Design Systems, Inc.

Field	Description
Via Shapes	Displays via attributes. When Via Shapes is set to off (the default), vias are displayed as the bounding box of each via. When set to on via attributes are graphically displayed, resulting in detailed images. For large designs, setting via shapes to on will increase the amount of time it takes to render a design. Environment variable: displayViaShapes
Dynamic Highlight	Marks the edge, object, or the point that is selected if you enter a point selection. In addition, it also marks the label of the object. When *Dynamic Highlight* is on and your cellview contains a large number of objects, pointer motion may slow down. You can increase the pointer motion speed by turning this off. Environment variable: dynamicHilightOn
Transparent Group	Enables editing a group without the need for EIP in that group. This setting is applicable only for the current window. By default, it is off. Environment variable: transparentGroup
Dragged Object Ghost	Enables you to view the ghost of the original object when it is being moved or stretched. This option is useful while working with instances. The example below shows an instance being moved. In the first figure, the *Dragged Object Ghost* check box is selected, whereas in the second, the check box is not selected. The check box is selected by default. Environment variable: ghostOfDraggedObjects

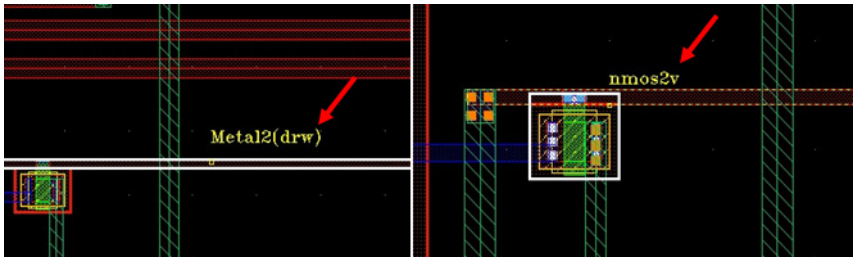


Details of the Original Object (Ghost) Visible

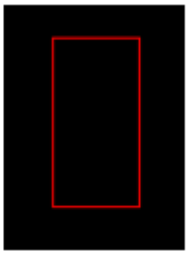
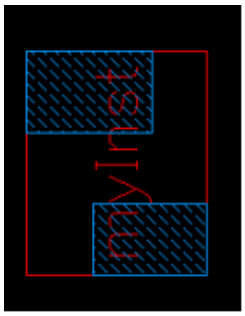


Only Outline of the Original Object (Ghost) Visible

© 2023 Cadence Design Systems, Inc.

Field	Description
Group Name Only	Lets you display only the group name for a figure group when *Display Levels*, *Start* and *Stop*, are set to `0`. The figure group should be set to *Visible* in the Palette Assistant. Environment variable: displayFigGroupName
Selection Hint	Enables displaying selection information when an object is selected by clicking on it. The information is displayed next to the pointer and it lets the you know which object was selected. You can select an object from several overlapping objects. If you want to display selection hint for a wire or via stack, ensure that selection options for Spine and Via Stack are enabled. 

© 2023 Cadence Design Systems, Inc.

Field	Description
Traversed instance BBox	Draws a bounding box around a traversed instance. A traversed instance has size greater than the *Size* set in the *Filter* section. If this check box is selected and the *Stop* display level is greater than zero, the bounding box of the instance displays even if you continue to zoom out of the design, as shown below.  If you zoom out when this check box is not selected, the instance becomes invisible and can be detected or selected only by hovering the pointer over the instance. Select the check box before zooming out.The bounding box does not appear if you select the check box after the instance becomes invisible. This check box is off by default. Environment variable: drawTraversedInstBbox
Traversed Instance Name	Displays the name of the instance for a traversed instance. This option is enabled only when the *Traversed instance BBox* option is selected. Environment variable: drawTraversedInstName 

© 2023 Cadence Design Systems, Inc.

Field	Description
Current Level PR Boundary	Display the PR boundary at the current display level only.
Transparent Clone	Clones the instances that are marked as transparent.
Scroll Percent	Enables you to specify or change the scroll percentage for a window. The default value is `25`, which means that if you scroll the middle mouse button up or down, the design display area will zoom in or zoom out, respectively, by 25 percent. Environment variable: scrollPercent
Instance Drawing Mode	Determines the type of boundary displayed in the top-level cellview. By default, the *BBox* displays. You can set *P&R Boundary* or *Snap Boundary* to display. Environment variable: instanceDrawingMode
Path Display	Determines whether you see path borders and centerlines, path borders only, or path centerlines only. Environment variable: pathCL
Snap Pattern Display	Controls the display of snap patterns. ■ *Full*: Displays the snap pattern boundary, tracks, fins, and width spacing patterns periods. ■ *Tracks*: Displays the snap pattern boundary and tracks. ■ *Boundary*: Displays only the snap pattern boundary. ■ *Periods*: Displays only the width spacing pattern periods. ■ *Periods and Tracks*: displays the width spacing patterns periods and snap pattern tracks. Environment variable: snapPatternDrawingStyle
Set LPP Visibility Mode	Enables you to control the visibility of a layer purpose pair (LPP) based on its visibility and validity. By default, this option is set to *Do not check validity*. With this setting, only the *Visibility* attribute of the LPP is used to determine the visibility of a shape on that LPP; the validity of the LPP is ignored. If you set this option to *Check validity*, only the shapes on valid and visible LPPs are displayed in the design window. Environment variable: lppVisibilityMode

© 2023 Cadence Design Systems, Inc.

Field	Description
Show Name Of	When *Display Levels* is set to show only instance outlines, sets whether the instance name (for example: I1) or the master cell name or both appear on each instance. Environment variable: instLabel
Array Display	■ *Full:* Displays all instances in the array. ■ *Border:* Displays only the instances around the outside edge of the array. ■ *Source:* Displays only the instance at the origin of the array. Environment variable: arrayDisplay
Display Levels	Sets the first (*Start*) and last (*Stop*) levels in the design hierarchy that can be seen in detail. The hierarchy levels are numbered 0 to 32. The top cellview is level 0, instances inside of it are level 1, and so forth. Environment variables: startLevel, stopLevel
Glyph Transparency slider	Controls the transparency level of the glyph. You can choose a value from *0* to *100* using the slider. Environment variable: glyphTransparency

Zoom/Pan Controls

The following table describes the fields available on the *Zoom/Pan Controls* tab of the Display Options form:

Field	Description
Dynamic Zoom	Activates dynamic zoom (*View - Dynamic Zoom*) to objects on the design canvas. The default for this option is on. Environment variable: autoZoomPan

© 2023 Cadence Design Systems, Inc.

Field	Description
Mode	Specifies the dynamic zoom setting to be used. You can choose from: ■ *Pan To Selected:* Whenever an object, or a set of objects, is selected/cross-selected in the Navigator assistant, the view will pan to the selected objects without changing the zoom factor (equivalent to *View – Pan*). ■ *Zoom To Selected:* Whenever an object, or a set of objects, is selected/cross-selected in the Navigator assistant, the view will zoom to the selected object(s). The zoom scale applied is based on the value specified in the *Zoom Scale (%)* field. ■ *Zoom To Fit:* Fits the complete design in the view. The zoom scale applied is based on the value specified in the *Zoom Scale (%)* field. Environment variable: autoZoomMode
Zoom Scale (%)	Lets you set the scale of zoom in percentage. The default is `90%`. Environment variable: autoZoomScale

Grid Controls

The following table describes the fields available on the *Grid Controls* tab of the Display Options form:

Field	Description
Type	Controls the grid display. ■ *none:* Turns off the grid display. Environment variable: drawGridOn ■ *dots:* Displays a dot for each grid point. ■ *lines:* Makes a grid of lines, like a graph. Environment variable: drawDottedGridOn

© 2023 Cadence Design Systems, Inc.

Field	Description
Dim Major Dots	Controls the dimming of dots of the major grid. When this option is not selected, the major grid dots are visible in bold face. This is a window-specific option and is available only when *Type* is set to *dots*. Environment variable: dotStyleMajorGrid
Minor Spacing and Major Spacing	Sets the number of user units between the visible grid. Minor grid points are white, major grid points are green by default. Environment variables: gridSpacing and gridMultiple
X Snap Spacing and Y Snap Spacing	Sets the distance at which the pointer can snap between grid points along the X axis and the Y axis. This is your drawing grid. The layout default is 0.1 user units and the graphic default is 1.0 user units. Environment variables: xSnapSpacing and ySnapSpacing

Resolution

The following table describes the fields available on the *Resolution* tab of the Display Options form**:**

Field	Description
Resolution slider	Controls the level of detail of the design displayed in the cellview. The resolution options are *Low*, *Medium*, *High*, and *Very High*. The default value is *Medium*. Using the *Resolution* slider, the performance gets enhanced to display the design with a *Very High* resolution. Environment variables: displayResolution and filterSizeForResolution
Highest Resolution for P&R Objects	Enables to use a filter size for objects that is bigger or equal to the filter size used for shapes. This check box is available when the resolution is set to *High* or *Very High.* When this option is not selected, smallest objects are not drawn, which improves the redraw time. Environment variable: displayPRHighestResolution

© 2023 Cadence Design Systems, Inc.

Snap Modes

The following table describes the fields available on the *Snap Modes* tab of the Display Options form**:**

Field	Description
Create	Controls how line segments snap to the grid as you create objects. Environment variable: snapMode
Edit	Controls how line segments of objects snap to the grid as you move or copy them and how edges or corners move as you stretch them. Environment variable: segSnapMode

Display Abstraction Controls

The following table describes the fields available on the *Display Abstraction Controls* tab of the Display Options form:

Field	Description
Merge Shapes	Makes the display more readable by hiding all the extra edges in the design. Environment variable: mergeEnable
Scope	Lets you select the hierarchies for which the *Enable Merge* command applies. The choices are *Pcells*, *Except current*, and *All*. ■ *Pcells*: All Pcells in the hierarchy including the current cellview ■ *Except current*: All hierarchies except the current cellview ■ *All*: All hierarchies including the current cellview Environment variable: mergeScope
Trim Shapes	Hides the layer that is cut using the `Trim` command. Environment variable: trimEnable

© 2023 Cadence Design Systems, Inc.

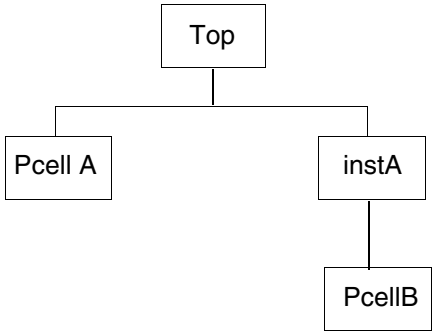
Field	Description
Slot Shapes	Lets you create holes in wires. You overlap the shape to slot with another shape in slot purpose that is displayed as a hole in the shape. Environment variable: slotEnable

Dimming

The following table describes the fields available on the *Dimming* tab of the Display Options form**:**

Field	Description
Dim Virtual Hierarchy	Lets you change the luminosity of the virtual hierarchy. Environment variable: vhDimming When the *Dim Virtual Hierarchy* option is selected, the *Enable Dimming* and *Dim Selected Object Content* options are selected automatically and cannot be changed. Also, the *Automatic Dimming* option is disabled.
Enable Dimming	Lets you change the color luminosity of objects in the cellview. Environment variable: dimmingOn

© 2023 Cadence Design Systems, Inc.

Field	Description
Scope	■ *none*: Does not apply dimming. This is the default. Even if you change the Dim Intensity from zero percent to one hundred percent, the objects remain undimmed. The only exception is that the background cellview dims if its Display field is set to dimmed. ■ *outside:* Dims everything but the instance, figure group, or any object you are editing in place. The dimmed objects represent the objects that are non-editable at the current edit level. With the outside setting on, ❑ If a Pcell exists at the edit level and is visible at the edit level, its contents are not dimmed. For example, in the following hierarchy, both `PcellA` and `PcellB` are be visible at *Display Stop Level* 32. However, at the `Top` edit level when the dimming scope is *outside*, the contents of `PcellA` are undimmed and those of `PcellB` are dimmed.  ❑ If the *Via Shapes* option is on, vias belonging to the edit level are not dimmed. ❑ If the *Instance Pins* option is on, instance pins belonging to edit level-1 are not dimmed.

© 2023 Cadence Design Systems, Inc.

Field	Description
	■ *all:* Dims the design completely as soon as you select this option. If you select an object and Dim Intensity is set to hundred percent, only the selection box is displayed. ■ *eipSurround:* Dims all the objects surrounding the EIP hierarchy. Environment variable: dimmingScope ■ *Dim Grid Lines:* Dims the grid lines on the canvas. Environment variable: dimGridLines ■ *Automatic Dimming:* Dims the design when you select an object or when the design contains highlights or probes. Only selected, highlighted, or probed objects remain undimmed. This option allows dimming even when the dimming Scope is set to none. This option is not available when Scope is set to all. Environment variable: dimmingAutoEnabled ■ *Dim Intensity:* Lets you set the luminosity of dimmed objects. When set to `0`, nothing is dimmed. When set to `100`, dimmed objects are not visible. The default is 50%. *Dim Intensity* has no effect if *Scope* is set to *none*, except when the *Display* field of the background cellview is set to *dimmed*. In that case, only the background cellview is dimmed. Environment variable: dimmingIntensity ■ *Dim Selected Object Content:* Dims the selected object. ■ *True Color Selection only:* Displays the selected objects in their original colors when other objects are dimmed. The selection highlight is not displayed. This option is disabled when *Scope* is set to *none* and *Automatic Dimming* is off. Environment variable: displaySchDashText
Cellview	Specifies that you want to store, load, or delete the display settings to or from the cellview.
Library	Specifies that you want to store, load, or delete the display settings to or from the library of the edit cellview.

© 2023 Cadence Design Systems, Inc.

Field	Description
Tech Library	Specifies that you want to store, load, or delete the display settings to or from the technology library of the edit cellview.
File	Specifies the file to which you want to store or from which you want to load the settings. When you select the file radio button the path field is editable so that you can enter the correct directory where you want to save the file. If you do not remember the name of the directory use the *Browse* button. Environment variable: displayOptionsFile
Save To	Saves the current settings to either the cellview, library of the cellview, technology library of the cellview, or a specified file. If you are saving to a file, the settings from both the Layout Editor Options and Display Options forms are saved.The selections do not take effect until you click *Apply* on the *Display Options* form.

Related Topics

Setting Display Options

leHiEditDisplayOptions

© 2023 Cadence Design Systems, Inc.

Dynamic Display Form

Use the Dynamic Display form to control the information displayed about objects.

The Dynamic Display form contains the following tabs and common options.

Tab/Option	Description
Setup	Lets you specify the type of information displayed for shapes and objects. You can choose to customize the display information for top-level categories or for individual shape and object types.
Info Balloon	Lets you specify display settings for information balloons. You can set the pause time, fade time, and transparency level of information balloons.
Dynamic Measurement	Lets you specify the height and color of the text displayed during dynamic measurement.
Highlight Aligned Edges	Lets you specify the width and color of the highlight displayed on the edges of aligned shapes.
Show Info Balloon	Displays information balloon showing information about an object or shape when the pointer is moved over the existing shape or object. The type of information can be customized and is object dependent. Environment variable: balloonOn
Measurement Display On	Displays information about objects and shapes dynamically, during creation or editing of objects. The type of information can be customized and is object dependent. Environment variable: measurementDisplayOn
Highlight Aligned Edges On	Highlights edges of the shapes aligned to the pointer in horizontal and vertical directions. Environment variable: highlightAlignedEdges

© 2023 Cadence Design Systems, Inc.

Setup

The following table describes the fields available on the *Setup* tab of the Dynamic Display form.

Field	Description
Object Display Setting	Allows you to expand categories to display shape and object types. You can display and customize information about existing objects or dynamically during the creation or editing of objects. You must first turn on the controls for information balloons or textual display of measurements.
Virtuoso XL	Displays an expandable tree that contains the *Shapes* and *Objects* categories in which data types are organized. ■ *Shapes*: Displays the types of shapes that you can customize for information balloon and dynamic measurement display. ■ *Objects*: Displays the types of objects that you can customize for information balloon and dynamic measurement display.
Configuration	Enables you to save and load the customized display of shapes and objects information.
Save	Opens the Save Configuration form where you can specify the details to save the configuration.
Delete	Removes the selected configuration.
Default	Saves the selected configuration as the default configuration.
Contents	Lists the available parameters for a shape or object. For information about parameters, see Parameter Types for Shapes and Objects.

© 2023 Cadence Design Systems, Inc.

Field	Description
Select	Enables you to select or deselect all available parameters for a shape or object. ■ *All*: Enables you to select all available parameters for each shape and object rather than selecting parameters individually. By default, all parameters are selected for the *default* configuration. ■ *None*: Enables you to deselect all available parameters for each shape and object rather than deselecting parameters individually.

Info Balloon

The following table describes the fields available on the *Info Balloon* tab of the Dynamic Display form.

Field	Description
Pause Time	Specifies the amount of time before the information balloon will appear when the pointer is over an object. Valid values are between 0 and 10. Environment variable: balloonPauseTime
Fade Time	Specifies the amount of time (in seconds) for the information balloons to fade away after the balloon appears. Valid values are between .001 and 100. Environment variable: balloonFadeTime
Transparency	Specifies the transparency of the balloon information box. When set to 0, the box is completely transparent. When set to 100, objects behind the box are not visible. The default is 50. Environment variable: balloonTransparency

© 2023 Cadence Design Systems, Inc.

Dynamic Measurement

The following table describes the fields available on the *Dynamic Measurement* tab of the Dynamic Display form.

Field	Description
Text Height (pixels)	Specifies the height of the text displayed during dynamic measurement. Valid values are any integer between `4` and `50`. Environment variable: measurementTextHeight
Color	Specifies the color of the text displayed during dynamic measurement. Environment variable: dynamicMeasurementTextColor

Highlight Aligned Edges

The following table describes the fields available on the *Highlight Aligned Edges* tab of the Dynamic Display form.

Field	Description
Width	Specifies the width of the highlight displayed on the edges of aligned shapes. You can choose from: *Thin*, *Medium*, or *Thick*. Environment variable: highlightAlignedEdgesWidth
Color	Specifies the color of the highlight displayed on the edges of aligned shapes. Environment variable: highlightAlignedEdgesColor

© 2023 Cadence Design Systems, Inc.

Parameter Types for Shapes and Objects

The following table describes the parameters available for shapes and objects on the *Setup* tab of the Dynamic Display form.

Distance is the absolute distance as expressed in a straight line between the last point entered and the current pointer location. Delta X and Y is the difference between the coordinates of a reference point and the current pointer location on the x or y axis. This reference point is the point entered to create part of an object, or the selection point used to begin a move or a copy.

Shape/ Object	Available Parameters	Dynamic Measurement-Specific Parameters
Rectangle	*X-Length, Y-Length, Area, Perimeter, XL Status, LPP, Label, NetName* (if applicable), *LPP-Color*	*Delta XY*
Polygon	*Area, X-Length, Y-Length, Perimeter, XL Status, LPP, Label, NetName* (if applicable), *LPP-Color*	*Delta XY, Distance*
Path	*XL Status, Area, Width, Length, LPP, Label, NetName* (if applicable), *LPP-Color* The length of a path is the total centerline path length and does not include beginning or ending extensions. The info balloon also displays the area information.	*Delta XY, Distance*
MPP	*XL Status, Area, Width, Length, LPP, Label, NetName* (if applicable), *LPP-Color* The length of a multipart path is the total centerline path length and does not include beginning or ending extensions or subparts.	*Delta XY, Distance*
Pathseg	*XL Status, Area, Width, Length, LPP, Label, NetName* (if applicable), *LPP-Color* The info balloon also displays the area and total length information.	*Delta XY, Distance*
Circle	*Area, Perimeter, Radius, XL Status, LPP, Label, LPP-Color*	*Delta XY*

© 2023 Cadence Design Systems, Inc.

Shape/ Object	Available Parameters	Dynamic Measurement-Specific Parameters
Ellipse	*Area, Perimeter, XL Status, LPP, Label, LPP-Color*	*Delta XY*
Donut	*Area, Perimeter, Radius, XL Status, LPP, Label, LPP-Color*	*Delta XY*
Label	*LPP, Label* (text of label)	*Delta XY*
TextDisplay	*LPP, Label* (text of label)	
Instance	*XL Status, Label* (lib/cell/view names)	*Delta XY*
Blockage	*Area, X-Length, Y-Length, Perimeter, LPP, LPP-Color, Distance, Label* (any label attached to the blockage)	*Delta XY, Distance*
Row	*X-Length, Y-Length, Area, Row Sites, Label* (row base name)	*Delta XY*
Boundary	*Area, Perimeter, LPP, Label* (boundary name) For the PR boundary of a soft block, ‘Area’ displays the soft block utilization information.	*Delta XY, Distance*
Via	*XL Status, NetName, LPP-Color, Label* (viaDefName)	*Delta XY*
Custom Placement	*Area, Perimeter*	*Delta XY*
Marker	*Message* (type tool, short message, severity)	
Pin	*XL Status, LPP, Label* (pin name), *NetName, LPP-Color*	*Delta XY*
Modgen	*Label* (any label attached to the modgen)	*Delta XY*
Ruler	*Length, Points*	*Delta XY*
Row Region	*Area, Perimeter, Label, Custom SKILL Function*	

© 2023 Cadence Design Systems, Inc.

Shape/ Object	Available Parameters	Dynamic Measurement-Specific Parameters
Virtual	*Area, Label, Stop Level, Type, Placement Status* (These parameters are selected by default) *Perimeter, Distance, Delta XY, Custom SKILL Function* (These parameters are not selected by default) The parameters are also displayed in the info balloon. If *Delta XY* is selected for dynamic measurement, it is also displayed in the info balloon.	*Delta XY*

Related Topics

Object Information Display

Customizing Information Balloon Using Custom SKILL Functions

leHiEditObjectInfo

© 2023 Cadence Design Systems, Inc.

Dynamic Selection Assistant Options Form

Use the Dynamic Selection Assistant Options form to specify how selected objects are displayed in the assistant and the mode used when opening a cellview from the shortcut menu of the assistant.

Field	Description
Display Controls	Enables you to specify how the selected objects are displayed in the Dynamic Selection assistant. ■ *Automatically zoom to selected*: Zooms to the selected objects in the design display area automatically. If this option is selected, you can zoom shapes in the hierarchy by selecting the shapes in the assistant. You can also zoom to the corresponding shape on the canvas if you double-click the shape in the assistant. ■ *Automatically expand hierarchy*: Expands the hierarchical instances, groups, and ROD objects shown in the assistant, automatically. The hierarchy is expanded down to the display level set in the Display Options form. If the display level is set to `0`, the hierarchy cannot be expanded. The assistant does not necessarily list all the objects contained in a particular hierarchical object, only those that were under the pointer when the display was frozen. ■ *Hide non-selectable objects*: Removes from the assistant any objects that cannot be selected. ■ *Reverse LPP order*: Displays the layer-purpose pairs (LPPs) for selected objects from bottom to top. By default, LPPs of selected objects are displayed from top to bottom. ■ *Flat view*: Displays the flattened version of the hierarchy. If this option is selected, *LPP*, *Data*, and *Hierarchy* columns are displayed in the assistant.
Open mode	Specifies whether the cellview is opened for in the *edit* or *read* mode.
Open in	Specifies whether the cellview is opened in a *new tab*, the *current tab*, or a *new window*.

© 2023 Cadence Design Systems, Inc.

Related Topics

Dynamic Selection Shortcut Menu

Dynamic Selection Assistant

Display Options Form

Selecting Objects Using the Dynamic Selection Assistant

© 2023 Cadence Design Systems, Inc.

Dynamic Selection Shortcut Menu

The shortcut menu of the Dynamic Selection assistant contains the following items.

Command	Description
Open Instance (<mode>/<tab>)	Opens the master cellview of the instance currently under the mouse pointer with the mode specified in the command name. For example, *Open (edit/new tab)* opens the cellview in edit mode in a new tab; *Open (read/current tab)* opens the cellview in read mode in the current tab. You specify the default behavior for this command in the Dynamic Selection Assistant Options form. This command is available only if the mouse pointer is over an instance. To open a different view of the current cell, use the *Open Cellview* command instead.
Open Cellview	Displays the Open form where you can specify: ■ Which view of the current cell you want to open ■ Whether that view is opened in edit or read mode ■ Whether it is opened in a new tab, the current tab, or in a new window. This command is available only if the mouse pointer is over an instance.
Edit In Place	Enables you to edit in place the currently selected cellview in the layout window. This command is available only if the mouse pointer is over an instance, a group, or an object contained in a hierarchy.
Expand All Below	Expands the hierarchy to display the hierarchical objects under the current node. This command is available only for hierarchical objects. The hierarchy is expanded down to the display level set in the Display Options form. So if, for example, the display level is set to 0, the hierarchy cannot be expanded. The assistant does not necessarily list all the objects contained in a particular hierarchical object, only those that were under the pointer when the display was frozen.

© 2023 Cadence Design Systems, Inc.

Command	Description
Collapse All Below	Collapses the hierarchy to hide the hierarchical objects under the currently selected node. This command is available only for hierarchical objects.
Options	Opens the Dynamic Selection Assistant Options form where you can specify: ■ How selected objects are displayed in the assistant ■ Default open mode
Freeze DSA/UnFreeze DSA	Freezes or unfreezes the display in the assistant.
Settings	Enables you to save and load your display preferences for the Dynamic Selection assistant. This ensures that the order of fields that you choose for the assistant and the show or hide states are retained even after you close the current Virtuoso session,

Related Topics

Dynamic Selection

Dynamic Selection Assistant Options Form

Display Options Form

© 2023 Cadence Design Systems, Inc.

Edit Cellview Properties Form

Use the Edit Cellview Properties form to edit the properties of a cellview.

Field	Description
Attribute	Represents the characteristics of the object. None of these fields can be changed. ■ *Library:* Displays the name of the library containing this cellview. ■ *Cell:* Displays the name of this cell. ■ *View:* Displays the name of this view. ■ *bBox:* Displays the coordinates of the invisible box surrounding the design in the cellview. ■ *Type:* Displays the view type that corresponds to the view name for this cellview. View types are defined in your technology file. ■ *DBU per micron:* Displays the number of database units (DBU) per user unit (usually microns). Database units and user units are defined in the `.cdsenv` file or by the library manager. ■ *Mode:* Shows whether this cellview is opened in edit mode (`Edit`) or read-only mode (`Read`). ■ *Save Needed:* Shows whether you made changes to this cellview that need to be saved to disk. ■ *CellType:* Displays the various cell types available. You can select one from the list to set the cell type of the current cellview. For information about cell types, see Cell Types.

© 2023 Cadence Design Systems, Inc.

Field	Description
	■ *Symmetry Type:* Defines the orientation of this cell. The valid values include `no`, `X`, `Y`, `XY`, and `any`. ■ *SitePattern* ❑ *Site Definition* cyclic field: Displays all the `siteDefs` defined in the technology file. Select a site definition from the list to set it for the current cellview. ❑ *Origin:* displays the position associated with the sitePattern. ❑ *Rotation:* Lists the possible rotation values that can be associated with the sitePattern. **Note:** You can alternatively set the sitePattern for a cellview using SKILL in the CIW. For example, to set the sitePattern for a design with site definition as `Core`, type the following in the CIW: `geGetEditCellView()~>sitePattern = list(list("Core" list(0.0 0.0) "R0"))`
Property	Shows the properties for this cellview. Most properties correspond to settings in the Display Options form. Other Cadence applications such as DRC or Extract can also add properties. ■ *Add:* Opens the Open Property form, which lets you add a new cellview property. ■ *Delete:* Deletes a selected property created with *Add*. ■ *Modify:* Opens the Modify Property form, which lets you change the definition for any property created with *Add*.

Certain options in the Edit Cellview Properties form appear disabled, such as the Connectivity, Parameter, and ROD tabs and the Next and Previous buttons. These options are not available when you edit cellview properties, and are applicable when you edit objects within a cellview.

Related Topics

Attribute Tab of Various Objects

© 2023 Cadence Design Systems, Inc.

Site Definitions

Adding a New Cellview Property

Cell Types

The following table describes the type of cells that you select in the *CellType* field in the Edit Cellview Properties form.

Blocks/Cells	Cell Type Attribute	LEF MACRO Class	Description
Custom cells	*none*		A cell without any cell type definition.
Soft cells	*blockBlackBox*	`BLOCK BLACKBOX`	A block that sometimes only contains a size statement that estimates its total area. The block can optionally contain pins. This cell type does not have any sub-block implementation available.
	softMacro	`BLOCK SOFT`	A block that contains a version of the sub-block that is not fully implemented. Certain parts of the block can still be modified, such as the aspect ratio or pin locations, because the sub-block is not yet fully implemented.
Hard cells	*block*	`BLOCK`	A predefined cell used in a hierarchical design.
	blockRing	`RING`	A large cell that has an internal power mesh and only exposes power-pin shapes that form a ring along the cell boundary. It can also be used for power-switch cells that are abutted together to form a power-ring around a power-domain.

© 2023 Cadence Design Systems, Inc.

Blocks/Cells	Cell Type Attribute	LEF MACRO Class	Description
	cover	`COVER`	A cell with data that is fixed to the floorplan and cannot change, such as power routing (ring pins) around the core.
Core cells	*core*	`CORE`	A standard cell used in the core area. The cell must contain a site definition so that standard cell placers can correctly align the cell to the standard cell rows.
	coreSpacer	`CORE SPACER`	A cell in the core area used to fill in space between regular core cells.
	coreAntenna	`CORE ANTENNACELL`	A cell in the core area used for solving process antenna violations. It has a single input to a diode to bleed off charge that builds up during manufacturing.
	coreFeedThru	`CORE FEEDTHRU`	A cell in the core area used for connecting to another cell.
	coreTieHigh	`CORE TIEHIGH`	A cell in the core area used for connecting unused I/O terminals to the power bus.
	coreTieLow	`CORE TIELOW`	A cell in the core area used for connecting unused I/O terminals to the ground bus.
	coreWellTap	`CORE WELLTAP`	A standard cell in the core area that connects N and P diffusion wells to the correct power or ground wire. The cell provides a tap for the N and P wells to the power or ground wires.
	coreSpacerEndCapPre	`ENDCAP PRE`	A cell placed at the left end of core rows to connect to power wiring.

© 2023 Cadence Design Systems, Inc.

Blocks/Cells	Cell Type Attribute	LEF MACRO Class	Description
	coreSpacerEndCapPost	`ENDCAP POST`	A cell placed at the right end of core rows to connect to power wiring.
Bump cells	*coverBump*	`COVER BUMP`	A physical-only cell that has bump geometries and pins. Typically, a bump cell has geometries only on the top-most bump metal layer, although it might contain a via and pin to the metal layer below.
I/O cells	*pad*	`PAD`	An I/O pad cell.
	padInput	`PAD INPUT`	An input pad cell for I/O rows and I/O corner pads.
	padOutput	`PAD OUTPUT`	An output pad cell for I/O rows and I/O corner pads.
	padInOut	`PAD INOUT`	An I/O pad cell for I/O rows and I/O corner pads.
	padPower	`PAD POWER`	A power pad cell for I/O rows and I/O corner pads.
	padSpacer	`PAD SPACER`	An I/O pad cell for I/O rows. Unlike a core spacer cell, a pad spacer cell can contain logic pins.
	padAreaIO	`PAD AREAIO`	An area I/O driver cell that does not have the bump built in as part of the cell, and therefore requires routing to a cover bump cell for connection to the IC package.
	cornerTopLeft	`ENDCAP TOPLEFT`	A top-left I/O corner cell.
	cornerTopRight	`ENDCAP TOPRIGHT`	A top-right I/O corner cell.
	cornerBottomLeft	`ENDCAP BOTTOMLEFT`	A bottom-left I/O corner cell.

© 2023 Cadence Design Systems, Inc.

Blocks/Cells	Cell Type Attribute	LEF MACRO Class	Description
	cornerBottomRight	`ENDCAP BOTTOMRIGHT`	A bottom-right I/O corner cell.

Related Topics

Edit Cellview Properties Form

Macros

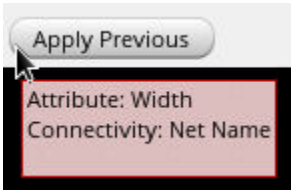
© 2023 Cadence Design Systems, Inc.

Edit Properties Form

Use the Edit Properties form to edit the properties of the objects that you have selected. The following table lists the tabs and common options available in the Edit Properties form.

Tab/Option	Description
Attribute Tab	Displays characteristics of an object. The available attributes vary, depending on the type of object selected. For information about the attributes of various objects, such as area boundary, instance, label, path, polygon, row, or via, see Attribute Tab of Various Objects.
Connectivity Tab	Displays routing and net information about selected pins and shapes on a net. Only pins display *I/O Type* or *Access Direction*.
Parameter Tab	Displays any parameters defined for a parameterized cell (Pcells) or the selected instance, such as fluid guard ring. For CDF instances, you can display the prompt names or CDF parameter names by selecting the option *Prompt Name* or *Display CDF Parameter Name*. By default, the prompt name is displayed for CDF instances. For information about the Edit Instance Properties form for a Fluid Guard Ring, see Edit Instance Properties Form.
Property Tab	Sets the properties of the object and lets you add your own properties to an object.
ROD Tab	Displays the ROD properties of an object.
Hide/Show Selection Tree	Hides or displays the selection tree, which is displayed on the left in the form. The selection tree enables you to select the objects whose properties you want to edit.
Deselect highlighted ones from canvas	Deselects objects from the canvas that you selected in the selection tree. Also, removes those objects from the selection tree in the form. Note that this option changes the selection set on the canvas.

© 2023 Cadence Design Systems, Inc.

Tab/Option	Description
Keep highlighted ones and deselect others from canvas	Retains the objects selected in the selection tree and removes all other objects from the selection tree. Also, the removed objects are deselected from the canvas. Note that this option changes the selection set on the canvas.
Previous	Highlights the previous object in the selection tree and resets the form to show that object's properties.
Next	Highlights the next object in the selection tree and resets the form to show that object's properties.
Select all objects in the tree	Selects all the objects listed in the selection tree.
Apply Previous	Reapplies the last-applied property editor data of the current session. When you hover over this button, the previously applied data, if any, is displayed as a tooltip. For example, if you modified the width of a rectangle on the *Attribute* tab and the net name on the *Connectivity* tab, the following tooltip is displayed, listing the names of the tabs and modified fields.  Only the properties applicable to the selected object are applied.

© 2023 Cadence Design Systems, Inc.

Connectivity Tab

The following table describes the fields available on the *Connectivity* tab of the Edit Properties form. The fields displayed might vary depending on the object selected. For example, only pins display *I/O Type* or *Access Direction*.

Field	Description
Net Name	Displays the name of the net to which the pin is connected. This field can be used to rename or delete the net. When you edit the net name of an object (via or pathseg) that is a route element, the *Net Name* field appears as the *Route Net Name* field. This indicates that editing the net of an object in a route changes the net of the entire route.
Terminal Name	Sets the name of the terminal associated with this pin. The terminal name should always be the same as the net name.
Net Expression	Assigns a net expression of the terminal listed in the *Terminal Name* field. ■ *Property:* Sets the override property name to the net expression. ■ *Default:* Defines the net to be used if no override property is defined in the hierarchy above this point (in the schematic view). Unless a different signal name is entered, the terminal name is used.
I/O Type	Assigns a property used by routers to identify the direction of the signal into or out of this cellview. The signal can be input, output, inputOutput (bidirectional), switch (carries data either in or out, but not simultaneously), or jumper (passes data through this cellview).
Net Criticality	Sets a weighting factor that determines the routing priority for this net for the Cadence place-and-route tools.
Access Direction	Assigns a property used to identify the part of the pin to which the routers can connect routing. Applies only to rectangle pins. The selections are: top, left, bottom, right, any or none.

© 2023 Cadence Design Systems, Inc.

Field	Description
Signal Type	Specifies the signal type of the net for a pin. The options available for signal type are *signal*, *ground*, *power*, *clock*, *analog*, *tieOff*, *tieHi*, *tieLo*, *scan*, or *reset*. The default is *signal*. For a photonics pin, only the *optical*, *singleModeOptical*, or *multiModeOptical* options are available.

Instance Connectivity

When you edit an instance that has terminals, the Connectivity tab displays the following fields. It enables you to map the terminals on an instance or a homogenous mosaic to the nets in a cellview. For information about updating instance connectivity or propagating nets, see Propagating Nets.

Field	Description
Terminal Name	Specifies the terminal name. Several pins can be defined on one terminal.
Net Name	Specifies the top-level net name. If a net name field is blank, the corresponding terminal is not currently mapped to a top-level net.
Set Net Name	Sets the net name same as the terminal name, when you click the *Set Net Name* button.
Clear All Net Names	Clears the text fields.
Set All Net Names to Terminal Name	Sets all net names to the terminal names.
Show All Must-Join Terminals	Displays the must-join terminal names that exist in the cellview of the instance.
Defaults	Displays the existing top-level net names of terminals.

© 2023 Cadence Design Systems, Inc.

Property Tab

The following table describes the fields available on the *Property* tab of the Edit Properties form.

Field	Description
Add	Opens the Add Property form, which lets you add a new cellview property.
Delete	Deletes a selected property created with *Add*.
Modify	Opens the Modify Property form, which lets you change the definition for any property created with *Add*.

ROD Tab

The following table describes the fields available on the *ROD* tab of the Edit Properties form.

Field	Description
ROD Name	Displays the name of the ROD object.

© 2023 Cadence Design Systems, Inc.

Field	Description
Handle	Displays the system handle and user handle information. A handle is an attribute of, or item of information about, a ROD object, such as the coordinates of a point on the bounding box around an object, the width of the bounding box of an object, or the resistance of an object. ■ *System handle:* Displays handle names assigned to the object. When you choose a different handle from the cyclic field, the *Value* field updates to reflect the current handle. ■ *Value:* Specifies a non-editable field displaying the location of the system handle displayed in the *System handle* field. These values are predefined reference points. ■ *User handle:* Displays handle names assigned by the user to the object. When you choose a different handle from the cyclic field, the *Value* field updates to reflect the current handle. If a user handle is not assigned, the fields are empty. ■ *Value:* Displays the current location of the system handle displayed in the *User handle* field. You can change these values by typing new coordinates in this field.

© 2023 Cadence Design Systems, Inc.

Field	Description
Alignment	Displays the alignment information about two ROD objects. You can edit these fields if the objects have alignment values assigned to them, otherwise the fields are empty. When looking at the alignment information for two objects, the selected object is *Align object*. The separation is the distance from *Reference object* (unselected) to *Align object* (selected). ■ *Reference object:* Displays the name of the object assigned to be the reference object of *Align object*. If there is more than one object assigned to be *Reference object*, you can choose that object from the cyclic field. ■ *Align object:* Displays the name of the object aligned to *Reference object*. This field is not editable. ■ *Reference handle:* Displays the name of the handle *Align object* is aligned to. ■ *Align handle:* Displays the name of the handle *Reference object* is aligned to. ■ *X separation:* Displays the distance in the X direction from either the reference point handle or the reference point on *Reference object* to the alignment point handle on *Align object*. The value can be positive or negative or a Cadence SKILL language expression that evaluates to a positive or negative number. ■ *Y separation:* Displays the distance in the Y direction from either the reference point handle or the reference point on *Reference object* to the alignment point handle on *Align object*. The value can be positive or negative or a SKILL expression that evaluates to a positive or negative number.

© 2023 Cadence Design Systems, Inc.

Property Editor Shortcut Menu

Right-clicking in the selection tree of the Edit Properties form displays the property editor shortcut menu. The table describes the options available on the shortcut menu.

Option	Description
Select All/Clear Selection	Selects or deselects all the objects in the selection tree.
Deselect	Deselects the highlighted object from the canvas.
Keep	Retains the highlighted objects and deselects the other objects from the canvas.
Zoom To Selected	Zooms to the selected object on the canvas.
Highlight Modified Fields	Highlights the modified fields in the Edit Properties form.
Dynamic Cross Highlight	Dynamically highlights the objects on the canvas.
Collapse All	Collapses all expanded tree nodes in the selection tree.
Expand Selected	Expands the selected node in the selection tree.

Related Topics

Editing Object Attributes

leHiEditProp

Add Property Form

Modify Property Form

© 2023 Cadence Design Systems, Inc.

Attribute Tab of Various Objects

Different objects support different attributes. The following table lists the Edit Properties forms in which you can modify the attributes of various objects.

Attributes in the Edit Area Boundary Properties Form	Attributes in the Edit Instance Pin Properties Form	Attributes in the Edit Polygon Pin Properties Form
Attributes in the Edit Blockage Properties Form	Attributes in the Edit Label Properties Form	Attributes in the Edit PR Boundary Properties Form
Attributes in the Edit Cluster Boundary Properties Form	Attributes in the Edit Line Properties Form	Attributes in the Edit Rectangle Pin Properties Form
Curvy Shapes Attributes	Attributes in the Edit Modgen Properties Form	ROD Multipart Path - Master Path Fields Attributes
Attributes in the Edit Donut Properties Form	Attributes in the Edit Mosaic Properties Form	Routing, Slot, Pin, or Feed Through Blockage Attributes
Attributes in the Edit Donut Pin Properties Form	Attributes in the Edit Path Properties Form	Attributes in the Edit Row Properties Form
Attributes in the Edit Dot Pin Properties Form	Attributes in the Edit Path Pin Properties Form	Attributes in the Edit Snap Boundary Properties Form
Attributes in the Edit Ellipse Properties Form	Attributes in the Edit PathSeg Properties Form	Attributes in the Edit Via Properties Form
Attributes in the Edit Ellipse Pin Properties Form	Attributes in the Edit PathSeg Pin Properties Form	Attributes in the Edit Via Pin Properties Form
Attributes in the Edit Group Properties Form	Pin Name or Other Text Display Attributes	Attributes in the Edit Via Stack Properties Form
Attributes in the Edit Group Array Properties Form	Placement Halo Blockage Attributes	Virtual Hierarchy / VH Clone Attributes
Attributes in the Edit Instance Properties Form	Attributes in the Edit Polygon Properties Form	

For information about tabs in the property editor, see Edit Properties Form.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Area Boundary Properties Form

The following table describes the fields available on the Attributes tab of the Edit Area Boundary Properties form.

Field	Description
Name	Sets the name of the area boundary.
Points	Sets the coordinates of each point of the area boundary.
bBox	Displays the coordinates of the box surrounding the area boundary.
Area	Displays the total area in user units of the area boundary.
Rectangle and Rectilinear	Displays whether a boundary is a rectangle or a polygon.

Attributes in the Edit Blockage Properties Form

The following table describes the fields available on the Attributes tab of the Edit Blockage Properties form.

Field	Description
Owner Information	Sets owner as *Cluster*, *Instance*, or *No Owner*.
Points	Sets the coordinates of each point of the blockage.
Area	Displays the total area of the blockage in user units.
Type	Displays the blockage type.
Layer	Selects the layer on which the blockage is enforced.
Material	Selects the material on which the blockage is enforced.
Use Effective Width	Enables the *Effective Width* field, when clicked.
Effective Width	Sets the value which is checked against the layer spacing rule or table spacing rules, if defined.
Spacing	Displays the row offset value in user units. *Spacing* attribute is disabled if the *Use Effective Width* attribute is selected.
Allow PGNet	Allows shapes with signal types power and ground to be placed within the blockage area.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Cluster Boundary Properties Form

The following table describes the fields available on the Attributes tab of the Edit Cluster Boundary Properties form.

Field	Description
Name	Sets the name of the cluster boundary.
Cluster Name	Displays the name of the associated cluster.
Points	Sets the coordinates of each point of the cluster boundary.
bBox	Displays the coordinates of the box surrounding the cluster boundary.
Area	Displays the total area in user units of the cluster boundary.
Rectangle and Rectilinear	Displays whether a boundary is a rectangle or a polygon.
Max Voltage	Displays the maximum voltage if the cluster is a power domain cluster.

Curvy Shapes Attributes

(Virtuoso Photonics Option) The following table describes the fields available for curvy shapes on the Attributes tab of the Edit Properties form.

Field	Description
Curvy Line	■ *Layer:* Displays the layer on which the curvy line is drawn. ■ *bBox:* Displays the coordinates of the box surrounding the curvy line. ■ *Points:* Displays the coordinates of each point of the curvy line. ■ *Begin Point:* Displays the coordinates of the beginning point of the curvy line. ■ *End Point:* Displays the coordinates of the end point of the curvy line.

© 2023 Cadence Design Systems, Inc.

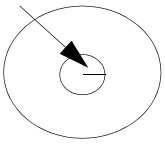
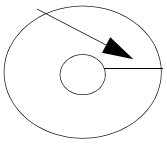
Field	Description
Curvy Path	■ *Segment*: Displays the coordinates of the segment of the curvy line. ■ *Layer*: Displays the layer on which the curvy path is drawn. ■ *bBox*: Displays the coordinates of the box surrounding the curvy path. ■ *Points*: Displays the coordinates of each point of the curvy path. ■ *Begin Point*: Displays the coordinates of the beginning point of the curvy path. ■ *End Point*: Displays the coordinates of the end point of the curvy path. ■ *Begin Facet Angle*: Displays the beginning facet angle for the curvy path. ■ *End Facet Angle*: Displays the end facet angle for the curvy path. ■ *Begin Facet Radius*: Displays the beginning facet radius for the curvy path. ■ *End Facet Radius*: Displays the end facet radius for the curvy path. ■ *Segments*: Displays the coordinates of the segments of the curvy path. ■ *Segments Count*: Displays the number of segments.

© 2023 Cadence Design Systems, Inc.

Field	Description
Curvy Surface	■ *Layer*: Displays the layer on which the curvy surface is drawn. ■ *bBox*: Displays the coordinates of the box surrounding the curvy surface. ■ *Points*: Displays the coordinates of each point of the curvy surface. ■ *Area*: Displays the area of the curvy surface. ■ *Boundary*: Displays the coordinates of the boundaries of segments the curvy surface. ■ *Boundary Count*: Displays the number of boundaries. ■ *Holes*: Displays the coordinates of the holes of segments of the curvy surface. ■ *Number of holes*: Displays the number of holes.

Attributes in the Edit Donut Properties Form

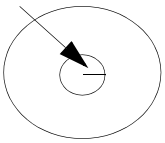
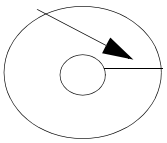
The following table describes the fields available on the Attributes tab of the Edit Donut Properties form.

Field	Description
Layer	Sets the layer on which the donut is drawn.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the donut.
Center: x and y	Sets the X and Y coordinates of the center of the donut.
Inner Radius and Outer Radius	Sets the inner and outer radius of the donut. The inner radius of the donut  The outer radius of the donut 

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Donut Pin Properties Form

The following table describes the fields available on the Attributes tab of the Edit Donut Pin Properties form.

Field	Description
Layer	Sets the layer on which the donut is drawn.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the donut.
Center: x and y	Sets the X and Y coordinates of the center of the donut.
Inner Radius and Outer Radius	Sets the inner and outer radius of the donut. The inner radius of the donut  The outer radius of the donut 
Placement Status	Displays the placement status of the instance in a cyclic field. The choices are Unplaced, None, Placed, Fixed, and Locked/Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.
Pin Name	Displays the name of the pin.
Terminal Name	Displays the name of the terminal.
Route Method	Displays the routing method of the pin in a cyclic field. The choices are symbolic and geometric. When *Update Net Signal Type* is enabled, terminals with nets that have a *Signal Type* of Power or Ground in the schematic are automatically given a *Route Method* of Geometric when either the *Generate All From Source* or *Update Components and Nets* commands are run.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Dot Pin Properties Form

The following table describes the fields available on the Attributes tab of the Edit Dot Pin Properties form:

Field	Description
Layer	Sets the layer of the dot.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the bounding box of the dot.
Center x and y	Sets the X and Y coordinates of the dot origin.
Width	Sets the width of the dot in user units (usually microns).
Height	Sets the height of the dot in user units (usually microns).
Placement Status	Displays the placement status of the instance in a cyclic field. The choices are Unplaced, None, Placed, Fixed, and Locked/ Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.
Pin Name	Displays the name of the pin.
Terminal Name	Displays the name of the terminal.
Route Method	Displays the routing method of the pin in a cyclic field. The choices are symbolic and geometric. When *Update Net Signal Type* is enabled, terminals with nets that have a *Signal Type* of Power or Ground in the schematic are automatically given a *Route Method* of Geometric when either the *Generate All From Source* or *Update Components and Nets* commands are run.
ROD Name	Displays the name of the dot if it is a ROD object.

Attributes in the Edit Ellipse Properties Form

The following table describes the fields available in the Attributes tab of the Edit Ellipse Properties form.

Field	Description
Layer	Sets the layer of the ellipse or circle.

© 2023 Cadence Design Systems, Inc.

Field	Description
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
Left, Right, Bottom, and Top	Sets the coordinates of the box that defines the ellipse or circle.

Attributes in the Edit Ellipse Pin Properties Form

The following table describes the fields available in the Attributes tab of the Edit Ellipse Pin Properties form.

Field	Description
Layer	Sets the layer of the ellipse or circle.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
Left, Right, Bottom, and Top	Sets the coordinates of the box that defines the ellipse or circle.
Placement Status	Displays the placement status of the pin in a cyclic field. The choices are Unplaced, None, Placed, Fixed, and Locked/Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.
Pin Name	Displays the name of the pin.
Terminal Name	Displays the name of the terminal.
Route Method	Displays the routing method of the pin in a cyclic field. The choices are symbolic and geometric. When *Update Net Signal Type* is enabled, terminals with nets that have a *Signal Type* of Power or Ground in the schematic are automatically given a *Route Method* of Geometric when either the *Generate All From Source* or *Update Components and Nets* commands are run.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Group Properties Form

The following table describes the fields available on the Attributes tab of the Edit Group Properties form.

Field	Description
Name	Displays the name of the group.
Origin X and Origin Y	Sets the origin point of the group.
Orientation	Sets whether the group is rotated or mirrored. See Orientation Valid Values.
Query Group Member button	Displays the appropriate Edit Properties form for the types of objects contained in the group. For example, if the first object in the group is a rectangle, the Edit Rectangle Properties form appears. To continue through the objects in the group, use the Next and Previous buttons on the form. You can edit the properties of each member of the group using these Edit Properties forms. This chapter describes all of the Edit Properties forms that might appear as a result of the query.

Attributes in the Edit Group Array Properties Form

The following table describes the fields available on the *Attributes* tab of the Edit Group Array Properties form.

Field	Description
Name	Displays the name of the selected group array.
Origin	Sets a specific point in a group array as its origin. You can specify the values in the *X* and *Y* fields.
Rows, Columns	Specifies the dimensions of the array. It determines the number of cells or members in a group array.
Lock X,Y	Disables the *Mode*, *X*, and *Y* fields. When this option is selected, spacing mode and spacing values for the group array cannot be edited.

© 2023 Cadence Design Systems, Inc.

Field	Description
Mode	Lets you select the spacing mode for the group array. ■ *Spacing*: Lets you specify the distance that is measured between the adjacent or closest edges of cells. For example, the distance between the top edge of a cell and the lower edge of another cell. ■ *Pitch*: Lets you specify the distance that is measured between the same-edge corners of cells. For example, the distance between the left edges of two cell. You can specify the distance in the *X* and *Y* fields. Based on the spacing mode selected, the specified distance is kept between the cells of a group array.
Reference LPP	Specifies the layer-purpose pair (LPP) used for computing the bounding box of the cells in a group array. The bounding box of the geometries on the specified reference LPP is used for calculating the spacing between cells. When the value is set to *None*, the combined bounding box of all the geometries in a cell is used for determining the spacing between cells.
Rotation	Specifies whether a group array is rotated or mirrored. For information, see Orientation Valid Values.
Array Pattern	Lets you specify the orientation of individual cells of a group array. You can select a predefined or custom pattern. You can specify the orientation values directly in the text field by enclosing them in quotation marks. You can also select a predefined or custom pattern from the drop-down list. Alternatively, you can use the Array Pattern form, which is accessed by clicking the *Open Array Pattern editor* button  . For more information, see Predefined Orientation Patterns for a Group Array.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Instance Properties Form

The following table describes the fields available on the Attributes tab of the Edit Instance Properties form.

Field	Description
Library, Cell, and View	Sets the library, cell, and view names of the master cell for this instance. **Note:** You can use the SKILL function leEnableInstPropEditFields to enable or disable the *Library* or *Cell* fields for an instance in the Property Editor form.
Browse	Lets you select the library, cell, and view names by clicking on them in the browser.
Origin: X and Y	Sets the X and Y coordinates of the origin of the instance.
Name	Sets the name assigned to this instance. The layout editor automatically assigns instance names that begin with the letter `I`, followed by a number.
Rotation	Sets whether the instance is rotated or mirrored. See Orientation Valid Values.
Cell Type (read only field)	Displays the master cell type used to describe the nature of a physical block or cell. Layout editor does not support any level one editing on any of these design objects. These cell types are used in Virtuoso Layout Suite XL and Virtuoso Floorplanning. See Orientation Valid Values.
Cluster	Specifies the name of the cluster associated with the instance.
Max Voltage	Displays the maximum voltage if the cluster is a power domain cluster.
Placement Status	Displays the placement status of the instance in a cyclic field.The choices are Unplaced, None, Placed, Fixed, and Locked/Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Instance Pin Properties Form

The following table describes the fields available on the Attributes tab of the Edit Instance Pin Properties form.

Field	Description
Library, Cell, and View	Sets the library, cell, and view names of the master cell for this instance. **Note:** You can use the SKILL function leEnableInstPropEditFields to enable or disable the *Library* or *Cell* fields for an instance in the Property Editor form.
Browse	Lets you select the library, cell, and view names by clicking on them in the browser.
Origin: X and Y	Sets the X and Y coordinates of the origin of the instance.
Name	Sets the name assigned to this instance. The layout editor automatically assigns instance names that begin with the letter `I`, followed by a number.
Rotation	Sets whether the instance is rotated or mirrored. See Orientation Valid Values.
Cell Type (read only field)	Displays the master cell type used to describe the nature of a physical block or cell. Layout editor does not support any level one editing on any of these design objects. These cell types are used in Virtuoso Layout Suite XL and Virtuoso Floorplanning. See Orientation Valid Values.
Cluster	Specifies the name of the cluster associated with the instance.
Max Voltage	Displays the maximum voltage if the cluster is a power domain cluster.
Placement Status	Displays the placement status of the instance in a cyclic field.The choices are Unplaced, None, Placed, Fixed, and Locked/Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.
Pin Name	Displays the name of the pin.
Terminal Name	Displays the name of the terminal.

© 2023 Cadence Design Systems, Inc.

Field	Description
Route Method	Displays the routing method of the pin in a cyclic field. The choices are symbolic and geometric. When *Update Net Signal Type* is enabled, terminals with nets that have a *Signal Type* of Power or Ground in the schematic are automatically given a *Route Method* of Geometric when either the *Generate All From Source* or *Update Components and Nets* commands are run.

Attributes in the Edit Label Properties Form

The following table describes the fields available on the Attributes tab of the Edit Label Properties form.

Field	Description
Layer	Sets the layer on which the label is drawn.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the label.
Text	Sets the text that appears in the label.
Origin: x and y	Sets the X and Y coordinates of the label origin.
Height	Sets the height of the label, in user units (usually microns).
Rotation	Sets whether the label is rotated or mirrored. See Orientation Valid Values.
Font	Sets the text style of the label.
Justification	Sets the location of the label origin. The origin appears as a small square on the label when you place or select it.
Drafting	Prevents the label from being rotated more than 90 degrees.

© 2023 Cadence Design Systems, Inc.

Field	Description
Overbar	Specifies a display option that determines how text strings containing underscore characters are displayed in a layout window. When the overbar is disabled (default), the software displays underscore characters (_) as part of the text string. When the overbar is enabled, the software interprets underscore characters (_) in the text string name as toggle switches that control where overbars begin and end. Overbars appear above the text string, as shown in the examples. **Property Name** → **Appears in Design Window As** _abcde → $\overline{abcde}$ ab_cde → $ab\overline{cde}$ _abc_de → $\overline{abc}de$ ab_cd_ef_gh_ij → $ab\overline{cd}ef\overline{gh}ij$

Attributes in the Edit Line Properties Form

The following table describes the fields available on the Attributes tab of the Edit Line Properties form.

Field	Description
Layer	Sets the layer of the line.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Sets the coordinates of the box surrounding the line.
Points	Sets the coordinates of each point of the line.
ROD Name	Displays the name of the line if it is a ROD object.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Modgen Properties Form

The following table describes the fields available on the Attributes tab of the Edit Modgen Properties form.

Field	Description
Name	Sets the name of the modgen.
Origin	Sets the origin point of the modgen.
Rotation	Sets whether the modgen is rotated or mirrored. See Orientation Valid Values.
Rows and Columns	Specifies the number of rows and columns in the modgen.
Horizontal and Vertical Alignment	*Alignment:* Sets the alignment of the modgen.
Custom Spacing	Sets the custom spacing of the modgen.
Reference	*Layer and Purpose:* Sets the layer and purpose of the reference layer of the modgen.
Number of Members	Specifies the number of members in the modgen.

Attributes in the Edit Mosaic Properties Form

The following table describes the fields available on the Attributes tab of the Edit Mosaic Properties form.

Field	Description
Library, Cell, and View	Sets the library, cell, and view names of the master cell for this instance.
Name	Sets the name assigned to this instance. The layout editor automatically assigns instance names that begin with the letter `I`, followed by a number.
Origin: x and y	Sets the X and Y coordinates of the origin of the instance.
Rows and Columns	Sets the number of rows and columns in an array of instances.
Delta X and Delta Y	Sets the spacing between rows and columns in an array of instances.
Rotation	Sets whether the mosaic is rotated or mirrored. See Orientation Valid Values.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Path Properties Form

The following table describes the fields available on the Attributes tab of the Edit Path Properties form.

Field	Description
Layer	Sets the layer of the path.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the path.
Points	Sets the coordinates of each point of the path.
Width	Specifies the path width in user units (usually microns).
End Type	Controls how the path ends are drawn. ■ *truncate:* The path terminates at the end point with no overlap. ■ *extend:* The path terminates one-half the path width past the end point. ■ *round:* The path terminates one-third the path width past the end point, in an octagonal (chamfered) shape. Tapering starts at the endpoint of the path and the end segment is one-third the path width, with the other two segments calculated using one-third the width times the square root of two. ■ *variable:* The path terminates beyond the end point at the distance you enter in the *Begin Extension* and *End Extension* fields.
Begin Extension and End Extension	Sets the length of the beginning and ending extension in user units if *Type* is set to *variable*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Route Status	Indicates the status of the route containing the path. ■ *Routed:* Indicates normal routing status. ■ *Fixed:* Indicates a route that should not be edited automatically, though it can be edited manually. ■ *Locked/Cover:* Indicates a route that should not be edited automatically or manually. The route forms a part of the base array. To edit the route, change the routing status to *Routed* or *Fixed*.
Part of a Route	Indicates whether or not the path is part of a route. By default, it is FALSE for paths.
ROD Name	Appears if the path is a ROD object.
Subpart	Appears if the path is a multipart path and has subparts. **(Virtuoso RF Option)** For curved paths, *End Type*, *Begin Extension*, *End Extension*, *Part of a Route*, and *Route Status* attributes are disabled.

Attributes in the Edit Path Pin Properties Form

The following table describes the fields available on the Attributes tab of the Edit Path Pin Properties form.

Field	Description
Layer	Sets the layer of the path.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the path.
Points	Sets the coordinates of each point of the path.
Width	Specifies the path width in user units (usually microns).

© 2023 Cadence Design Systems, Inc.

Field	Description
End Type	Controls how the path ends are drawn. ■ *truncate:* The path terminates at the end point with no overlap. ■ *extend:* The path terminates one-half the path width past the end point. ■ *round:* The path terminates one-third the path width past the end point, in an octagonal (chamfered) shape. Tapering starts at the endpoint of the path and the end segment is one-third the path width, with the other two segments calculated using one-third the width times the square root of two. ■ *variable:* The path terminates beyond the end point at the distance you enter in the *Begin Extension* and *End Extension* fields.
Begin Extension and End Extension	Sets the length of the beginning and ending extension in user units if *Type* is set to *variable*.
Route Status	Indicates the status of the route containing the path. ■ *Routed:* Indicates normal routing status. ■ *Fixed:* Indicates a route that should not be edited automatically, though it can be edited manually. ■ *Locked/Cover:* Indicates a route that should not be edited automatically or manually. The route forms a part of the base array. To edit the route, change the routing status to *Routed* or *Fixed*.
Part of a Route	Indicates whether or not the path is part of a route. By default, it is FALSE for paths.
ROD Name	Appears if the path is a ROD object.
Subpart	Appears if the path is a multipart path and has subparts.
	(Virtuoso RF Option) For curved paths, *End Type*, *Begin Extension*, *End Extension*, *Part of a Route*, and *Route Status* attributes are disabled.

© 2023 Cadence Design Systems, Inc.

Field	Description
Placement Status	Displays the placement status of the pin in a cyclic field. The choices are Unplaced, None, Placed, Fixed, and Locked/Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.
Pin Name	Displays the name of the pin.
Terminal Name	Displays the name of the terminal.
Route Method	Displays the routing method of the pin in a cyclic field. The choices are symbolic and geometric. When *Update Net Signal Type* is enabled, terminals with nets that have a *Signal Type* of Power or Ground in the schematic are automatically given a *Route Method* of Geometric when either the *Generate All From Source* or *Update Components and Nets* commands are run.

Attributes in the Edit PathSeg Properties Form

The following table describes the fields available on the Attributes tab of the Edit PathSeg Properties form.

Field	Description
Layer	Sets the layer of the path.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the path.
Width	Specifies the path width in user units (usually microns).This field always displays the width in orthogonal units, even though a path segment may be diagonal.
Begin Point	Sets the coordinates of the beginning point of the path segment.
End Point	Sets the coordinates of the beginning point of the path segment.
Begin Style	Controls how the path segment beginning is drawn. You can choose from *truncate*, *extend*, *variable*, or *custom*. For more information, see PathSeg Begin and End Styles.

© 2023 Cadence Design Systems, Inc.

Field	Description
End Style	Controls how the path segment end is drawn. You can choose from *truncate*, *extend*, *variable*, or *custom*.
Begin Extension and End Extension	Sets the length of the beginning and ending extension in user units if *Type* is set to *variable*.
Route Status	Indicates the status of the route containing the pathseg. ■ *Routed***:** Indicates normal routing status. ■ *Fixed:* Indicates a route that should not be edited automatically, though it can be edited manually. ■ *Locked/Cover:* Indicates a route that should not be edited automatically or manually. The route forms a part of the base array. To edit the route, change the routing status to *Routed* or *Fixed*.
Part of a Route	Indicates whether or not the pathseg is part of a route.
Topology	Indicates the topology pattern of the pathseg. The valid topology values are *None, RING, STRIPE, FOLLOWPIN, IOWIRE, COREWIRE, BLOCKWIRE, PADRING, and BLOCKRING*. The default value is `None`. The Topology field is visible only if the Part of a Route field is set to FALSE. When the Part of a Route field is set to TRUE, the Edit Properties form title is updated to Edit PathSeg in Route Properties.
ROD Name	Appears if the path is a ROD object.
Subpart	Appears if the path is a multipart path and has subparts.
Setup Constraint Group	Displays the constraint group from which the data is initialized.

Attributes in the Edit PathSeg Pin Properties Form

The following table describes the fields available on the Attributes tab of the Edit PathSeg Pin Properties form.

Field	Description
Layer	Sets the layer of the path.

© 2023 Cadence Design Systems, Inc.

Field	Description
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the path.
Width	Specifies the path width in user units (usually microns).This field always displays the width in orthogonal units, even though a path segment may be diagonal.
Begin Point	Sets the coordinates of the beginning point of the path segment.
End Point	Sets the coordinates of the beginning point of the path segment.
Begin Style	Controls how the path segment beginning is drawn. You can choose from *truncate*, *extend*, *variable*, or *custom*. For more information, see PathSeg Begin and End Styles.
End Style	Controls how the path segment end is drawn. You can choose from *truncate*, *extend*, *variable*, or *custom*.
Begin Extension and End Extension	Sets the length of the beginning and ending extension in user units if *Type* is set to *variable*.
Route Status	Indicates the status of the route containing the pathseg. ■ *Routed*: Indicates normal routing status. ■ *Fixed:* Indicates a route that should not be edited automatically, though it can be edited manually. ■ *Locked/Cover:* Indicates a route that should not be edited automatically or manually. The route forms a part of the base array. To edit the route, change the routing status to *Routed* or *Fixed*.
Part of a Route	Indicates whether or not the pathseg is part of a route.

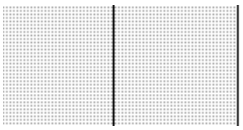
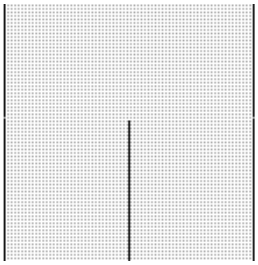
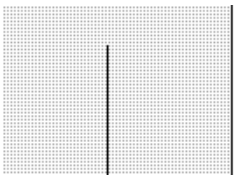
© 2023 Cadence Design Systems, Inc.

Field	Description
Topology	Indicates the topology pattern of the pathseg. The valid topology values are *None, RING, STRIPE, FOLLOWPIN, IOWIRE, COREWIRE, BLOCKWIRE, PADRING, and BLOCKRING*. The default value is `None`. The Topology field is visible only if the Part of a Route field is set to FALSE. When the Part of a Route field is set to TRUE, the Edit Properties form title is updated to Edit PathSeg in Route Properties.
ROD Name	Appears if the path is a ROD object.
Subpart	Appears if the path is a multipart path and has subparts.
Setup Constraint Group	Displays the constraint group from which the data is initialized.
Placement Status	Displays the placement status of the pin in a cyclic field. The choices are Unplaced, None, Placed, Fixed, and Locked/Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.
Pin Name	Displays the name of the pin.
Terminal Name	Displays the name of the terminal.
Route Method	Displays the routing method of the pin in a cyclic field. The choices are symbolic and geometric. When *Update Net Signal Type* is enabled, terminals with nets that have a *Signal Type* of Power or Ground in the schematic are automatically given a *Route Method* of Geometric when either the *Generate All From Source* or *Update Components and Nets* commands are run.

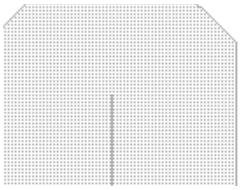
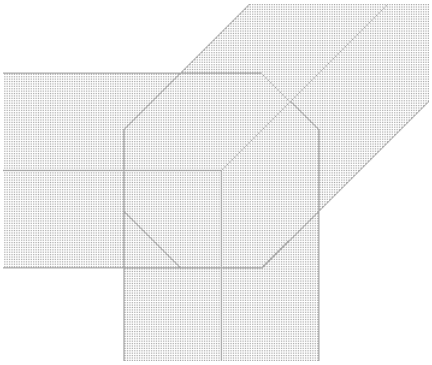
© 2023 Cadence Design Systems, Inc.

PathSeg Begin and End Styles

The following table describes the options available to draw the beginning and end of a pathSeg. You can choose them from the *Begin Style and End Style* attributes of a pathSeg.

Field	Description
truncate	The path segment begins at the begin point with no overlap. 
extend	The path segment extends from the begin point by one-half the path width. 
variable	The path segment extends from the begin point by whatever value you enter in the *Begin Extension* field, in the example below by 0.25 microns from the begin path point. 

© 2023 Cadence Design Systems, Inc.

Field	Description
custom	The path beginning extends in an octagonal (chamfered) shape. Each dimension of the chamfer is equidistant, except where the distance adjusts to accommodate 45-degree angled paths.  The custom path style adjusts to the path width when a 45-degree path end overlaps with other custom path ends. The path ends match when the end points overlap, as in the example below. 

Pin Name or Other Text Display Attributes

The following table describes the fields available for pin name or other text display attributes on the Edit Properties form.

Field	Description
Layer	Sets the layer of the text display.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the text display.

© 2023 Cadence Design Systems, Inc.

Field	Description
Text	Displays the text of the text display for the associated pin and is not editable.
Height	Sets the height of the text display in user units (usually microns).
Rotation	Sets whether the text display is rotated or mirrored. See Orientation Valid Values.
Font	Sets the text style of the text display.
Justification	Sets the location of the origin of the text display origin. The origin appears as a small square on the pin name when you place or select it.
Drafting	Prevents the text from being rotated more than 90 degrees.
Overbar	Specifies a display option that determines how text strings containing underscore characters are displayed in a layout window. When the overbar is disabled (default), the software displays underscore characters (_) as part of the text string. When the overbar is enabled, the software interprets underscore characters (_) in the text string name as toggle switches that control where overbars begin and end. Overbars appear above the text string, as shown in the examples. **Property Name** → **Appears in Design Window As** _abcde → $\overline{abcde}$ ab_cde → $ab\overline{cde}$ _abc_de → $\overline{abc}de$ ab_cd_ef_gh_ij → $ab\overline{cd}ef\overline{gh}ij$

© 2023 Cadence Design Systems, Inc.

Placement Halo Blockage Attributes

The following table describes the fields available on the Attributes tab of the Edit Properties form for placement halo blockage.

Field	Description
Left Offset	Specifies the number of user units the placement halo is offset from the left of the instance bBox.
Right Offset	Specifies the number of user units the placement halo is offset from the right of the instance bBox.
Bottom Offset	Specifies the number of user units the placement halo is offset from the bottom of the instance bBox.
Top Offset	Specifies the number of user units the placement halo is offset from the top of the instance bBox.

Attributes in the Edit Polygon Properties Form

The following table describes the fields available on the Attributes tab of the Edit Polygon Properties form.

Field	Description
Layer	Sets the layer of the polygon.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Sets the coordinates of the box surrounding the polygon.
Points	Sets the coordinates of each point of the polygon.
ROD Name	Displays the name of the polygon if it is a ROD object.

Attributes in the Edit Polygon Pin Properties Form

The following table describes the fields available on the Attributes tab of the Edit Polygon Pin Properties form**.**

Field	Description
Layer	Displays the layer of the rectangle.

© 2023 Cadence Design Systems, Inc.

Field	Description
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the pin.
Points	Displays the coordinates of each point of the pin.
Placement Status	Displays the placement status of the pin in a cyclic field. The choices are Unplaced, None, Placed, Fixed, and Locked/Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.
Pin Name	Displays the name of the pin.
Terminal Name	Displays the name of the terminal.
Route Method	Displays the routing method of the pin in a cyclic field. The choices are symbolic and geometric. When *Update Net Signal Type* is enabled, terminals with nets that have a *Signal Type* of Power or Ground in the schematic are automatically given a *Route Method* of Geometric when either the *Generate All From Source* or *Update Components and Nets* commands are run.
MPT Coloring: Color and State	Displays the color and state of the pin. **(Virtuoso RF Option)** For curved polygon pins, you can view the attributes described in this table.

Virtuoso Photonics Option

The following table describes the additional options that are displayed for a photonics pin on the *Attribute* tab of the Edit Properties form.

Field	Description
Width	Lets you specify the width of the facet of the waveguide that connects to this pin.
Angle	Lets you specify the angle of the facet of the waveguide at the intersection of the pin.
Radius	Lets you specify the radius of the curve of the waveguide center at the point it meets the facet.

© 2023 Cadence Design Systems, Inc.

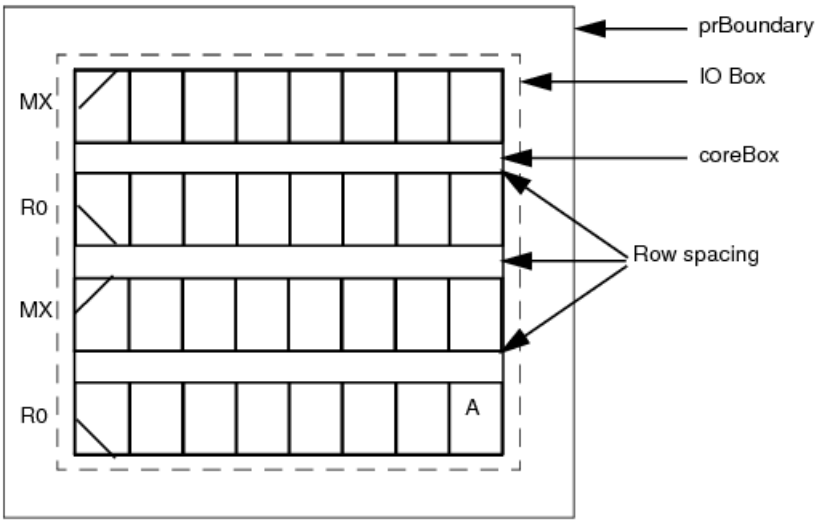
You need the `Virtuoso_Photonics_Option` license to use this functionality. For information on obtaining the required license, contact your local Cadence representative.

Attributes in the Edit **PR Boundary** Properties Form

The following table describes the fields available on the Attributes tab of the Edit PR Boundary Properties form.

Field	Description
Points	Displays the coordinates of each point of the PR boundary.
BBox	Displays the coordinates of the box surrounding the PR boundary.
Area	Displays the total area of the PR boundary in user units.
Rectangle and Rectilinear	Displays whether a boundary is a rectangle or a polygon. The PR boundary is always a rectangle.
Halo	Allows you to optionally create a new or edit an existing area halo associated with the PR boundary object. This area halo is always on offset to the bounding box of the PR boundary object.
IOBox	Displays the *Left, Bottom, Right, and Top* coordinates of the bounding box of the IOBox. IOBox represents the area reserved for pads.
CoreBoxSpec	Displays the *Left, Bottom, Right, and Top* coordinates of the bounding box. CoreBoxSpec is a rectangular grid of sites where multiple rows can be placed in horizontal or vertical direction. This is the area where standard cells can be placed. An IOBox, together with the CoreBox, indicates the area for the IO cells.
Site Pattern	Displays the site definition as core or pad, as defined in the technology file.
Horizontal	Specifies whether the row orientation is horizontal or not.
Spacing	Displays the row offset value in user units.
Flip Type	Displays which rows in the site can be flipped. Cyclic choices include *Even*, *Odd*, *None*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Spacing Type	Displays the rows to which the spacing value applies. Cyclic choices include *Every*, *Even*, *Odd*. In the example below, the prBoundary has an IO box and a coreBox. The coreBox has four horizontal rows of siteDef A. All rows are separated by a distance row spacing. Every odd numbered row is flipped. The number of rows and row geometry is computed based on the overall bBox of the coreBox and the siteDef of the coreBox. API is available to retrieve the number of rows and bounding box of each row. For information about accessing boundary attributes, see Generic Boundary Attributes and Boundary SKILL Functions. 

Attributes in the Edit Rectangle Pin Properties Form

The following table describes the fields available on the Attributes tab of the Edit Rectangle Pin Properties form.

Field	Description
Layer	Displays the layer of the rectangle.

© 2023 Cadence Design Systems, Inc.

Field	Description
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
Left, Right, Bottom, and Top	Displays the coordinates of the pin.
Edit	Specifies a radio button that lets you edit the coordinates. This is the topmost edit button.
Width and Height	Displays the width and height of the pin.
Placement Status	Displays the placement status of the pin in a cyclic field. The choices are Unplaced, None, Placed, Fixed, and Locked/Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.
Pin Name	Displays the name of the pin.
Terminal Name	Displays the name of the terminal.
Route Method	Displays the routing method of the pin in a cyclic field. The choices are symbolic and geometric. When *Update Net Signal Type* is enabled, terminals with nets that have a *Signal Type* of Power or Ground in the schematic are automatically given a *Route Method* of Geometric when either the *Generate All From Source* or *Update Components and Nets* commands are run.
MPT Coloring: Color and State	Displays the color and state of the pin.

ROD Multipart Path - Master Path Fields Attributes

The following table describes the fields available on the Attributes tab for ROD Multipart Path.

Field	Description
Layer	Sets the layer of the master path.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
bBox	Displays the coordinates of the box surrounding the path.
Points	Sets the coordinates of each point of the master path.

© 2023 Cadence Design Systems, Inc.

Field	Description
Width	Specifies the path width in user units (usually microns).
Type	Controls how the path ends are drawn.
Begin Extension and End Extension	Sets the length of the beginning and ending extension in user units if *Type* is set to *variable*.
Choppable	Indicates whether or not a ROD path can be chopped. The value must be `t` or `nil`. When a path has subparts and the master path is choppable, all subpaths and sets of subrectangles must be choppable also. When a path has subparts and the master path is not choppable, each subpath and/or set of subrectangles can be choppable or not. The default is choppable.
ROD Name	Displays the name of the MPP.
Subpart	Opens one of the three Edit ROD Subpart forms, Offset Subpath, Enclosure Subpath, or Subrectangle. See the ROD Subpart Form for a description of the fields and options.

Routing, Slot, Pin, or Feed Through Blockage Attributes

The following table describes the fields available on the Attributes tab of the Edit Properties form for routing, slot, pin, or feed through blockage attributes.

Field	Description
Owner Information	Sets the name of the (optional) owner of the blockage.
Points	Sets the coordinates of each point of the cluster boundary.
Area	Displays the total area in user units of the blockage.
Type	Displays the type of blockage you are editing. In this case, the blockage would be of the type: routing, slot, pin, feed, or placement.
Placement Blockage Type	Displays the type of placement blockage: Hard, Partial, Soft or Macro-Only.

© 2023 Cadence Design Systems, Inc.

Field	Description
Max Density	Displays the maximum percentage of the total design area that the metal in the specified layer occupies. Based on the placement blockage type, the following are the valid values of the Max Density field: ■ Hard: 0% ■ Partial: Value greater than 0% and less than 100%. The value can be a floating point number. Default value is 50%. ■ Soft: 0% ■ Macro-Only: 100%
Layer	Selects the layer on which the blockage is enforced.
Layer Filter	Enables you to search for a layer-purpose pair and narrow down the available choices in the *Layer* list.
Material	Selects the material on which the blockage is enforced.
Use Effective Width	Enables the *Effective Width* field, when clicked.
Effective width	Sets the value which is checked against the layer spacing rule or table spacing rules, if defined.
Allow PGNet	Allows shapes with signal types power and ground to be placed within the blockage area.
Color	Lets you select the color in which the blockage is created. The default value is *grayColor*. The options available in the *Color* field are based on the layer selected in the *Layer* field. If you select a layer in the *Layer* field that support *mask1Color* and *mask2Color*, the *Color* field will have the options, *grayColor*, *mask1Color*, *mask2Color*, *blackColor*, and *multiColor*.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Row Properties Form

The following table describes the fields available on the Attributes tab of the Edit Row Properties form.

Field	Description
Base Name	Allows you to enter a unique name for each row object you create. You enter the base name and the software adds an underscore followed by a number which is greater than or equal to 1. As you enter new sites, each site is incremented by 1. For example, if your base name is ROW_1, the name assigned to the first site is ROW_1_1, the next is ROW_1_2, the next is ROW_1_3, and so on. If you do not enter a custom row name the software assigns the name row_number, which increments by 1 each time you add a site and rows.
Row Width	Displays the width of the row.
Rotation	Allows you to set the row orientation. See Orientation Valid Values.
Origin X and Y	Allows you to set a corner of the row at a specific point in the cellview.
Site Definition Name	Specifies a read only field displaying the name of the current row.
Technology Library	Specifies a read only field displaying the technology library the original row was defined in.
Site Information	Specifies a read-only field displaying the type of site definition (core or pad) that is the current site in the Site Definition Name field. This information contains the type (core or pad), the width and height of the site definition, and three types of symmetry. This information is specified in the technology library site definitions. The symmetry can be SymmetricInX, SymmetricInY, or SymmetricInR90.
Number of Sites	Sets the number of sites in each row. The number of sites is any integer equal to or greater than zero.
Templates	Allows you to place rails and/or components automatically into the rows.

© 2023 Cadence Design Systems, Inc.

Field	Description
Type radio buttons	Displays Rails, Allowed Components, or Both in the Pre-Defined section of the form. You move them into the Selected section of the form using the arrows. Once they are in the Selected section of the form, you add the components individually selected from the Selected field.

Attributes in the Edit Snap Boundary Properties Form

The following table describes the fields available on the Attributes tab of the Edit Snap Boundary Properties form.

Field	Description
Points	Sets the coordinates of each point of the snap boundary.
bBox	Displays the coordinates of the box surrounding the snap boundary.
Area	Displays the total area in user units of the snap boundary.

Attributes in the Edit Via Properties Form

The following table describes the fields available on the Attributes tab of the Edit Via Properties form.

Field	Description
Via Definition cyclic field	Lists all valid via definitions available in the effective technology file. Choosing a via from the cyclic field selects the via from the effective technology file.
Net Name	Applies a net name to the via you are placing in the cellview.
Technology Library / Via Type	Specifies a read only field that displays the technology library from which the selected via definition is obtained. If you select a via definition from the list which is defined in a different library, the field is updated to that library name. The field also displays the type of `viaDef` displayed in the *Via Definition* field. The `viaDefs` are `standard` or `custom`.
Rotation	Displays the current rotation of the via.

© 2023 Cadence Design Systems, Inc.

Field	Description
Origin: X and Y	Sets the X and Y coordinates of the via origin.
Justification	Sets the origin of a via. Choose `offset` to manually set the X and Y origin points. Cyclic field choices are: `offset`, `lowerLeft`, `centerLeft`, `upperLeft`, `lowerCenter`, `centerCenter`, `upperCenter`, `lowerRight`, `centerRight`, `upperRight`. The value `offset` is not available for a cdsVia.
Reset Parameters to	Enables you to reset the values in the form to the technology database values if you want to revert from the manually updated parameter values to the defaults. You can reset the values to *Minimum Rules then ViaDef*, *Minimum Rules*, or *Minimum Rule Enclosures*.
Compute From	This section provides options available in *Compute From* section.
Rows and Columns	Sets the number of rows or the number of columns of via cuts in a via array. If you change the values in the *Rows* and *Columns* fields, the values in the other fields, such as *Width*, *Length*, *Row Spacing*, and *Column Spacing*, are dynamically updated.When you update the value in the *Row* or *Column* field, this will the update other values in the form and apply the constraints. This will happen when you finish editing the values in the *Row* or *Column* fields and click at some other area in the form.
Swap button	Swaps the values in the *Rows* and *Columns* fields. If the swapped values violate the minimum rules, the spacing and enclosure values are re-calculated based on the *Reset Parameters to* setting.
Width and Height	Sets the width and height of the via, in user units (typically microns)
Cut Section	This section provides options available in Cut Section.

© 2023 Cadence Design Systems, Inc.

Field	Description
Cut Class cyclic field	Displays cut class choices only if a `cutClasses` constraint has been defined for the cut layer of the selected viaDef in the `layerRules` section of the technology file. The following are examples of cut classes: ■ *Small* is a single cut via where both dimensions are short. ■ *Bar* is a large via that has one long and one short dimension. The shorter dimension is equal to the width or length of `Small`. ■ *Large* is a large via that has two long dimensions, equal to the longer dimension of *Bar*. If you update the *Cut Class* selection, the values in the other fields, such as *Rows*, *Columns*, *Width*, *Length*, *Row Spacing*, and *Column Spacing*, are dynamically updated.
Width and Length	Sets the width and length of the via cut, in user units (typically microns). If a *Cut Class* is selected, the cut *Width* and *Length* are automatically updated to the cut size values defined for that cut class in the `layerRules` section of the technology file.
Swap button	Swaps the values in the *Width* and *Length* fields.
Row Spacing	Sets the distance between the horizontal edge to edge cuts in a via.
Column Spacing	Sets the distance between the vertical edge to edge cuts in a via.
Swap button	*Swaps the values in the Row Spacing and Column Spacing fields.*
Edit Cut Pattern	Opens the Edit Cut Pattern Form and enables you to edit cut patterns on via arrays.
Enclosures	Sets directional offset distances on Layer 1, the bottom routing layer, and Layer 2, the top routing layer.
Compute Enclosures	Calculates enclosures automatically. The *Compute Enclosures* setting does not impact the via you create in the *Auto* mode.

© 2023 Cadence Design Systems, Inc.

Field	Description
Left	■ *Layer 1:* Sets the horizontal distance on the left side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:*Sets the horizontal distance on the left side of the via from the cut edge to the enclosure edge on the layer designated layer 2 (top routing layer).
Right	■ *Layer 1:* Sets the horizontal distance on the right side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the horizontal distance on the right side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (top routing layer).
Top	■ *Layer 1:* Sets the vertical distance on the top side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the vertical distance on the top side of the via from the cut edge to the enclosure edge on the layer designated layer 2 (top routing layer).
Bottom	■ *Layer 1:* Sets the vertical distance on the bottom side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the vertical distance on the bottom side of the via from the cut edge to the enclosure edge on the layer designated layer 2 (top routing layer).
Implant X	■ *Layer 1:* Sets the horizontal distance from the left and right cut edge to the implant enclosure edge on the layer designated layer 1 (bottom routing layer). This is a read-only field. ■ *Layer 2:* Sets the horizontal distance from the left and right cut edge to the implant enclosure edge on the layer designated layer 2 (top routing layer). This is a read-only field.

© 2023 Cadence Design Systems, Inc.

Field	Description
Implant Y	■ *Layer 1:* Sets the vertical distance from the top and bottom cut edge to the implant enclosure edge on the layer designated layer 1 (bottom routing layer). This is a read-only field. ■ *Layer 2:* Sets the vertical distance from the top and bottom cut edge to the implant enclosure edge on the layer designated layer 2 (top routing layer). This is a read-only field.
Route Status	Indicates the status of the route containing the via. ■ *Routed:* Indicates normal routing status. ■ *Fixed:* Indicates a route that should not be edited automatically, though it can be edited manually. ■ *Locked/Cover:* Indicates a route that should not be edited automatically or manually. The route forms a part of the base array. To edit the route, change the routing status to *Routed* or *Fixed*.
Part of a Route	Indicates whether or not the via is part of a route. The following fields are available only for a cdsVia: ■ *Metal Purpose:* Displays the metal layer purpose. ■ *Cut Purpose:* Displays the via layer purpose. ■ *Other Purpose:* Displays the implant and well layer purpose.
MPT Coloring	For a selected via, you can set the colors for Layer1, CutLayer, and Layer2. For all the via layers, the following MPT colors are available: grayColor, mask1Color, mask2Color, and mask3Color.

© 2023 Cadence Design Systems, Inc.

Attributes in the Edit Via Pin Properties Form

The following table describes the fields available on the Attributes tab of the Edit Via Pin Properties form.

Field	Description
Via Definition cyclic field	Lists all valid via definitions available in the effective technology file. Choosing a via from the cyclic field selects the via from the effective technology file.
Net Name	Applies a net name to the via you are placing in the cellview.
Technology Library / Via Type	Specifies a read only field that displays the technology library from which the selected via definition is obtained. If you select a via definition from the list which is defined in a different library, the field is updated to that library name. The field also displays the type of `viaDef` displayed in the *Via Definition* field. The `viaDefs` are `standard` or `custom`.
Rotation	Displays the current rotation of the via.
Origin: X and Y	Sets the X and Y coordinates of the via origin.
Justification	Sets the origin of a via. Choose `offset` to manually set the X and Y origin points. Cyclic field choices are: `offset`, `lowerLeft`, `centerLeft`, `upperLeft`, `lowerCenter`, `centerCenter`, `upperCenter`, `lowerRight`, `centerRight`, `upperRight`. The value `offset` is not available for a cdsVia.
Reset Parameters to	Enables you to reset the values in the form to the technology database values if you want to revert from the manually updated parameter values to the defaults. You can reset the values to *Minimum Rules then ViaDef*, *Minimum Rules*, or *Minimum Rule Enclosures*.
Compute From	This section provides options available in *Compute From* section.

© 2023 Cadence Design Systems, Inc.

Field	Description
Rows and Columns	Sets the number of rows or the number of columns of via cuts in a via array. If you change the values in the *Rows* and *Columns* fields, the values in the other fields, such as *Width*, *Length*, *Row Spacing*, and *Column Spacing*, are dynamically updated.When you update the value in the *Row* or *Column* field, this will the update other values in the form and apply the constraints. This will happen when you finish editing the values in the *Row* or *Column* fields and click at some other area in the form.
Swap button	Swaps the values in the *Rows* and *Columns* fields. If the swapped values violate the minimum rules, the spacing and enclosure values are re-calculated based on the *Reset Parameters to* setting.
Width and Height	Sets the width and height of the via, in user units (typically microns)
Cut Section	This section provides options available in Cut Section.
Cut Class cyclic field	Displays cut class choices only if a `cutClasses` constraint has been defined for the cut layer of the selected viaDef in the `layerRules` section of the technology file. The following are examples of cut classes: ■ *Small* is a single cut via where both dimensions are short. ■ *Bar* is a large via that has one long and one short dimension. The shorter dimension is equal to the width or length of `Small`. ■ *Large* is a large via that has two long dimensions, equal to the longer dimension of *Bar*. If you update the *Cut Class* selection, the values in the other fields, such as *Rows*, *Columns*, *Width*, *Length*, *Row Spacing*, and *Column Spacing*, are dynamically updated.
Width and Length	Sets the width and length of the via cut, in user units (typically microns). If a *Cut Class* is selected, the cut *Width* and *Length* are automatically updated to the cut size values defined for that cut class in the `layerRules` section of the technology file.
Swap button	Swaps the values in the *Width* and *Length* fields.

© 2023 Cadence Design Systems, Inc.

Field	Description
Row Spacing	Sets the distance between the horizontal edge to edge cuts in a via.
Column Spacing	Sets the distance between the vertical edge to edge cuts in a via.
Swap button	*Swaps the values in the Row Spacing and Column Spacing fields.*
Edit Cut Pattern	Opens the Edit Cut Pattern Form and enables you to edit cut patterns on via arrays.
Enclosures	Sets directional offset distances on Layer 1, the bottom routing layer, and Layer 2, the top routing layer.
Compute Enclosures	Calculates enclosures automatically. The *Compute Enclosures* setting does not impact the via you create in the *Auto* mode.
Left	■ *Layer 1:* Sets the horizontal distance on the left side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:*Sets the horizontal distance on the left side of the via from the cut edge to the enclosure edge on the layer designated layer 2 (top routing layer).
Right	■ *Layer 1:* Sets the horizontal distance on the right side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the horizontal distance on the right side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (top routing layer).
Top	■ *Layer 1:* Sets the vertical distance on the top side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the vertical distance on the top side of the via from the cut edge to the enclosure edge on the layer designated layer 2 (top routing layer).

© 2023 Cadence Design Systems, Inc.

Field	Description
Bottom	■ *Layer 1:* Sets the vertical distance on the bottom side of the via from the cut edge to the enclosure edge on the layer designated layer 1 (bottom routing layer). ■ *Layer 2:* Sets the vertical distance on the bottom side of the via from the cut edge to the enclosure edge on the layer designated layer 2 (top routing layer).
Implant X	■ *Layer 1:* Sets the horizontal distance from the left and right cut edge to the implant enclosure edge on the layer designated layer 1 (bottom routing layer). This is a read-only field. ■ *Layer 2:* Sets the horizontal distance from the left and right cut edge to the implant enclosure edge on the layer designated layer 2 (top routing layer). This is a read-only field.
Implant Y	■ *Layer 1:* Sets the vertical distance from the top and bottom cut edge to the implant enclosure edge on the layer designated layer 1 (bottom routing layer). This is a read-only field. ■ *Layer 2:* Sets the vertical distance from the top and bottom cut edge to the implant enclosure edge on the layer designated layer 2 (top routing layer). This is a read-only field.
Route Status	Indicates the status of the route containing the via. ■ *Routed:* Indicates normal routing status. ■ *Fixed:* Indicates a route that should not be edited automatically, though it can be edited manually. ■ *Locked/Cover:* Indicates a route that should not be edited automatically or manually. The route forms a part of the base array. To edit the route, change the routing status to *Routed* or *Fixed*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Part of a Route	Indicates whether or not the via is part of a route. The following fields are available only for a cdsVia: ■ *Metal Purpose:* Displays the metal layer purpose. ■ *Cut Purpose:* Displays the via layer purpose. ■ *Other Purpose:* Displays the implant and well layer purpose.
MPT Coloring	For a selected via, you can set the colors for Layer1, CutLayer, and Layer2. For all the via layers, the following MPT colors are available: grayColor, mask1Color, mask2Color, and mask3Color.
Pin	This section provides options available in *Pin* Section.
Placement Status	Displays the placement status of the pin in a cyclic field. The choices are Unplaced, None, Placed, Fixed, and Locked/Cover. If the placement status is *Unplaced*, the instance will not be visible. See Placement Status Valid Values.
Pin Name	Displays the name of the pin.
Terminal Name	Displays the name of the terminal.
Route Method	Displays the routing method of the pin in a cyclic field. The choices are symbolic and geometric. When *Update Net Signal Type* is enabled, terminals with nets that have a *Signal Type* of Power or Ground in the schematic are automatically given a *Route Method* of Geometric when either the *Generate All From Source* or *Update Components and Nets* commands are run.

Attributes in the Edit Via Stack Properties Form

The following table describes the fields available on the *Attributes* tab of the Edit Via Stack Properties form.

Field	Description
Rotation	Displays the current rotation of the via stack.

© 2023 Cadence Design Systems, Inc.

Field	Description
Justification	Sets the origin of a via stack. Cyclic field choices are*: bottom-left, center-left, top-left, bottom-center, center-center, top-center, bottom-right, center-right, or top-right.*
Origin: X and Y	Sets the X and Y coordinates of the via stack origin.
Compute From Section	■ *Top Rows* and *Top Columns:* Sets the number of rows and columns for the top via in the via stack. ■ *Width and Height:* Sets the width and height of the via stack, in user units (typically microns).
Definition Section	■ *Start Layer:* Lets you select one of the two layers between which you want to place the via stack. The layer selected in the *End Layer* list does not appear in the *Start Layer* list. ■ *End Layer:* Lets you select one of the two layers between which you want to place the via stack. The layer selected in the *Start Layer* list does not appear in the *End Layer* list. ■ *Transitions: Via Definitions :*Lets you select viaDefs for each layer pair between the specified *Start Layer* and *End Layer*.
Cut Class	Displays cut class choices only if a cutClasses constraint has been defined for the cut layer of the selected viaDef in the layerRules section of the technology file.

Virtual Hierarchy / VH Clone Attributes

The following table describes the fields available on the Attributes tab of the Edit Properties form for a virtual hierarchy or clone.

Field	Description
Name	Displays the name of the virtual hierarchy or virtual hierarchy clone.
Master	Displays the library, cell, and view name of the virtual hierarchy or virtual hierarchy clone.
Type	Displays the type of virtual hierarchy or virtual hierarchy clone, *Generated* or *Created*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Frozen	Sets whether the virtual hierarchy or virtual hierarchy clone is frozen or not.
Family	Displays the family of the virtual hierarchy or virtual hierarchy clone.
Origin X and Origin Y	Sets the origin point of the virtual hierarchy or virtual hierarchy clone.
Rotation	Sets whether the virtual hierarchy or virtual hierarchy clone is rotated or mirrored. See Orientation Valid Values.
Number of Members	Displays the number of members in the virtual hierarchy or virtual hierarchy clone.

© 2023 Cadence Design Systems, Inc.

Placement Status Valid Values

Value	Description
Unplaced	Unplaced instances can be placed and moved. Indicates that the current placement is arbitrary and should be changed. Unplaced objects do not contribute to the bounding box of the containing design, and they will not be returned by a region query.
None	Default value.
Placed	Placed instances can be moved by automatic and interactive commands. Indicates that an effort has been made by the software to place this object but it may be placed in future attempts.
Fixed	Fixed instances can be moved by interactive commands.
Locked/Cover	Locked instances cannot be moved by any commands. The placement of this object should not be changed.

Orientation Valid Values

Value	Description
RO	no rotation
R90	rotated 90 degrees
R180	rotated 180 degrees
R270	rotated 270 degrees
MY	mirrored over the Y axis
MYR90	mirrored over Y, rotated 90 degrees
MX	mirrored over the X axis
MXR90	mirrored over X, rotated 90 degrees

Related Topics

Edit Properties Form

© 2023 Cadence Design Systems, Inc.

Edit Cut Pattern Form

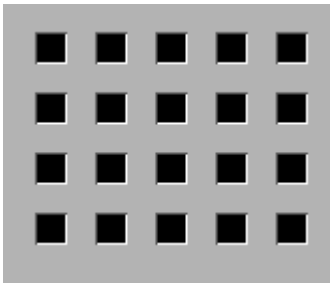
Use the Edit Cut Pattern form to edit cut patterns for standard and custom vias.

Field	Description
Quick Edit	Allows you to edit the cut pattern in conjunction with the *Patterns* cyclic field to select a predetermined pattern. The form displays a maximum of ten rows and ten columns when you select this option. You can create a cut pattern with rows and columns that are fewer than the rows and columns of the array, since the selected pattern repeats across the entire array, starting from the lower left corner of the array and repeating upwards and to the right.
Wall Paper	Allows you to edit the edit the cut spaces cut by cut or use the Patterns cyclic field to select a predetermined pattern. To edit cut by cut click the square to turn off the cut. This leaves a space in the cuts when you create the via. The form displays a maximum of ten rows and ten columns when you select this option. You can create a cut pattern with rows and columns that are fewer than the rows and columns of the array, since the selected pattern repeats across the entire array, starting from the lower left corner of the array and repeating upwards and to the right.
Manual Edit	Displays all of the rows and columns. You can edit the cut spaces cut by cut or use the *Patterns* cyclic field to select a predetermined pattern like those shown below. To edit cut by cut, click the square to turn off the cut. This leaves a space in the cuts when you create the via.

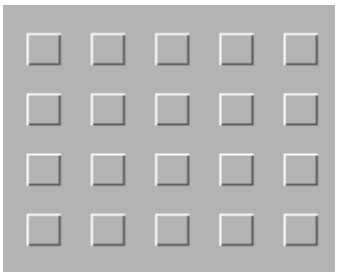
© 2023 Cadence Design Systems, Inc.

Field	Description
Patterns	Lets you select a predetermined pattern, such as All On, All Off, Top, Bottom, Left, Right, H Stripes, V Stripes, Chess Board, U-L Corner, U-R Corner, L-L Corner, L-R Corner, Cross, Farm, and As Is.

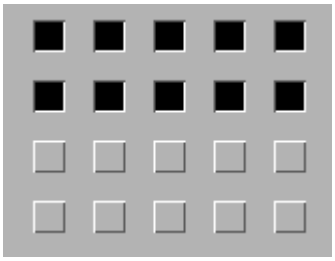
All On



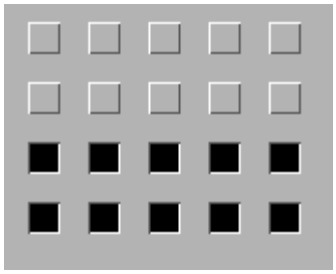
All Off



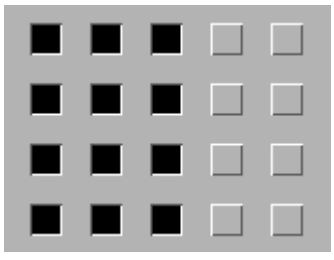
Top



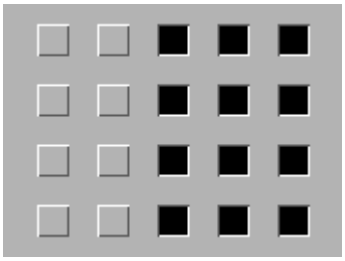
Bottom



Left



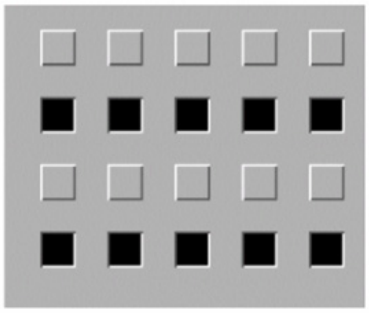
Right



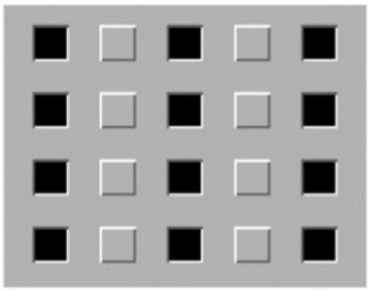
© 2023 Cadence Design Systems, Inc.

Field	Description

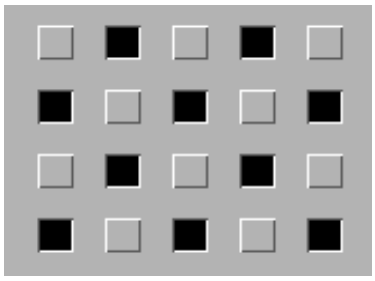
H Stripes (Horizontal Stripes)



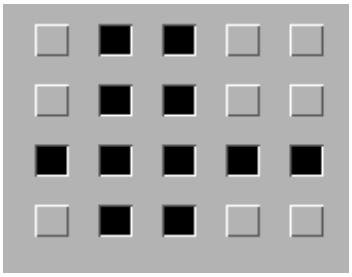
V Stripes (Vertical Stripes)



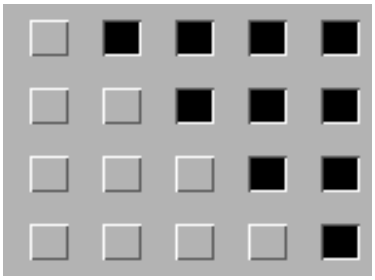
Chess Board



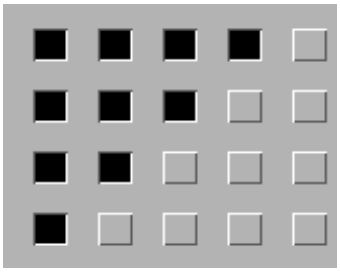
Cross



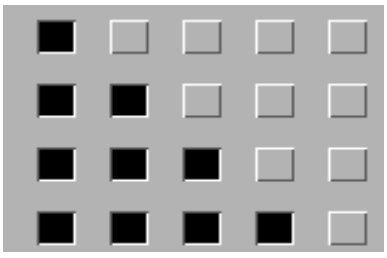
U-R Corner (Upper Right Corner)



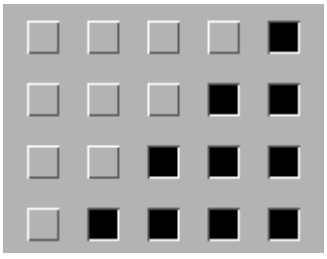
U-L Corner (Upper Left Corner)



L-L Corner (Lower Left Corner)



L-R Corner (Lower Right Corner)



© 2023 Cadence Design Systems, Inc.

Field	Description
Paper Size	Allows you add or delete rows or columns to the cut pattern when *Wall Paper* is selected. The form displays a maximum of ten rows and ten columns when you select this option. ■ Left arrow: Deletes a column with each click. ■ Right arrow: Adds a column with each click. ■ Up arrow: Adds a row with each click. ■ Down arrow: Deletes a row with each click.

Related Topics

Editing Cut Patterns

© 2023 Cadence Design Systems, Inc.

Edit Synchronous Clone Form

Use the Edit Synchronous Clone form to automatically replicate changes in all the clones in the synchronous figure group. If you make a change to the physical topology of one member of a synchronized family, that change is propagated to all the other clones in the family.

Field	Description
Do not show me this dialog again	Prevents the Edit Synchronous Clone form from being displayed the next time you edit in place (EIP) a synchronous clone figure group.

Related Topics

Editing a Clone in a Synchronized Family

© 2023 Cadence Design Systems, Inc.

Edit Via Array Pattern Form

Use the Edit Via Array Pattern form to specify the spacing to be applied between the arrays of cuts.

Edit Via Array Pattern is applicable only for a cdsGenVia.

Field	Description
Spacing X	Specifies the spacing to be applied between the arrays of cuts horizontally.
Spacing Y	Specifies the spacing to be applied between the arrays of cuts vertically.
Swap button 	Swaps the values in the *Spacing X* and *Spacing Y* fields.
Pattern X	Specifies the number of cuts in the array horizontally.
Pattern Y	Specifies the number of cuts in the array vertically. For *Pattern X* and *Pattern Y* fields, you can provide multiple numbers separate by spaces. For example, if you specify 3 2 4 in the *Pattern X* field, then the via engine horizontally creates 3 cuts (separate by a “small” cut spacing), then a “large” *Spacing X*, then 2 cuts (separate by a “small” cut spacing), then a “large” spacing X, then 4 cuts (separate by a “small” cut spacing). This pattern is repeated or truncated depending on the number of cut columns specified on the *System* tab. For a 15x15 via, with 3 2 4 specified in the *Pattern X* field, the via engine will create a 3 2 4 3 2 1via.
Swap button 	Swaps the values in the *Pattern X* and *Pattern Y* fields.

Related Topics

Creating a Via with Arrayed Via Cuts

© 2023 Cadence Design Systems, Inc.

Enter Points Form

Use the Enter Points form to create an object at specific coordinates in the design window.

All point adjustment commands, such as DRD, Grid, Snap Mode, Acute Angle, and Gravity are disabled when the points are entered using the Enter Points form.

Field	Description
XY	Enables you to specify the points for an object, such as a rectangle, a polygon, an arc, a path, a circle, an ellipse, a donut, geometric wires, instances, pins, labels, vias, multipart paths, fluid guard rings, MPP guard rings, slots, and PR objects. The object appears on the canvas at the points specified in this field. In this field you can also specify points for commands such as, Pan, Zoom, Move, Copy, Chop, and Edit In Place.
Use Relative XY	Indicates that the points specified in the *XY* field are the points relative to the last entered points of the command (delta X and Y). Environment variable: useRelativeXY

Related Topics

Points Formats Supported in the Enter Points Forms

Entering Points Using the Enter Points Form

leGetCoordinateForm

© 2023 Cadence Design Systems, Inc.

Export Image Form

Use the Export Image form to take a screen capture of the design contained in the specified window and save it to a standard image format.You can access the form by selecting *File – Export Image*.

Field	Description
Export Region	Specifies the area of the design to capture: ■ *Visible area*: Select this option if you want to capture only the part of the design that is currently visible in the window. ■ *Entire design:* Select this option if you want to capture the entire design, not just the part that is currently visible in the window. In other words, this option will disregard the current zoom level and capture the entire design. ■ *Selected area:* Select this option if you want to capture a region. Click Select and then click-and-drag the mouse in the graphics window to create a bounding box around the area you want to capture. The screen coordinates of the selected area are then displayed in this field.

© 2023 Cadence Design Systems, Inc.

Field	Description
Size	Specifies the desired size of the resulting image. You can either provide the width and height for the image or specify a scale factor. ■ *Scale exported region by*: Specify the factor by which to scale the image. A factor greater than 1.0 scales the image up (for example, a factor of 2.0 doubles the size of the image) and a factor less than 1.0 scales the image down (for example, a factor of 0.5 reduces the size by half). The default factor is 1.0. **Note:** An image requires <width>*<height>*<scaleFactor>*32 bits of memory, therefore scaling an image to a high factor can cause an out of memory condition. If insufficient memory to save an image is an issue, a warning message will be displayed advising that you scale down the image size before attempting to save it again. ■ *Custom*: Specify a custom size for the image. You can specify either the width or the height, or both. If you specify only the width or only the height, then the other value is calculated automatically according to the aspect ratio of the region being captured. ■ *Width and Height:* If you specify both the width and height, then they are taken to be the maximum values and, in order to maintain the aspect ratio of the original region, either one might be reduced. The resulting image is never larger than the width and height specified. For example, if you specify a width of 400 and a height of 400, and the area being captured is not as high as it is wide, then the resulting image will be 400 pixels wide but less than 400 pixels high to maintain the aspect ratio. ■ *Maintain aspect ratio:* If Maintain aspect ratio is unchecked, then an image of the specified size is created. The aspect ratio of the original region is still maintained and the image is padded with blank space as required.

© 2023 Cadence Design Systems, Inc.

Field	Description
Appearance	■ *Full color/Monochrome/Bi-color*: Specify the color format to be used. ■ *Transparent Background*: Select for the image to have a transparent background. ■ *Background/Foreground*: Specify the color to be used for the image background or foreground using the Select Color form. ■ *Swap*: Switches the Background and Foreground settings.
Halos	■ *Style*: Select the halo style to be export, *Gradient* (default) or *Flat*. ■ *Scale with image size*: Select to re-size halos in accordance with image export size.
Output	■ *Name:* Specify the name of the file for the image to be saved as. ■ *Type:* Specify the format to save the image in. ■ *Directory Icon*: Specify the location to save the image to. ■ *Save to File*: Saves the image to the specified location. ■ *Copy to Clipboard*: Copies the image to the clipboard for re-use. ■ *Close dialog after saving image*: By default, the Export Image dialog closes after you capture an image. If you want the dialog box to remain open in order to capture more images, uncheck this option. ■ *Reset to Defaults:* Restores field settings to original values.

Related Topics

hiExportImageDialog

© 2023 Cadence Design Systems, Inc.

Export Label Form

Use the Export Label form to export the name, coordinates, and layer-purpose pair for some or all of the labels in your design to a text file.

Field	Description
Export only selected	Exports information only for the currently selected labels. If this option is not selected, information for all the labels in the current design is exported.
Output File	Specifies the name of the file containing the label information. Type the path and name of the file into the field or use the *Browse* button to navigate to the required location on disk.

Related Topics

Label Creation

© 2023 Cadence Design Systems, Inc.

Find Marker Form

Use the Find Marker form to search for and highlight each error or warning marker in a layout to help you correct errors.

Field	Description
Severity	Specifies the type of markers displayed. Select *all* or *none* to select all types of markers or none of the markers to be displayed. ■ *fatal error* tool generated error. Can not be signed off, as it will always result in a non-functioning design or a design that can not be manufactured. Such error are generated usually from the process end (design rule) where the far/ foundry does not want to allow sign-off. ■ *critical error* tool generated error. Usually requires a design review to sign- off. ■ *signed off critical error* displays error that has been reviewed by the design team, and has been signed-off. ■ *error* displays error markers. ■ *signed off error* displays error that has been signed-off. ■ *warning* displays warning markers. ■ *acknowledged warning* displays warning that has been acknowledged by the designer. ■ *info* tool generated information. ■ *annotation* user generated information. Environment variable: markerSeverity

© 2023 Cadence Design Systems, Inc.

Field	Description
Search Scope	Specifies what part of the design hierarchy you want to search. ■ *cellview:* Searches only for objects in the cellview displayed in the current window. ■ *hierarchy starting from:* Searches all cellviews specified in the design hierarchy. The hierarchy either starts with the *top cellview* or the *current cellview*. ■ *library:* Searches all cellviews in the specified library. You must type in the library name, and the library must exist in the current search path. Environment variable: markerSearchScope
Options	*Zoom To Markers:* Enlarges the position of the marker in the window by zooming in on the specified marker; when off, displays the marker but does not zoom in. Environment variable: markerZoom
Cellview Access Mode	Specifies the mode for opening a cellview. You base your selection on what you want to do in the cellview. ■ *read:* Displays the cellview but does not permit you to edit and save. Find marker searches for the markers with the criteria specified in the find marker form and opens the cellview containing the first desired marker in read mode. However, if that cellview is already opened in Edit mode, the cellview is not re-opened in read mode. ■ *write:* Displays the cellview and permits you to edit and save. Find marker searches for the markers with the criteria specified in the find marker form and opens the cellview containing the first desired marker in Edit mode. If the cellview is already opened in read mode, the cellview is re-opened in Edit mode so that you can delete markers if required. Environment variable: markerAccessMode

© 2023 Cadence Design Systems, Inc.

Field	Description
Find by	Specifies which markers will display. ■ *master cell only:* Searches only for markers in the master cell. ■ *all occurrences:* Searches for all occurrences of markers. Environment variable: markerSearchBy

Related Topics

Finding Markers in a Design

© 2023 Cadence Design Systems, Inc.

Find/Replace Form

Use the Find/Replace form to search for and replace objects with specific attributes or property values.

Field	Description
Search	This section enables you to specify the search criteria.
Search for	Sets the type of object to be searched, such as a label or pin.
Add Criteria	Adds a line of fields to the form that lets you make your search more specific. ■ Attribute list: Indicates the specific attributes, instance, library, view name, or properties for the selected object type. The list values change to reflect the attributes of the type of object you select in the *Search for* field. ■ Condition/logic list: Searches for character strings or numerals that are an exact match (==) or not a match (!=), to get matches that contain a string, a leading and/or trailing asterisk must be used. The symbols <, <=, >, >= are used where applicable. *EXIST* and *!EXIST* locate or exclude discrete blocks of information anywhere in the string. ■ Value list or text field: Sets the value of the attribute or property to use in the search. In the text field, you can specify the search string using the wildcard characters. If a layer is selected in the attribute list and the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form, only valid layers displayed in the *Layers* assistant are populated in the value list. You can search for a specific layer-purpose pair using the search filter. The search feature does not accept ranges in the Add Criteria fields. For example, if you enter two path widths, you get the error message, “No figures found that meet the current search criteria.”
Delete	Deletes the corresponding line of criteria from the form.
Match all of the above	Displays objects that meet all the search criteria specified in the Add Criteria box.

© 2023 Cadence Design Systems, Inc.

Field	Description
Match any of the above	Displays objects that meet any of the search criteria specified in the Add Criteria box.
Case Sensitive	Makes the search case sensitive.
Search Hierarchy Range	Sets the levels of hierarchy to be considered for the search. ■ *current cellView*: Searches in the current edit cellview in the layout window. This is the default option. ■ *current to bottom*: Searches from the current edit cellview level to the bottom of the hierarchy. ■ *current to stop level*: Searches from the current edit cellview level to the stop level specified in the *Display Levels Stop* field in the Display Options form. ■ *current to user level*: Searches from the current edit cellview level to the level that you specify.
Region	Specifies the area of the cellview that needs to be searched. ■ *Entire CellView*: Searches the entire cellview. This is the default option. ■ *Viewing Area:* Searches the portion of the cellview that is displayed in the canvas at that instant. ■ *Draw a Rectangle Region on the Canvas* or *Draw a Polygon Region on the Canvas:* Enables you to specify a rectangular or polygonal area for the search. Select the required option, and then click in the canvas to specify the vertices of the rectangle or polygon.
Search	Specifies the scope of object search. This option is available only when the viewing area or the rectangle or polygon area is selected as *Region*. ■ *Enclosed Figures:* Searches the objects completely enclosed by the viewing area or the rectangle or polygon area. ■ *Enclosed and Crossed Figures*: Searches the objects completely or partially enclosed by the viewing area or the rectangle or polygon area.

© 2023 Cadence Design Systems, Inc.

Field	Description
Replace	This section enables you to specify the values with which search results need to be replaced. The Replace section is available only in Layout XL and higher tiers.
Replace	Enables you to replace an object's search results with a new object, attribute value, or property value. In the replace box, the list contains frequently used properties. In the text field, you can specify the value of the property with which you want to replace the original value. The value specified replaces the property values of all the objects matching the search criteria. For example, to replace parameter values for a parameterized cell, set *Replace* to *property* and type the parameter name.
Add Criteria	Adds a line of fields to the form that lets you make the replace operation more specific. For more information, see Object Search Criteria.
Search Results	This section enables you to specify how the search results are displayed.
Zoom To Figure	Centers the cellview around the current object in the search group when you click *Find*, *Previous*, or *Next*.
Current Figure	Enables you to specify the number of the objects in the search group that you want to highlight as you click *Previous* or *Next*. The value in this field is the range of the current figures found in the search group. Press `Tab` after you specify the figure number to highlight the figure.
Figure Count	Displays the number of objects found, after you click *Find*. When a current-to-bottom search is done, the figure count returns the total number of figures found. In this case, the *Current Figure* field returns the figures with unique database IDs (dbIds).
Add Select	Selects the current object in the search group.
Deselect	Deselects the current object in the search group.
Replace	Edits or replaces the current object with the settings shown in the *Replace* section.
Previous	Highlights the previous object in the search group.

© 2023 Cadence Design Systems, Inc.

Field	Description
Next	Highlights the next object in the search group.
Reports	This section enables you to save the report for find and replace operations to a specified file.
Show Detailed Report in CIW	
	Displays a detailed report for find and replace operations in the CIW. Environment variable: showFindReplaceReport
Save Report To	Saves the report of the find and replace operations to a specified file. The default file is "`vlsSearchReport.csv`". Environment variable: saveFindReplaceReportToFile, findReplaceReportFileName
Browse	Lets you specify the path where you want to save the report of the find and replace operation.

You can use the following buttons in the Find/Replace form to perform the find and replace operations.

Button	Description
Find	Searches for objects matching the settings in the Find/Replace form.
Select All	Selects all objects found by *Find/Replace*.
Deselect All	Deselects all objects found by *Find/Replace*.
Replace All	Replaces all objects found by *Find/Replace* with the settings shown in the *Replace* section.

Related Topics

Calculating Area and Density

Object Search Criteria

© 2023 Cadence Design Systems, Inc.

Flatten Form

Use the Flatten form to explode a cell instance or array, move the contents of the cell or array up one or more levels into the current level of the hierarchy.

Field	Description
Hierarchy Level	Controls the number of design hierarchy levels that are flattened. You can move an instance up one level or move the contents of all displayed instances up to the current cellview level. You can select *one level*, *displayed level*, or *user level*. You can select a value if you select the *user level* option. Environment variable: flattenMode
Pcells	Flattens any selected parameterized cells. Environment variable: flattenPCells
Vias	Flattens the selected vias. This option is selected by default. If this option is not selected, geometries in the selected via instances are preserved. Environment variable: flattenVias

© 2023 Cadence Design Systems, Inc.

Field	Description
Preserve	■ *Pins:* Preserves the connectivity information of flattened pins. Environment variable: preservePins ■ *Pin geometries:* Preserves the geometric information of flattened pins. Deselect the *Preserve: Pins* option to select this option. Environment variable: preservePinFigs Use the environment variable flattenKeepLabels to retain labels on shapes when an instance is flattened. ■ *Term name:* Preserves the terminal name of flattened pins. Environment variable: preserveTermName ■ *ROD objects:* Preserves the attributes of ROD objects (such as object name, alignments, multipart path subparts, multipart rectangles and subrectangles) when an instance is flattened. The option is off by default. When this option is off, ROD objects become ordinary unnamed objects, the sub-parts of multipart paths become ordinary paths and rectangles, the master rectangles in multipart rectangles become ordinary, unnamed rectangles, and the subrectangles of multipart rectangles become ordinary rectangles. Environment variable: preserveRodObjects
Selections	Retains the selection of all objects at instance level after the objects are flattened.
Detached blockages	Deletes detached blockages which would otherwise remain after an instance containing detached blockages is flattened. Environment variable: delDetachedBlockages
Labels	Controls whether labels on shapes are retained when an instance is flattened. Environment variable: flattenKeepLabels

© 2023 Cadence Design Systems, Inc.

Related Topics

Instance Hierarchy Flattening

Transparent Layout Hierarchy Generation

leHiFlatten

© 2023 Cadence Design Systems, Inc.

Flip Form

Use the Flip form to flip an object horizontally or vertically in the specified window..

Field	Description
Specify reference and direction	Activates the options in the form for specifying the reference and the direction to be used for flipping an object. Environment variable: useFlipReference
Reference	Specifies the point around which an object is flipped. The default is *CenterCenter*. Environment variable: flipReference
X,Y	Specifies the X and Y coordinates around which an object is flipped. This field is available when you select *Custom* from the *Reference* list.
Flip around combined bbox	Flips multiple selected objects around the specified reference point of the combined bounding box of the objects. Environment variable: flipAroundCombinedCenter
Direction	Specifies whether an object is flipped horizontally along the X axis or vertically along the Y axis.

Related Topics

Object Flipping

useRotateReference

leHiFlip

© 2023 Cadence Design Systems, Inc.

Flip Vertical Form

Use the Flip Vertical form to define how objects are flipped vertically.

Field	Description
Reference	Specifies the reference point around which an object is mirrored along the Y axis. Valid values are *LowerLeft*, *CenterLeft*, *UpperLeft*, *LowerCenter*, *CenterCenter*, *UpperCenter*, *LowerRight*, *CenterRight*, and *UpperRight*. Environment variable: flipReference
Flip around combined bbox	Flips multiple selected objects around the specified reference point of the combined bounding box of the objects. Environment variable: flipAroundCombinedCenter

© 2023 Cadence Design Systems, Inc.

Flip Horizontal Form

Use the Flip Horizontal form to define how objects are flipped horizontally.

Field	Description
Reference	Specifies the reference point around which an object is mirrored along the X axis. Valid values are *LowerLeft*, *CenterLeft*, *UpperLeft*, *LowerCenter*, *CenterCenter*, *UpperCenter*, *LowerRight*, *CenterRight*, and *UpperRight*. Environment variable: flipReference
Flip around combined bbox	Flips multiple selected objects around the specified reference point of the combined bounding box of the objects. Environment variable: flipAroundCombinedCenter

Related Topics

Object Flipping

© 2023 Cadence Design Systems, Inc.

Freeze Pcell Instance Form

Use the Freeze Pcell Instance form to specify the library, cell, and view of the non-Pcell instance of the new master created on the disk to freeze a Pcell instance.

Field	Description
Library	Lets you select the name of the library in which you want to create the non-Pcell instance. If one Pcell instance is selected for freeze, the library name in which the Pcell instance is saved is automatically populated in the *Library* field. If multiple Pcell instances are selected, the value `Default` is populated in the *Library* field. The `Default` value implies that each non-Pcell instance will be created in its respective library.
Cell	Lets you type the name of the non-Pcell instance. A unique name is automatically populated in the *Cells* field. However, you can update the name of the cell. Also, when multiple Pcell instances are selected, the value `Default` (non-editable) is automatically populated in the *Cells* field.
View	Displays the view of the non-Pcell instance.
Use Existing Master	Lets you use the existing frozen Pcell master, if available, when converting a Pcell instance to a non-Pcell instance. The *Use Existing Master* and *Overwrite Existing Master* options are mutually exclusive; only one of them can be enabled at a time. Environment variable: reuseExistingFrozenMaster
Overwrite Existing Master	Enables you to overwrite the existing master.

Related Topics

Freezing and Unfreezing Pcell Instances

© 2023 Cadence Design Systems, Inc.

Group Return Form

Use the Group Return form to define the return mode used when exiting editing-in-place.

Field	Description
Return Mode	Selects the type of return mode. The choices are group or group and instance.
Remember return mode	Uses the value selected in the Return Mode when the *Group Return* command is run again.

Related Topics

Editing a Cell in Place

© 2023 Cadence Design Systems, Inc.

Guard Ring Form

Use the Guard Ring form to create an MPP guard ring.

Field	Description
Guard Ring Template	Specifies the template to be used for creating the guard ring.
Technology Library cyclic field	Lets you choose the technology library in which the template you want to use is defined.
Net Name	Specifies the net with which you want to associate the master path or subpart.
Enclose by	Specifies the distance between the guard ring and the objects that the guard ring will surround. Environment variable: grEnclosedBy
Rectangular	Sets the shape of the guard ring to a rectangle.
Rectilinear	Allows the shape of the guard ring to be a polygon.

Related Topics

Creating an MPP Guard Ring

© 2023 Cadence Design Systems, Inc.

Halo Editor Form

Use the Halo Editor form to create and modify halos. You can access the form by clicking *Define Halo* in the Create PR Boundary and Create Instance forms.

Field	Description
Existing Halos	Displays all halo blockages you have created from the fields at the bottom of the form. In the case displayed above one placement, two routing, and one slot halo have been created and added to the field using the Add/Update button.
Delete	Deletes all blockages selected in the Existing Halos field.
Add/Update	Adds any newly created halo. Update changes any selected halo to the offsets entered in the Offsets fields.
Blockage Type	Selects the type of blockage to create. The choices are: placement, routing, fill, slot, pin, feedthru, or screen. Environment variable: blockageType
Material	Selects the material on which the blockage is enforced. The choices are: all, other, nwell, pwell, nplus, pplus, ndiff, pdiff, poly, metal, diffusion, li, metal, cut or diffusion. Not all materials are legal for all types of blockages. When Blockage Type is routing all materials are available except cut and li. When Blockage Type is placement the Material field is grayed out. Other denotes undefined materials. Environment variable: blockageMaterial
Layer	Selects the layer associated with the selected material. The layers displayed are those defined in the technology file as having one of the above materials. When Blockage Type is placement the Layer field is grayed out. Environment variable: blockageLayer

© 2023 Cadence Design Systems, Inc.

Field	Description
Offsets	Specifies the number of user units the placement halo is offset from the instance master's prBoundary. ■ *Top:* Specifies the number of user units the placement halo is extended from the top of the instance bBox. Environment variable: haloTopOffset ■ *Bottom:* Specifies the number of user units the placement halo is extended from the bottom of the instance bBox. Environment variable: haloBottomOffset ■ *Right:* Specifies the number of user units the placement halo is extended from the right of the instance bBox. Environment variable: haloRightOffset ■ *Left:* Specifies the number of user units the placement halo is extended from the left of the instance bBox. Environment variable: haloLeftOffset

Related Topics

Blockage Halos

Create P&R Boundary Form

Create Instance Form

© 2023 Cadence Design Systems, Inc.

Highlight Options Form

Use the Highlight Options form to control the objects highlighted when a net is selected or probed. You can choose to selectively display or hide probes for specified nets.

Field	Description
Highlight	This section lets you specify the objects that are highlighted when a net is selected or probed.
Pins	Specifies that the pins on probed nets are highlighted. Environment variable: probeNetDisplayCheckPins
Instances	Specifies that the instances on probed nets are highlighted. Environment variable: probeNetDisplayCheckInstance
Visible Shapes Only	Specifies that shapes, pins, and vias on the visible layers are highlighted. Environment variable: probeVisibleLpp
FlightLines	Enables you to choose one of the following settings for flight lines: ■ *Physical*: Specifies that all shapes on the probed net are highlighted. ■ *Logical*: Specifies that segments linking all pins on the probed net are highlighted. Note: Flight lines are visible in VLS XL and higher tiers. Environment variable: probeNetDisplayFlightLines
From Selected Objects (Pins, Instances, Shapes)	Specifies that the nets associated with probed or selected pins, instances, and shapes are highlighted. Environment variables: probeTermDisplayRadio, probeInstDisplayRadio, probeShapeDisplay
Hierarchy Depth	Specifies the hierarchy level you want to display for net probing. Environment variable: probeNetDisplayHierarchyLevel
Dynamic Highlight	Specifies that any object over which the pointer hovers is probed. Environment variable: dynamicProbe

© 2023 Cadence Design Systems, Inc.

Field	Description
Drawing Mode	Enables you to choose the drawing mode for the next probe. ■ *True Color and Highlight Color*: Specifies the probed shapes to be redrawn with their layer-purpose pair (LPP) and the highlight color. ■ *Highlight Color Only*: Specifies the probed shapes to be redrawn only with the LPP chosen from *Highlight Color* list. ■ *True Color Only*: Specifies the probed shapes to be redrawn only with their LPP. The drawing mode with True Color enables you to visualize the different layers of the connected shapes in the cellview. Environment variable: trueColorProbe Note: To highlight probed nets, it is recommended that display dimming is enabled. When the drawing mode is *True Color and Highlight Color* or *True Color Only,* the *Enable Dimming* option on the Display Options form is selected automatically. However, you can deselect the *Enable Dimming* option. Video For a video demonstration of the *True Color* option, see *True Color Probe*.
Highlight Color	Shows the highlight drawing layers to be used on the next probe that will be drawn. You can choose *Cyclic* or select an LPP. Environment variable: probeHighlight
Name Filtering	This section lets you selectively display or hide the probes only for the specified list of nets.
Filter net names	Lets you enable or disable probe filtering by net name. Environment variable: probeNetNameFilterByName
Filter Mode	Lets you show or hide the probes for specified nets. Environment variable: probeNetNameFilterByNameMode

© 2023 Cadence Design Systems, Inc.

Field	Description
Net Names	Lets you specify multiple net names and add them to the list. The net names should be separated by a space. This is a case-sensitive field. Environment variable: probeNetNameFilterByNameDefaultList
Add To List	Lets you add to the list the net names that you specified in *Net Names*.
Remove From List	Removes the selected net names from an already existing list.
Remove All	Removes all net names from the list.
Add Selected Nets	Lets you add nets to the list by selecting the nets in the *Navigator* assistant or from the layout canvas.
Save To Cellview	Lets you save the current filtering options to the cellview. This option is available only when the current cellview is editable.
Load From Cellview	Lets you load the saved filtering options of the cellview in a different window.
Delete From Cellview	Lets you delete the saved filtering options from the cellview

Related Topics

Probing Nets, Instances, Pins, and Shapes

© 2023 Cadence Design Systems, Inc.

Instance to Mosaic Form

Use the Instance to Mosaic form to convert selected instances into mosaics.

Field	Description
Mode	Sets the mode for conversion of the selected instances into mosaics. You can choose from *Convert Instances into Mosaics* or *Convert Instances into Single Mosaic*. Environment variable: convertEachInstanceIntoMosaic ■ *Convert Instances into Mosaics:* Specifies the default and converts a number of selected instances into an equivalent number of 1x1 mosaics. ■ *Convert Instances into Single Mosaic:* Converts a number of selected instances of the same master into a single mosaic.

Related Topics

Converting Instances to Mosaics

© 2023 Cadence Design Systems, Inc.

Layer Generation Form

Use the Layer Generation form to create new shapes by performing logical operations on selected objects in the specified layers.

Field	Description
Input Layers (the first and third fields)	Sets the layer or layers you want to modify with *Layer Generation*. You can search for specific layer-purpose pair using the search filter in the first and third fields.
Boolean Operators (second field)	Combines the selected objects on the input layers. ■ *GROW BY*: Redraws the Layer Generation form so you can enter a resizing value. Enter the number of user units (resizing value) by which to enlarge selected objects on the input layer in the field to the right.
Output Layer (fourth field)	Sets the layer on which the new shapes will appear. You can search for specific layer-purpose pair using the search filter. The layer selection using the search filter is available for *Input Layers* and *Output Layer* fields only if the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form.

Related Topics

Layer Generation

leHiLayerGen

© 2023 Cadence Design Systems, Inc.

Layout Editor Options Form

Use the Layout Editor Options form to change the layout editor environment. The form contains the following sections and the common fields.

Section/Field	Description
Editor Controls	Enables you to set various editor controls such as whether to allow commands to be repeated and connections to be maintained.
Foundry Constraints	Specifies the name of the constraint group that contains foundry constraints.
Gravity Controls	Enables you to define gravity controls to set the pointer to automatically snap to objects in a cellview.
Tap	Enables you to define controls for tapping objects.
Snapping	Enables you to define controls for snapping objects.
Wire Editing	Enables you to specify the default wire constraint group and via parameters calculation mode.
File	Specifies the file to which you want to store or from which you want to load the settings. You can enter the correct directory where you want to save the file. If you do not remember the name of the directory use the *Browse* button. Environment variable: editorOptionsFile
Save To	Saves the current setting from the *Layout Editor Options* form to the file specified in the *File* field.
Load From	Sets the *Layout Editor Options* form settings to the values saved in the specified file.

© 2023 Cadence Design Systems, Inc.

Editor Controls

The following table describes the fields available in the *Editor Controls* section of the Layout Editor Options form.

Field	Description
Repeat Commands	Sets all commands to automatically repeat, if you first select the command and then the object to edit. Does not affect commands if you first select an object, then the command. Environment variable: modalCommands
Display Reference Point	Displays a bold asterisk (*) on the current reference point in your layout. Environment variable: displayRefPoint
Auto Set Reference Point	Sets a reference point automatically whenever you enter a new point. You can set the reference point to a particular point by pressing + key on the keypad. The pointer changes to an asterisk (✱). The point you click is the new reference point. You can set the layout editor so that the reference point does not change by selecting *Auto Set Reference Point*. This allows you to place several objects using a single reference point. You use reference points to measure the distance between points you enter. Environment variable: autoSetRefPoint
Recursion Check	Turns recursion checking on. Recursion checking prevents the creation of recursive hierarchy (where a cell has an instance of itself at some level) by commands such as *Create Instance*, *Make Cell*, and *Find/Replace*. Environment variable: recursionCheck
Maintain Connections	Turns on an automatic re-connection mode for pins, paths, and pathsegs that have been separated during any editing activity that moves pins. The check box is off by default. See Pin-Path and Pin-PathSeg Connections Preserved. Environment variable: maintainConnections

© 2023 Cadence Design Systems, Inc.

Field	Description
Abut Server	Turns automatic abutment and automatic unabutment on and off. When on, unabutment automatically restores abutted devices when moved apart. This ensures that dropped vias are restored when chained devices are moved apart. It also eliminates errors such as devices with no diffusion overlap on one side. When off, automatic abutment and automatic unabutment are off. Also, when set off in the layout editor, unabutment retains the abutted device configuration even when the device is unabutted.
Select Created Object	Ensures that the object you create remains selected after you complete the create, copy and paste commands. If you create multiple objects, all of them remain selected after being created. With this option selected, any object(s) that you may have selected before starting the create command remains selected after the create command completes. The option is off by default.The Repeat commands should be disabled for this option to work for copy and paste commands. Environment variable: selectCreatedObj
Rotate Around Combined BBox	Rotates multiple selected objects around the specified reference point of the combined bounding box of the objects. This check box is selected by default. Environment variable: rotateAroundCombinedCenter
Flip Around Combined BBox	Flips multiple selected objects around the specified reference point of the combined bounding box of the objects. This check box is selected by default. Environment variable: flipAroundCombinedCenter
Turn Off Infix in Smart Snapping	Turns off infix when commands, such as *Create Measurement* and *Quick Align*, that use smart snapping are active. Environment variable: turnInfixOffWhenSmartSnapping

© 2023 Cadence Design Systems, Inc.

Field	Description
Use Layer Palette LPPs Only	Ensures only valid layers displayed in the *Layers* assistant are populated in the layer field of the *Property Editor*, *Move*, *Copy*, *Label,* and *Find/Replace* forms. When this option is selected, you can select an LPP not present in the filtered layers of the Palette but exists in the technology file. The name of the LPP is case-sensitive and should match the name specified in the technology file. Environment variable: validLppFilterOn
Check Large Coordinates:	Provides control for checking for large coordinates during the copy operation. The *Copy* command does not allow objects to go out of the INT_MAX boundary and hence does not distort the copied image. The value of INT_MAX is 2147483647 (2^{31} -1) (signed int max value). When you copy an object, until the copy operation is committed, an image of the copied object is displayed when you drag. This image is referred as the copied image. Environment variable: checkLargeCoordinates
Palette LPPs Display Order	Enables you to change the display order of layer-purpose pairs in the canvas. You can do this by dragging up and down the layer-purpose pairs listed in the Palette when Palette mode is set to *Multi-select*. Environment variable: pteLppsDisplayOrderControl
Display Active LPP in Foreground	Displays the shapes on the active layer in the foreground, on top of the shapes drawn on all other layers. Environment variable: pteDisplayActiveLppInForeground

© 2023 Cadence Design Systems, Inc.

Field	Description
Instance Editing Mode	Specifies whether the *Flip, Rotate,* and *Create Instance* (*when creating an array of instances*) commands use the instance bounding box (*BBox*), the place and route boundary (*P&R Boundary*), or the *Snap Boundary* as the instance border when operating on instances or mosaics. For example, if you have a standard cell with a large label that extends well beyond the PR boundary, the bounding box of that cell will be much larger than its place and route boundary. Set this option to *P&R Boundary* to have *Flip* and *Rotate* ignore the bounding box and instead use the PR boundary when operating on the cell. The default is *BBox*. If you set this environment variable to *P&R Boundary* or *Snap Boundary* and the instance or mosaic does not have the corresponding boundary, then *BBox* is used instead. Environment variable: editInstanceBoundaryMode
Half Width Check	Sets whether or not you can create or edit a path that has a segment with length less than half the path width. You can either disable the checking (*off*) or enable it (*notify* or *enforce*). If you set it to *off*, a warning message is displayed only if the first and last segments of the path are less than half the path width. If you set the checking to *never*, no warning message is displayed if the length of any segment of the path is less than half the path width. Environment variable: checkHalfWidthPathSeg
Conic Sides	Sets the number of segments to use when you convert conics to polygons. The maximum number of sides is 4000. Environment variable: numSides

© 2023 Cadence Design Systems, Inc.

Foundry Constraints

The following table describes the field available in the *Foundry Constraints* section of the Layout Editor Options form.

Field	Description
Constraint Group	Specifies the name of the constraint group that contains the foundry constraints. This field is available only when the `AlternateFoundryCG` environment variable or the `afcgDefaultFoundry` parameter is set to a non-empty string. In case an invalid constraint group is specified, the value in this field is set to *foundry*. You can display a foundry description by setting the `afcgDefaultFoundry` parameter to a non-empty string in the `techParams` subsection of a technology file. For example, the following image shows how the `afcgDefaultFoundry "My Foundry Description"` parameter value appears in the *Foundry Constraints* section. 

Gravity Controls

The following table describes the fields available in the *Gravity Controls* section of the Layout Editor Options form.

Field	Description
Gravity On	Sets the drawing pointer to automatically snap to objects in the cellview. The pointer snaps to all the visible objects at any hierarchy level. Environment variable: gravityOn

© 2023 Cadence Design Systems, Inc.

Field	Description
Distance slider	Sets the snap distance between the pointer and an object. You can choose from *Small*, *Medium*, *Large*, or *Very Large* using the slider. Environment variable: gravityDistance
Selectable Objects Only	Sets the drawing pointer to snap only to the selected objects in the cellview. Environment variable: gravitySelectableOnly

Tap

The following table describes the fields available in the *Tap* section of the Layout Editor Options form.

Field	Description
Auto Tap	Sets the layer of the wire being created to be the same as that of an existing clicked wire automatically. Attributes of objects are also tapped. In addition to object attributes, an existing net name can be tapped. ■ *Wire:* Controls whether auto tap occurs on the first entry point when you use the *Create – Wire* command. When *Auto Tap Wire* is on, the current entry layer will seed the form and update attributes of the wire you create based on the first point you enter. *Auto Tap Wire* is also available from the shortcut menu when the *Create – Wire* command is active. Environment variable: autoTap ■ *Shape*: Controls whether auto tap occurs on the first entry point when you create rectangles, polygons, paths, circles, donuts, ellipses, and blockages. During the create command, when *Auto Tap Shape* is on, the current entry layer will seed the form and update attributes of the object you create based on the first point you enter. Environment variable: autoTapShape For more information, see Tapping Wires and Object Tapping.

© 2023 Cadence Design Systems, Inc.

Field	Description
Tap Layer	Sets the tapped layer as the layer for the wire or shape being created, and updates the Layer Selection Window. *Layer* option is enabled only when one of *Wire* or *Shape* from *Auto Tap* option is on. Environment variable: autoTapType
Tap Attributes	Captures attributes from the wire or shape you click. When tapping a wire or path, you can tap the LPP, extension type, and width of the wire. The *Attributes* option is enabled only when one of *Wire* or *Shape* from *Auto Tap* option is on. Environment variable: autoTapType
Select from Overlaps	Prompts you to choose a layer or net name from the Choose Object to Tap form when clicking on a coordinate with objects on overlapping layers. The option is enabled only when one of *Wire* or *Shape* from *Auto Tap* option is on. For more information about tapping LPPs, see Tapping Wires.
Include All (Wire) Via Layers	Prompts you with the Choose Object to Tap form when tapping a via in Layout XL and higher tiers. For more information, see Tapping Vias Environment variable: layerTapPickWireVia
Tap Purpose List	Prioritizes the purposes considered when changing the current layer. For example, if you have a `metal1 pin` layer, but you want to create a wire on `metal1 drawing` layer, you must have the purpose list set to `drawing`. You can specify an ordered list of purposes in the field. For more information, see Customizing Tap Settings Using the Layout Editor Options Form. Environment variables: useTapPurposeList, tapPurposeList

© 2023 Cadence Design Systems, Inc.

Snapping

The following table describes the fields available in the *Snapping* section of the Layout Editor Options form.

Field	Description
Snap Pattern Snapping	Enables snap pattern snapping. Environment variable: snapPatternSnapping
Instance to Row	Snaps instances to rows during the commands *Edit – Move*, *Edit – Copy, Edit – Stretch*, and *Create – Instance*. Instances must be associated with a legal site. To set the site definition for an instance master, use the Edit Cut Pattern Form. Ensure that the site definition set for the instance master and the site definition corresponding to the row to which you want to snap the instance are the same. Instances snap to the nearest site within a row and match the orientation of the site. The instance must be either over a legal row or the distance between the instance and the legal row must be within the specified aperture in *Layout Editor Options* form. Rows can be horizontal or vertical and instances can be any orientation. If the instance has a snap boundary, the snap boundary snaps to the row site. If the instance has a prBoundary but no snap boundary, the prBoundary snaps to the row site. If the instance does not have either a snap boundary or a prBoundary, then the lower left corner of the instance snaps to the row site. If multiple instances are selected, the instance closest to the reference point will be snapped to the row. Environment variable: rowSiteEnable This option supports the row types - row region, dbRow and dbPlaceArea. Snapping rows to manufacturing grid is not supported.

© 2023 Cadence Design Systems, Inc.

Field	Description
Layer Manufacturing Grid	(Virtuoso Photonics Option) Determines in Layout EXL or a higher tier whether objects are snapped to coarse manufacturing grids instead of the base manufacturing grid. This feature is supported for the *Create Instance*, *Create Rectangle*, *Create Polygon*, *Create Path*, *Create Pin*, *Create Via*, *Move*, *Copy*, and *Stretch* commands. Rectangles, polygons, paths, and pins are snapped to the manufacturing grid defined for the layer in the technology file. Instances or vias are snapped to the least common multiple value of manufacturing grids of all the layers in the instance or via. Environment variable: multiMfgResolutionSnap In the technology file, the base mfgGridResolution is defined inside controls section and the layer-specific mfgGridResolution is defined in mfgResolutions sub-section in the `layerRules` section.
Wire Snapping	This section controls the snapping of wires to a grid.

© 2023 Cadence Design Systems, Inc.

Field	Description
Wires	Lets you select the snapping mode.The available options are: ■ *Manufacturing Grid* snaps the wires to the *Manufacturing Grid*. This is the default option. ■ *Routing Grid* snaps the segments and vias other than the ones directly connecting to the off-grid pins to the routing grid. ■ *Track Pattern* snaps the segments and vias other than the ones directly connecting to the off-grid pins to track. The track patterns are saved on the design and so each design can have its own set of non-uniform track patterns. ■ *Snap Pattern* enables snapping of wires to the nearest snap pattern track. For *Routing Grid* and *Track Patterns*, the pin snapping has precedence over the grid or track snapping. Therefore, if pins are off grid, it is still possible to use *Routing Grid* and Tr*ack Patterns*. The segments connecting the pins will be off grid and the other segments will be on grid. If there are track patterns in the design, then *Routing Grid* does not snap the segments on the routing grid but on the manufacturing grid. Environment variable: snapWireGrid
Enforce	Snaps the wire to the centerline of the track pattern. Also, if this option is selected, the wire inherits the width of the active width spacing pattern and the direction in which the wire is created is the same as the direction of the track pattern. Environment variable: weEnforceDirFromRoutingPattern
Instance Pins Patterns	Lets you view the patterns available in an instance. Environment variable: weInstancePinPattern

© 2023 Cadence Design Systems, Inc.

Wire Editing

The following table describes the fields available in the *Wire Editing* section of the Layout Editor Options form.

Field	Description
Wire	This section enables you to specify the default wire constraint group and various wire controls.

© 2023 Cadence Design Systems, Inc.

Field	Description

Default Wire Constraint Group	Allows the selection of one of the available constraint groups, which may be defined at the technology or the design level. The constraint group defines the appropriate layers and vias for the application being used. Only the layers, vias, and any rules specified in the selected constraint group are used when creating wires. Selection of a constraint group in the *Wire* cyclic field specifies the application default constraint group for wire creation and editing commands. The default active constraint group is virtuosoDefaultSetup.
	
	For more information about defining constraint groups, see Constraint Groups.
	
	Environment variable: wireConstraintGroup
	
	Existing wires are not adjusted based on changes to the constraint group in the *Wire Editing* section. The recommended flow is to set the constraint group at the beginning of a session and to not change the constraint group while you are working on a design. To vary the constraints on different nets, set desired constraints on specific nets by using the *Process Rule Editor*.
	
	If you update the constraint group while working on a design, remove the existing results (delete all routing) that would be effected by the constraint group change and re-route.
	
	When the constraintGroupCustomFilterMode environment variable is set to `t`, the user-defined `constraintGroupCustomFilter` API is called to filter the list of constraint groups and display the constraint groups in the *Default Wire Constraint Group* cyclic field. In the following example, the constraint groups whose names start with `cg_` are filtered and displayed in *Default Wire Constraint Group*.
	
	`procedure(constraintGroupCustomFilter(editedCellView cg)`
	`  prog((result)`
	`  result = nil`
	`  if(strncmp(cg~>name "cg_" 3) == 0 then`
	`  result = t`
	`  )`
	`  return(result)`
	`  )`
	`)`

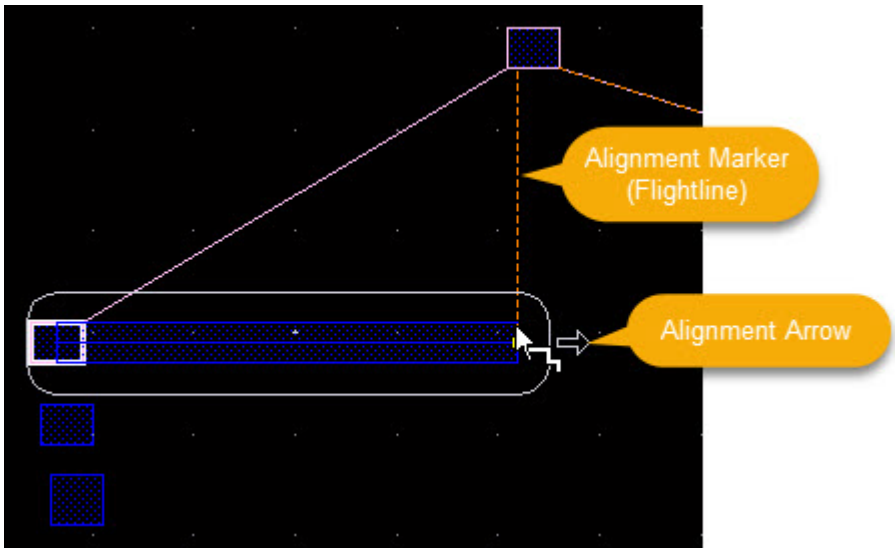
© 2023 Cadence Design Systems, Inc.

Field	Description
Via Parameters Calculation Mode	Sets the calculation mode for via parameters to one of the following: ■ *Min Rules & ViaDef*: For all the via parameter fields, such as the number of cut rows and columns, and cut sizes, the minimum rule or the viaDef default values, whichever is greater, are used. If minimum rules are not defined, viaDef defaults are used. The spacing and enclosure values are calculated automatically from the minimum rules. The minimum rules are picked from the constraint group that you select from the Via list in the *Create Via* section. The calculated values for the spacing and enclosure fields are never less than the viaDef defaults. For example, if the minExtensionDistance rule is 0.05 and the default enclosure in viaDef is 0.1, 0.1 is used as the enclosure value in the *Create Via* form. Similarly, if the minSpacing rule is 0.2 and the default cut spacing in viaDef is 0.18, 0.2 is used as the cut spacing X and Y in the *Create Via* form. ■ *Minimum Rules*: For all the via parameter fields, only the minimum rules, such as minWidth and minSpacing, are used. For example, if the minExtensionDistance rule is 0.05 and the default enclosure in viaDef is 0.1, 0.05 is used as the enclosure value in the *Create Via* form. The viaDef default enclosure value is ignored. ■ *ViaDef Defaults*: For all the via parameter fields, viaDef defaults are used. When this option is selected, the via engine does not check any rules. Environment variable: viaParamCalcMode The assisted routing commands, such as *Point to Point Routing*, *Guided Routing*, *Finish Wire*, and *Finish Entire Net*, use the *Via Parameters Calculation Mode* options (*Min Rules & ViaDef*, *Minimum Rules*, and *ViaDef Defaults*) for via creation. For more information on assisted routing commands, see Using Assisted Routing Commands.

© 2023 Cadence Design Systems, Inc.

Field	Description
Auto Merge Wires	Merges the connected geometric paths automatically. Also, allows you to choose whether or not to merge the collinear geometric pathSegs while creating or stretching a geometric wire. Environment variable: weAutoMergeWires
Allow Loops	Permits creation of loops when creating or editing wires. The option is off by default. When off, the *Create Geometric Wire* command (*Create Geometric Wire* and *Create Wire* commands in XL/GXL) does not allow the creation of loops on the same layer. When the layer is changed while wire creation, loop checking is reset and applies only to the new layer. For information about loop removal, see Removing Loops From Wires. Environment variable: allowLoops
Adjust Edited Vias Parameters	Enables the re-calculation of via alignment of the vias if you stretch the wires connected to them. This check box is not selected by default. Environment variable: adjustEditedViasParams

© 2023 Cadence Design Systems, Inc.

Field	Description
Show Alignment Markers	Displays an alignment arrow and an alignment marker. The arrow points to the closest object on the active flightline. The marker snaps the cursor to the target object edges and center as you move the cursor. Environment variable: showAlignmentMarkers 
Show Hints	Controls the display of the *Interactive Routing Bindkey* hint box in Layout EXL and higher tiers when you start the *Create Wire*, *Create Bus*, or *Create Stranded Wire* command. In Layout XL, the hint box is displayed only for the *Create Stranded Wire* command. The *Interactive Routing Bindkey* hint box describes the useful bindkeys and shortcuts for the interactive routing commands. This option is selected by default.
Create Via	This section enables you to specify the default constraint group and via parameters calculation mode.

© 2023 Cadence Design Systems, Inc.

Field	Description
Via	Specifies the constraint group from which vias will be available when adding vias using *Create – Via* or while editing vias using the *Edit Via Properties* form, the *Property Editor* assistant, or the *Create Via* form. The via spacing values in the constraint group selected here populate the *Create Via* form, as well as reflect on the *Edit Via Properties* form and *Property Editor* assistant. For more information about defining constraint groups, see Constraint Groups. Environment variable: viaConstraintGroup The spacing on the via does not always match the spacing value defined in the constraints. The via engine tries both an odd and an even spacing to avoid asymmetric enclosures.

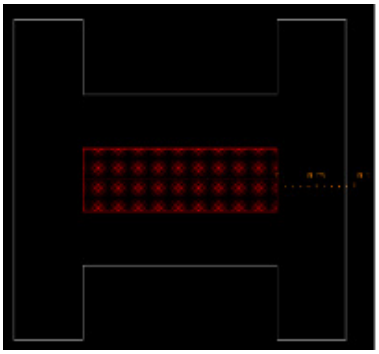
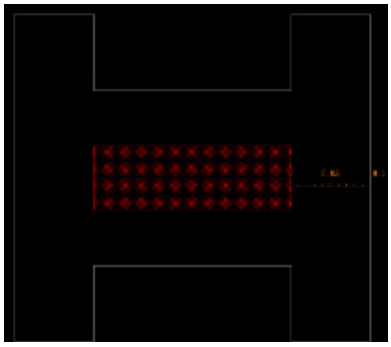
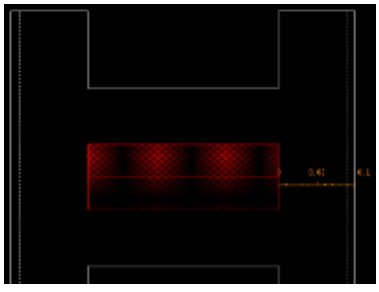
© 2023 Cadence Design Systems, Inc.

Field	Description
Via Parameters Calculation Mode	Seeds the *Create Via* form the first time when a specified viaDef is used. Then onwards, the *Create Via* form is populated from Parameter Caching. If required, you can later reset the calculation mode by using the *Reset Parameters to* list on the Create Via Form. The cyclic field values for this option include: ■ *Min Rules & ViaDef*: For all the via parameter fields, such as the number of cut rows and columns, and cut sizes, the minimum rule or the viaDef default values, whichever is greater, are used. The spacing and enclosure values are calculated automatically from the minimum rules. The minimum rules are picked from the constraint group that you select from the *Via* list in the *Create Via* section. The calculated values for the spacing and enclosure fields are never less than the viaDef defaults. For example, if the minExtensionDistance rule is 0.05 and the default enclosure in viaDef is 0.1, 0.1 is used as the enclosure value in the *Create Via* form. Similarly, if the minSpacing rule is 0.2 and the default cut spacing in viaDef is 0.18, 0.2 is used as the cut spacing X and Y in the *Create Via* form. ■ *Minimum Rules*: For all the via parameter fields, only the minimum rules, such as minWidth and minSpacing, are used. For example, if the minExtensionDistance rule is 0.05 and the default enclosure in viaDef is 0.1, 0.05 is used as the enclosure value in the *Create Via* form. The viaDef default enclosure value is ignored. ■ *ViaDef Defaults*: For all the via parameter fields, viaDef defaults are used.
Halo	This section enables you to specify how halos are represented in the layout canvas in order to make it easier to view them in the design.

© 2023 Cadence Design Systems, Inc.

Field	Description
Enable	Enables or disables a halo display in the layout canvas. When the option is unselected, the halo is not shown when the wire is created. Environment variable: weHaloEnabled

© 2023 Cadence Design Systems, Inc.

Field	Description
EOL Spacing list	Lets you select the spacing for the halo. ■ *End to Side EOL Spacing shows the end-to-side halo spacing for the* `minEndOfLineSpacing` *constraint.*  ■ *End to End EOL Spacing shows the end-to-end halo spacing for the* `minEndOfLineSpacing` *constraint.*  ■ *Both EOL Spacing shows both end-to-side and end-to-end halo spacing for the* `minEndOfLineSpacing` *constraint.*  Environment variable: weHaloEOLSpacing

© 2023 Cadence Design Systems, Inc.

Field	Description
Maskcolor list	Lets you select a scope for creating a halo from the drop-down list. ■ *Min Same & Diff Maskcolo*r shows the minimum `diffMask` or the `sameMask` spacing for the `minEndOfLineSpacing` constraint. ■ *Max Same and Diff Maskcolor* shows the maximum `diffMask` or the `sameMask` spacing for the `minEndOfLineSpacing` constraint. ■ *Both Maskcolors* shows the minimum (`diffMask`, `sameMask`) spacing and the maximum (`diffMask`, `sameMask`) spacing of the available `minEndOfLineSpacing` constraint. Environment variable: weHalo
Dashed	Creates a dashed halo around a wire, bus, or stranded wire while creating or editing an object that violates a spacing constraint. By default, this option is deselected. Environment variable: weHaloDashStyle
True Color	Creates a halo in the color of that layer for each layer. By default, this option is deselected. Environment variable: weHaloTrueColor

Related Topics

virtuosoDefaultSetup

Parameter Caching

leHiEditEditorOptions

Snap Pattern Snapping for FinFET Devices in Layout Editor

© 2023 Cadence Design Systems, Inc.

Layout Integrity Check Form

Use the Layout Integrity Check form to perform integrity checks in the layout and view a summary of results.

Field	Description
Check	Allows you to select the checks you want to perform. You can select *cells without library*, *cells without master*, *vias without definition*, *shapes without layer and purpose definition*, *Pcell evaluation failure*.
Hierarchy Range	Sets the levels of hierarchy to be considered for layout integrity check. ■ *current to bottom*: Performs the check from the current edit cellview level to the bottom of the hierarchy. ■ *current cellView:* Performs the check in the cellview currently open in the layout window. ■ *top to bottom:* Performs the check through the entire hierarchy of the cellview. ■ *current to user level:* Performs the check from the current edit cellview level to the level that you specify in the *User level* field.
Add markers in	Sets where you want to add markers. ■ *no markers:* Does not add markers. ■ *current cellview:* Adds markers in the current cellview. ■ *cellviews in the hierarchy:* Adds markers in the cellviews in the hierarchy.
Delete Markers	Deletes markers from the cellview.

Related Topics

Display Options Form

Checking Layout Integrity

enableLayoutIntegrityCheck

© 2023 Cadence Design Systems, Inc.

Load Multipart Path Template Form

Use the Load Multipart Path Template form to load ASCII template files.

Field	Description
Template File	Specify the ASCII template file that you want to load. You can use the *Browse* button to search for the file.

Related Topics

Loading an ASCII Multipart Path Template File

© 2023 Cadence Design Systems, Inc.

Magnifier Options Form

Use the Magnifier Options form to magnify (zoom in) a specific part of the design display area instead of zooming in the entire design display area.

Field	Description
Anchored at Cursor Position	Fixes the magnifier at its last position. The magnifier does not move along with the pointer. You can move the pointer and the design under and around the pointer is magnified in the magnifier. This option is always grayed out and cannot be set by using the form. Use the hash bindkey (#) to select this option. Even when selected, the option remains grayed out. The # bindkey toggles between the *Anchored at Cursor Position* and *Not Anchored* options. Environment variable: magDisplayMode
Anchored at	Specifies where the magnifier should be positioned with respect to the pointer position. By default, the magnifier appears at the *top left corner* of the design display area with respect to the pointer. The available choices include *top right corner*, *bottom left corner*, and *bottom right corner*. If you select *Anchored at*, the *Not Anchored* option is turned off. Environment variables: magDisplayMode, magAnchorPosition

© 2023 Cadence Design Systems, Inc.

Field	Description
Not Anchored	Enables you to customize the position of the magnifier either to the center of the area being magnified or as an offset value from the pointer position. If you select *Not Anchored*, the *Anchored at* option is turned off. Environment variables: magDisplayMode, magFloatPosition ■ *Center:* Positions the magnifier such that its center point coincides the pointer position. ■ *Position Magnifier:* Enables you to specify the magnifier position with respect to the pointer position. The available choices for positioning the magnifier with respect to the pointer position include *top left*, *top right*, *bottom right*, and *bottom left*. Environment variable: magOffsetCorner ■ *offset by:* Specifies the distance of the magnifier from the pointer position. You can set any value from `0` through `100`. The default value is `15`. For example, with the default *top left* setting and offset `15`, the magnifier is positioned at the top left side of the pointer and offset both on X and Y axis by a distance of 15 units. If you set the *offset by* to `0`, one of the vertices of the magnifier coincide the pointer position. Environment variable: magCursorOffset
Magnifier Size	Enables you to set the pixel size of the square magnifier window. You can set any value from `50` through `250`. The default value is `200`. Environment variable: magSize
Keep Within Window	Enables you to display the magnifier in the design display area. This check box is selected by default. If the check box is not selected, the magnifier might not display completely based on the other settings in the form. Environment variable: magConfined
Save as Defaults	Saves the specified magnifier settings in the `.cdsenv` file for the current root application. You can retrieve the updated default values by using the *Get Defaults* button.

© 2023 Cadence Design Systems, Inc.

Field	Description
Get Defaults	Retrieves the default values saved in the `.cdsenv` file for the current root application and resets the values updated, if any, in the form.

Related Topics

Magnifying the Design Area

© 2023 Cadence Design Systems, Inc.

Make Cell Form

Use the Make Cell form to create a cell from the selected objects in the specified window.

Field	Description
Library	Sets the library in which to create the new cellview.
Cell	Sets the name of the new cellview.
View	Sets the view name of the new cellview.
Type	Lets you choose the type of cell you are going to create. The valid choices are *none* (which lets you choose an unspecified object), *block*, *blockRing*, *cover*, *coverBump*, *pad*, *padSpacer*, *core*, *coreSpacer*, *coreAntenna*, *corner*, *softMacro*, *via*, *blockBlackBox*, *padAreaIO*. Environment variable: makeCellCellType
Replace selected figures	Replaces the selected objects in the cellview with an instance of the new cell.
Origin	Origin drop-down list lets you select the origin point. You can select from *LowerLeft*, *LowerRight*, *UpperLeft*, *UpperRight*, or *Custom*. When you select *Custom*, the *X, Y* and *Set by cursor* fields are enabled.
X, Y	Sets an X and Y origin point for the new cell when you enter the points in the *X, Y* fields.
Set by cursor	Sets an X and Y origin point for the new cell when you enter a new point in the cellview with the pointer.

© 2023 Cadence Design Systems, Inc.

Field	Description
Hierarchy	■ *Transparent instance:* Generates made cell instances that are transparent. This means the binder can recognize the instances and nets within a made cell instance and can bind them in a one-to-one correspondence with the schematic instances. Environment variable: makeCellTransparent ■ *Preserve connectivity:* Retains the correct logical connectivity on the made cell. Environment variable: makeCellPreserveConn ■ *Create pins:* Creates pins for nets that are external to the newly made cell, if the nets do not already have a pin created. Environment variable: makeCellCreatePins ■ *Pins below boundary:* Positions the created pins outside the design's boundary. Environment variable: makeCellPinsBelowBoundary
P&R Boundary	Generates a rectangular or a polygonal place and route boundary for the made cell. ■ If the selected PR boundary shape is *Rectangle*, layout editor automatically calculates the PR boundary dimensions and creates the boundary. ■ If the selected PR boundary shape is *Polygon*, the layout editor opens the new made cell in the same window with the *Create – P&R Objects – P&R Boundary* (*Polygon*) command selected. You can manually draw the PR boundary on the canvas and the design automatically returns to the top level when you finish creating the polygon. Environment variable: makeCellPRBoundary
Cluster	Connects or places the selected objects in or with a cluster or instance/cluster boundary. The cellview cannot already contain a cluster or instance/cluster boundary with the same name as the one you have selected. The association between the cluster and the instances/cluster boundary is retained.

© 2023 Cadence Design Systems, Inc.

Field	Description
Type	Specifies a read-only field that displays what type of cluster you have selected in the Cluster Name cyclic Field inclusive, exclusive, or suggested.

Related Topics

Making Cells

Transparent Layout Hierarchy Generation

leHiMakeCell

Cell Types

© 2023 Cadence Design Systems, Inc.

Make Readonly/Editable Form

Use the Make Readonly/Editable form to switch to the edit mode for updating the hierarchy.It lists master cellviews of all the instances existing in the current cellview.

The left pane of the form displays a tree-like list of the master cellviews whose instances exist in the current cellview. It displays the library/cell/view information of these master cellviews. The last column in the pane displays the mode of the listed master cellviews.

Note: The *Make Read only/Editable* form does not list the masters of the Pcell instances in the cellview.

Field	Description
Make Read Only	Changes the mode of the selected master cellview to read-only.
Make Editable	Changes the mode of the selected master cellview to editable.

© 2023 Cadence Design Systems, Inc.

Field	Description
Filter section	Specifies a criteria for listing only specific master cellviews in the left pane. ■ *Use:* Searches for master cellviews of all or of the instances selected in the cellview. ❑ *masters:* Displays in the left pane master cellviews of all the instances in the cellview. ❑ *masters in the selected set:* Displays in the left pane master cellviews of only the instances selected in the cellview. ■ *Name:* Specifies a name pattern to use while searching for the master cellviews. ❑ *condition/logic cyclic field == or !=*: Lets you search for character strings or numerals that are an exact match (==) or are not a match (!=); to get matches that contain a string, a leading and/or trailing asterisk must be used. ❑ *value field:* Sets the value of the name of the master cellview to use in the search. ■ *Type:* Specifies the type of the master cellview to use in the search. ❑ *condition/logic cyclic field == or !=*: Lets you search for character strings or numerals that are an exact match (==) or are not a match (!=); to get matches that contain a string, a leading and/or trailing asterisk must be used. ❑ *cyclic field:* Lets you search for specific type of master cellview, such as a *blockRing* or a *softMacro*. You can also select to search based on *all* types of master cellviews or based on *none* of the types. ■ *Filter*: Searches for master cellviews matching the criteria specified in the *Filter* section and displays the result in the left pane.

© 2023 Cadence Design Systems, Inc.

Related Topics

Making Hierarchy Editable

© 2023 Cadence Design Systems, Inc.

MarkNet Options Form

Use the Mark Net command lets you visually trace the physical connectivity of a net(s) in a layout design without having to use a schematic. The command extracts the metal, via, poly and diffusion layer information from the technology file and highlights these in the layout as the net passes from one layer to the next through the hierarchy.

The form comprises the following tabs:

Tab	Description
Options Tab	Provides options to specify hierarchy, region and highlight color.
Via Layers Tab	Defines via layers in the (*layer1 viaLayer layer2*) format, where each component layer can be specified as an LPP. If the *Customized Via Layers* check box on the Options Tab is not selected, the *Via Layers* tab is disabled and the via layers defined in the technology file are used.
Connected Layers Tab	Defines connected layers in the (layer1 purpose) (layer2 purpose) format. The connected layers map to the equivalent layers defined in the technology file. If the Customized Connected Layers check box is not selected on the Options Tab, the Connected Layers tab is disabled and connected layers are not used for tracing.
Stop Layers Tab	Defines the layers on which the *Mark Net* command must stop tracing a net. The stop criterion is specified in the *layer stopLayer* format, where both layers can be specified as LPPs. The *Mark Net* command stops tracing the connected shapes on the *layer* LPP if it finds an overlapping shape on the *stopLayer* LPP. If the *Enable Stop Layers* check box is not selected on the Options tab, the *Stop Layers* tab is disabled and the *Mark Net* command highlights all the connected shapes.

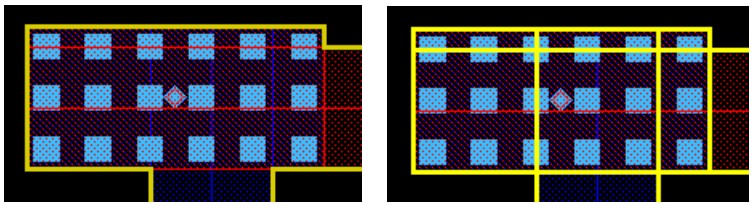
© 2023 Cadence Design Systems, Inc.

Options Tab

The following table describes the fields available on the *Options* tab of the Mark Net Options form:

Field	Description
Hierarchy Range	Sets how many levels of the hierarchy are highlighted when a net is selected. ■ *Top to bottom:* Highlights the selected net through the entire hierarchy of the window cellview. ■ *Current to bottom:* Highlights the net from the current edit cellview level to the bottom of the hierarchy. ■ *Top to current:* Highlights the net from the window cellview to the current edit cellview level. ■ *Top to stop level:* Highlights the net from the window cellview to the stop level specified in the *Display Levels* fields in the Display Options Form. ■ *Current to stop level:* Highlights the net from the current edit cellview level to the stop level specified in the *Display Levels Stop* field in the Display Options form. ■ *User defined:* Highlights the net through the user defined levels of hierarchy starting from the window cellview. This option makes the *Start* and *Stop* fields editable. Environment variable: markNetRange
Start	Sets the hierarchy level at which the highlighting starts. This field is editable only when *Mark Net Hierarchy Range* is set to *User defined*. Environment variable: markNetStartLevel
Stop	Sets the level at which the highlighting stops. This field is editable only when *Mark Net Hierarchy Range* is set to *User defined*. Environment variable: markNetStopLevel

© 2023 Cadence Design Systems, Inc.

Field	Description
Region	■ *Entire CellView:* Traces nets in the entire cellview. This option is selected by default. ■ *Viewing Area:* Traces nets in the portion of the cellview that is displayed in the canvas at that instant. ■ *Draw a Rectangle Region on the Canvas or Draw a Polygon Region on the Canvas:* Enables you to specify a rectangular or polygonal area for tracing nets. Select the required option, and then click in the canvas to specify the vertices of the rectangle or polygon.
Auto Highlight Color	Marks up to ten nets with a different color by automatically cycling through Cadence-defined layer purpose pairs y0 drawing through y9 drawing. If the appearance of these layer purpose pairs is not what you want, you can change them using the *Display Resource Editor*. Environment variable: markNetAutoColorCycle
Highlight Color	Enables you to manually change the color used to highlight the nets. You can select one of the Cadence-defined layer purpose pair y0 drawing through y9 drawing. If the appearance of these layer purpose pairs is not what you want, you can change them using the *Display Resource Editor*.
Thick Line	Enables you to display the traced nets in bold. Environment variable: markNetThickline
Highlight Incrementally	Displays the traced connectivity in a progressive manner. When this check box is selected, the connectivity is highlighted after processing every 10,000 shapes.
Highlight Individual Shapes	Adds a separate highlight for each shape.  With merged highlights With separate highlights

© 2023 Cadence Design Systems, Inc.

Field	Description
Retain Via Info for Save MarkNets	Enables you to save via shapes when you later run the *Connectivity – Nets – Save All Mark Nets* command. The via shapes are retained in the cellview created for storing the traced shapes. This option is not selected by default. Therefore, by default, the via shapes are not included in the separate cellview generated by the *Save All Mark Nets* command. The “Number of shapes processed by Mark Net”, as reported in the CIW, is not impacted by this check box setting. Environment variable: markNetSaveViaInfo
Select From Overlaps	Enables you to select the LPP from where the *Mark Net* command starts tracing the net when multiple overlapping shapes are selected. If this check box is selected, clicking overlapping shapes opens the *Choose LPP to Mark* form in which you can select the required LPP. If the check box is not selected, the *Mark Net* command starts tracing the net from any one of the overlapping shapes. Environment variable: markNetSelectOverlapLayer
Customized Connected Layers, Customized Via Layers, and Enable Stop Layers	Enables the settings defined on the Via Layers, Connected Layers, and Stop Layers tabs to be used for tracing a net. These check boxes are selected by default. If the *Customized Via Layers* check box is not selected, the via layers defined in the technology are used. If the *Customized Connected Layers* or the *Enable Stop Layers* check box is not selected, the connected layers or stop layers are not used. Environment variables: markNetViaLayer, markNetConnectedLayer, markNetStopLayer
Exclude Purposes	Enables you to exclude the specified purposes from tracing. The default purpose values that are listed are *annotate*, *boundary*, *error*, *flight*, *label*, and *warning*. You can delete any of the default purpose values. This check box is selected by default. Environment variable: markNetExcludePurposes

© 2023 Cadence Design Systems, Inc.

Field	Description
Save/Load custom settings for Via, Connected and Stop Layers	■ *Browse:* Opens the UNIX Browser window. Use this window to specify the name and location of the file to which you want to save the layer settings. You can also use this window to locate an existing file containing customized layer settings. The path and file name that you specify are displayed in the *Browse* field. If a markNetOptions file has been specified using the markNetOptionsFile environment variable, the settings for the via, connected, and stop layers are read from this file at the start of a new Virtuoso session. ■ *Save To:* Saves the layer settings to the file specified in the *Browse* field. ■ *Load From:* Populates the *Via Layers*, *Connected Layers*, and *Stop Layers* tabs with the layer settings stored in the file specified in the *Browse* field. ■ *Via Layers:* Enables you to save or load via layer settings to or from the file specified in the *Browse* field. ■ *Connected Layers:* Enables you to save or load connected layer settings to or from the file specified in the *Browse* field. ■ *Stop Layers:* Enables you to save or load via layer settings to or from the file specified in the *Browse* field. ■ *Exclude Purposes:* Enables you to save or load exclude purposes settings to or from the file specified in the *Browse* field. By default, settings for all the three layers — via, connected, and stop — are saved and loaded to or from the file specified in the *Browse* field.

© 2023 Cadence Design Systems, Inc.

Via Layers Tab

The following table describes the fields available on the *Via Layers* tab of the Mark Net Options form:

Field	Description
Via Layers list box	Displays all via layers — those defined by you and ones that have been picked from the technology file. ■ *Enable:* Lets you select the via layers that you want to use for tracing a net. To select all the via layers, select the check box at the top of the column. To select individual via layers, select the check box against each required via layer. ■ *Delete:* Lets you delete via layers that are invalid or not required. To delete all the via layers, select the check box at the top of the column and click *Delete*. To delete individual via layers, select the check box against a via layer and click *Delete*.  ***You cannot undo the deletion.***
Add Via Layers	Lets you define the via layers in the (*layer1 viaLayer layer2*) format. For each layer, specify the layer name in the *layer* column and the purpose name in the *purpose* column. It is optional to specify the purpose name.
Add	Displays the newly added via layer in the (*layer1 viaLayer layer2*) format in the *Via Layers* list box.
Reset to TechDB Defaults	Lets you use the via layers defined in the technology file. Clicking this button populates the list box with the via layers from the technology file and deletes any via layers that you may have added.

© 2023 Cadence Design Systems, Inc.

Connected Layers Tab

The following table describes the fields available on the *Connected Layers* tab of the Mark Net Options form:

Field	Description
Connected Layers list box	Displays all connected layers — those defined by you and ones that have been picked from the technology file. ■ *Enable:* Lets you select the connected layers that you want to use for tracing a net. To select all the connected layers, select the check box at the top of the column. To select individual connected layers, select the check box against each required connected layer. ■ *Delete:* Lets you delete connected layers that are invalid or not required. To delete all the connected layers, select the check box at the top of the column and click *Delete*. To delete individual connected layers, select the check box against a connected layer and click *Delete*.  *Caution* ***You cannot undo the deletion.***
Add More Entries	Displays a text field in the *Connected Layers* list box. In this field, you specify the values in the *(layer1 purpose) (layer2 purpose)* format.
Reset to TechDB Defaults	Lets you use the connected layers defined in the technology file. Clicking this button populates the list box with the connected layers from the technology file and deletes any connected layers that you may have added.

© 2023 Cadence Design Systems, Inc.

Stop Layers Tab

The following table describes the fields available on the *Stop Layers* tab of the Mark Net Options form:

Field	Description
layer [purpose] stopLayer [purpose] list box	Displays the stop layers that you have defined.
layer [purpose] [color] [lock] stopLayer [purpose] list box	Displays the stop layers that you have defined. ■ *Enable:* Lets you select the stop layers that you want to use for tracing a net. To select all the stop layers, select the check box at the top of the column. To select individual stop layers, select the check box against each required stop layer. ■ *Delete:* Lets you delete a stop layer that is invalid or not required. To delete all the stop layers, select the check box at the top of the column and click *Delete*. To delete individual stop layers, select the check box against a stop layer and click *Delete*.  ***You cannot undo the deletion.***
Add Stop Layers	Enables you to define the stop layers in the (*layer stopLayer*) format. For each layer, specify the layer name in the *layer* column and the purpose name in the *purpose* column. The purpose name is optional. Appropriate messages are generated if you try to add duplicate entries for a stop layer or specify the same value for *layer* and *stopLayer*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Add Stop Layers	Enables you to define the stop layers in the (*layer stopLayer*) format. For each layer, specify the layer name in the *layer* column, the purpose name in the *purpose* column, select the color (‘*any*’, ‘*gray*’, ‘*mask1*’ or ‘*mask2*’ ....’*maskn*’) in the *color* column, select the lock state (‘*any*’, ‘*yes*’, or ‘*no*’) in the *lock* column. The purpose name is optional. Appropriate messages are generated if you try to add duplicate entries for a stop layer or specify the same value for *layer* and *stopLayer*. **Note:** To stop tracing a net at a particular via layer, Cadence recommends that you remove or disable that via layer entry on the Via Layers tab, instead of defining it as a stop layer on the *Stop Layers* tab. Having the entry only on one of the tabs prevents ambiguity and undesirable results.
Add	Displays the newly added stop layer in the *layer [purpose] stopLayer [purpose]* list box.

Related Topics

Net Tracing Using the Mark Net Command

Tracing Nets

© 2023 Cadence Design Systems, Inc.

Marker Text Window

Use the marker text window to view information about markers.

Field	Description
Location	Displays the library, cell, and cellview names.
Reason	Displays the reason associated with a marker. This information relates to the specific technology file rules that have been broken or connectivity errors.

Related Topics

Displaying Marker Information

© 2023 Cadence Design Systems, Inc.

Modify Corner Form

Use the Modify Corner form to modify corners of rectangles and polygons in the specified window.

Field	Description
Type of Corner	Controls whether you create a rounded corner (radial) or a 45-degree edge (chamfer). If you choose to create a chamfer corner, the *Radius* field changes to *Distance*.
Radius	Sets the radius of the rounded corner, in user units. This field is available if you set the *Type of Corner* to *radial*.
Distance	Sets the distance in user units from the vertex to begin beveling. This field is available if you set the *Type of Corner* to *chamfer*.
Number of Sides	Sets the number of straight segments used to create a rounded corner. This setting is used only if *Type of Corner* is set to *radial*.
Snap to manufacturing grid	Snaps the geometry to the manufacturing grid.
Optimize sides for smooth curve	Optimizes the number of sides to get a smooth curve for the corner. This option is enabled only if the *Snap to manufacturing grid* option is selected.

Related Topics

Modifying Corners of a Polygon

leHiModifyCorner

© 2023 Cadence Design Systems, Inc.

Modify Property Form

Use the Modify Property form to change a user-created property. You can modify and delete only those properties that were added using the Add Property form.

Field	Description
Name	Displays the name of the property.
Type	Controls the value type of the property. ■ *String* (any text, displayed in a cyclic field) ■ *Int* (integer) ■ *Float* (floating-point number) ■ *Time* (date and time) ■ *Boolean* (on or off) ■ *ILList* (list of IL values)
Value	Sets the default value for this property. This line can show a text entry field, a Boolean button, or a cyclic field, depending on the *Type* setting.
Choices	Sets the possible values that appear in the cyclic field for a string property. One of the values listed here must match the default shown in *Value*.
Minimum and Maximum	Specifies the minimum and maximum values for integer, floating-point number, and time properties. These ranges appear in parentheses after the property name in the Edit Properties form and the Edit Cellview Properties form.

Related Topics

Modifying a Property

Edit Cut Pattern Form

Edit Properties Form

© 2023 Cadence Design Systems, Inc.

Move Form

Use the Move form to move an object in the specified window to another location or layer in the same window or in a different window.

When constraint-aware editing is on, the *Move* command honors the following physical constraints: *Alignment*, *Fixed*, *Locked*, *Matched Parameters*, *Orientation*, *Matched Orientation*, and *Symmetry*.

Field	Description
Snap Mode	Controls the direction in which you can move the object. Environment variable: snapMode
Change To Layer	Lets you move an object to another layer. You can search for specific layer-purpose pair using the search filter in the *Change To Layer* list box.
Treat as absolute point	Specifies the points in Delta X,Y fields will be considered as absolute points.
Delta X, Y	Lets you specify the number of user units to move the point of origin of an object on the X and/or Y axes.
Apply	Moves the origin point of the selected object as specified in the Delta X and Delta Y fields.
Display Draglines	Controls whether draglines are displayed during the *Move and Stretch* commands. By default, the draglines indicate connections from the pins of the object you are moving to pins of the nearest objects. Environment variable: flightLineEnable
Connections	■ *Chain Mode:* Enables you to move device chains. ■ *All:* Moves all instances in a chain. ■ *Selected:* Moves only selected instances. ■ *Selected Plus Left:* Moves the instance selected and all instances to the left of the selected instance in the chain. ■ *Selected Plus Right:* Moves the instance selected and all instances to the right of the selected instance in the chain.

© 2023 Cadence Design Systems, Inc.

Field	Description
Orientation	Click the buttons at the bottom to change the orientation of the selected instance. ■ *Rotate:* Rotates the component 90 degrees counterclockwise and clockwise. ■ *Sideways:* Mirrors the component on the y axis (flips it horizontally). ■ *Upside Down:* Mirrors the component on the x axis (flips it vertically). In the Virtuoso Layout Suite MXL Only, the Move form includes an additional *Wire Bond* section. The options in this section let you control the move behavior of bond wires and bond fingers. For more information, see Moving Bond Wires and Bond Fingers.
Snap	■ *Pins and boundary to grid:* Snaps the moved object to a grid depending on the block type. If it is a digital block, the place and route boundary is snapped to the placement grid and pins with placement status `placed` are snapped to the routing grid. If it is an analog block, both place and route boundary and `placed` pins are snapped to the manufacturing grid. Pins with placement status `unplaced`, `fixed`, `locked`, or `unknown` are not snapped, even if they were touching the boundary before the move. Environment variable: snapToGrid ■ *Pins to boundary:* Snaps pins to the place and route boundary automatically. ■ *Allow pin resizing:* Specifies that the pins need to be resized during pin snapping. Environment Variable: allowPinResizing The *Allow pin resizing* setting that you make in the *Move* form is applicable only for the current instance. For all other instances, the *Allow pin resizing* setting in the *F*loorplan Global Options form is considered as the default setting. Changes that you make to the setting in the Move form cannot overwrite the global setting.

© 2023 Cadence Design Systems, Inc.

Related Topics

Object Moving

leHiMove

Snapping Soft Block Pins to Grid

Constraint-Aware Editing

Default Constraint Types

© 2023 Cadence Design Systems, Inc.

Open Form

Use the Open form to specify the settings for opening a cellview.

Field	Description
View	Specifies the view of the cell to open.
Open for	Specifies whether the cellview is opened in edit or read mode.
Open in	Specifies whether the cellview is opened in a new tab, the current tab, or a new window.

Related Topics

Annotation Browser Shortcut Menus

© 2023 Cadence Design Systems, Inc.

Pad Opening Info Form

Use the Pad Opening Info form to search for bond pad shapes in a hierarchical cellview, generate reports and labels for the pad shapes that are found, and promote the generated labels to the top level of the cellview.

The form offers three modes to support the main tasks you can perform: *Generate Report*, *Generate Labels*, and *Promote Labels To Top*.

The options available depend on the task you are performing. The table below summarizes which options are available in which modes.

Options/Mode	Generate Report	Generate Labels	Promote Labels
Search Criteria	X	X	X
Sort	X	X	
Reporting	X		
Label Control		X	
Promote Label			X

Delete Bond Pad Labels is available in all modes and deletes any existing bond pad labels at the top level of the current cellview. Only bond pad labels generated by the *Pad Opening Info* command are deleted; labels generated by other tools or commands are not affected.

Search Criteria

The *Search Criteria* options let you specify the layer-purpose pair and hierarchy levels to be searched for pad shapes, and the dimensions of the pad shapes you are interested in. You

© 2023 Cadence Design Systems, Inc.

can also search for existing labels in the design and limit the search to specific regions of the cellview.

Field	Description
Bond Pad	■ *Layer/Purpose:* Specifies the layer-purpose pair to search for bond pad shapes. The default is "Bondpad drawing". You can search for specific layer-purpose pair using the search filter in the *Layer/Purpose* list box. The layer selection using the search filter is available only if the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form. Environment variable: bondPadLPP ■ *Search Depth:* Specifies the number of hierarchical levels to traverse when searching for pad shapes. For example, if you specify 5, the command searches for shapes on the specified layer-purpose pair at hierarchy levels 0 through 5. The maximum value that you can specify in this field is 32. ■ The *Dimension* fields: Lets you refine the search further by specifying the dimensions of the pads you want to label. The X dimension is required and defaults to > and 0. You can optionally specify an operator and value for the Y dimension as well. The command returns only shapes with a bounding box that meets the specified criteria. For non-rectangular (e.g., octagonal) pads, use the width of the enclosing bounding box as the pad dimension.

© 2023 Cadence Design Systems, Inc.

Field	Description
Region	■ *Specify Search Regions:* Lets you narrow the search area to specific regions of the cellview. ■ *Clear All:* Clears any search regions currently defined. ■ *Add Region:* Adds another region definition to the list. ■ *Create Rectangle:* Lets you click and drag in the layout canvas to draw a rectangular search region. The corresponding coordinates are automatically entered in the *Region* field. You can edit these coordinates manually if required. ■ *Create Polygon:* Lets you draw a polygonal search region by clicking on each point of the polygon in the layout canvas. To finish a polygon, either double-click in the canvas or click and press Return. The corresponding coordinates are automatically entered in the *Region* field. You can edit these coordinates manually if required.
Label/Name	Lets you specify the layer and hierarchy level to search for existing labels. ■ *Object Type:* Lets you select the type of object you want to search, *Label*, *Text Display*, or *Both*. ■ *Layer/Purpose:* Specifies the layer-purpose pair to search for existing labels. ■ *Search Depth:* Sets the hierarchy level on which to search for existing labels. The command returns only the labels found at the hierarchy level you specify. For example, if you set the *Bond Pad Search Depth* to 5 and the *Label/Name Search Depth* to 2, then the command shows all the pad shapes found in levels 0 through 5, but returns only the text labels found at level 2. Existing labels on other levels are not displayed.
Highlight	Draws the pad shapes found in the appropriate highlight color in the canvas.

© 2023 Cadence Design Systems, Inc.

Sort

The *Sort* options let you specify the location for the first pad label to be generated and the direction in which subsequent pads are labeled.

Field	Description
1st PAD Coords	Sets the location for the first label generated by the command. Click the button and then the required location in the canvas to enter the corresponding coordinates in the field. You can also edit the coordinates manually using the following format: `(1031.975 877.84)`.
Sort Direction	Specifies the direction to use when extracting pad locations. Choose between *Clockwise* and *Counter Clockwise*.

Reporting

The *Reporting* options are available only in *Generate Report* mode and let you specify the type of report generated. If the predefined standard report does not meet your requirements, you can customize the report as required.

Field	Description
Make Standard Report	Generates a report in standard format. Select the box and use the *Browse* button to specify the location and name for the report file.
Make Customized Report	Lets you define and generate reports that are customized to your requirements in terms of content and filtering criteria. ■ *Add:* Opens the Create Encore Report form where you define the format of your customized report. You can add as many customized reports as required. ■ *Del:* Removes the selected customized report from the list. ■ *Edit:* Opens the Create Encore Report form where you can modify the selected customized report.
File Format	Specifies whether the report is generated as plain text or in CSV format.

© 2023 Cadence Design Systems, Inc.

Label Control

The *Label Control* options are displayed only in Generate Labels mode. They let you specify what label names are generated and how they will be drawn in the canvas.

Field	Description
Read Label	■ *Label (Pattern)* specifies a naming convention to be used when generating labels. The label pattern feature supports standard bus label notation. The following are examples of label patterns and the labels generated for each pattern. For example, for label pattern `A<5:2>`, the `A5, A4, A3, A2` labels are generated. ❑ `A<5:2> - A5, A4, A3, A2` ❑ `B<0:6:2> - B0, B2, B4, B6` ❑ `A<1:2>B<1:2> - A1B1, A1B2, A2B1, A2B2` ❑ `A<1:2>C - A1C, A2C` ❑ `A[0:1]-B[0:1] - A0, B0, A1, B1` ❑ `A<<0:1>> - A<0>, A<1>` ❑ `A(<0:2>) - A(<0>), A(<1>), A(<2>)` ❑ `A([0:2]) - A(0), A(1), A(2)` ❑ `A{<0:2>} - A{<0>}, A{<1>}, A{<2>}` ❑ `A{[0:2]} - A{0}, A{1}, A{2}` ❑ `A([0:6:2])C - A(0)C, A(2)C, A(4)C, A(6)C` ■ *From File*: Reads the label names from a file. Enter the path to the file in the field provided, or use the *Browse* button to locate it on your system.

© 2023 Cadence Design Systems, Inc.

Field	Description
Label Options	■ *Font:* Specifies the font used to draw the labels on the canvas. ■ *Height:* Sets the height of the label text. ■ *Justification:* Specifies the label alignment in relation to the pad shape (for example, *lowerLeft*, *centerCenter*, *upperRight*, etc). ■ *Layer/Purpose:* Specifies the layer-purpose pair on which the label is drawn. The default is `"text drawing"`. Environment variable: padLabelLPP

Promote Label

The Promote Label options are shown only in Promote Label To Top mode. They let you specify the appearance of the promoted labels in the canvas and control whether the corresponding lower-level labels are removed from the cellview.

Field	Description
Label Options	■ *Font:* Specifies the font used to draw the labels on the canvas. ■ *Height:* Sets the height of the label text. ■ *Justification:* Specifies the label alignment in relation to the pad shape (for example, *lowerLeft*, *centerCenter*, *upperRight*, etc). ■ *Layer/Purpose:* Specifies the layer-purpose pair on which the label is drawn. The default is `"text drawing"`. Environment variable: padLabelLPP ■ *Remove Original Labels/Text-Displays:* Deletes the original labels at the lower level after they have been promoted to the top level.

Related Topics

Generating a Standard Report for Pad Shapes

© 2023 Cadence Design Systems, Inc.

Generating a Customized Report for Pad Shapes

Generating Labels for Bond Pads

Promoting Labels to the Top Level

Create Encore Report Form

Pad Opening Info: Filter Form

leHiDisplayPadOpeningInfoForm

© 2023 Cadence Design Systems, Inc.

Pad Opening Info: Filter Form

Use the Pad Opening Info: Filter form to set the filter for the customized report that you want to generate for pad shapes in a hierarchical cellview. You can access this form by using the *Set Filter* button in the Create Encore Report form.

Field	Description
Filter Options	■ Mode: Select a mode from the first list as *Pad Opening Name*, *X*, *Y*, or *Area*. You can set filter for bond pads based on their name, width, height, and area. ■ Operator: Select an operator from the second list. The valid operators for *Pad Opening Name* are *==*, *!=*, *contains*, and *doesn't contain*. For *X*, *Y*, and *Area*, the valid operators are: *==*, *!=*, *<*, *<=*, *>*, and *>=*. ■ Input: Enter the value in the input field based on which you want to filter the data. ■ Logical operator: Select an operator from the last list for logical operation with any subsequent filters defined. The valid operators for logical operation are *and* and *or*.

Related Topics

Generating a Customized Report for Pad Shapes

Setting Filters for a Customized Report

Create Encore Report Form

Pad Opening Info Form

© 2023 Cadence Design Systems, Inc.

Palette Options Form

Use the Palette Options form to set local and global options for panels in the Palette assistant.

Field	Description
Keep windows synchronized	Initializes all new windows to synchronized state. Environment Variable: pteWindowSynchronizationMode
Search Mode	Enables you to switch between *Filter* and *Find* search modes. Environment Variable: pteFindModeOn
Multi-select mode	Lets you select multiple layer-purpose pairs at one time. Only one layer—the one selected first—is set as active. Environment Variable: pteMultiSelectionMode
Auto Redraw	Sets the scope of the auto redraw feature. ■ *Off*: Turns off the auto redraw feature. Manual redraw is needed to update all windows. ■ *Current Window*: Updates in real time the current window canvas whenever a change is made in a Palette panel. ■ *All Windows*: Updates in real time all synchronized window canvases whenever a change is made in a Palette panel. Environment Variable: pteAutoRedrawMode
Layer Set Activation	Specifies how visibility and selectability status of layer-purpose pairs is affected when a layer set is made active. ■ *Filter only*: Displays all layer-purpose pairs present in the selected layer sets. The visibility and selectability status of the layer-purpose pairs remains unchanged. ■ *Filter and apply visibility, selectability:* Displays all layer-purpose pairs present in the selected layer sets. The visibility and selectability status of layer-purpose pairs is reset to that defined in the layer set files. ■ *Force visibility and selectability only*: Displays all layer-purpose pairs present in the selected layer sets. The visibility and selectability of all layer-purpose pairs is turned on. Environment Variable: pteLayerSetActivationMode

© 2023 Cadence Design Systems, Inc.

Field	Description
Scope of 'All Layers' Layer Set	Defines the scope of the layer-purpose pairs displayed in the *Layers* panel. ■ *LSW layers*: Displays layer-purpose pairs defined in the leLswLayers section of the technology file. If this section is not available, all layer-purpose pairs defined in the technology file are displayed. ■ *Techfile layers*: Displays all valid layer-purpose pairs defined in the technology file, even if the leLswLayers section is available. Environment Variable: pteAllLayersScope
Header Column Click	Defines the action to be performed when you click the Visibility (V), Selectability (S), Membership (m), and Validity (v) column headers in the Layers panel. ■ *Toggle on/off*: Turns on or off the Visibility (V), Selectability (S), Membership (m), and Validity (v) status. ■ *Sort*: Sorts layer-purpose pairs based on the Visibility (V), Selectability (S), Membership (m), and Validity (v) status. Environment Variable: pteCheckableColumnClickMode
Scope of Used Layers	Lets you specify the layers to display in the Layers panel when EIP mode is active. If different technology files are associated with cellviews at different hierarchical levels in the design, only the layers defined in the technology file associated with the cellview being edited in place are displayed in the Layers panel. ■ *Current to bottom*: Displays layers used in the design at the current hierarchy level and below. This is the default option. ■ *Top to bottom*: Displays all layers used in the design. Environment Variable: pteUsedLayersEipScope
Depth for Visibility	Determines the number of layer-purpose pairs for which the visibility attribute is affected when you hold down `Ctrl` or both `Ctrl` and `Shift` and click a layer-purpose pair with the middle mouse button in the *Layers* panel. Environment Variable: pteLayersVisibilityDepth

© 2023 Cadence Design Systems, Inc.

Field	Description
Depth for Selectability	Determines the number of layer-purpose pairs for which the selectability attribute is affected when you hold down `Ctrl` or both `Ctrl` and `Shift` and right-click a layer-purpose pair in the *Layers* panel. Environment Variable: pteLayersSelectabilityDepth
Sync visibility/ selectability by parent	Propagates changes in the visibility and selectability status of parent objects and grids to child objects and grids. Environment Variable: pteApplyParentState
Shape visibility by color visibility	Makes colored shapes invisible if you deselect the Visibility (*V*) check box corresponding to the layer-purpose pairs on which they are drawn. If deselected, only the colored outline is hidden; the shapes remain visible. Environment Variable: pteLinkColorVisibilityToShapeVisibility
Color mixed sort	Sorts all layer-purpose pairs, colored and uncolored, together and lists them in the order defined by the specified attribute. Environment Variable: pteColorMixedSort
Compact MPT mode	Hides the Color (*C*) column and displays on the *Active* (*Layer*) toolbar a set of options that you can use to control the color and lock state combination for a layer-purpose pair and also the visibility of colored shapes based on their lock state. Environment Variable: pteCompactMPT
Inverse color scheme	Inverses the color scheme of the Palette panels. Environment Variable: pteInverseColorScheme
Movable tools	Specifies whether Palette panels toolbars can be moved. If selected, three vertical lines appear to the left of the toolbar name. When you point to the vertical lines, the pointer becomes a four-headed arrow. You can drag the pointer to move a toolbar to a different location within the Palette panel to which it belongs. Environment Variable: pteMovableTools

Related Topics

Visibility and Selectability of Parent-Child Objects

© 2023 Cadence Design Systems, Inc.

Paste Form

Use the Paste form to place the contents of the yank buffer in the specified window.

Field	Description
Reference Points	Lets you select the reference point of the bounding box. You can select one of the ten reference points when pasting. The reference points are *lowerLeft*, *centerLeft*, *upperLeft*, *lowerCenter*, *centerCenter, upperCenter*, *lowerRight*, *centerRight*, *upperRight*, *origin*, *user specified*. Environment Variable: pasteRef
Rotate	Turns the object 90 degrees counterclockwise. You can also click the middle mouse button to rotate or mirror objects you paste.
Sideways	Mirrors the object along the X axis.
Upside Down	Mirrors the object along the Y axis.

Related Topics

Object Yanking and Pasting

leHiPaste

© 2023 Cadence Design Systems, Inc.

Print Hierarchy Tree Form

Use the Print Hierarchy Tree form to specify details of the cellview for which you want to print the hierarchy of cellview instances.

Field	Description
Library	Lets you select the name of the library for which you want to print the hierarchy of cellview instances.
Cell	Lets you specify the name of the cell for which you want to print the hierarchy of cellview instances. Alternatively, you can select the name from the *Cells* list box.
View	Lets you specify the name of the view for which you want to print the hierarchy of cellview instances. Alternatively, you can use the *Browse* button to specify the library, cell, and view for which you want to print the hierarchy tree.
Type	Displays the type of the view.
Library path file	Displays the path to your library definitions file. If the path is too long to be displayed, you can put the pointer in this field and use the arrow keys to display the rest of the path. You cannot edit this field.
Cells	Displays names of all the cells in the library selected in the *Library* drop-down list.

Related Topics

Viewing the Hierarchy Tree

© 2023 Cadence Design Systems, Inc.

Quick Align Form

Use the Quick Align form to select a reference edge, centerline, or point. In the post-select mode, you can select the reference edge, centerline, or point after selecting the reference set objects.

Field	Description
Copy reference	Enables you to copy an object during alignment. Environment variable: quickAlignMode The *Copy* option will only copy fully selected objects. It will not copy partially selected objects.
Snap Target	Enables you to select method to enable the pointer to the reference set and target object. ■ *Edges and Points:* Snaps the pointer to edges, centerlines, vertices, nexus, and end points in the reference set and target object. ■ *Edges:* Snaps the pointer to only the edges and centerlines in the reference set and target object. ■ *Points:* Snaps the pointer to vertices, midpoint of edges, nexus, end points, and origin point of objects in reference set and target. Environment variable: quickAlignSnapTarget

© 2023 Cadence Design Systems, Inc.

Field	Description
Offset	■ *No Spacing:* Does not apply any spacing between the reference edge, centerline, or point and the target edge, centerline, or point. Environment variable: quickAlignSpacingType ■ *Min Spacing:* Applies the minimum spacing value of the reference object layer, as specified in the technology file, while aligning the reference set to the target edge, centerline, or point. Environment variable: quickAlignSpacingType ■ *User Spacing:* Applies the spacing you specify to the distance between the reference edge, centerline, or point and the target edge, centerline, or point. You can specify an integer. Environment variable: quickAlignSpacingType, quickAlignUserSpacing
Snap Target Visualization Settings	Customizes the appearance of the edge and point highlights that appear on the potential reference and target edges and points. ■ *Width:* Enables you to specify the width of the highlight. The default is 0. Environment variable: targetWidth ■ *Color:* Enables you to select the color of the highlight. The default is yellow. Environment variable: targetColor ■ *Line Style:* Enables you to select the type of highlight. The default is solid. Environment variable: targetLineStyle

Related Topics

Object Alignment by Using the Quick Align Command

Quick Align Shortcut Menu

© 2023 Cadence Design Systems, Inc.

leHiQuickAlign

© 2023 Cadence Design Systems, Inc.

Quick Align Shortcut Menu

The following table displays the options available on the Quick Align shortcut menu.

Field	Description
Target Axis	Sets the *Vertical* or *Horizontal* axis for aligning the reference set of objects. By default, the target axis is created at the point of click in the design.
Spacing	Enables you to specify the spacing between the reference set and target object after alignment. The options available on the *Spacing* menu map to the *No Spacing*, *Min Spacing*, and *User Spacing* options on the Quick Align form.
Switch to copy mode or Switch to move/stretch mode	Enables you to switch between *Copy* and *Move/Stretch* mode of alignment. If the *Move/Stretch* mode is selected in the Quick Align form, then the *Switch to copy mode* option is available in this menu. However, if the *Copy* mode is selected in the Quick Align form, then the *Switch to move/stretch mode* option is available in this menu.
Snap to targets in reference layer	Enables you to snap the pointer to only those targets that are on the same layer as the object with the reference point, edge, or centerline. If selected, this option on the shortcut menu toggles to *Snap to targets on all layers*. ■ The *Snap to targets in reference layer* option maps to the selected *On Same Layer* check box on the Quick Align form. ■ The *Snap to targets on all layers* option maps to the unselected *On Same Layer* check box.
Snap to Edges and Points	Snaps the pointer to edges, centerlines, vertices, nexus, and end points in the reference set and target object.
Snap to Edges	Snaps the pointer to only the edges and centerlines in the reference set and target object.
Snap to Points	Snaps the pointer to vertices, midpoint of edges, nexus, end points, and origin point of objects in reference set and target.
Options	Opens the Quick Align form.

© 2023 Cadence Design Systems, Inc.

Field	Description
De(Select) Under Cursor submenu	Provides a list of all the objects currently selected and enables you to select the one you want to deselect. This command is available on the *Quick Align* context-sensitive menu only when you right-click over an object.

Related Topics

Quick Align Form

Object Alignment by Using the Quick Align Command

leHiQuickAlign

© 2023 Cadence Design Systems, Inc.

Refresh Cellviews Form

Use the Refresh Cellviews form to edit a master cellview that is instantiated in the cellview you currently have open.

Field	Description
Refresh	Shows the library, cell, and view name of cells that have been edited since you opened this cellview. You can choose to refresh, in memory, your copies of any of the listed master cells.
All or None	Lets you choose to update memory for all or none of the listed cellviews.

Related Topics

Refreshing Cellviews to Reflect Updated Masters

© 2023 Cadence Design Systems, Inc.

Remaster Instances Form

Use the Remaster Instances form to specify instances to search for based on the view name, cell name, and the library name for a given cellviewID, then re-masters those instances with the master that matches the update library, update cell and update view names you provide in the form.

Field	Description
Search for	■ *Library:* Specifies the name of the library that contains the cellviews you want to re-master. Environment variable: remasterInstSearchLibText ■ *Cell Name:* Specifies the name of the cellview that contains the cellviews you want to re-master. Environment variable: remasterInstUpdateCellText ■ *View Name:* Specifies the name of the view to be re-mastered. Environment variable: remasterInstSearchViewText
Update to	■ *Library:* Specifies the name of the library that contains the replacement master cellviews. Environment variable: remasterInstUpdateLibText ■ *Cell Name:* Specifies the name of the cellview that contains the replacement master cellviews. Environment variable: remasterInstUpdateCellText ■ *View Name:* Specifies the name of the view to replace the view in the *Search* for *View Name* field. Environment variable: remasterInstUpdateViewText
Check Terminals	Checks all corresponding cellviews for matching or a subset of matching instTerm information. Environment variable: remasterInstCheckTerms

Related Topics

Instance Remastering

© 2023 Cadence Design Systems, Inc.

leHiRemasterInstances

© 2023 Cadence Design Systems, Inc.

Remove Shape From Net Form

Use the Remove Shape From Net form to remove the selected shape from the net.

Field	Description
Net Name	Shows the net of the selected shape. If more than one shape is selected, Net Name shows the net of the first shape. If the net name shown is not the net you want the shapes deleted from, type the correct net name.

Related Topics

Removing Shapes from a Net

leHiDeleteShapeFromNet

© 2023 Cadence Design Systems, Inc.

Rename Command Name Form

In Layout EXL or a higher tier, use the Rename Command Name form to change the name, description, or group category of any recording.

Field	Description
Command Name	Specifies the name of the recording.
Description	Specifies information about the recording. The tooltip for a saved recording displays a brief description in the Capture and Relay assistant.
Group	Specifies the group under which you want to save your recording for easy access. When you rename a recording, you can specify a new group category or you can choose a group from the existing group categories for the recording. You can also change the group name or description of any recording without changing the command name.

Related Topics

Capture and Replay Assistant

Recording Your Actions in a Design

Renaming a Recording in the Capture and Replay Assistant

© 2023 Cadence Design Systems, Inc.

Repeat Copy Form

Use the Repeat Copy form to repeatedly place copies of the selected object.

Field	Description
Release Direction	Resets the paste direction and allows you to paste the copied object in any direction and at any distance with respect to the previous pasted object. When off, the direction of pasting the object is fixed, which is the direction in which the copied object was last pasted. The check box is available only after you have placed the first copy
Snap Mode	Controls how the mouse pointer snaps to the grid as you place the copies. ■ *anyAngle:* Enables you to move the pointer at any angle. ■ *diagonal:* Enables you to move the pointer parallel to the X or Y axis or at a 45-degree angle to the axes. ■ *orthogonal:* Enables you to move the pointer parallel to the X and Y axis. ■ *horizontal*: Enables you to move the pointer parallel to the X axis. ■ *vertical:* Enables you to move the pointer parallel to the Y axis.
Delta	■ *X and Y:* Enables you to specify the exact distance from the object at which you want to place the copy. ■ *Apply XY:* Places the copy at the specified distance from the object.
Copy Connectivity	Copies the connectivity information on path, pathseg, fluid guard ring, and via to the new object. Environment variable: copyConnectivity

© 2023 Cadence Design Systems, Inc.

Field	Description
Snap To Grid	Snaps the pin edges to the manufacturing or the routing grid, depending on the selected block type. ■ If the block type is custom, the pin edges are snapped to the manufacturing grid. ■ If the block type is digital, the pin edges are snapped to the routing grid.
Create Synchronous Copy	Creates a synchronized copy of the selected layout objects.
Rotate	Turns the object 90 degrees counterclockwise each time it is pressed.
Sideways	Mirrors the object along the X axis.
Upside Down	Mirrors the object along the Y axis.

Related Topics

Copying an Object in Repeat Mode

leHiRepeatCopy

Generating a Synchronous Copy

© 2023 Cadence Design Systems, Inc.

Reshape Form

Use the Reshape form to change the shape of a selected object.

Field	Description
Mode	Sets the geometry to use for reshaping a selected object. ■ *rectangle:* Lets you add or remove a rectangle shape. ■ *line:* Lets you add a polygon to a shape or reshape a path. Only line can be used to reshape a path.
Snap Mode	Controls the shape of line segments; applies only when *Mode* is set to *line*. Environment variable: snapMode
Snap	■ *Pins and boundary to grid:* Snaps the stretched object to a grid depending on the block type. If it is an *analog* block, the place and route boundary and pin edges are snapped to the manufacturing grid. If it is a *digital* block, the place and route boundary is snapped to the placement grid and the pin centers are snapped to the routing grid. Pins with placement status unplaced, fixed, locked, or unknown are not snapped, even if they were touching the boundary before the reshape. Environment variable: snapToGrid During the stretch operation, in case the soft block is stretched outside the top-level PR boundary, then all the instance pins on the soft block, with the placement status as *placed*, are snapped to the manufacturing grid. This is irrespective of the block type because the routing grid is not initialized outside the PR boundary but the manufacturing grid exists outside the PR boundary. ■ *Pins to boundary:* Snaps pins to the place and route boundary automatically. ■ *Snap Blocks Pins to:* Snaps soft block pins automatically either to the *Modified Edge* or to the *Closest Edge* after the reshape is complete.

© 2023 Cadence Design Systems, Inc.

Related Topics

Object Reshaping

Soft Block Editing

Reshaping an Object in Level-1 Editing Mode

© 2023 Cadence Design Systems, Inc.

Restore View Form

Use the Restore View form to choose the view that you want to restore. The list box displays the last three saved views.

Related Topics

Saving and Restoring a View

© 2023 Cadence Design Systems, Inc.

Return To Level Form

The text box lists the levels above the current one in the hierarchy and the cell associated with each level.

Related Topics

Returning to a Different Hierarchy Level

© 2023 Cadence Design Systems, Inc.

ROD Subpart Form

Use the ROD Subpart form to define, change, or delete subparts for the associated master path. You can access this form from the Create Multipart Path form.

The following table describes the sections and common options available in the ROD Subpart form.

Field	Description
Offset Subpath Fields	Lets you create a new offset sub-path. It lets you view one line of data for each existing offset sub-path in the scroll window. When you select the *Offset Subpath* check box, the form displays the fields for offset subpaths.
Enclosure Subpath Fields	Lets you create a new enclosure sub-path. When you select the *Enclosure Subpath* check box, the form displays the fields for enclosure subpaths.
Subrectangle Fields	Lets you create a new set of sub rectangles. When you select the *Subrectangle* check box, the form displays the fields for subrectangles.
Scroll window	Shows the list of offset, enclosure, or sub-rectangle sub-paths that already exist for the MPP that you are creating, depending on whether you select the *Offset Subpath*, *Enclosure Subpath*, or *Subrectangle* check box. There is one line of data for each sub-path of the type selected. Each data element in a line corresponds to a field in the lower part of the ROD subpart form. When you click an existing subpart in the scroll window, the system fills the fields in the rest of the ROD Subpart form with the values for that subpart. You can add a new subpart, delete an existing subpart, or change the values for an existing subpart.
Add	Lets you create a new subpart from the values currently in the ROD Subpart form.
Delete	Lets you delete the highlighted subpart.
Edit	Lets you change the values for an existing subpart. Highlight the subpart that you want to change, specify the updated values in the ROD Subpart form, and then click the *Edit* button. The system replaces the field values of the highlighted subpart with the values currently in the ROD Subpart form.

© 2023 Cadence Design Systems, Inc.

Offset Subpath Fields

The following table describes the fields available in the *Offset Subpath* section of the ROD Subpart form.

Field	Description
Scroll window	Displays a list of offset subpaths that already exist for the current MPP. There is one line of data for each offset subpath. Each data element in a line corresponds to a field in the ROD subpart form. The data for an offset subpath is displayed in the following sequence: **1.** Layer **2.** Width **3.** Choppable **4.** Separation **5.** Justification **6.** Begin Offset **7.** End Offset **8.** List of connectivity elements, if any.

© 2023 Cadence Design Systems, Inc.

Field	Description
	When a subpart has connectivity, the connectivity data is listed in the following sequence. **1.** Net Name **2.** I/O Type **3.** Connectivity = Pin **4.** Access Direction **5.** Display Pin Name **6.** Height **9.** Layer **7.** Justification **8.** Font **9.** Text Options **10.** Orientation **11.** Reference Handle **12.** Offset X:Offset Y When Display Pin Name is equal to t, the following elements correspond to fields on the Pin Name Display form: *Height*, *Layer*, *Justification*, *Font*, *Text Options*, and *Orientation*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Layer	Lets you choose the layer-purpose pair for the subpart. The default purpose is `drawing`. Choosing a layer-purpose pair for a subpart does not change the current layer in the *Layers* assistant. If the layer-purpose pair you choose is not valid in the *Layers* assistant, the system displays a dialog box asking if you want to add the layer-purpose pair to the *Layers* assistant. If you agree, the system adds the layer-purpose pair, but does not change the current layer-purpose pair. If do not want to add the layer-purpose pair to the *Layers* assistant, choose a layer-purpose pair that is already in the Layers assistant or the system will use the current layer-purpose pair. The layer selection using the search filter is available only if the *Use Layer Palette LPPs Only* option is selected in the Layout Editor Options form.
Choppable	Indicates whether or not the subpart can be chopped. The default is choppable. When the master path is choppable, all of its subparts must be choppable. When the master path is not choppable, each of its subparts can be choppable or not.
Begin Offset	Specifies the offset of the starting edge of the subpath from the starting edge of the master path.The value must be a signed integer or floating-point number. A positive number extends the beginning of the subpath beyond the beginning of the master path; a negative number retracts the beginning of the subpath from the beginning of the master path. The default is *End Offset*, if specified; otherwise `0`.
End Offset	Specifies the offset of the ending edge of the subpath from the ending edge of the master path.The value must be a signed integer or floating-point number. A positive number extends the end of the subpath beyond the end of the master path; a negative number retracts the end of the subpath from the end of the master path. The default is *Begin Offset*, if specified; otherwise `0`.

© 2023 Cadence Design Systems, Inc.

Field	Description
Width	Specifies the width of the offset subpath. The value must be a positive integer or floating-point number. If you do not specify the width, the system uses the `minWidth` rule for the offset subpath layer from the technology file. If the `minWidth` rule is not defined in the technology file, the system uses the width of the master path. If the `minWidth` rule is not defined in the technology file, and ■ This is the first time you have used this form in your current editing session and you do not specify a width, the system supplies a default and displays an warning saying that the default width is being used. ■ You have used this form earlier in your current editing session, the previous value for this field is displayed and the system displays a warning.
Justification	Specifies from which part of the master path to separate the subpath, in relation to the direction of the master path. When the value is *left*, the right edge of the subpath is separated from the left edge of the master path. When the value is *right*, the left edge of the subpath is separated from the right edge of the master path. When the value is *center*, the centerline of the subpath is separated from the centerline of the master path. The default is *center*.
Separation	Specifies the distance between the centerline or an edge of the subpath and the centerline or an edge of the master path, depending on the value of the *Justification* field. The value must be a signed integer or floating-point number; the default is `0`.

© 2023 Cadence Design Systems, Inc.

Field	Description
Connectivity	Determines whether to associate the subpart with a net and whether to make it into a pin. The default is *None*. When you choose *Net*, the ROD Subpart form expands to show the net connectivity fields. When you choose *Pin*, the form expands to show both the net and pin connectivity fields. ■ *None*: Specifies that you do not want to define connectivity. ■ *Net*: Specifies that you want to associate the subpart with a specific net. Type a value in the Net Name field. ■ *Pin*: Specifies that you want the subpart to be a pin. To define a pin, you must also specify a value for the Net Name field.

Enclosure Subpath Fields

The following table describes the fields available in the *Enclosure Subpath* section of the Rod Subpart form.

Field	Description
Scroll window	Lists the enclosure subpaths that already exist for the current MPP. There is one line of data for each enclosure subpath. Each data element in the line corresponds to a field on the ROD subpart form.
Enclosure	Specifies the enclosure of the subpath in relation to the master path. To calculate the width of an enclosure subpath, the system subtracts two times the enclosure value from the width of the master path: `Width of Enclosure Subpath = Width of Master Path - (2 * Enclosure Value)` The value must be a signed integer or floating-point number. If you do not specify this field, the system uses the minExtensionDistance rule from the technology file for the master path layer to the subpath layer. If the `minExtensionDistance` rule is not defined in the technology file, the system sets the value to zero. Connectivity is the same as for offset subpaths.

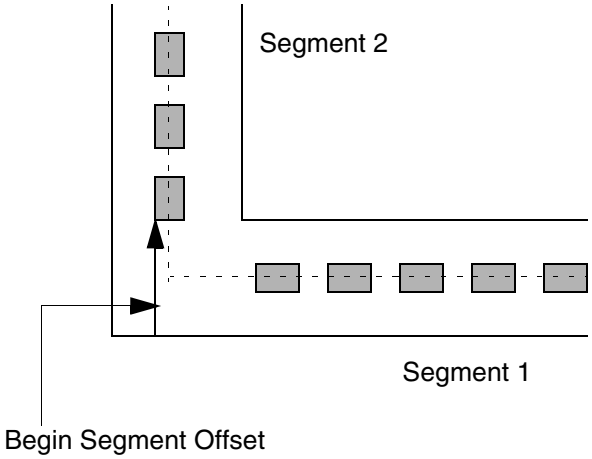
© 2023 Cadence Design Systems, Inc.

Subrectangle Fields

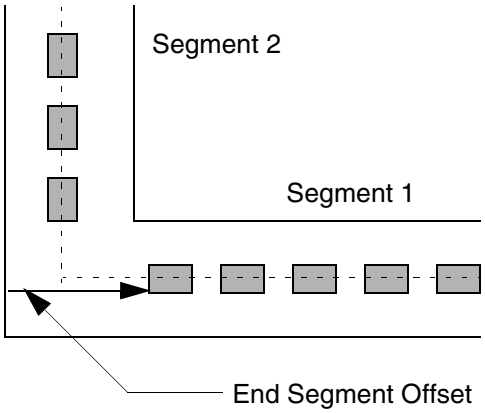
The following table describes the fields available in the *Subrectangle* section of the ROD Subpart form.

Field	Description
Scroll Window	Lists the sets of subrectangles that already exist for the current MPP. There is one line of data for each set of sub-rectangles. Each data element in the line corresponds to a field in the lower part of the ROD subpart form.
Begin Offset	Specifies the offset of the starting edge of the first sub rectangle from the starting edge of the master path.The value must be a signed integer or floating-point number. A positive number extends the beginning of the sub rectangles beyond the beginning of the master path; a negative number retracts the beginning of the sub-rectangles from the beginning of the master path. The default is *End Offset*, if specified; otherwise `0`.
End Offset	Specifies the offset of the ending edge of the last sub rectangle from the ending edge of the master path.The value must be a signed integer or floating-point number. A positive number extends the end of the sub-rectangles beyond the end of the master path; a negative number retracts the end of the sub-rectangles from the end of the master path. The default is *Begin Offset*, if specified; otherwise `0`.

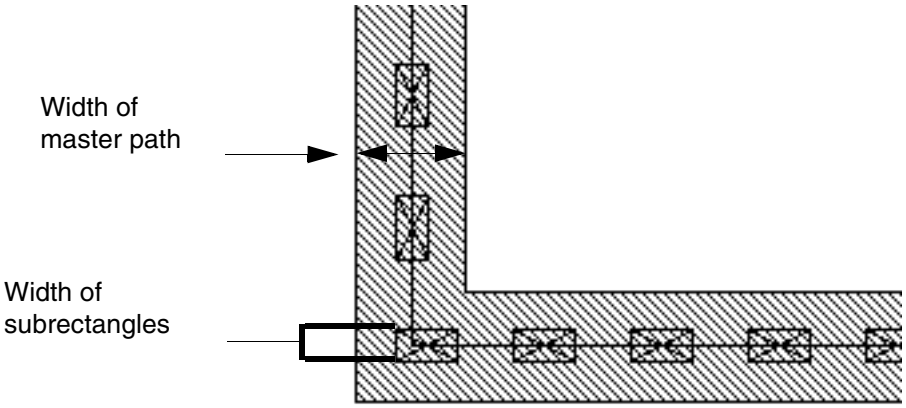
© 2023 Cadence Design Systems, Inc.

Field	Description
Begin Segment Offset	Specifies the offset of the first rectangle in a segment from the edge of the master path.The value must be a positive integer or floating-point number. Does not apply to the first segment of a sub rectangle sub path; the position of the first rectangle in the first segment is determined by the value of the *Begin Offset* option.  If the segment begins before the full distance specified by *Begin Segment Offset*, then it is not applied, and a sub rectangle could be placed at the corner of the sub rectangle path. The system will not create a spacing error; if the first sub rectangle in a segment would be too close to the last sub rectangle in the previous segment (if there would be less space than you specified for the *Space* option), the system discards the last sub rectangle in the previous segment. The default is 0.

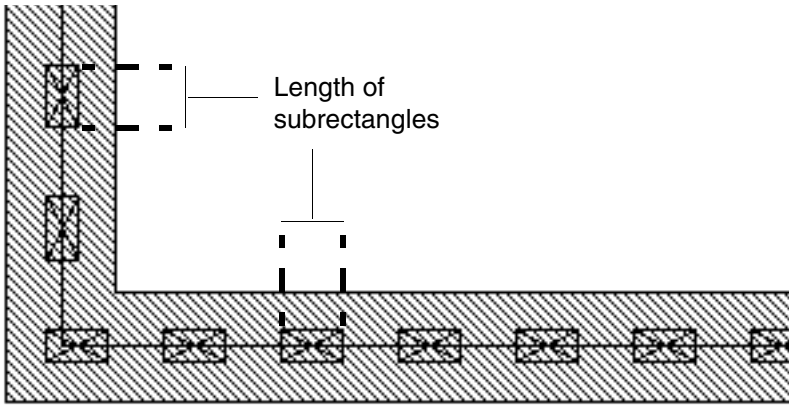
© 2023 Cadence Design Systems, Inc.

Field	Description
End Segment Offset	Specifies the offset of the last rectangle in a segment from the edge of the master path.The value must be a positive integer or floating-point number. Does not apply to the last segment of a sub rectangle sub path; the position of the last rectangle in the last segment is determined by the value of the *End Offset* option.If the segment ends before the full distance specified by *End Segment Offset*, then it is not applied, and a subrectangle could be placed at the corner of the subrectangle path. The system will not create a spacing error; if the first subrectangle in a segment would be too close to the last subrectangle in the previous segment (if there would be less space than you specified for the *Space* option), the system discards the last subrectangle in the previous segment. The default is `0`. 

© 2023 Cadence Design Systems, Inc.

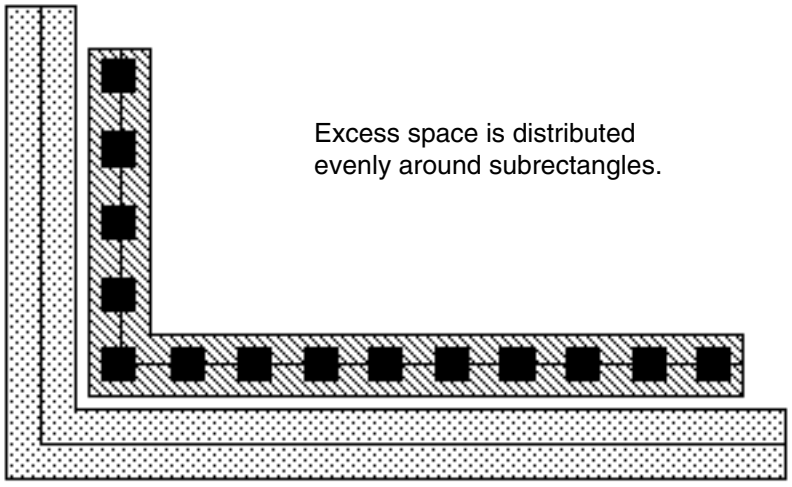
Field	Description
Width	Specifies the width of the rectangle(s). The width is parallel to the width of the master path. The value must be a positive integer or floating-point number. If not specified, the system uses the value of the *Length* field; if neither is specified, the system uses the minWidth rule for the subrectangle layer from the technology file. If the minWidth rule is not defined in the technology file, the system uses the width of the master path. 

© 2023 Cadence Design Systems, Inc.

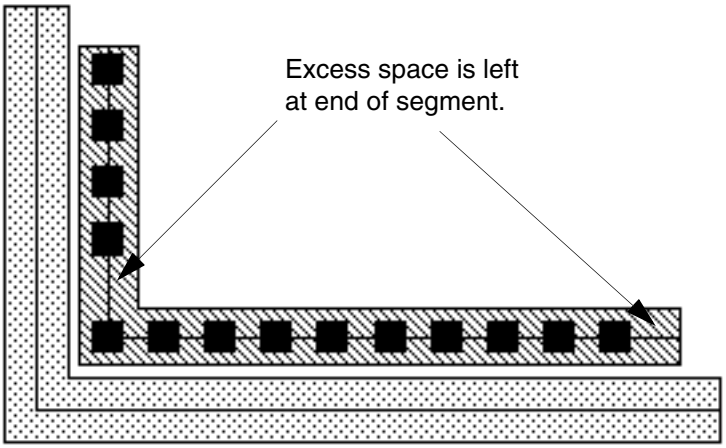
Field	Description
Length	Specifies the length of the subrectangle(s). The length is parallel to the master path centerline. If not specified, the system uses the minWidth rule for the subrectangle layer from the technology file. If the minWidth rule is not defined in the technology file, you must specify a length; otherwise, the length remains zero, the system displays a warning in the CIW, and no subrectangles are created. 
Space	Specifies the distance between the edges of adjoining rectangles. The value must be a positive integer or floating-point number. If not specified, the system uses the `minSpacing` rule for the subrectangle layer from the technology file. If the `minSpacing` rule is not defined in the technology file, you must enter a number or the system displays an error in the CIW.

© 2023 Cadence Design Systems, Inc.

Field	Description
Gap	Specifies the method the system uses to place subrectangles within each segment. The default is *distribute*. The system always uses the value of the *Space* field for minimum space between subrectangles and calculates the maximum number of rectangles that fit in the segment. When the value of *Gap* is *distribute*, the system distributes the space around subrectangles as evenly as possible, in multiples of the grid space specified by `mfgGridResolution`. Any remaining space is left after the last subrectangle in the segment.



When the value of *Gap* is *minimum*, the system places subrectangles *Space* apart until there is no space for another rectangle, then leaves the excess space after the last subrectangle in the segment.



© 2023 Cadence Design Systems, Inc.

Field	Description
Justification	Specifies from which part of the master path to separate the sub-rectangles, in relation to the direction of the master path. The default is *center*. When the value is *left*, the right edge of the sub-rectangles is separated from the left edge of the master path. When the value is *right*, the left edge of the sub rectangles is separated from the right edge of the master path. When the value is *center*, the centerline of the sub rectangles is separated from the centerline of the master path.
Separation	Specifies the distance between the centerline or an edge of the sub rectangles and the centerline or an edge of the master path, depending on the value of the *Justification* field. The default is `0`.
Diagonal	Specifies whether to create sub rectangles on diagonal portions of the path. Setting this option to `t` works only when diagonal path segments are 45 degrees and all sub rectangles on the path are square.

For *Separation*, whether you enter a positive or negative number depends on where you want to define the set of subrectangles in relation to the master path.

Separation	Center Justification	Left Justification	Right Justification
Zero	Center of width of subrectangles on master path centerline	Left edge of master path coincident with right edge of subrectangles	Right edge of master path coincident with left edge of subrectangles
Positive number	Center of width of subrectangles on left side of master path centerline	Left edge of master path on right side of right edge of subrectangles	Right edge of master path on left side of left edge of subrectangles
Negative number	Center of width of subrectangles on right side of master path centerline	Left edge of master path on left side of right edge of subrectangles	Right edge of master path on right side of left edge of subrectangles

© 2023 Cadence Design Systems, Inc.

Related Topics

Subparts of a Multipart Path

About Sets of Subrectangles

About Enclosure Subpaths

Create Multipart Path Form

© 2023 Cadence Design Systems, Inc.

Rotate Form

Use the Rotate form to change the orientation of geometric objects.

Field	Description
Specify reference and angle	Activates the following options in the form for specifying the reference for rotating an object: *Reference*, *X,Y*, and *Flip around combined bbox*. Environment variable: useRotateReference
Reference	Specifies the point around which an object is rotated. The default is *CenterCenter*. Environment variable: rotateReference
X,Y	Specifies the X and Y coordinates around which an object is rotated. This field is available when you select *Custom* from the *Reference* list.
Rotate around combined bbox	Rotates multiple selected objects around the specified reference point of the combined bounding box of the objects. Environment variable: rotateAroundCombinedCenter
Angle of Rotation	Sets the angle for rotating the object you select. Valid values are: ■ *90*: Turns the object, instance, or label 90 degrees counterclockwise. ■ *180*: Mirrors the object about the Y axis, or turns an instance or label 180 degrees counterclockwise. ■ *270*: Mirrors the object about the X axis, or turns an instance or label 270 degrees counterclockwise.
Snap Mode	Specifies the snap mode for rotating the objects. Valid snap modes are *anyAngle*, *diagonal*, *orthogonal*, *horizontal*, or *vertical*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Precision	Controls the precision of the angle of the snap. ■ *1 Degree:* Lets you use whole numbers; for example, 10. ■ *.1 Degree:* Lets you use one decimal place; for example, 12.5. ■ *Any:* Lets you use three decimal places; for example, 15.125, assuming 1,000 database units/user units.

Related Topics

Object Rotation

useRotateReference

leHiRotate

© 2023 Cadence Design Systems, Inc.

Rotate Left Form

Use the Rotate Left form to define how objects are rotated by 90 degrees in the counterclockwise direction.

Field	Description
Reference	Specifies the point around which an object is rotated in 90-degree increments, counterclockwise. Valid values are *LowerLeft*, *CenterLeft*, *UpperLeft*, *LowerCenter*, *CenterCenter*, *UpperCenter*, *LowerRight*, *CenterRight*, and *UpperRight*. Environment variable: rotateReference
Rotate around combined bbox	Rotates multiple selected objects around the specified reference point of the combined bounding box of the objects. Environment variable: rotateAroundCombinedCenter

© 2023 Cadence Design Systems, Inc.

Rotate Right Form

Use the Rotate Right form to define how objects are rotated by 90 degrees in the clockwise direction.

Field	Description
Reference	Specifies the point around which an object is rotated in 90-degree increments, clockwise. Valid values are *LowerLeft*, *CenterLeft*, *UpperLeft*, *LowerCenter*, *CenterCenter*, *UpperCenter*, *LowerRight*, *CenterRight*, and *UpperRight*. Environment variable: rotateReference
Rotate around combined bbox	Rotates multiple selected objects around the specified reference point of the combined bounding box of the objects. Environment variable: rotateAroundCombinedCenter

Related Topics

Object Rotation

© 2023 Cadence Design Systems, Inc.

Save a Copy Form

Use the Save a Copy form to save a copy of the currently open design.

Field	Description
Name	Shows the name of the cell. The other columns in the list box show the current library, cell, and view in which the cell exists. The following three fields let you specify a different name and location to save the cellview: ■ The first cyclic field lists the libraries and you can choose the one in which you want to save the design. By default, the library of the current design is selected. Environment variable: saveACopyLibName ■ The second text field lets you specify a different name for the cell. If you retain the name and also do not change the library and view locations, the existing cellview will be overridden. ■ The third and the last cyclic field lets you select the view type for the cell you are saving.
Open Saved Copy	Opens the saved copy of the design after you create it. The *Open for* and *Open in* fields are enabled when this option is selected. Environment variable: openSavedCellViewInWindow
Open for	Specifies whether to open the saved copy of the design for reading or editing.
Open in	Specifies whether the saved copy of the design is opened in a new tab, the current tab, or a new window.
Open Connectivity Reference	Specifies whether the schematic view for connectivity reference is opened when you open the saved copy of the design. This setting overrides the `lxSchematicDefaultApp` environment variable and the Connectivity form.

Related Topics

Saving a Design

© 2023 Cadence Design Systems, Inc.

Save All Mark Nets Form

Use the Save All Mark Nets form to save a traced net**.**

Field	Description
Library	Lets you specify the library in which you want to save the mark nets.
Cell	Lets you specify the cell in which you want to save the mark nets in the specified library.
View	Lets you specify the view name in which you want to save the mark nets in the specified library and cell.
Browse	Opens the *Library Browser* to let you select the library, cell, and view in which to save the generated mark nets.

Related Topics

Saving Marked Nets

leHiSaveAllHighLightMarkNet

© 2023 Cadence Design Systems, Inc.

Save Configuration Form

Use the Save Configuration form to save the configuration for displaying information about shapes and objects.

Field	Description
Select configuration name	Specifies the name for the object display configuration you are saving.
Select path	Specifies the path for the configuration display file. The configuration is saved to `odc.il` at the location *`selectedPath`*`/dfII/ODC/odc.il`. The path that you choose to save the configuration determines the sequence in which a configuration is loaded and is derived from the `setup.loc` file.

Related Topics

Managing Object Information Configuration

© 2023 Cadence Design Systems, Inc.

Save Multipart Path Templates Form

Use the Save Multipart Path Templates form to save the values currently in the Create Multipart Path and ROD Subpart forms as a template with a unique name.

Field	Description
Template List	Lists the existing templates.
ASCII Fie	Saves the template definition to an ASCII file. The extension of the template file can be `.il` or you can use any other extension.
Technology Library	Saves the template definition to the selected technology library.

Related Topics

Multipart Path Values Saved as a Template

Create Multipart Path Form

ROD Subpart Form

© 2023 Cadence Design Systems, Inc.

Save/Restore Selection Set Form

Use the Save/Restore Selection Set form to save objects selected in the canvas as a selection set so that you can restore them later.

Field	Description
Name	Enables you to specify the name of the selection set under which you want to save the objects selected in the canvas.
Store	Enables you to save the selection set information to a file for later retrieval. If the Store button does not work, the selection set is empty or the name you entered in the *Name* field already exists. A warning is issued in CIW.
Stored Selection Sets	Lists the saved selection sets. The selection set name you entered in the *Name* field is listed here once you click the *Store* button.
Select	Retrieves the stored information and displays it on the originally selected set. You must click the stored selection set prior to clicking the *Select* button to bring up the stored selection set you want to display. Each selection you make adds additional selection to the display unless you click the *Deselect* button in the form before making another selection.
Deselect	Removes selection from the stored set from the objects in the stored selection set you have selected to display in the cellview.
Delete	Removes the set that you have selected in the Stored Selection Sets list box. If the deleted set is the current selection set, the selection is removed from the form, but remains displayed until you change the selection.

Related Topics

Object Selection

© 2023 Cadence Design Systems, Inc.

Save Via Variant Form

Use the Save Via Variant form to save a standard or custom via definition as a via variant.

Field	Description
Save to	Specifies where to save the via variant - in the technology library or in the design itself. If you select *Technology Library*, select the library from the drop-down list in which to save the via variant.
Save as	Specifies the name of the new via variant. The drop-down list shows the earlier saved, if any, via variants. A warning message is generated if you overwrite an existing via variant.
Summary	The list box is read-only and displays all the parameters, including the updated ones, for the new via variant. `viaDef` at the top indicates the parent viaDef used for the creation of the via variant.

Related Topics

Create Via Form

Via Creation

© 2023 Cadence Design Systems, Inc.

Save View Form

Use the Save View form to save the image shown in a cellview window to a file that lasts for the current editing session.

Field	Description
Name	Sets the name you want to assign to the view. This name will appear in the Restore View form.

Related Topics

Saving and Restoring a View

Restore View Form

© 2023 Cadence Design Systems, Inc.

Save Recording Form

Use the Save Recording form to save the recording of your actions, which you can replay later to perform those actions quickly. You can record your actions by using the Capture and Relay assistant in Layout EXL or a higher tier.

Field	Description
Command Name	Specifies the name of the recording.
Description	Specifies information about the recording. The tooltip for a saved recording shows a brief description in the Capture and Relay assistant.
Group	Specifies the group under which you want to categorize your recording for easy access.
Save To	Specifies the location where the recording is saved. The list of locations is derived from `setup.loc`. Your recording is saved as a `.car` file under the `dfII/captureReplay` directory at the selected location.
Save	Saves the recording at the specified location.

Related Topics

Capture and Replay Assistant

Recording Your Actions in a Design

leCARStartCapture

leCARStopCapture

© 2023 Cadence Design Systems, Inc.

Select Area Shape Type Form

Use the Select Area Shape Type form to specify the type of shape within which you want to select objects. You can choose to draw a rectangle or a polygon to define the area for selecting objects.

Related Topics

Selecting Objects

© 2023 Cadence Design Systems, Inc.

Select Array Member Form

Use the Select Area Member form to select objects within an enclosed area**.**

Field	Description
Row	Indicates the row the array member is in.
Column	Indicates the column the array member is in.

Related Topics

Descending into a Cellview

Editing a Cell in Place

© 2023 Cadence Design Systems, Inc.

Selection Options Form

Use the Selection Options form to set controls for object selection.

Field	Description
Selection Controls	This section enables you to set the selection mode as partial or full and specify how objects are selected.
Mode	Controls whether the selection is full or partial. ■ *Full:* Selects objects fully. ■ *Partial:* Selects objects fully if you completely enclose them within the area selection box, and partially selects objects if you partially enclose them within the area selection box.
Spine	Selects all the contiguous pathsegs that have the same width as the first-clicked pathseg. It excludes the pathsegs that branch off from the main spine at a T junction. This option is selected by default. If you do not select this option, then only the pathseg you click at gets selected. Environment variable: spineSelection ■ *Allow different width:* Enables spine selection for pathsegs with different widths. This option is available only when *Spine* is selected.

© 2023 Cadence Design Systems, Inc.

Field	Description
Via Stack	Allows the selection of a via stack with a single click. Environment variable: viaStackSelection ■ *Vias with matching sizes between common layers:* Lets you select vias of the same via stack when there are two or more super-imposed via stacks on adjacent layers with the same origin. Vias of the same via stack have the same size on common layers. If this option is off, multiple super-imposed via stacks are selected as a single via stack and are available for editing as a single via stack. This option is available only when Via Stack is selected. For each pair of adjacent vias in a via stack, the bounding box of the top layer in the lower via is equal to the bounding box of the bottom layer in the higher via. This refers to the matching sizes of the vias between the common layers and is used for identifying vias of the same via stack. For example, the two vias M1M2 and M2M3 in the via stack M1M3 are on adjacent layers and share the same origin. In addition, the bounding box of Metal2 shape in M1M2 is equal to the bounding box of the Metal2 shape in M2M3. This testifies that the two vias belong to the same via stack. Environment variable: viaStackSelectionMatchingSizes
Mosaic Partial Selection	Allows partial selection of mosaics.
Via Partial Selection	Allows partial selection of vias. You can control the status of this option using the following SKILL command: `hiGetCurrentWindow()~>selectPartialVia`
Net Based Selection	Allows selection of nets in the layout editor. Environment variable: netBasedSelectionOn
Repeat Commands	Allows the commands to automatically repeat, if you first select the command and then the object. This option does not affect commands if you first select an object, then the command.
Point Selection Controls	This section enables you to set overlap mode and controls for object selection.

© 2023 Cadence Design Systems, Inc.

Field	Description
Overlap Mode	■ *Toggle mode*: Allows you to change selection between two objects. When you click the first object using the left mouse button, the software toggles back and forth between the smaller object and the larger object. You can add more objects to the selected set by holding down the Shift key and selecting additional objects. You can remove selections by using the Control key and clicking the left mouse button on the original selected object. Objects are deselected in the opposite order in which they were selected. Once you have selected the objects you want in the set, you can also click the selected edge of one of the objects. This removes all of the selections except the one you have clicked on. ■ *Cycle mode:* Allows you to select overlapped objects one by one by using the left mouse button to select any object and continuing to select them by clicking on the original object until you reach the largest overlapped object. This mode is for use when three or more objects overlap at the same point. The cycle mode always cycles from the smallest to the largest object.
User Aperture Mode	Defines a box around the initial point. You can select multiple points within this aperture range and the software will treat it as one point. The default is five pixels. You can specify the size of the box in the *Size* field. Environment variable: pointSelectAperture ■ *off:* Specifies that point selection does not use the specified aperture. ■ *on:* Specifies that point selection uses the specified aperture. ■ *visible:* Specifies that point selection uses the specified aperture, which displays on the canvas.

© 2023 Cadence Design Systems, Inc.

Field	Description
Select objects larger than the window	Enables you to control whether you can select an object that has bounding box or boundary greater than the size of the design canvas. When this option is not selected, you cannot select objects larger in size than the design canvas. This check box is selected by default. Environment variable: selectObjectLargerThanWindow
Select objects of edit cellview first	Enables you to point-select objects belonging to the edit cellview in preference over objects from another cellview, such as, shapes of instances, even though objects of the edit cellview might be on a higher layer. Environment variable: selectObjectInEditCvFirst
Area Selection Controls	This section enables you to define the area selection in full and partial selection modes.
Full Mode	Controls the area selection in *Full* selection mode. Environment variable: fullSelectionMode ■ *Enclosed Figures:* Selects the objects you completely enclose with the area selection box. ■ *Enclosed and Crossed Figures:* Selects the objects you completely or partially enclose with the area selection box. ■ *Crossed Figures:* Selects the objects you partially enclose with the area selection box and does not select any fully-enclosed objects.

© 2023 Cadence Design Systems, Inc.

Field	Description
Partial Mode	Controls the area selection in *Partial* selection mode. Environment variable: partialSelectionMode ■ *Vertex:* Selects the objects you completely enclose with the area selection box, the vertices of any objects you partially enclose with the area selection box, and the edges of any objects you enclose with the area selection box. ■ *Enclosed Edges:* Selects the objects you completely enclose with the area selection box and the edges of any objects you partially enclose with the area selection box. ■ *Crossed Edges:* Selects the edges of any objects you enclose with the area selection box and does not select any fully-enclosed objects. ■ *Enclosed and Crossed Edges:* Selects the objects you completely enclose with the area selection box, the vertices of any objects you partially enclose with the area selection box, and the edges of any objects you partially enclose with the area selection box.
Auto Store Selection Controls	This section enables you to limit the number of selection sets saved and the objects included in a selection set.
Auto Store	Enables storage of selection sets, which can contain fully or partially selected figures. This option is disabled by default. The option is not compatible with the Undo or Redo commands. If a subview is already open, the option does not work with Edit In Place and Descend commands.
Max number of Stored sets	Limits the number of sets which can be retained by the Save/ Restore Selection Set form. The maximum and default number of stored sets is 10. You can change the limit as required. The oldest saved selection set is removed, without any warning or error message. Environment variable: maxAutoStoreSelectionSet

© 2023 Cadence Design Systems, Inc.

Field	Description
Max number of selected elements within a set	Limits the number of objects you can enter into one selection set. The maximum and default number of stored sets is 100. You can change the limit as required. The oldest saved selection set is not recorded and generates a warning message in CIW. Environment variable: maxAutoStoreSelectionObj

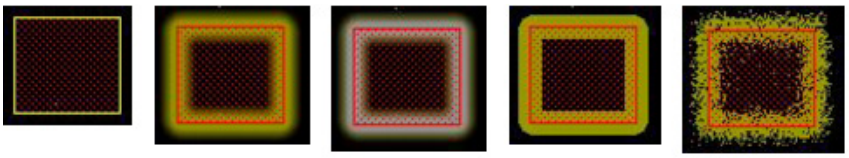
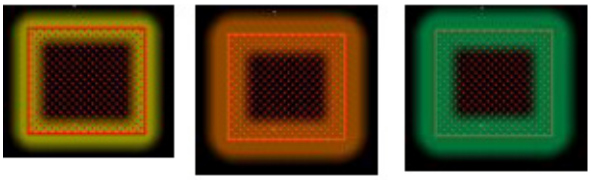
Related Topics

Selecting a Group of Objects

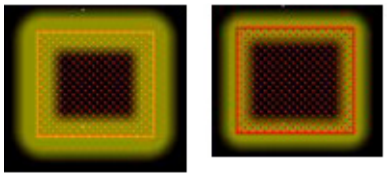
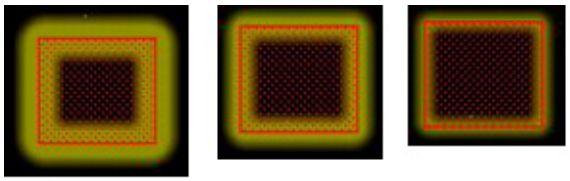
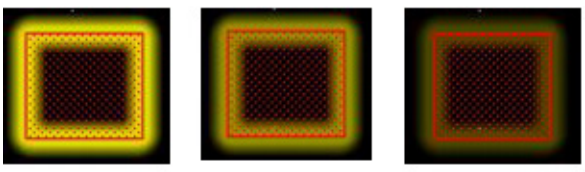
© 2023 Cadence Design Systems, Inc.

Selection Protection Highlight Options Form

Use the Selection Protection Highlight Options form to specify the protection highlight options during the selection of objects.

Field	Description
Options	This section lets you specify the halo mode, color, level and width.
Halo Mode	Determines the type of halo around a protected object. The *normal* mode does not generate any halo. The default mode is *fadeout*. The figures below show the effect of different modes on objects marked protected. Environment variable: protectedObjHighlightHaloMode  *normal* *fadeout* *glow* *plain* *random*
Halo Color	Determines the color of the halo. The default halo color is yellow. When the *Halo Mode* is set to *glow*, the *Halo Color* appears on the outer edges of the halo. The figures below show examples of the halo in three colors.  yellow orange green

© 2023 Cadence Design Systems, Inc.

Field	Description
Level	Determines the placement of the halo relative to the object boundary box. The default level is *under*. Environment variable: protectedObjHighlightHaloLevel  *over* *under*
Width	Specifies the thickness of the halo on a protected object. The default width is *normal*. The figures below show halos with different thickness. Environment variable: protectedObjHighlightHaloWidth  *thick* *normal* *thin*
Transparency	Controls the visibility of the halo. The visibility of the halo is the maximum at the lowest transparency setting. The default transparency value is `40`. The figures below show halos at different transparency settings. Environment variable: protectedObjHighlightHaloTrans  *10* *40* *70*

Related Topics

Object Selectability Controls

© 2023 Cadence Design Systems, Inc.

Set Area View Level Form

Use the Set Area View Level form to set the display detail for an area.

Field	Description
Display Levels	Specifies the first (*From*) and last (*To*) levels in the design hierarchy that can be seen in detail. The hierarchy levels are numbered 0 to 32. Shapes in the current cellview are at level 0, shapes in the masters of instances inside of it are level 1, and so forth.

Related Topics

Setting an Area Display

leHiSetAreaViewLevel

© 2023 Cadence Design Systems, Inc.

Set Default Application Form

Use the Set Default Application form to set the default application for opening the current cellview and all cellviews with the same view type as the current cellview.

Field	Description
Current application	Displays the current application in which the cellview is open.
Current cellview default	Displays the default application set for the current cellview.
Current default for all maskLayout cellviews	Specifies the default application set for all cellviews with the view type of the current cellview.
Set default application for current cellview to	Sets the default application for the current cellview. ■ *As Is*: Retain the current application setting. ■ *None*: Removes any default application setting. ■ <*anyRegisteredApplication*> Sets the selected registered application, such as Layout XL, as the default application for the current cellview. To apply this setting, the `saveViewSubType` environment variable must be set to `nil`, otherwise the Layout cellview automatically opens in the same application in which it was last saved.
Set default application for all layout cellviews to	Sets the default application for all cellviews with the view type of the current cellview. ■ *As Is*: Retain the current application setting. ■ *None*: Removes any default application setting. ■ <*anyRegisteredApplication*> Sets the selected registered application, such as Layout XL, as the default application. Environment variable: maskLayoutDefaultApp To apply this setting, the `saveViewSubType` environment variable must be set to `nil`, otherwise the Layout cellview automatically opens in the same application in which it was last saved.

© 2023 Cadence Design Systems, Inc.

Related Topics

saveViewSubType

Specifying New Default Values for a Virtuoso Session

Setting the Default Application for a Cellview Type

© 2023 Cadence Design Systems, Inc.

Set Pin Label Text Style Form

Use the Set Pin Label Text Style form to specify pin name characteristics.

Field	Description
Height	Sets the height of the name in user units, usually microns.
Font	Sets the text style of the name.
Text Options	Displays the label for the pin only if the *Pin Names* option in the Display Options form is enabled. ■ *Drafting:* Prevents the label from being rotated more than 90 degrees. ■ *Overbar:* Determines how text strings containing underscore characters are displayed. ■ *Create As Label:* Determines whether the pin is created as a label or text display. If this option is selected, a label is created, else a text display is created. Environment variable: pinTextCreateAsLabel
Layer Name	Sets the layer for the pin name. ■ *Same as Pin:* Puts the pin name on the same layer as the pin. Environment variable: pinTextSameLayer
Layer Purpose	Puts the pin name on the same layer as the pin being created. ■ *Same as Pin:* Puts the pin name on the same layer as the pin. Environment variable: pinTextSamePurpose
Justification	Sets the location of the name origin.
Orientation	Sets the orientation of the pin name. Environment variable: pinTextOrientation

Related Topics

Setting Pin Label Text Style Options

© 2023 Cadence Design Systems, Inc.

Show Selected Set Window

The *Show Selected Set* section shows information about the each type of object that is selected in the current cellview.

Field	Description
Net Statistics	Lists the number of terminals, instance terminals, guides, and steiners for each net in the selected set. For each constituent figure of the net, the report lists the layers, wire length, horizontal and vertical wire length, 45-degree wire length, vias, routes, and wire segments.
Connectivity Summary	Lists connectivity statistics for the selected set based on signal types, such as signal, power, ground, and clock.
Routing Statistics	Displays routing statistics for the selected set based on layers. For each layer, the report lists total wire length, horizontal and vertical wire length, 45-degree wire length, guide length, and the number of down vias and guides.
Total Shapes Selected	Lists the selected objects, their layers, the number of points used to create them, and the coordinates of the bounding box that surrounds each object.
Total Instances Selected	Lists all selected cell instances, including each instance name, placement orientation, and the coordinates of its origin.
Total Mosaics Selected	Displays information about arrays of cell instances are selected.
Total Boundaries Selected	Lists the selected boundaries, their types, the number of edges used to create them, the area, and the boundary names.
Total Blockages Selected	Lists the selected blockages, their types, their layers, their owners, the number of points used to create them, and their areas.
Total Vias Selected	Lists the selected vias, their types (standard or custom), their names, their nets, their directions, and in the case of custom vias their library, cell and view names.

Related Topics

leHiShowSelSet

© 2023 Cadence Design Systems, Inc.

Size Form

Use the Size form to reduce or enlarge objects by stretching each edge in or out by the given number of user units

Field	Description
Increase Size By Value	Controls the amount by which you enlarge or reduce the object. Positive numbers enlarge, negative numbers reduce.
Increase Size By Direction	Controls the amount by which you enlarge or reduce the object in the *Left*, *Right*, *Top*, or *Bottom* direction. Positive numbers enlarge, negative numbers reduce.

Related Topics

Modifying Object Size

leHiSize

© 2023 Cadence Design Systems, Inc.

Split Form

Use the Split form to split and stretch a portion of an object.

Field	Description
Lock angles	Prevents you from changing the angle of a corner or edge as you stretch it.
Snap Mode	Controls the direction in which you can stretch an edge. Environment variable: snapMode
Keep Wires Connected to	Splits paths, pathsegs, and via vertices coincident with a selected vertex. These are split as a set. ■ *All wires and vias:* Maintains pathseg-to-pathseg and pathseg-to-via connections during split. ■ *Spine wires only:* Maintains connections to spine pathsegs but not to connecting vias when splitting a spine. This option is selected by default. ■ *No wires and vias:* Does not maintain pathseg-to-pathseg and pathseg-to-via connections during split.
Keep wires connected to shapes	Maintains connection with shapes, such as pin figures, rectangles, polygons, and MPP, coincident with the selected vertex during the split operation. This check box is selected by default.
Lock Unselected Vias	Locks the vias that are not selected while splitting a wire. By default, this option is disabled. It is enabled only when the *All wires and vias* option is selected from the *Keep Wires Connected To* field.

Related Topics

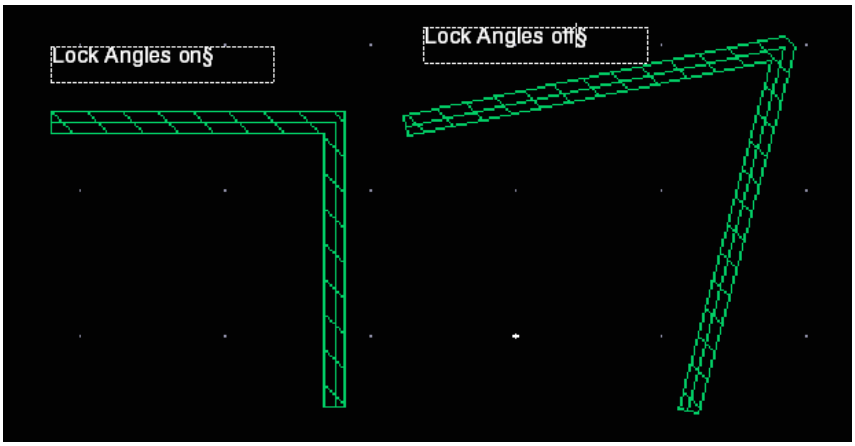
Object Splitting

leHiSplit

© 2023 Cadence Design Systems, Inc.

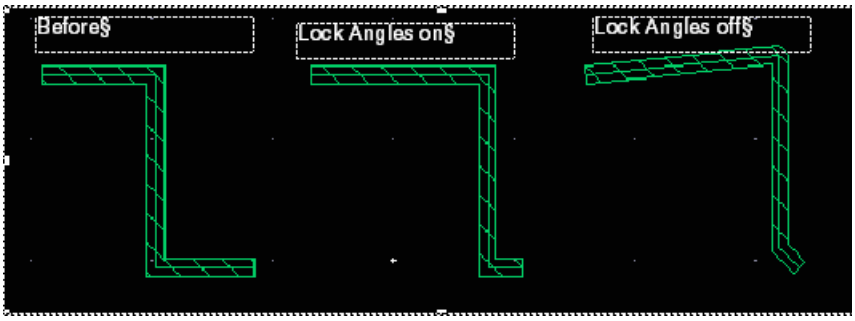
Stretch Form

Use the Stretch form to stretch objects in the specified window.

Field	Description
Lock Angles	Prevents you from changing the angle of a corner or edge as you stretch it. When off, *Lock Angles* may snap a pathSeg to a different legal angle; a path may take on any angle. In all cases, the non-selected end does not move. ■ Stretching a corner of a path: When *Lock Angles* is on, stretching a corner does nothing. When *Lock Angles* is off, you can stretch the corner of a path by moving the corner to any new location desired, possibly creating any-angle segments and acute angles.  ■ Stretching a segment in a path: When Lock Angles is on, the lengths of the neighboring segments are adjusted. When Lock Angles is off, you can move the segment to any new location desired, possibly creating any-angle segments and acute angles.

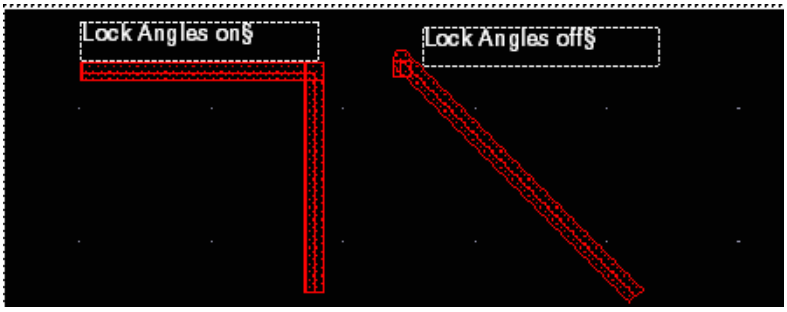
© 2023 Cadence Design Systems, Inc.

Field	Description



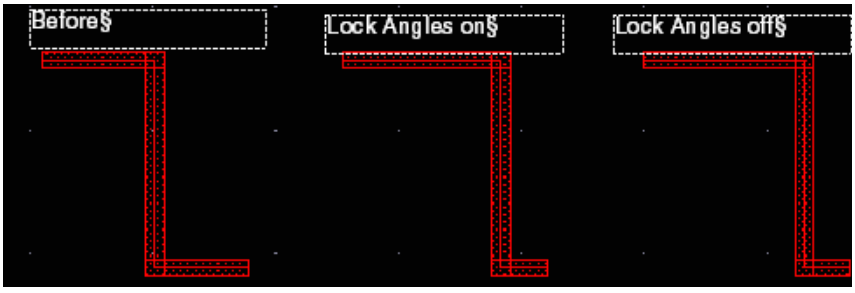
- Stretching a corner of a pathSeg:

  When *Lock Angles* is on, stretching a corner does nothing. When *Lock Angles* is off, you can stretch the corner of a pathSeg corner to any new location desired on a 45 degree grid.



- Stretching a segment in a pathSeg:

  When *Lock Angles* is on or off, the lengths of the neighboring segments are adjusted on a 45 degree grid.



© 2023 Cadence Design Systems, Inc.

Field	Description
Snap Mode	Controls the direction in which you can stretch an edge. Environment variable: snapMode
Delta X,Y	Lets you specify the number of user units the object will stretch on the X and/or Y axes relative to its point of origin.
Apply	Stretches the origin point of the selected object as specified in the Delta X and Delta Y fields.
Display draglines	Controls whether drag-lines are displayed during the *Stretch* command. By default, the drag-lines indicate connections from the pins of the object you are moving to pins of the nearest objects. Environment variable: flightLineEnable  *Tip* When constraint-aware editing is on, the *Stretch* command honors the following physical constraints: *Alignment*, *Fixed*, *Locked*, *Matched Parameters*, *Orientation*, *Matched Orientation*, and *Symmetry*.
Connections Section	This section lets you specify the wire connection settings.
Move Connected Instances and Wires	Controls whether the instances and wires connected to an instance are also moved or stretched to maintain connectivity when you move the instance. (Layout EXL Only) Environment variable: moveConnectedInstancesAndWires

© 2023 Cadence Design Systems, Inc.

Field	Description
Keep Wires Connected to	Stretches paths, pathsegs, and via vertices coincident with a selected vertex. These are stretched as a set. It also allows overlapping wires on the same net to remain connected during the stretch command. ■ *All wires and vias:* Maintains pathseg-to-pathseg and pathseg-to-via connections during stretch. ■ *Spine wires only:* Maintains connections to spine pathsegs but not to connecting vias when stretching a spine. This option is selected by default. ■ *No wires and vias:* Does not maintain pathseg-to-pathseg and pathseg-to-via connections during stretch. Environment variable: maxNumConnWireElements
Keep wires connected to shapes	Maintains connection with shapes, such as pin figures, rectangles, polygons, and MPP, coincident with the selected vertex during the stretch operation. This check box is selected by default. Environment variable: keepWiresConnectedToNonWires
Lock Unselected Vias	Locks the vias that are not selected while stretching a wire. Since the via is locked, the wire is modified to keep the shape connected. In case, the wire could not be modified to maintain the connection, the via is anyway moved to maintain the connection. By default, this option is disabled. It is enabled only when the *All wires and vias* option is selected from the *Keep Wires Connected To* list.
Keep guard ring ends connected	Enables you to keep the two touching ends of a guard ring together while stretching the guard ring. If the check box is not selected, only the selected end of the guard ring is stretched and results in opening up the guard ring. This check box is not selected by default. This check box is common in the Quick Align Form. The setting in one form is preserved in the other form.

© 2023 Cadence Design Systems, Inc.

Field	Description
Chain Mode	Lets you move device chains. You can only completely select instances, not parts of instances, so that the stretch command will only allow you to move them using the stretch command. To stretch parts of instances you must use the stretch handles. ■ *All:* Moves all the instances in a chain. ■ *Selected:* Moves only selected instances. ■ *Selected Plus Left:* Moves the selected instance and all instances to the left of it in the chain. ■ *Selected Plus Right:* Moves the selected instance and all instances to the right it in the chain.
Special Objects Section	This section lets you specify the settings for selecting vias and mosaics.
Mosaic Partial Selection	Allows partial selection of mosaics.
Via Partial Selection	Allows partial selection of vias. When you select this option, the *Via Mode* option is enabled.
Via Mode	Lets you defines the mode of stretching a via, when the *Via Partial Selection* option is enabled. ■ *Stretch Metal:* Fills the metal in the via as per the dimensions of the new box created after the stretch operation. ■ *Stretch Row/Col:* Maintains the number of rows when stretching horizontally and the number of columns when stretching vertically. ❑ *Allow opposite metal extend:* Ensures that the cut box stays at the same location even if the enclosure is increased after the stretch operation. The metal of the new via goes beyond the existing metal. This option is available only when the *Stretch Row/Col* via mode is selected. Environment variable: stretchViaMode

© 2023 Cadence Design Systems, Inc.

Field	Description
Snap Section	■ *Pins and boundary to grid:* Snaps the stretched object to a grid depending on the block type. If it is an *analog* block, the place and route boundary and pin edges are snapped to the manufacturing grid. If it is a *digital* block, the place and route boundary is snapped to the placement grid and the pin centers are snapped to the routing grid. Pins with placement status `unplaced`, `fixed`, `locked`, or `unknown` are not snapped, even if they were touching the boundary before the stretch. Environment variable: snapToGrid ■ *Pins to boundary:* Snaps pins to the place and route boundary automatically. ■ *Soft Block Pins to:* Snaps snaps soft block pins either to the *Modified Edge* or to the *Closest Edge* automatically after the reshape is complete. During the stretch operation, in case the soft block is stretched outside the top-level PR boundary, then all the instance pins on the soft block, with the placement status as *placed*, are snapped to the manufacturing grid. This is irrespective of the block type because the routing grid is not initialized outside the PR boundary but the manufacturing grid exists outside the PR boundary.

© 2023 Cadence Design Systems, Inc.

Field	Description
Boundary Section	■ *Constant area stretch:* Maintains the area of a soft block constant while its place and route boundary is stretched. Additional options let you specify the adjustable edges, that is, the edges that can be moved automatically in order to maintain a constant area. ■ *Left edge, Right edge, Top edge, Bottom edge:* Determines the direction in which the boundary is expanded, whether left, right, top, or bottom edge, after the reshape is complete. For example, when you select *Right edge* and *Top edge*, only these edges are stretched keeping the left and bottom edges anchored. This helps to maintain a constant area while having control over the edges to be moved. ■ *Stretch with P&R boundary:* Stretches along with the PR boundary, the shapes intersecting an edge of a PR boundary or abutted to an edge of a PR boundary. The supported shapes are orthogonal pathSegs, rectilinear pins, rectangles and row regions. Environment variable: stretchWithPRBoundary

Related Topics

Object Stretching

Stretching an Object in Level-1 Editing Mode

Stretching Multipart Paths

Stretching Wires and Vias

Removing Loops from Wires

leHiStretch

Default Constraint Types

Constraint-Aware Editing

© 2023 Cadence Design Systems, Inc.

Summary Window

Use the Summary window to view statistics related to the current cellview.

Field	Description
Show Environment	Displays information about the cellview and its edit mode, display levels, and current entry layer.
Layer Object Statistics	Lists the number of shapes created in the cellview, the layer on which shapes are present, and the color and lock state of the shapes. The shape types listed are Arc, Bend, Donut, Dot, Ellipse, Label, Line, Path, PathSeg, Polygon, Rect, Trl, Taper, and TextDisplay.
Routing Grid Statistics	Displays the x and y steps, offsets and the manufacturing grid information of the valid routing layers. Valid routing layers belong to a valid constraint group. The default constraint group for the design is considered to be the valid constraint group. The following constraint groups are considered in the given order: default constraint group, active constraint group, or foundry constraint group. If one does not exist, the next one in the order is considered. If the foundry constraint group also does not exist, then `Metal` and `Poly` are considered as valid routing layers.
Instance Statistics	Displays names of the cell instances and arrays of instances (mosaics) that have been placed in the current cellview, as well as the total number of unplaced instances.
Design Statistics	Displays the layer summary that lists the total number of metal and routing layers. The Component Summary portion of this section reports the placement status of components.
Wire Statistics	Displays the following that have been placed in the current cellview: ■ Total number of standard and custom vias ■ Total number and length of paths, path segments, 45-degree paths, and 45-degree path segments
Marker Object Statistics	Displays the number of error markers that have flagged design violations in the current cellview.
Row Object Statistics	Displays the number of horizontal and vertical rows that have been placed in the current cellview.

© 2023 Cadence Design Systems, Inc.

Field	Description
Track Pattern Statistics	Displays the number of horizontal and vertical tracks that have been placed in the current cellview.
Blockage Object Statistics	Displays the number of blockage objects, placement area blockage objects, placement area halo objects and layer blockage objects that have been placed in the cellview. It displays the total number of blockage objects, the blockage count of each type of blockage in the cellview, and is sorted by layers.
PR Boundary Object Statistics	Displays whether a PR boundary object has been placed in the cellview, shows the origin point of the PR boundary and its area measurement in user units, and the shape of the PR boundary.
Net Statistics	Lists the number of terminals, instance terminals, guides, and steiners for each net in the cellview. For each constituent figure of the net, the report lists the layers, wire length, horizontal and vertical wire length, 45-degree wire length, vias, routes, and wire segments.
Connectivity Summary	Lists connectivity statistics for a cellview based on signal types, such as signal, power, ground, and clock.
Routing Statistics	Displays routing statistics for the selected set based on layers. For each layer, the report lists total wire length, horizontal and vertical wire length, 45-degree wire length, guide length, and the number of down vias and guides.

Related Topics

Viewing Cellview Summary

Design Summary Form

Show Selected Set Window

© 2023 Cadence Design Systems, Inc.

Tap Form

Use the Tap form to access information from the following objects: paths, wires, rectangles, polygons, circles, ellipses, donuts, microwave objects, blockages, instances, vias, rows, labels and text displays.

Field	Description
Layer	Sets the tapped layer as the entry layer, and updates the *Layer* assistant. This selection includes paths, path segments, wires, rectangles, polygons, circles, ellipses, donuts, microwave objects, labels, and textDisplays.
Net	Captures net names from rectangles, polygons, and paths, wires, wire segments, labels, or text displays. You can capture the net names to automatically insert the net names when applying connectivity to a via.
Attributes	Captures attributes from paths, instances, vias, rows and microwave objects.
Pop up selection window	Displays the *Choose Object to Tap* form. When you click an object with multiple layers, the *Choose Object to Tap* form appears with a list of layers on which you can create the object.
Command (shapes, wires)	Starts the create command for the tapped shape or wire. The *Command (shapes/wires)* option works only for shapes and wires. It does not work for other object types such as instances, vias, and blockages. When you select this option, the following commands are started for each object type: ■ Rectangle: leHiCreateRect ■ Polygon: leHiCreatePolygon ■ Path: leHiCreatePath ■ Ellipse: leHiCreateEllipse ■ Donut: leHiCreateDonut ■ PathSeg: leHiCreateGeometricWire

© 2023 Cadence Design Systems, Inc.

Related Topics

Object Tapping

Quick Align Form

leHiLayerTap

© 2023 Cadence Design Systems, Inc.

Track Pattern Editor Form

Use the Track Pattern Editor form to add, update or delete track patterns.

For more information about creating track patterns for Virtuoso® Multi-Patterning Technology using the fields *Color Pattern*, *Name*, *Pattern Group Name*, and *Width*, see Track-based Coloring.

Field	Description
Filter by Routing Layer	Filters the routing layers for track patterns to be displayed in the form for the current design. ■ *all:* Updates the form to show all routing layers to which tracks patterns are assigned. ■ *layerName:* Updates the form to only show track patterns that are assigned to the specified routing layer. Toggle selection of track patterns using the left and right mouse buttons.
Delete	Deletes the selected track pattern.
Layer	Specifies the routing layer to associate with the new track pattern.
Direction	■ *Horizontal:* Defines horizontal tracks ■ *Vertical:* Defines vertical tracks
Spacing	Specifies the spacing between tracks. Also know as the routing pitch.
Origin Offset	Offsets the tracks up and to the right of the origin point by the specified distance.
Number of Tracks	Specifies the number of tracks (horizontal or vertical) associated with the selected track pattern.
Color Pattern	Specifies the color pattern for track patterns. The color pattern is used when there is more than one mask for a layer.
Name	Assigns the track pattern as a constraint to a constraint group that can then be assigned to a net. This causes the net to be routed on the specified tracks.

© 2023 Cadence Design Systems, Inc.

Field	Description
Pattern Group Name	Specifies a constraint group that contains the `allowWidthRanges` constraint to determine the width associated with the track.
Width	Specifies the width associated with the track.

Related Topics

Creating a Track Pattern

leHiCellviewTrackPatterns

© 2023 Cadence Design Systems, Inc.

Transmission Line Bend Form

Use the Transmission Line Bend form to create a two-segment transmission line with two different width settings.

Field	Description
Snap Mode	Controls the shape of the transmission line segments.
Bend Style	Sets the type of corner to use in the bend. You can choose from: *bend*, *chamfer*, or *radial*.
Side 1 Width and Side 2 Width	Sets the width of the two segments in user units (typically microns).
Bend Factor	Sets the maximum width allowed in a standard bend before the bend is automatically chamfered. (Available only when *Bend Style* is *bend*.)
Chamfer Factor	Sets the fraction of the corner that is removed in a chamfer bend. (Available only when *Bend Style* is *chamfer*.)
Radius Factor	Sets the ratio of the centerline bend width to the radial bend width. (Available only when *Bend Style* is *radius*.)
Resolution	Sets the number of segments used in a 180-degree radial bend. Environment variable: trlNPtsToPI

Related Topics

Creating Bends

leHiCreateBend

© 2023 Cadence Design Systems, Inc.

Transmission Line Form

Use the Transmission Line form to create a transmission line.

Field	Description
Mode	Controls how transmission line segments snap to the grid.
Bend Style	Sets the type of corner to use for the bends in the transmission line. You can choose from: *bend*, *chamfer*, or *radial*.
Width	Specifies the transmission line width in user units (typically microns).
Bend Factor	Sets the maximum width allowed in a standard bend before the bend is automatically chamfered. (Available only when *Bend Style* is *bend*.)
Chamfer Factor	Sets the fraction of the corner that is removed in a chamfer bend. (Available only when *Bend Style* is *chamfer*.)
Radius Factor	Sets the ratio of the centerline bend width to the radial bend width. (Available only when *Bend Style* is *radius*.)
Resolution	Sets the number of segments used in a 180-degree radial bend. Environment variable: trlNPtsToPI

Related Topics

Creating Transmission Lines

leHiCreateTrl

© 2023 Cadence Design Systems, Inc.

Transmission Line Taper Form

Use the Transmission Line Taper form to create a one-segment transmission line that connects two transmission lines of different width settings.

Field	Description
Snap Mode	Controls how the points of the taper snap to the grid.
Taper Style	Sets the taper style you create:  linear  exponential
Side 1 Width and Side 2 Width	Specifies the width of the taper ends in user units (typically microns).
Resolution	Sets the number of segments used in an exponentially tapered line. Environment variable: trlNPtsToPI

Related Topics

Creating Tapers

leHiCreateTaper

© 2023 Cadence Design Systems, Inc.

Transparency Options Form

Use the Transparency Options form to control the transparency of tracks, FinFET/WSP shapes, regions, and wire editor tracks.

Field	Description
All Linked	Enables the outline and fill values of *Tracks*, *FinFET/WSP Shapes*, *Regions*, and *Wire Editor Tracks* to be changed together by tuning only a single slider at a time.
Transparency Enabled	Displays transparency options as active.
Tracks	Enables you to set transparency options for tracks by tuning the slider. ■ *Linked*: If selected, enables you to change the outline and fill values of tracks by tuning only a single slider at a time. If not selected, you need to set the outline and fill values individually. ■ *Outline*: Sets transparency for the outline of tracks. ■ *Fill*: Sets transparency for the file of tracks.
FinFET/WSP Shapes	Enables you to set transparency options for FinFET/WSP shapes by tuning the slider. ■ *Linked*: If selected, enables you to change the outline and fill values of FinFET/WSP shapes by tuning only a single slider at a time. If not selected, you need to set the outline and fill values individually. ■ *Outline*: Sets transparency for the outline of FinFET/WSP shapes. ■ *Fill*: Sets transparency for the file of FinFET/WSP shapes.

© 2023 Cadence Design Systems, Inc.

Field	Description
Regions	Enables you to set transparency options for regions by tuning the slider. ■ *Linked*: If selected, enables you to change the outline and fill values of regions by tuning only a single slider at a time. If not selected, you need to set the outline and fill values individually. ■ *Outline*: Sets transparency for the outline of regions. ■ *Fill*: Sets transparency for the file of regions.
Wire Editor Tracks	Enables you to set transparency options for wire editor tracks by tuning the slider. ■ *Linked*: If selected, enables you to change the outline and fill values of wire editor tracks by tuning only a single slider at a time. If not selected, you need to set the outline and fill values individually. ■ *Outline*: Sets transparency for the outline of wire editor tracks. ■ *Fill*: Sets transparency for the file of wire editor tracks.

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

Tree Form

Use the Tree form to set the levels of instances hierarchy to be listed for the current cellview. The following table describes the fields available in the Display section of the form.

Field	Description
Top to stop level	Lists the entire hierarchy of instances in the cellview, from the top level to the level specified in the *Display Levels: Stop* field on the Display Options form.
Current to stop level	Lists the hierarchy of instances from the current level of hierarchy to the level specified in the *Display Levels: Stop* field on the Display Options form.
Top to current	Lists the hierarchy of instances from the top level to the current level of hierarchy.
Current to user level	Lists the hierarchy of instances from the current level of hierarchy to the level that you specified in the user level spin box.

Related Topics

Making Hierarchy Editable

leHiTree

© 2023 Cadence Design Systems, Inc.

Uniquify Form

Use the Uniquify form to specify the library, cell, and view name of the cellview to be uniquified.

Field	Description
Library	Lets you specify the name of the library that contains the cellviews you want to uniquify.
Cell	Lets you specify the name of the cell that contains the view to be uniquified.
View	Lets you specify the name of the view to be uniquified.

Related Topics

Uniquifying a Cellview

leUniquifyCellView

© 2023 Cadence Design Systems, Inc.

Unmark Net Form

Use the Unmark Net form to unmark a traced net.

Field	Description
Unmark All	Removes the highlighting from all marked nets. If you want to delete the highlighting of a single marked net, click that net.

Related Topics

Unmarking Nets

leUnmarkNet

© 2023 Cadence Design Systems, Inc.

View By Form

Use the View By form to specify multiple parameters for displaying layer-purpose pairs in a tree structure in the *Layers* panel.

Field	Description
View items by	Enables you to select the top-level column based on which layer-purpose pairs are grouped. You can specify the display order of layer-purpose pairs by choosing *Ascending* or *Descending* in the *View items by section*. By default, layer-purpose pairs are displayed in ascending order.
Then by	Specifies the next level grouping column based on which layer-purpose pairs are displayed. There are three *Then by* sections in the View By form using which you can define the second, third, and fourth level of grouping columns. For example, you can choose to view layer-purpose pairs based on four columns in the following order: Layer, Purpose, Priority, and Color. You can specify the display order of layer-purpose pairs by choosing *Ascending* or *Descending* in each *Then by* section. By default, layer-purpose pairs are displayed in ascending order.

Related Topics

Viewing Layer-Purpose Pairs in a Tree Structure

© 2023 Cadence Design Systems, Inc.

Yank Form

Use the Yank form to copy objects, and parts of objects, into a temporary buffer.

Field	Description
Yank Shape	Sets the type of shape you draw around, or through, the objects you want to copy.
Yank Levels	Sets the number of levels of design hierarchy through which *Yank* can copy shapes. The current cellview is level 0, instances inside it are level 1, and so forth.
Snap Mode	Controls the shape of line segments when *Yank Shape* is set to *polygon*.

Related Topics

Object Yanking and Pasting

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

B

Layout XL Basic Editing Environment Variables

This appendix provides information on the names, descriptions, and values of the layout editor and graphics editor environment variable used in Virtuoso® Layout Suite.

Graphic environment variables control characteristics of window display, and layout environment variables control how various layout editor commands work. Many graphic environment variables have duplicate layout environment variables. In these cases, a layout environment variable supersedes a graphic variable unless the graphic variable is stored in the cellview. You can set both types of environment variables.

Only the environment variables documented are supported for public use. All other environment variables, regardless of their name or prefix, and undocumented aspects of any environment variables, are private and are subject to change at any time.

Related Topics

System Evaluation of Environment Variables

Ways to Set Environment Variables

System Evaluation of Environment Variables

The system looks for the value of an environment variable as follows:

- If no layout license is available, the system looks for the system-defined default settings in the following file:

  install_dir/etc/tools/graphic/.cdsenv

- If a layout license is available, the system looks for the system-defined default settings in the following file:

  install_dir/etc/tools/layout/.cdsenv

© 2023 Cadence Design Systems, Inc.

- The system loads any values in the local `.cdsenv` file in your home directory,

  ~/.cdsenv

  The layout variables supersede any matching graphic variables. Environment variables defined in your `.cdsenv` supersede any default values in the graphic or layout `.cdsenv`.

Note: There is a precedence of window settings that might override environment variable settings. The settings are read in the following order: cellview properties, library properties, technology library properties, and environment variable values. Therefore, any override window properties stamped on a cellview will take precedence over a setting in the `.cdsenv` file.

Related Topics

Ways to Set Environment Variables

Specifying Environment Settings

Ways to Set Environment Variables

You can set graphic and layout environment variables.

- Ways to Set Graphic Environment Variables
- Ways to Set Layout Editor Environment Variables

Ways to Set Graphic Environment Variables

Graphic environment variables control window display. These variables can be stored in the cellview or the `.cdsenv` file. You can use any of the following methods to change default settings of graphic environment variables:

- Set the variable in the Display Options form.

  If you want to retain these settings for future sessions, save the settings using the *CIW – Options – Save Defaults* command.

- Copy the variables that you want to change from the default `.cdsenv` file to the `.cdsenv` file in your home directory.

  It is there that you edit the default value. The default file is:

  your_install_dir`/tools/dfII/samples/.cdsenv`

  Note: Do not use the `.cdsenv` system files located in the following directory:

© 2023 Cadence Design Systems, Inc.

your_install_dir/tools/dfII/etc/tools/*/.cdsenv. These files contain system default values for all applications and have extra arguments that do not work in your environment. Also, do not copy lines from the graphic/.cdsenv or layout/.cdsenv default files to the .cdsenv file in your home directory. The file formats are slightly different, so the lines that result from copying do not work.

- Set the variable in CIW using the following command:

  ```
  envSetVal("graphic" "name" type value)
  ```

  For example:

  ```
  envSetVal("graphic" "iconsOn" 'boolean t)
  ```

  To determine the current value of a graphic environment variable, type the following in CIW:

  ```
  envGetVal("graphic" "name")
  ```

  For example:

  ```
  envGetVal("graphic" "iconsOn")
  ```

Ways to Set Layout Editor Environment Variables

Layout environment variables control how various layout editor commands work. You can use any of the following ways to change default settings of layout environment variables:

- Set the variable in the Layout Editor Options form.
  If you want to retain the updated settings for future sessions, save the settings using the *CIW – Options – Save Defaults* command.
- Include the environment variables in your .cdsenv file in your home directory. The default file is:

  your_install_dir/tools/dfII/samples/.cdsenv
- Include the envSetVal() function in your .cdsinit file.
- Include the envSetVal() function in any other SKILL file you load.
- Use the envSetVal() function in CIW. For example, to set the gravityOn variable, type the following command in CIW or include it in a file:

  ```
  envSetVal("layout" "gravityOn" 't)
  ```

  To determine the current value of any layout editor environment variable, type the envGetVal() function in CIW as follows:

  ```
  envGetVal("layout" "variable_name")
  ```

  For example:

  ```
  envGetVal("layout" "gravityOn")
  ```


© 2023 Cadence Design Systems, Inc.

If you set an environment variable by using the `envSetVal()` function in CIW prior to opening a cellview window, the next window opened in the session will reflect the change. Any environment variable reset with this method takes effect after any open cellview window is closed and reopened, and in all subsequent windows that are opened in the current session. You must reset the environment variable in the current session to change it. Changes are effective for the current session only.

Note: There are a number of graphic and layout environment variables that are no longer compatible with `leSetEnv` and `leGetEnv`. Cadence recommends that you use only `envSetVal` and `envGetVal` to set, reset, or determine the setting of environment variables.

Related Topics

Display Options Form

Layout Editor Options Form

Specifying Environment Settings

Changing Graphic Environment Variables That Are Cellview Properties

Some graphic environment variables are cellview properties, and you can make the changes made to such environment variables to be visible in the cellview window immediately.

To change a graphic environment variable setting and view the changes instantly:

➡ Type the following in CIW:

```
window = hiGetCurrentWindow()
window~>environmentVariable = value
```

For example:

```
window = hiGetCurrentWindow()
window~>displayStretchHandles = t
```

This changes the environment variable for the current window only, and you must reset it if you open new windows. The change is effective for the current session only.

Related Topics

Specifying Environment Settings

© 2023 Cadence Design Systems, Inc.

abeFixShapePartitioning

```
layout abeFixShapePartitioning boolean { t | nil }
```

Description

Enables proper functioning of non-rectangle tiles. It adds the non-rectangle tiles to the output layers without changing them to partially overlapped shapes. This environment variable works only with shapes, not with point arrays.

The default is `nil`.

Set this environment variable before initializing an ABE session for a cellview.

GUI Equivalent

None

Examples

```
envGetVal("layout" "abeFixShapePartitioning")
envSetVal("layout" "abeFixShapePartitioning" 'boolean t)
```

Related Topics

Advanced Boolean Engine Functions

© 2023 Cadence Design Systems, Inc.

abutServerOn

```
layout abutServerOn boolean { t | nil }
```

Description

Turns automatic abutment and automatic unabutment on and off. Unabutment automatically restores abutted devices when moved apart. This ensures that dropped vias are restored when chained devices are moved apart. It also eliminates errors such as devices with no diffusion overlap on one side.

The default is `t`. When set to `nil`, automatic abutment and unabutment are turned off in layout XL, and unabutment retains the abutted device configuration even when the device is unabutted in the layout editor.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Abut Server*

Examples

```
envGetVal("layout" "abutServerOn")
envSetVal("layout" "abutServerOn" 'boolean nil)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

accessEdgesOn

```
layout accessEdgesOn boolean { t | nil }
graphic accessEdgesOn boolean { t | nil }
```

Description

Checks if the access edges of rectangular pins are displayed.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Access Edges*

Examples

```
envGetVal("layout" "accessEdgesOn")
envSetVal("layout" "accessEdgesOn" 'boolean t)
envGetVal("graphic" "accessEdgesOn")
envSetVal("graphic" "accessEdgesOn" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

acuteAnglePath

```
layout acuteAnglePath boolean { t | nil }
```

Description

Allows creation of acute angle paths.

The default is `t`. When set to `nil`, it allows creation of paths from `90` degree to `270` degree angles. You must set *Snap Mode* to *anyAngle* for this environment variable to take effect.

GUI Equivalent

Command:	*Create – Path*
Field:	*Acute Angle*

Examples

```
envGetVal("layout" "acuteAnglePath")
envSetVal("layout" "acuteAnglePath" 'boolean nil)
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

addDesignLppsToTech

```
graphic addDesignLppsToTech boolean { t | nil }
```

Description

Creates a layer-purpose pair (LPP) in the technology file when a shape is present in the design but its layer number and purpose number do not correspond to an existing LPP in the technology file. A default packet is used to display shapes on this new LPP.

The default is `nil`.

If layer and purpose names associated with either the layer or purpose numbers, or both are not defined in the technology file attached to the sub cellview of the hierarchy, then:

- Name of the new layer will be `L`*`XXX`*, where *`XXX`* is the layer number.
- Name of the new purpose will be `P`*`YYY`*, where *`YYY`* is the purpose number.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "addDesignLppsToTech")
envSetVal("graphic" "addDesignLppsToTech" 'boolean t)
```

Related Topics

Instance Snapping

Checking Layout Integrity

© 2023 Cadence Design Systems, Inc.

alignApplySpacing

```
layout alignApplySpacing boolean { t | nil }
```

Description

Sets the toggle button for the *Apply Spacing* field on the *Edit – Advanced – Align* form to on or off.

The default is `nil`. When set to `t`, it enables the data entry area.

To set the value for the data entry area, use the `alignSpacingValue` environment variable.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Apply Spacing*

Examples

```
envGetVal("layout" "alignApplySpacing")
envSetVal("layout" "alignApplySpacing" 'boolean t)
```

Related Topics

alignSpacingValue

Align Form

© 2023 Cadence Design Systems, Inc.

alignDirection

```
layout alignDirection string { "horizontal" | "vertical" }
```

Description

Sets the *Horizontal* and *Vertical* radio buttons for the *Align Direction* field in the *Reference* section of the *Edit – Advanced – Align* form.

The default is `vertical`.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Align Direction*

Examples

```
envGetVal("layout" "alignDirection")
envSetVal("layout" "alignDirection" 'string "horizontal")
```

Related Topics

Align Form

© 2023 Cadence Design Systems, Inc.

alignLayer

```
layout alignLayer string "layerPurpose"
```

Description

Lets you set up the default layer-purpose pair for the *Layer* list in the *Reference* and *Align Using* sections of the *Edit – Advanced – Align* form.

By default, no layer purpose is set up.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Layer*

Examples

```
envGetVal("layout" "alignLayer")
envSetVal("layout" "alignLayer" 'string "Metal1 drawing")
```

Related Topics

Align Form

© 2023 Cadence Design Systems, Inc.

alignMatchRefLayer

```
layout alignMatchRefLayer boolean { t | nil }
```

Description

Lets you change the target pin layer purpose of the pin to the reference pin layer purpose.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Match layer/purpose with the reference pin*

Examples

```
envGetVal("layout" "alignMatchRefLayer")
envSetVal("layout" "alignMatchRefLayer" 'boolean t)
```

Related Topics

Align Form

© 2023 Cadence Design Systems, Inc.

alignMatchRefSize

```
layout alignMatchRefSize boolean { t | nil }
```

Description

Lets you change the size of the target pin to that of the reference pin.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Match size with the reference pin*

Examples

```
envGetVal("layout" "alignMatchRefSize")
envSetVal("layout" "alignMatchRefSize" 'boolean t)
```

Related Topics

Align Form

© 2023 Cadence Design Systems, Inc.

alignMethod

```
layout alignMethod cyclic { "top" | "bottom" | "left" | "right" | "center" |
     "origin" | "upperLeft" | "upperRight" | "upperCenter" | "lowerLeft" |
     "lowerRight" | "lowerCenter" | "centerLeft" | "centerRight" | "centerCenter" }
```

Description

Sets the values in the cyclic *Use* field for *Object* or *Layer* in the *Align Using* section on the *Edit – Advanced – Align* form, depending on how the *Reference* and *Align Direction* fields are set, as shown in the table below.

Reference	Align Direction	Align Using	Use list contains	Default
Point	*Horizontal* or *Vertical*	*Object*	`Upper Left`, `Upper Center`, `Upper Right`; `Center Left`, `Center Center`, `Center Right`; `Lower Left`, `Lower Center`, `Lower Right`; `Origin`	`Lower Left`
		Layer	`Upper Left`, `Upper Center`, `Upper Right`; `Center Left`, `Center Center`, `Center Right`; `Lower Left`, `Lower Center`, `Lower Right`	`Lower Left`
Object	*Horizontal*	*Object*	`Top Edge`, `Bottom Edge`, `Center`; `Origin`	`Center`
		Layer	`Top Edge`, `Bottom Edge`, `Center`	`Center`
	Vertical	*Object*	`Left Edge`, `Right Edge`, `Center`; `Origin`	`Center`
		Layer	`Left Edge`, `Right Edge`, `Center`	`Center`
Layer	*Horizontal*	*Object*	`Top Edge`, `Bottom Edge`, `Center`; `Origin`	`Center`
Layer	*Vertical*	*Object*	`Left Edge`, `Right Edge`, `Center`; `Origin`	`Center`
	Horizontal	*Layer*	`Top Edge`, `Bottom Edge`, `Center`	`Center`

© 2023 Cadence Design Systems, Inc.

Reference	Align Direction	Align Using	Use list contains	Default
	Vertical	*Layer*	`Left Edge`, `Right Edge`, `Center`	`Center`

You can set the *Align Using* field with the `alignUsing` environment variable and the *Reference* field with the `alignReference` environment variable.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Align Using – Object – Use* *Align Using – Layer – Use*

Examples

```
envGetVal("layout" "alignMethod")
envSetVal("layout" "alignMethod" 'cyclic "top")
envSetVal("layout" "alignMethod" 'cyclic "centerLeft")
```

Related Topics

alignUsing

alignReference

Align Form

© 2023 Cadence Design Systems, Inc.

alignObjType

```
layout alignObjType cyclic { "instance" | "row" | "cluster boundary" | "any shape"
     | "pins" }
```

Description

Sets the object type to *Instance*, *Any Shape, Row*, *Cluster Boundary*, *Group*, or for Layout GXL, *Pins* when the *Align Using* field on the *Edit – Advanced – Align* form is set to *Object*.

The default is `instance`. The valid values are: `instance`, `row`, `cluster boundary`, `group`, and `any shape` for the floor planning mode in Layout GXL pins.

You can set the *Align Using* field with the `alignUsing` environment variable.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Align Using – Object*

Examples

```
envGetVal("layout" "alignObjType")
envSetVal("layout" "alignObjType" 'cyclic "cluster boundary")
envSetVal("layout" "alignObjType" 'cyclic "any shape")
```

Related Topics

alignUsing

Align Form

© 2023 Cadence Design Systems, Inc.

alignReference

```
layout alignReference string { "point" | "object" | "layer" }
```

Description

Sets the *Reference* field on the *Edit – Advanced – Align* form to *Point*, *Object*, or *Layer*. The valid values are `point`, `object`, or `layer`. For object, instances, rows, cluster boundaries, and groups are supported in layout editor; additionally, pins are supported in Layout EXL and higher tiers.

The default is `point`.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Reference*

Examples

```
envGetVal("layout" "alignReference")
envSetVal("layout" "alignReference" 'string "object")
envSetVal("layout" "alignReference" 'string "layer")
```

Related Topics

Align Form

© 2023 Cadence Design Systems, Inc.

alignRefMethod

```
layout alignRefMethod cyclic { "center" | "top" | "bottom" | "left" | "right" |
     "origin" }
```

Description

Sets the values in the cyclic *Use* field when the *Reference* field on the *Edit – Advanced – Align* form is set to *Object* or *Layer*, based on whether *Align Direction* is set to *Horizontal* or *Vertical*, as shown in the table below.

Reference	Align Direction	Use list contains	Default
Object	*Horizontal*	`Top Edge`, `Bottom Edge`, `Center`, `Origin`	`Center`
	Vertical	`Left Edge`, `Right Edge`, `Center`, `Origin`	`Center`
Layer	*Horizontal*	`Top Edge`, `Bottom Edge`, `Center`, `Origin`	`Center`
	Vertical	`Left Edge`, `Right Edge`, `Center`, `Origin`	`Center`

The valid values are: `top`, `bottom`, `center`, `left`, `right`, and `origin`. You can set the *Reference* field with the `alignReference` environment variable, and the *Align Direction* field with the `alignDirection` environment variable.

The default is `center`.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Reference – Object – Use* *Reference – Layer – Use*

Examples

```
envGetVal("layout" "alignRefMethod")
envSetVal("layout" "alignRefMethod" 'cyclic "right")
```


© 2023 Cadence Design Systems, Inc.

Related Topics

alignReference

alignDirection

Align Form

© 2023 Cadence Design Systems, Inc.

alignRefObjType

```
layout alignRefObjType cyclic { "instance" | "row" | "cluster boundary" | "group"
     | "pins"}
```

Description

Sets the object type to *Instance*, *Row*, *Cluster Boundary*, or *Group* when the *Reference* field on the *Edit – Advanced – Align* form is set to *Object*.

The default is `instance`. The valid values are: `instance`, `row`, `cluster boundary`, `group`, and when in floor planning mode in Layout GXL, `pins`. You can set the *Reference* field with the `alignReference` environment variable.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Reference – Object*

Examples

```
envGetVal("layout" "alignRefObjType")
envSetVal("layout" "alignRefObjType" 'cyclic "cluster boundary")
```

Related Topics

alignReference

Align Form

© 2023 Cadence Design Systems, Inc.

alignReverseSort

```
layout alignReverseSort boolean { t | nil }
```

Description

Sets the toggle button for the *Reverse Sort* field on the *Edit – Advanced – Align* form to on or off

The default is `nil`.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Reverse Sort*

Examples

```
envGetVal("layout" "alignReverseSort")
envSetVal("layout" "alignReverseSort" 'boolean t)
```

Related Topics

Align Form

© 2023 Cadence Design Systems, Inc.

alignSnapToGridType

```
layout alignSnapToGridType cyclic { "None" | "XYSnap" | "MfgGrid" }
```

Description

Snaps objects to XY snap spacing, manufacturing grid, or none. The valid values are: `None`, `XYSnap`, `MfgGrid`.

The default is `XYSnap`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "alignSnapToGridType")
envSetVal("layout" "alignSnapToGridType" 'cyclic "None")
envSetVal("layout" "alignSnapToGridType" 'cyclic "MfgGrid")
```

Related Topics

Setting Up Grid Controls

Align Form

© 2023 Cadence Design Systems, Inc.

alignSortMethod

```
layout alignSortMethod cyclic { "alignDirection" | "horizontal" | "vertical" |
     "orderSelected" | "instanceName" | "netName"}
```

Description

Sets the sort order for the *Sort Objects by* field on the *Edit – Advanced – Align* form.

- `alignDirection`: Default. Sorts objects by the horizontal or vertical alignment direction selected in the Reference section of the Align form.
- `horizontal`: Sorts objects aligned horizontally, retaining their original horizontal order.
- `vertical`: Sorts objects aligned vertically, retaining their original vertical order.
- `orderSelected`: Sorts objects in the order they were selected.
- `instanceName`: Sorts objects alphanumerically by instance name.
- `netName`: Sorts objects alphanumerically by net name.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Sort Objects By*

Examples

```
envGetVal("layout" "alignSortMethod")
envSetVal("layout" "alignSortMethod" 'cyclic "horizontal")
envSetVal("layout" "alignSortMethod" 'cyclic "netName")
```

Related Topics

Align Form

alignReference

© 2023 Cadence Design Systems, Inc.

alignSpacingType

```
layout alignSpacingType string { "spacings" | "pitches" }
```

Description

Sets the *Between Objects and Pitches* radio buttons on the *Edit – Advanced – Align* form, when the *Reference* field is set to *Point*, and the *Apply Spacing* field is turned on. The valid values are `spacings` (for the *Between Objects* radio button) or `pitches`.

The default is `spacings`. You can set the *Apply Spacing* field with the `alignApplySpacing` environment variable.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Apply Spacing*

Examples

```
envGetVal("layout" "alignSpacingType")
envSetVal("layout" "alignSpacingType" 'string "pitches")
```

Related Topics

alignApplySpacing

Align Form

© 2023 Cadence Design Systems, Inc.

alignSpacingValue

```
layout alignSpacingValue float spacingValue
```

Description

Sets the value when the *Apply Spacing* field on the *Edit – Advanced – Align* form is set to on. The value must be a integer point number. You can set the *Apply Spacing* field with the `alignApplySpacing` environment variable.

The default is `1.0`.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Apply Spacing*

Examples

```
envGetVal("layout" "alignSpacingValue")
envSetVal("layout" "alignSpacingValue" 'float 1.0)
```

Related Topics

alignApplySpacing

Align Form

© 2023 Cadence Design Systems, Inc.

alignToolBarReferenceObject

```
layout alignToolBarReferenceObject cyclic { "left" | "top" | "bottom" | "right" |
     "vertical" | "horizontal" | "click" }
```

Description

Selects the reference object from the selection set when using the *Align* toolbar.

- `left`: Default. The leftmost object from the selection set is used as the reference object
- `right`: The rightmost object from the selection set is used as the reference object
- `top`: The topmost object from the selection set is used as the reference object
- `bottom`: The bottommost object from the selection set is used as the reference object
- `horizontal`: The object closest to the horizontal center of the selection set bounding box is used as the reference object
- `vertical`: The object closest to the vertical center of the selection set bounding box is used as the reference object
- `click`: The clicked object is used as the reference object

GUI Equivalent

None

Examples

```
envGetVal("layout" "alignToolBarReferenceObject")
envSetVal("layout" "alignToolBarReferenceObject" 'cyclic "top")
envSetVal("layout" "alignToolBarReferenceObject" 'cyclic "click")
```

Related Topics

Aligning Objects by Using the Align Toolbar

Align Form

© 2023 Cadence Design Systems, Inc.

alignUpdateLabelLayer

```
layout alignUpdateLabelLayer boolean { t | nil }
```

Description

Updates the layer of labels to that of the pin. This option can be used when the layers of the pin and label are different.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Update pin label layer*

Examples

```
envGetVal("layout" "alignUpdateLabelLayer")
envSetVal("layout" "alignUpdateLabelLayer" 'boolean t)
```

Related Topics

Align Form

© 2023 Cadence Design Systems, Inc.

alignUsing

```
layout alignUsing string { "object" | "layer" }
```

Description

Sets the *Align Using* field on the *Edit – Advanced – Align* form to *Object* or *Layer*. The valid values are `object` or `layer`.

The default is `object`.

- `object`: The target objects of the type specified by the cyclic field will align to the reference point, object, or layer specified in the *Reference* section of the form.
- `layer`: The target objects on the selected layer will align with the reference point, object, or layer specified in the *Reference* section of the form.

You can set the *Reference* field with the `alignReference` environment variable, the *Align Using* field with the `alignObjType` environment variable, and the *cyclic* field for object type with the `alignObjType` environment variable.

GUI Equivalent

Command:	*Edit – Advanced – Align*
Field:	*Align Using*

Examples

```
envGetVal("layout" "alignUsing")
envSetVal("layout" "alignUsing" 'string "layer")
```

Related Topics

alignReference

alignObjType

Align Form

© 2023 Cadence Design Systems, Inc.

alignUsingInstanceDrawingMode

```
layout alignUsingInstanceDrawingMode boolean { t | nil }
```

Description

Aligns the instances based on how they are rendered on the canvas. Instances can be rendered on the canvas as bounding box, true bounding box, PR boundary, or snap boundary.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "alignUsingInstanceDrawingMode")
envSetVal("layout" "alignUsingInstanceDrawingMode" 'boolean t)
```

Related Topics

Align Form

© 2023 Cadence Design Systems, Inc.

allowPartialDelete

```
layout allowPartialDelete boolean { t | nil }
```

Description

Specifies if deletion of partially selected objects is allowed.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "allowPartialDelete")
envSetVal("layout" "allowPartialDelete" 'boolean nil)
```

Related Topics

Object Deletion

© 2023 Cadence Design Systems, Inc.

alphaBlendingBoundaryFillValue

```
graphic alphaBlendingBoundaryFillValue int FillValue
```

Description

Sets the fill value for the alpha blending boundary.

The default is `1000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "alphaBlendingBoundaryFillValue")
envSetVal("graphic" "alphaBlendingBoundaryFillValue" 'int 100)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

alphaBlendingBoundaryHaloValue

```
graphic alphaBlendingBoundaryHaloValue int haloValue
```

Description

Sets the halo value for the alpha blending boundary.

The default is `1000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "alphaBlendingBoundaryHaloValue")
envSetVal("graphic" "alphaBlendingBoundaryHaloValue" 'int 100)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

alphaBlendingBoundaryOutLineValue

```
graphic alphaBlendingBoundaryOutLineValue int OutlineValue
```

Description

Sets the outline value for the alpha blending boundary.

The default is `1000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "alphaBlendingBoundaryOutLineValue")
envSetVal("graphic" "alphaBlendingBoundaryOutLineValue" 'int 100)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

alphaBlendingFinFillValue

```
graphic alphaBlendingFinFillValue int fillValue
```

Description

Sets the fill value for the alpha blending fin.

The default is `1000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "alphaBlendingFinFillValue")
envSetVal("graphic" "alphaBlendingFinFillValue" 'int 100)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

alphaBlendingFinHaloValue

```
graphic alphaBlendingFinHaloValue int haloValue
```

Description

Sets the halo value for the alpha blending fin.

The default is `1000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "alphaBlendingFinHaloValue")
envSetVal("graphic" "alphaBlendingFinHaloValue" 'int 100)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

alphaBlendingFinOutLineValue

```
graphic alphaBlendingFinOutLineValue int outlineValue
```

Description

Sets the outline value for the alpha blending fin.

The default is `1000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "alphaBlendingFinOutLineValue")
envSetVal("graphic" "alphaBlendingFinOutLineValue" 'int 100)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

alphaBlendingTrackFillValue

```
graphic alphaBlendingTrackFillValue int fillValue
```

Description

Sets the fill value for the alpha blending track.

The default is `1000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "alphaBlendingTrackFillValue")
envSetVal("graphic" "alphaBlendingTrackFillValue" 'int 100)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

alphaBlendingTrackHaloValue

```
graphic alphaBlendingTrackHaloValue int haloValue
```

Description

Sets the halo value for the alpha blending track.

The default is `1000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "alphaBlendingTrackHaloValue")
envSetVal("graphic" "alphaBlendingTrackHaloValue" 'int 100)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

alphaBlendingTrackOutLineValue

```
graphic alphaBlendingTrackOutLineValue int outlineValue
```

Description

Sets the outline value for the alpha blending track.

The default is `1000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "alphaBlendingTrackOutLineValue")
envSetVal("graphic" "alphaBlendingTrackOutLineValue" 'int 100)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

anchor

```
layout anchor cyclic { "unknown" | "left" | "right" }
```

Description

Adjusts the cellview placement of a chain of instances after the *Subtract Finger* or *Add Finger* command is used.

The default is `left`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "anchor")
envSetVal("layout" "anchor" 'cyclic "unknown")
envSetVal("layout" "anchor" 'cyclic "right")
```

Related Topics

Global Grid Anchoring

© 2023 Cadence Design Systems, Inc.

applyBounce

```
layout applyBounce boolean { t | nil }
```

Description

Specifies if gravity offset bounce values are used.

The default is `nil`. When set to `t`, `gravityOn` is automatically set to `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "applyBounce")
envSetVal("layout" "applyBounce" 'boolean t)
```

Related Topics

gravityOn

© 2023 Cadence Design Systems, Inc.

arrayDisplay

```
layout arrayDisplay string { "full" | "border" | "source" }
graphic arrayDisplay string { "full" | "border" | "source" }
```

Description

Determines how arrays are displayed.

The default is `full`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Array Display*

Examples

```
envGetVal("layout" "arrayDisplay")
envSetVal("layout" "arrayDisplay" 'string "border")
envGetVal("graphic" "arrayDisplay")
envSetVal("graphic" "arrayDisplay" 'string "source")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

askSaveOnReturn

```
graphic askSaveOnReturn boolean { t | nil }
```

Description

Displays a dialog box after the *Edit In Place* and *Descend Edit* commands are enabled and changes have been made in the *Edit* mode. The dialog box asks if you want to save your changes to a cellview.

The default is `t`. When set to `nil`, the dialog box is not displayed and the cellview stays open in edit mode.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "askSaveOnReturn")
envSetVal("graphic" "askSaveOnReturn" 'boolean nil)
```

Related Topics

Hierarchical Designs

© 2023 Cadence Design Systems, Inc.

autoCopyTemplateCellViewList

```
layout autoCopyTemplateCellViewList string "any_cellview"
```

Description

Copies the WSP and row region templates from the specified cellview to the new cellview being opened in the layout window. The copy operation is not performed when opening an existing layout or creating the cellview using database SKILL functions.

The default is `nil`. In the example below, when you open a new cellview, the WSPs and row region specifications from lib1:cell1:layout will be copied to the cellview being created.

GUI Equivalent

None

Examples

```
envGetVal("layout" "autoCopyTemplateCellViewList")
envSetVal("layout" "autoCopyTemplateCellViewList" 'string "(lib1 cell1 layout)")
```

Related Topics

Object Copying

© 2023 Cadence Design Systems, Inc.

autoCircleCreation

```
layout autoCircleCreation boolean { t | nil }
```

Description

Sets the *Mode* field to *Click To Place* when set to `t`, and to *Draw* when set to `nil`.

The default is `nil`.

GUI Equivalent

Command:	*Create – Shape – Circle*
Field:	*Mode*

Examples

```
envGetVal("layout" "autoCircleCreation")
envSetVal("layout" "autoCircleCreation" 'boolean t)
```

Related Topics

Create Circle Form

© 2023 Cadence Design Systems, Inc.

autoExtractSaveEnabled

```
layout autoExtractSaveEnabled boolean { t | nil }
```

Description

Saves the hierarchical cellviews extracted by the *Extract Layout* command automatically when the extraction scope is set to *Current Cellview* and *Cellviews in Hierarchy*.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Update – Extract Layout*
Field:	*Save Extracted Cellviews*

Related Topics

Hierarchical Extraction

© 2023 Cadence Design Systems, Inc.

autoLoadBindKeys

```
layout autoLoadBindKeys boolean { t | nil }
```

Description

Loads the `Layout.il` file automatically when you run `virtuoso`. The `Layout.il` file is searched for in the following locations:

- *<install_dir>*`/share/cdssetup/dfII/bindkeys/Layout.il`
- `~/.cadence/dfII/bindkeys/`*<customBindKey>*`.il`
- `./.cadence/dfII/bindkeys/`*<customBindKey>*`.il`

The `Layout.il` file, if found, is loaded from each location in the order specified above. Each `Layout.il` file is appended to the one that was loaded last. If the `Layout.il` files contain any overlapping bindkey settings, the bindkey settings defined in the `Layout.il` file loaded last take effect.

The default is `t`. When set to `nil`, the `Layout.il` files are not automatically loaded from the above specified paths.

GUI Equivalent

None

Examples

```
envGetVal("layout" "autoLoadBindKeys")
envSetVal("layout" "autoLoadBindKeys" 'boolean nil)
```

Related Topics

Bindkeys in Virtuoso Layout Suite

© 2023 Cadence Design Systems, Inc.

autoModePinSize

```
layout autoModePinSize boolean { t | nil }
```

Description

Specifies whether pin size needs to be set for the rectangular pin created in auto mode.

The default is `nil`.

GUI Equivalent

Command:	*Create – Pin – Auto*
Field:	*Pin Size*

Examples

```
envGetVal("layout" "autoModePinSize")
envSetVal("layout" "autoModePinSize" 'boolean t)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

autoModePinSizeFromPath

```
layout autoModePinSizeFromPath boolean { t | nil }
```

Description

Specifies that the shape of the pin is the same as that of underlying path or pathSeg.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "autoModePinSizeFromPath")
envSetVal("layout" "autoModePinSizeFromPath" 'boolean t)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

autoNetName

```
layout autoNetName boolean { t | nil }
```

Description

Determines whether shapes are added to nets attached to pins.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Nets – Add Shape*
Field:	*Auto*

Examples

```
envGetVal("layout" "autoNetName")
envSetVal("layout" "autoNetName" 'boolean nil)
```

Related Topics

Add Shape to Net Form

© 2023 Cadence Design Systems, Inc.

autoPermutePins

```
layout autoPermutePins boolean { t | nil }
```

Description

Specifies whether pins are permuted automatically during manual routing or editing.

The default is `t`.



Important

Although `autoPermutePins` is defined as a layout environment variable, it has no effect in layout editor. It is used only by XL.

GUI Equivalent

Command:	*Options – Connectivity – Generation* tab
Field:	*Permute pins*

Examples

```
envGetVal("layout" "autoPermutePins")
envSetVal("layout" "autoPermutePins" 'boolean nil)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

autoPinLayerFunctionsList

```
layout autoPinLayerFunctionsList string "list_of_layers"
```

Description

Specifies the list of layers for auto pin.

The default is an empty list.

GUI Equivalent

None

Examples

```
envGetVal("layout" "autoPinLayerFunctionsList")
envSetVal("layout" "autoPinLayerFunctionsList" 'string "metal1 metal2")
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

autoPinRectHeight

```
layout autoPinRectHeight float height
```

Description

Specifies the height of the rectangle.

The default is `0.0`.

GUI Equivalent

Command:	*Create – Pin – Auto – Pin Size*
Field:	*Height*

Examples

```
envGetVal("layout" "autoPinRectHeight")
envSetVal("layout" "autoPinRectHeight" 'float 1.0)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

autoPinRectWidth

```
layout autoPinRectWidth float width
```

Description

Specifies the width of the rectangle.

The default is `0.0`.

GUI Equivalent

Command:	*Create – Pin – Auto – Pin Size*
Field:	*Width*

Examples

```
envGetVal("layout" "autoPinRectWidth")
envSetVal("layout" "autoPinRectWidth" 'float 1.0)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

autoRefresh

```
layout autoRefresh boolean { t | nil }
```

Description

Specifies whether automatic checking of cellview needs refresh when *Edit In Place* or *Descend* commands are turned on.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "autoRefresh")
envSetVal("layout" "autoRefresh" 'boolean t)
```

Related Topics

Hierarchical Designs

© 2023 Cadence Design Systems, Inc.

autoSaveMarkNetSetupFile

```
layout autoSaveMarkNetSetupFile boolean { t | nil }
```

Description

Controls the saving of the `markNetOptions` file automatically.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "autoSaveMarkNetSetupFile")
envSetVal("layout" "autoSaveMarkNetSetupFile" 'boolean nil)
```

Related Topics

markNetOptionsFile

© 2023 Cadence Design Systems, Inc.

autoSetRefPoint

```
layout autoSetRefPoint boolean { t | nil }
```

Description

Checks if the reference points are set automatically whenever a new point is entered.

The default is `t`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Auto Set Reference Point*

Examples

```
envGetVal("layout" "autoSetRefPoint")
envSetVal("layout" "autoSetRefPoint" 'boolean nil)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

autoStoreSelectionSetOn

```
graphic autoStoreSelectionSetOn boolean { t | nil }
```

Description

Turns *Auto Store* on or off for *Auto Store Section Controls* in the Selection Options form, accessed by choosing *Options – Selection*.

The default is `nil`.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Auto Store Section Controls – Auto Store*

Examples

```
envGetVal("layout" "autoStoreSelectionSetOn")
envSetVal("layout" "autoStoreSelectionSetOn" 'boolean t)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

autoTapShape

```
layout autoTapShape boolean { t | nil }
```

Description

Controls whether a design-tap occurs on the first entry point or not, when you create rectangles, polygons, paths, circles, donuts, and ellipses. During the *Create* command, when *Shape Auto Tap* is on, the current entry layer will seed the form and update attributes of the object you create based on the first point you enter.

The default is `nil`.

GUI Equivalent

Command:	*Options – Layout Editor*
Field:	*Auto Tap: Shape*

Examples

```
envGetVal("layout" "autoTapShape")
envSetVal("layout" "autoTapShape" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

autoTapType

```
layout autoTapType string { "layer" | "attributes" | "layer attributes" }
```

Description

Controls how the auto tap command behaves. It is used to set whether attributes of the tapped objects will be used, and whether to switch to a different layer when tapping on an object that is on a different layer than the current layer.

- `layer`: Taps only layers of the object.
- `attributes`: Captures the beginning and ending extensions, the end type, and the width.
- `layer attributes`: Default. Taps both attributes and layers of the objects.

For trls, bends and tapers the tap command captures the bend factor (maximum width before they are chamfered), taper style (linear or exponential), number of points to taper in user units, the bend style (bend chamfer or radial), the width of a chamfered bend in a transmission line as a multiple of its standard width in user units, the number of segments in a radial bend in a transmission line as a function of the number of segments in a 180-degree radial bend, the radius of a radial bend in a transmission line as a multiple of the transmission-line width in user units, the width of the first end of a transmission line bend in user units, the width of the second end of a transmission line bend, and the width of the second end of a transmission line bend in user units.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Tap Layer* and *Tap Attributes*

Examples

```
envGetVal("layout" "autoTapType")
envSetVal("layout" "autoTapType" 'string "layer")
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

autoViaOverrideVariantParameters

```
layout autoViaOverrideVariantParameters boolean { t | nil }
```

Description

Lets you update the via variant parameter values in the *Single*, *Stack*, and *Auto* via creation modes.

The default is `nil`.

GUI Equivalent

Command:	*Create – Via, Options* button in the Create Via form
Field:	*Override Variant Parameters*

Examples

```
envGetVal("layout" "autoViaOverrideVariantParameters")
envSetVal("layout" "autoViaOverrideVariantParameters" 'boolean t)
```

Related Topics

Create Via Options Form

© 2023 Cadence Design Systems, Inc.

autoviaOverrideVariantParameters

```
layout.ava autoviaOverrideVariantParameters boolean { t | nil }
```

Description

Lets you to update a via variant and overwrite the existing via parameter values. If this check box is selected, it allows the via engine to update a via variant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant options*
Field:	*Override variants*

Examples

```
envGetVal("layout.ava" "autoviaOverrideVariantParameters")
envSetVal("layout.ava" "autoviaOverrideVariantParameters" 'boolean t)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

autoZoomMode

```
layout autoZoomMode cyclic { "Pan" | "Zoom" | "ZoomToFit" }
```

Description

Lets you specify the dynamic zoom setting to be used. You can specify `Pan`, `Zoom`, `ZoomToFit`.

The default is `Pan`.

GUI Equivalent

Command:	*Window – Assistants – Navigator* or *Options – Display*
Field:	*Mode*

Examples

```
envGetVal("layout" "autoZoomMode")
envSetVal("layout" "autoZoomMode" 'cyclic "Zoom")
envSetVal("layout" "autoZoomMode" 'cyclic "ZoomToFit")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

autoZoomPan

```
layout autoZoomPan boolean { t | nil }
```

Description

Turns on or off the automatic zooming or panning to objects you select in the *Navigator* assistant.

The default is `t`.

GUI Equivalent

Command:	*Window – Assistants – Navigator* or *Options – Display*
Field:	*Dynamic Zoom*

Examples

```
envGetVal("layout" "autoZoomPan")
envSetVal("layout" "autoZoomPan" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

autoZoomScale

```
layout autoZoomScale int zoomScale
```

Description

Lets you specify the zoom scale in percentage when using the dynamic zoom feature.

The default is `90`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Zoom Scale (%)*

Examples

```
envGetVal("layout" "autoZoomScale")
envSetVal("layout" "autoZoomScale" 'int 100)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

balloonFadeTime

```
layout balloonFadeTime float time
graphic balloonFadeTime float time
```

Description

Determines the amount of time (in seconds) for the information balloons to fade away after the balloon appears. Valid values are between `.001` and `100`.

GUI Equivalent

Command:	*Options – Dynamic Display – Info Balloon* tab
Field:	*Fade Time*

Examples

```
envGetVal("layout" "balloonFadeTime")
envSetVal("layout" "balloonFadeTime" 'float 5.0)
envGetVal("graphic" "balloonFadeTime")
envSetVal("graphic" "balloonFadeTime" 'float 5.0)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

balloonOn

```
layout balloonOn boolean { t | nil }
graphic balloonOn boolean { t | nil }
```

Description

Checks if the *Show Info Balloon* option is set to on.

The default is `nil`.

GUI Equivalent

Command:	*Options – Dynamic Display*
Field:	*Show Info Balloon*

Examples

```
envGetVal("layout" "balloonOn")
envSetVal("layout" "balloonOn" 'boolean t)
envGetVal("graphic" "balloonOn")
envSetVal("graphic" "balloonOn" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

balloonPauseTime

```
layout balloonPauseTime float time
```

Description

Calculates the amount of time before the information balloon appears when the pointer is over an object. Valid values are between 1 and 10.

GUI Equivalent

Command:	*Options – Dynamic Display – Info Balloon* tab
Field:	*Pause Time*

Examples

```
envGetVal("layout" "balloonPauseTime")
envSetVal("layout" "balloonPauseTime" 'float 5.0)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

balloonTransparency

`layout balloonTransparency int` *`transparency`*

`graphic balloonTransparency int` *`transparency`*

Description

Determines the transparency of the balloon information box.

When set to `0`, the box is completely transparent. When set to `100`, objects behind the box are not visible.

The default is `50`.

GUI Equivalent

Command:	*Options – Dynamic Display – Info Balloon* tab
Field:	*Transparency*

Examples

```
envGetVal("layout" "balloonTransparency")
envSetVal("layout" "balloonTransparency" 'int 70)
envGetVal("graphic" "balloonTransparency")
envSetVal("graphic" "balloonTransparency" 'int 25)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

bannerFields

```
layout bannerFields string { "X" | "Y" | "Mode" | "Select" | "SelI" | "SelO" |
     "SelN" | "Tpt" | "DRDEnforce" | "DRDNotify" | "DRDPost-Edit" | "CAE" | "dX" |
     "dY" | "Dist" | "Cmd" }
```

Description

Lets you configure the items that appear on the status banner. You can display all, some, or none of the items on the status banner. You can also change the order in which the items appear.

You can set the *Mode* option to full select (F), full select with spine (FS), partial select (P), or partial select with spine (PS).

- Select: Displays a combination of instances and objects.
- SelI: Displays the number of instances selected. In SelI, the last character is the letter **I** in uppercase, not one. The number following the setting is the number of instances selected.
- SelO: Displays the number of objects other than the instances selected. In SelO, the last character is the letter O in uppercase, not zero. The number following the setting is the number of selected objects, other than instances.
- SelN: Displays the number of selected nets.
- Tpt: Displays (T) if either the *Transparent Group* icon on the *Options* toolbar or the *Transparent Group* option in the Display Options form is turned on.
- DRDEnforce, DRDNotify, and DRDPost-Edit options: Displays the status set for DRD modes.
- CAE: The option for constraint aware editing, CAE, is available only in VLS XL and VLS GXL. It can be set to ON or OFF. By default, CAE is OFF.

Note: *X*, *Y*, and *Mode* are coupled. If you set one, the others are automatically set. The same is true for *dX*, *dY* and *Dist*, and Select, SelI and SelO.

By default, this environment variable displays Cmd and SnapMode. Cmd is also available on the *Status Toolbar*.

Important

The string selections are case sensitive.

© 2023 Cadence Design Systems, Inc.

GUI Equivalent

None

Examples

```
envGetVal("layout" "bannerFields")
envSetVal("layout" "bannerFields" 'string "DRDEnforce")
envSetVal("layout" "bannerFields" 'string "CAE")
```

Related Topics

Status Information

© 2023 Cadence Design Systems, Inc.

blockageAllowPGNet

```
layout blockageAllowPGNet boolean { t | nil }
```

Description

Lets you place shapes with signal types power and ground within the blockage area.

The default is `nil`.

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Allow PGNet*

Examples

```
envGetVal("layout" "blockageAllowPGNet")
envSetVal("layout" "blockageAllowPGNet" 'boolean t)
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageAttachOwner

```
layout blockageAttachOwner string { "none" | "instance" | "cluster" }
```

Description

Assigns one instance or one cluster as the blockage's owner, depending upon the type of blockage. In the case of a placement halo blockage, the owner should be an instance or PR boundary. An owner can have any number of blockages. Use the attach/detach command to add or remove an owner. Cyclical relationships are not allowed. For example, if object A is a parent to object B, then object B cannot be a parent for object A.

The default is `none`.

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Attach to Owner*

Examples

```
envGetVal("layout" "blockageAttachOwner")
envSetVal("layout" "blockageAttachOwner" 'string "instance")
envSetVal("layout" "blockageAttachOwner" 'string "cluster")
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageAttributeMode

```
layout blockageAttributeMode cyclic { "ignoreLPP" | "useLPP" }
```

Description

Determines the selectability of blockages.

- `ignoreLPP`: Default. Blockage visibility and selection is controlled by the status of the check box for the corresponding blockage object in the Objects panel.
- `useLPP`: Blockage visibility and selection is controlled by the status of the check box for the corresponding blockage object in the Objects panel and the status of the object's layer-purpose pair attribute in the Layers panel.

GUI Equivalent

None

Examples

```
envGetVal("layout" "blockageAttributeMode")
envSetVal("layout" "blockageAttributeMode" 'cyclic "useLPP")
```

Related Topics

Objects Panel

Layers Panel

© 2023 Cadence Design Systems, Inc.

blockageCreateMethod

```
layout blockageCreateMethod string { "manual" | "automatic" }
```

Description

Sets the method used to create a blockage.

- `Manual`: Default. Requires that you enter points to define the blockage.
- `Automatic`: Derives points from objects you select, then creates the blockage from these points.

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Create Method*

Examples

```
envGetVal("layout" "blockageCreateMethod")
envSetVal("layout" "blockageCreateMethod" 'string "automatic")
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageDerivePoints

```
layout blockageDerivePoints string { "Selected Set bBox" | "Instance Halo" |
     "Instance prBoundary" | "Instance bBox" }
```

Description

Sets the type of object or objects from which you can derive the points to create a blockage.

The default is `Selected Set bBox`.

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Derive From*

Examples

```
envGetVal("layout" "blockageDerivePoints")
envSetVal("layout" "blockageDerivePoints" 'string "Instance Halo")
envSetVal("layout" "blockageDerivePoints" 'string "Instance prBoundary")
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageIsPushedDown

```
layout blockageIsPushedDown boolean { t | nil }
```

Description

Indicates that the blockage has been pushed down from a higher level block.

The default is `nil`.

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Pushed Down*

Examples

```
envGetVal("layout" "blockageIsPushedDown")
envSetVal("layout" "blockageIsPushedDown" 'boolean t)
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageKeepOwner

```
layout blockageKeepOwner boolean { t | nil }
```

Description

Retains the owner information. When you create multiple blockages, the owner is automatically assigned to the additional blockages.

The default is `nil`

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Keep Owner*

Examples

```
envGetVal("layout" "blockageKeepOwner")
envSetVal("layout" "blockageKeepOwner" 'boolean t)
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageMaterial

```
layout blockageMaterial string { "all" | "other" | "nwell" | "pwell" | "nplus" |
     "pplus" | "ndiff" | "pdiff" | "poly" | "metal" | "diffusion" | "li" | "cut" |
     "passivationCut"}
```

Description

Selects the material on which the blockage is enforced.

The default is `all`.

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Material*

Examples

```
envGetVal("layout" "blockageMaterial")
envSetVal("layout" "blockageMaterial" 'string "other")
envSetVal("layout" "blockageMaterial" 'string "passivationCut")
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageMaxDensity

```
layout blockageMaxDensity float percentage
```

Description

Sets the maximum percentage of the total design area the metal in the specified layer may occupy.

The default is `100.0`

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Max Density*

Examples

```
envGetVal("layout" "blockageMaxDensity")
envSetVal("layout" "blockageMaxDensity" 'float 70.0)
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageSpacingType

```
layout blockageSpacingType string { "Blockage geometry" | "Effective width" |
     "Specify" }
```

Description

Sets the spacing type for the blockage.

The default is `Blockage geometry`.

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Compute spacing using*

Examples

```
envGetVal("layout" "blockageSpacingType")
envSetVal("layout" "blockageSpacingType" 'string "Effective width")
envSetVal("layout" "blockageSpacingType" 'string "Specify")
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageSpacingValue

```
layout blockageSpacingValue float width
```

Description

Specifies the value for effective width or spacing for blockage.

The default is `0.0`.

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Compute spacing using*

Examples

```
envGetVal("layout" "blockageSpacingValue")
envSetVal("layout" "blockageSpacingValue" 'float 1.0)
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

blockageType

```
layout blockageType string { "routing" | "placement" | "pin" | "feedthru" | "screen"
     }
```

Description

Selects the type of blockage to create.

The default is `routing`.

GUI Equivalent

Command:	*Create – P&R Objects – Blockage*
Field:	*Blockage Type*

Examples

```
envGetVal("layout" "blockageType")
envSetVal("layout" "blockageType" 'string "placement")
envSetVal("layout" "blockageType" 'string "pin")
```

Related Topics

Create Blockage Form

© 2023 Cadence Design Systems, Inc.

bondPadLPP

```
layout bondPadLPP string "layer_purpose_pair"
```

Description

Specifies the layer purpose pair on which the *Pad Opening Info* command is to search for pad shapes.

The default is `"Bondpad drawing"`.

GUI Equivalent

Command:	*Tools – Pad Opening Info*
Field:	*Layer/Purpose* drop-down in the *Bond Pad* section at the top of the form

Examples

```
envGetVal("layout" "bondPadLPP")
envSetVal("layout" "bondPadLPP" 'string "Metal1 drawing")
envSetVal("layout" "bondPadLPP" 'string "Via2 drawing")
```

Related Topics

Pad Opening Info Form

© 2023 Cadence Design Systems, Inc.

boundarySnappingLayer

`layout boundarySnappingLayer string "`*`layerName1 layerName2`*`"`

Description

Overrides the grid for PR boundary snapping. If this environment variable is specified, the PR boundary snaps to the active grids whose snapping layers are specified in this environment variable.

If this environment variable is not specified, snapping is done based on the grids specified in `boundarySnapXGrids` and `boundarySnapYGrids` environment variables.

In case `boundarySnapXGrids` and `boundarySnapYGrids` environment variables are not specified, snapping is done to the least common multiple of all active grids.

GUI Equivalent

None

Examples

```
envGetVal("layout" "boundarySnappingLayer")
envSetVal("layout" "boundarySnappingLayer" 'string "M1WSP")
```

Related Topics

boundarySnapXGrids

boundarySnapYGrids

© 2023 Cadence Design Systems, Inc.

boundarySnapXGrids

```
layout boundarySnapXGrids string "grid"
```

Description

Snaps the PR boundary to the specified grids in the X direction.

The grids specified through this environment variable need not be active global grids.

GUI Equivalent

None

Examples

```
envGetVal("layout" "boundarySnapXGrids")
envSetVal("layout" "boundarySnapXGrids" 'string "M1WSP M3WSP")
```

Related Topics

Global Grid Anchoring

© 2023 Cadence Design Systems, Inc.

boundarySnapYGrids

```
layout boundarySnapYGrids string "grid"
```

Description

Snaps the PR boundary to the specified grids in the Y direction.

The grids specified through this environment variable need not be active global grids.

GUI Equivalent

None

Examples

```
envGetVal("layout" "boundarySnapYGrids")
envSetVal("layout" "boundarySnapYGrids" 'string "M2WSP M4WSP")
```

Related Topics

Global Grid Anchoring

© 2023 Cadence Design Systems, Inc.

carDefaultGroup

```
layout carDefaultGroup string "groupName"
```

Description

Specifies the default group for the recording when you open the Capture and Replay assistant.

The default is `All`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "carDefaultGroup")
envSetVal("layout" "carDefaultGroup" 'string "Macros1")
```

Related Topics

Capture and Replay Assistant

© 2023 Cadence Design Systems, Inc.

chainMode

```
layout chainMode cyclic { "unknown" | "all" | "selected" | "selectLeft" |
     "selectRight" }
```

Description

Determines which instances in a chain of instances will move, copy, stretch or delete.

- `all`: Moves, copies, stretches, or deletes all instances in a chain together.
- `selected`: Default. Moves, copies, stretches, or deletes only selected instances irrespective of the chain contents.
- `selectLeft`: Moves, copies, stretches, or deletes the instance selected and all instances to the left of the selected instance in the chain.
- `selectRight`: Moves, copies, stretches, or deletes the instance selected and all instances to the right of the selected instance in the chain.

GUI Equivalent

Command:	*Edit – Move*, *Edit – Copy*, *Edit – Stretch*, *Edit – Delete*
Field:	*Chain Mode*

Examples

```
envGetVal("layout" "chainMode")
envSetVal("layout" "chainMode" 'cyclic "unknown")
envSetVal("layout" "chainMode" 'cyclic "selectLeft")
```

Related Topics

Move Form

Copy Form

Stretch Form

Delete Form

© 2023 Cadence Design Systems, Inc.

checkAllSPGridsEnclosures

```
layout checkAllSPGridsEnclosures boolean {t | nil}
```

Description

Searches all enclosures to locate the one that centers the shape on the tracks when snapping in detailed mode. The default is `t`.

Consider the following snap pattern definition:

```
snapPatternDefs(
    (GPG94              ("PPitch" "poly94")
    'step               0.094
    'stepDirection      "horizontal"
    'snappingLayers     (('layer "Poly" 'enclosures (0.009 0.01 0.012) \
                          'purposes ("drawing" "dummy")))
    'type               "local"
    )
)
```

When `checkAllSPGridsEnclosures` is set to `t`, all enclosure values (`0.009`, `0.01`, and `0.012`) are evaluated one by one. When the environment variable is set to `nil`, only the first enclosure value (`0.009`) is evaluated.

GUI Equivalent

None

Examples

```
envGetVal("layout" "checkAllSPGridsEnclosures")
envSetVal("layout" "checkAllSPGridsEnclosures" 'boolean nil)
```

Related Topics

Shape Snapping

© 2023 Cadence Design Systems, Inc.

checkHalfWidthPathSeg

```
layout checkHalfWidthPathSeg cyclic { "off" | "notify" | "enforce" | "never" }
```

Description

Provides control for checking the width of a path segment with respect to the width of the path. This environment variable applies to commands used for creating and editing paths, including ROD MPPs and their sub paths.

- `off`: Default. No warning appears when you create or edit a path that has a segment length less than half the path width, except if it is the first or last segment. If you create or edit the first or last segment of the path with a length less than half the path width, the endStyle extension is used to determine the segment length.
- `notify`: A warning dialog box appears when you create or edit a path that has a segment length less than half the path width. If you confirm to create or edit the short segment as an intended action, the path is created or updated, else the last attempted edit is ignored.
- `enforce`: It does not allow you to create a path segment with length less than half the path width.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Half Width Check*

Examples

```
envGetVal("layout" "checkHalfWidthPathSeg")
envSetVal("layout" "checkHalfWidthPathSeg" 'cyclic "notify")
envSetVal("layout" "checkHalfWidthPathSeg" 'cyclic "enforce")
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

checkLargeCoordinates

```
layout checkLargeCoordinates boolean { t | nil }
```

Description

Provides control for checking for large coordinates during the copy operation.

The default is `nil`. When set to `t`, the *Copy* command does not allow objects to exceed the INT_MAX boundary and hence does not distort the copied image. On a 64-bit system, INT_MAX = 2147483647.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Check Large Coordinates*

Examples

```
envGetVal("layout" "checkLargeCoordinates")
envSetVal("layout" "checkLargeCoordinates" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

chopArray

```
layout chopArray boolean { t | nil }
```

Description

Lets you specify whether to chop an array of instances (mosaic) or not. The array of instances is allowed to be chopped based on the shape (line, rectangle, polygon) selected for chopping.

The default is `t`. When set to `nil`, the array of instances is not chopped.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Chop array*

Examples

```
envGetVal("layout" "chopArray")
envSetVal("layout" "chopArray" 'boolean nil)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopCircleJustify

```
layout chopCircleJustify cyclic { "Center" | "Left" | "Right" | "Bottom" | "Top" }
```

Description

Specifies the justification of circle chop shape.

The default is `Center`.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Chop Shape: circle: Justification*

Examples

```
envGetVal("layout" "chopCircleJustify")
envSetVal("layout" "chopCircleJustify" 'cyclic "Left")
envSetVal("layout" "chopCircleJustify" 'cyclic "Top")
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopCircleRadius

`layout chopCircleRadius float` *`radius`*

Description

Specifies the radius of the circle chop shape.

The layout default is `0.0` user units.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Radius*

Examples

```
envGetVal("layout" "chopCircleRadius")
envSetVal("layout" "chopCircleRadius" 'float 2.0)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopLineJustify

```
layout chopLineJustify cyclic { "Center" | "Left" | "Right" }
```

Description

Specifies the justification of line chop shape.

The default is `Center`.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Chop Shape: line: Justification*

Examples

```
envGetVal("layout" "chopLineJustify")
envSetVal("layout" "chopLineJustify" 'cyclic "Left")
envSetVal("layout" "chopLineJustify" 'cyclic "Right")
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopLineType

```
layout chopLineType cyclic { "User Spacing" | "Min Spacing" }
```

Description

Specifies the type of spacing for line chop shape, `User Spacing` or `Min Spacing`.

The default is `User Spacing`.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Type*

Examples

```
envGetVal("layout" "chopLineType")
envSetVal("layout" "chopLineType" 'cyclic "Min Spacing")
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopLineUserWidth

```
layout chopLineUserWidth float width
```

Description

Specifies the width of the user spacing for line chop shape.

The layout default is `0.0` user units.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Type: User Spacing: Width*

Examples

```
envGetVal("layout" "chopLineUserWidth")
envSetVal("layout" "chopLineUserWidth" 'float 2.0)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopPRBDialog

```
layout chopPRBDialog boolean { t | nil }
```

Description

Displays a warning dialog box when the *Chop* command is run outside the PR boundary or there is no PR boundary in the design.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "chopPRBDialog")
envSetVal("layout" "chopPRBDialog" 'boolean t)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopRectHeight

```
layout chopRectHeight float height
```

Description

Specifies the height of the rectangle chop shape.

The layout default is `0.0` user units.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Height*

Examples

```
envGetVal("layout" "chopRectHeight")
envSetVal("layout" "chopRectHeight" 'float 2.0)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopRectJustify

```
layout chopRectJustify cyclic { "Bottom left" | "Bottom right" | "Top left" | "Top
     right" | "Center" }
```

Description

Specifies the justification of rectangle chop shape.

The default is `Bottom Left`.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Chop Shape: rectangle: Justification*

Examples

```
envGetVal("layout" "chopRectJustify")
envSetVal("layout" "chopRectJustify" 'cyclic "Bottom right")
envSetVal("layout" "chopRectJustify" 'cyclic "Center")
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopRectWidth

```
layout chopRectWidth float width
```

Description

Specifies the width of the rectangle chop shape.

The layout default is `0.0` user units.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Width*

Examples

```
envGetVal("layout" "chopRectWidth")
envSetVal("layout" "chopRectWidth" 'float 2.0)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopShape

```
layout chopShape cyclic { "unknown" | "rectangle" | "polygon" | "line" | "circle"
     | "wire" }
```

Description

Specifies the default shape of the cutter used by `leHiChop` or `leChopShape`.

The default is `rectangle`.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Chop Shape*

Examples

```
envGetVal("layout" "chopShape")
envSetVal("layout" "chopShape" 'cyclic "polygon")
envSetVal("layout" "chopShape" 'cyclic "line")
```

Related Topics

Chop Form

leHiChop

leChopShape

© 2023 Cadence Design Systems, Inc.

chopSpecifySize

```
layout chopSpecifySize boolean { t | nil }
```

Description

Specifies the size of the chop shape, rectangle or circle.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Specify Size*

Examples

```
envGetVal("layout" "chopSpecifySize")
envSetVal("layout" "chopSpecifySize" 'boolean t)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

chopSpecifySpacing

```
layout chopSpecifySpacing boolean { t | nil }
```

Description

Specifies the spacing of the line chop shape.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Specify Spacing*

Examples

```
envGetVal("layout" "chopSpecifySpacing")
envSetVal("layout" "chopSpecifySpacing" 'boolean t)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

circleRadius

`layout circleRadius float` *`radius`*

Description

Specifies the radius of the circle.

The default is `0.0`.

GUI Equivalent

Command:	*Create – Shape – Circle*
Field:	*Radius*

Examples

```
envGetVal("layout" "circleRadius")
envSetVal("layout" "circleRadius" 'float 3.0)
```

Related Topics

Create Circle Form

© 2023 Cadence Design Systems, Inc.

circleShapeMode

```
layout circleShapeMode cyclic { "normal" | "halfQ1" | "halfQ2" | "halfQ3" | "halfQ4"
     | "quarterQ1" | "quarterQ2" | "quarterQ3" | "quarterQ4"}
```

Description

Specifies the shape of circle that you want to create: full circle (`normal`), half circle (`halfQ1, halfQ2, halfQ3, halfQ4`), or quarter circle (`quarterQ1, quarterQ2, quarterQ3, quarterQ4`).

The default is `normal`.

GUI Equivalent

Command:	*Create – Shape – Circle*
Field:	*Shape Type*

Examples

```
envGetVal("layout" "circleShapeMode")
envSetVal("layout" "circleShapeMode" 'cyclic "halfQ1")
envSetVal("layout" "circleShapeMode" 'cyclic "quarterQ4")
```

Related Topics

Create Circle Form

© 2023 Cadence Design Systems, Inc.

clearAllMeasurementsUseEditMode

```
layout clearAllMeasurementsUseEditMode boolean { t | nil }
```

Description

Specifies that cellviews are to be opened in edit mode when deleting rulers.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "clearAllMeasurementsUseEditMode")
envSetVal("layout" "clearAllMeasurementsUseEditMode" 'boolean nil)
```

Related Topics

leHiClearMeasurementInHier

Saving and Deleting Rulers

© 2023 Cadence Design Systems, Inc.

clusterBoundaryAspectRatio

`layout clusterBoundaryAspectRatio float` *`aspectRatio`*

Description

Sets the aspect ratio of the cluster boundary during auto-creation in Layout XL.

Aspect Ratio W/L is the width-to-length ratio of the cluster boundary. For example, a value of 1 specifies a square boundary, 2 specifies a boundary twice as high as it is long, and 0.5 specifies a boundary half as wide as it is long.

The default is `1.0`.

GUI Equivalent

Command:	*Create – P&R Objects – Cluster Boundary*
Field:	*Aspect Ratio (W/H)*

Examples

```
envGetVal("layout" "clusterBoundaryAspectRatio")
envSetVal("layout" "clusterBoundaryAspectRatio" 'float 2.0)
```

Related Topics

Create Cluster Boundary Form

© 2023 Cadence Design Systems, Inc.

clusterBoundaryUtilization

```
layout clusterBoundaryUtilization float percentage
```

Description

Specifies the percentage of area within the cluster boundary that you want to fill in Layout XL.

The default is `80.0`.

GUI Equivalent

Command:	*Create – P&R Objects – Cluster Boundary*
Field:	*Utilization%*

Examples

```
envGetVal("layout" "clusterBoundaryUtilization")
envSetVal("layout" "clusterBoundaryUtilization" 'float 90.0)
```

Related Topics

Create Cluster Boundary Form

© 2023 Cadence Design Systems, Inc.

commandVerboseLevel

```
layout commandVerboseLevel int verboselevel
```

Description

Displays the verbosity message for commands such as *Copy, Move, Stretch, Delete, Flip, Create Measurement, Create Rectangle, and Create Instance,* in the log file or CIW. This message is based on the current verbose level. The default is `1`.

The supported levels are:

- Level 0: Verbosity output is not displayed.
- Level 1: Verbosity output is displayed in the log file as a comment.
- Level 2: Verbosity output is displayed in the CIW as an information message.
- Level 3: Verbosity output is displayed in the CIW in detail.

GUI Equivalent

None

Examples

```
envGetVal("layout" "commandVerboseLevel")
envSetVal("layout" "commandVerboseLevel" 'int 2)
```


© 2023 Cadence Design Systems, Inc.

confirmDelete

```
layout confirmDelete boolean { t | nil }
```

Description

Turns on and off a dialog box which reads *Delete Figure?* to confirm an object deletion.

You can choose the *Yes* or *No* option on the form. The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "confirmDelete")
envSetVal("layout" "confirmDelete" 'boolean t)
```

Related Topics

Delete Form

© 2023 Cadence Design Systems, Inc.

convertEachInstanceIntoMosaic

```
layout convertEachInstanceIntoMosaic 'boolean { t | nil }
```

Description

Specifies the mode for the *Convert to Mosaic* command. The command converts the currently selected instances into the equivalent number of individual mosaics.

The default is `t`. When set to `nil`, the command converts the currently selected instances into a single mosaic.

GUI Equivalent

Command:	*Edit – Convert – To Mosaic*
Fields:	*Convert Instances into Mosaics*
	Convert Instances into Single Mosaic

Examples

```
envGetVal("layout" "convertEachInstanceIntoMosaic")
envSetVal("layout" "convertEachInstanceIntoMosaic" 'boolean nil)
```

Related Topics

Converting Instances to Mosaics

© 2023 Cadence Design Systems, Inc.

copyColumns

`layout copyColumns int` *`numberOfColumns`*

Description

Specifies the default number of columns to copy.

The default is `1`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Columns*

Examples

```
envGetVal("layout" "copyColumns")
envSetVal("layout" "copyColumns" 'int 5)
```

Related Topics

Copy Form

© 2023 Cadence Design Systems, Inc.

copyConnectivity

```
layout copyConnectivity boolean { t | nil }
```

Description

Copies the connectivity information on path, pathseg, fluid guard ring, and via to the new object.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Copy Connectivity*

Examples

```
envGetVal("layout" "copyConnectivity")
envSetVal("layout" "copyConnectivity" 'boolean t)
```

Related Topics

Copy Form

© 2023 Cadence Design Systems, Inc.

copyCreateGroupArray

```
layout copyCreateGroupArray boolean { t | nil }
```

Description

Enables the creation of a group array when using the *Copy* command.

The default value is `nil`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Create as Group Array*

Examples

```
envGetVal("layout" "copyCreateGroupArray")
envSetVal("layout" "copyCreateGroupArray" 'boolean t)
```

Related Topics

Copy Form

Group Arrays

© 2023 Cadence Design Systems, Inc.

copyGroupArrayLockXY

```
layout copyGroupArrayLockXY boolean { t | nil }
```

Description

Enables the creation of a locked group array when using the *Copy* command.

When this environment variable is set to `t`, spacing mode and spacing values for a group array cannot be specified in the Copy form. In this case, spacing mode is read from the `copySpacingMode` environment variable and spacing values are read from the `groupArrayXSpacing` and `groupArrayYSpacing` environment variables.

The default value is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "copyGroupArrayLockXY")
envSetVal("layout" "copyGroupArrayLockXY" 'boolean t)
```

Related Topics

copySpacingMode

groupArrayXSpacing

groupArrayYSpacing

Group Arrays

Locked Group Arrays

© 2023 Cadence Design Systems, Inc.

copyIncrementInstName

```
layout copyIncrementInstName boolean { t | nil }
```

Description

Increments the instance name during the *Copy* command when selected.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "copyIncrementInstName")
envSetVal("layout" "copyIncrementInstName" 'boolean t)
```

Related Topics

Object Copying

© 2023 Cadence Design Systems, Inc.

copyIncrementNetName

```
layout copyIncrementNetName boolean { t | nil }
```

Description

Increments the bus bit on the copied object by one if the selected object has a net name with a bus bit.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Increment net name*

Examples

```
envGetVal("layout" "copyIncrementNetName")
envSetVal("layout" "copyIncrementNetName" 'boolean t)
```

Related Topics

Copy Form

© 2023 Cadence Design Systems, Inc.

copyOrder

```
layout copyOrder cyclic { "Single" | "Step" | "Array" | "Line" }
```

Description

Specifies the way in which the copies of objects are created.

- `Single`: Lets you create a single copy of objects.
- `Step`: Lets you copy the selected objects in an incremental manner.
- `Array`: Lets you copy the selected objects in an array form.
- `Line`: Supported for backward compatibility; it is the same as the `Step` option. The `Line` option is not available on the Copy form.

The default value is `Line`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Copies – Type*

Examples

```
envGetVal("layout" "copyOrder")
envSetVal("layout" "copyOrder" 'cyclic "Step" )
envSetVal("layout" "copyOrder" 'cyclic "Array" )
```

Related Topics

Copy Form

Copying an Object

Creating a Group Array by Using the Copy Form

© 2023 Cadence Design Systems, Inc.

copyResetRowCol

```
layout copyResetRowCol boolean { t | nil }
```

Description

Sets the default location for the Copy form's *Rows* and *Columns* fields each time you repeat the *Copy* command.

The default is `nil`, the location is set to the `.cdsenv` file. When set to `nil`, the last value entered in the form fields is retained.

GUI Equivalent

None

Examples

```
envGetVal("layout" "copyResetRowCol")
envSetVal("layout" "copyResetRowCol" 'boolean nil)
```

Related Topics

Copy Form

© 2023 Cadence Design Systems, Inc.

copyRows

```
layout copyRows int numberOfRows
```

Description

Specifies the default number of rows to copy.

The default is `1`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Rows*

Examples

```
envGetVal("layout" "copyRows")
envSetVal("layout" "copyRows" 'int 10)
```

Related Topics

Copy Form

© 2023 Cadence Design Systems, Inc.

copySpacingMode

```
layout copySpacingMode cyclic { "pitch" | "spacing" | "absolute" }
```

Description

Sets the spacing mode in the Copy form.

- `pitch`: Applies between objects the delta distance specified in the *Delta X, Y* fields of the Copy form. The distance is measured between the same-edge corners of the objects in the array; for example, the distance between the left edges of objects.
- `spacing`: Applies between objects the distance specified in the *Spacing X, Y* fields. The distance is measured between the adjacent-edge corners of the objects in the array; for example the distance between the bottom edge of an object and the top edge of another object.
- `absolute`: Enables you to copy an object at the exact location specified in the *X, Y* fields. The *absolute* mode is not available for group arrays.

The default value is `pitch`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Spacing – Mode*

Examples

```
envGetVal("layout" "copySpacingMode")
envSetVal("layout" "copySpacingMode" 'cyclic "absolute")
envSetVal("layout" "copySpacingMode" 'cyclic "spacing")
```

Related Topics

Copy Form

© 2023 Cadence Design Systems, Inc.

copyToLayer

```
layout copyToLayer boolean { t | nil }
```

Description

Specifies whether shapes are copied to a different layer with `leHiCopy`. This does not change the form field in the form.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Change To Layer*

Examples

```
envGetVal("layout" "copyToLayer")
envSetVal("layout" "copyToLayer" 'boolean t)
```

Related Topics

Copy Form

leHiCopy

© 2023 Cadence Design Systems, Inc.

copyXField

`layout copyXField float` *`spacingValue`*

Description

Sets the spacing value used as the distance between objects on the X-axis when making copies of objects. This value is applicable for all spacing modes.

The default value is `0`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Spacing – Mode – Pitch – Delta X*
	Spacing – Mode – Spacing – Spacing X
	Spacing – Mode – Absolute – X

Examples

```
envGetVal ("layout" "copyXField")
envSetVal ("layout" "copyXField" 'float 0.5)
```

Related Topics

copyYField

copySpacingMode

© 2023 Cadence Design Systems, Inc.

copyYField

`layout copyYField float` *`spacingValue`*

Description

Sets the spacing value used as the distance between objects on the Y-axis when making copies of objects. This value is applicable for all spacing modes.

The default value is `0`.

GUI Equivalent

Command:	*Edit – Copy*
Field:	*Spacing – Mode – Pitch – Delta Y*
	Spacing – Mode – Spacing – Spacing Y
	Spacing – Mode – Absolute – Y

Examples

```
envGetVal ("layout" "copyYField")
envSetVal ("layout" "copyYField" 'float 0.5)
```

Related Topics

copyXField

copySpacingMode

© 2023 Cadence Design Systems, Inc.

createAsLabel

```
layout createAsLabel boolean { t | nil }
```

Description

Specifies that a label is created as a label or a text display.

The default is `nil`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Mode: Auto: Create As Label*

Examples

```
envGetVal("layout" "createAsLabel")
envSetVal("layout" "createAsLabel" 'boolean t)
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

createInstResetCDF

```
layout createInstResetCDF boolean { t | nil }
```

Description

Specifies whether the *Create Instance* command retains the current effective CDF values for a component when the same component is placed at a later time.

The default is `nil`. When set to `t`, the editor CDF resets the CDF values to their default settings when you start the *Create Instance* command.

GUI Equivalent

None

Examples

```
envGetVal("layout" "createInstResetCDF")
envSetVal("layout" "createInstResetCDF" 'boolean t)
```

Related Topics

Create Instance Form

createInstResetCDFParamsFile

© 2023 Cadence Design Systems, Inc.

createInstResetCDFParamsFile

```
layout createInstResetCDFParamsFile string "pathAndFileName"
```

Description

Specifies the path to the file containing the CDF parameters that must be reset to default values during instance creation. Use this environment variable to reset selected parameters for selected libraries and cells to their default values.

The default value is `""`.

You need to specify each entry in the file using the following format:

"*library*" "*cell*" "*parameters*"

For example, in the following sample entry, in the *decap_cell* cell of the `gpdk090` library, the `leftShape rHM1_switch_S` parameter is reset to the default value during instance creation.

```
"gpdk090" "decap_cell" "leftShape rHM1_switch_S"
```

You can use the asterisk (*) to refer to all libraries, cells, or parameters, as shown in the following sample entries:

```
"gpdk090" "*" "leftShape rightShape HM1_switch_S HM1_switch_D"
"*" "analog_cell" "*"
```

The `createInstResetCDF` environment variable resets all parameters for all cells and takes precedence over `createInstResetCDFParamsFile` when both are specified.

GUI Equivalent

None

Examples

```
envGetVal("layout" "createInstResetCDFParamsFile")
envSetVal("layout" "createInstResetCDFParamsFile" 'string "/project1/params.txt")
```

Related Topics

Create Instance Form

createInstResetCDF

© 2023 Cadence Design Systems, Inc.

createPinLabel

```
layout createPinLabel boolean { t | nil }
```

Description

Specifies that a pin label be created when using the Create Pin form.

The default is `nil`.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Create Label*

Examples

```
envGetVal("layout" "createPinLabel")
envSetVal("layout" "createPinLabel" 'boolean t)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

createPinPathSpineAware

```
layout createPinPathSpineAware boolean { t | nil }
```

Description

Checks if the pins created in the *Single* mode are path spine aware.

When set to `t`, a single merged pin is created on the highest metal layer shape island. A metal layer shape island is made of overlapping shapes that belong to the same layer and net.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "createPinPathSpineAware")
envSetVal("layout" "createPinPathSpineAware" 'boolean t)
```

Related Topics

Creating Pins Automatically

© 2023 Cadence Design Systems, Inc.

createPinSchematicAware

```
layout createPinSchematicAware boolean { t | nil }
```

Description

Specifies whether pin creation must be schematic aware.

When set to `t`, the tool uses its *I/O Type* and *Signal Type* values from corresponding schematic reference.

When set to the default value, `nil`, or if the corresponding schematic reference is not found, the values specified in the Create Pin form are used.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Schematic Aware*

Examples

```
envGetVal("layout" "createPinSchematicAware")
envSetVal("layout" "createPinSchematicAware" 'boolean t)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

createRectilinearBoundary

```
layout createRectilinearBoundary boolean { t | nil }
```

Description

Specifies that a rectilinear PR boundary is created by merging the PR boundary of the overlapping instances in the selection set. In case, there is no overlap between the selected instances, a rectangular PR boundary is created instead of rectilinear PR boundary.

The default is `nil`, which means that a rectangular PR boundary is created from the bounding box of the selection set.

GUI Equivalent

None

Examples

```
envGetVal("layout" "createRectilinearBoundary")
envSetVal("layout" "createRectilinearBoundary" 'boolean t)
```

Related Topics

Attributes in the Edit PR Boundary Properties Form

© 2023 Cadence Design Systems, Inc.

createViaAlwaysUsesDefinedCalcMode

```
layout createViaAlwaysUsesDefinedCalcMode boolean { t | nil}
```

Description

Lets the via engine use the calculation mode defined in the Layout Editor Options form for creating vias.

The default is `nil`.The via engine uses pre-defined values for creating vias in *Single*, *Stack*, or *Auto* mode.

GUI Equivalent

None

Examples

```
envGetVal("layout" "createViaAlwaysUsesDefinedCalcMode")
envSetVal("layout" "createViaAlwaysUsesDefinedCalcMode" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

cursorShape

```
layout cursorShape boolean { t | nil }
graphic cursorShape boolean { t | nil }
```

Description

Checks if the pointer is displayed as dot shaped or as dotted-line cross shaped.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Cross Cursor*

Examples

```
envGetVal("layout" "cursorShape")
envSetVal("layout" "cursorShape" 'boolean t)
envGetVal("graphic" "cursorShape")
envSetVal("graphic" "cursorShape" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

crossSectionCutLineColor

`graphic crossSectionCutLineColor string "`*`colorName`*`"`

Description

(Layout MXL Only) Specifies the color of the cutline displayed on the canvas when viewing the cross section of a layout. You can set the cutline color to `red`, `blue`, `green`, `yellow`, or `white`.

The default value is `red`.

GUI Equivalent

Command:	*Tools – Cross Section –* `F3`
Field:	*Cutline Color*

Examples

```
envGetVal("graphic" "crossSectionCutLineColor")
envSetVal("graphic" "crossSectionCutLineColor" 'string "blue")
envSetVal("graphic" "crossSectionCutLineColor" 'string "green")
```

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

© 2023 Cadence Design Systems, Inc.

crossSectionDisplayForwardCut

```
graphic crossSectionDisplayForwardCut boolean { t | nil }
```

Description

(Layout MXL Only) Specifies whether the vias present between two adjacent layers are displayed in the cross section of a layout even if those vias are not directly under the cutline.

The default value is `nil`, implying that a via is displayed only when you move the pointer over the via.

GUI Equivalent

Command:	*Tools – Cross Section –* `F3`
Field:	*Display – Forward Cut*

Examples

```
envGetVal("graphic" "crossSectionDisplayForwardCut")
envSetVal("graphic" "crossSectionDisplayForwardCut" 'boolean t)
```

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

© 2023 Cadence Design Systems, Inc.

crossSectionDisplayLayerName

```
graphic crossSectionDisplayLayerName boolean { t | nil }
```

Description

(Layout MXL Only) Specifies whether the names of layers are displayed in the cross section viewer.

The default value is `t`.

GUI Equivalent

Command:	*Tools – Cross Section –* `F3`
Field:	*Display – Layer Name*

Examples

```
envGetVal("graphic" "crossSectionDisplayLayerName")
envSetVal("graphic" "crossSectionDisplayLayerName" 'boolean nil)
```

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

© 2023 Cadence Design Systems, Inc.

crossSectionDisplayNetName

```
graphic crossSectionDisplayNetName boolean { t | nil }
```

Description

(Layout MXL Only) Specifies whether the net names of objects are displayed in the cross section viewer.

The default value is `nil`.

GUI Equivalent

Command:	*Tools – Cross Section –* `F3`
Field:	*Display – Net Names*

Examples

```
envGetVal("graphic" "crossSectionDisplayNetName")
envSetVal("graphic" "crossSectionDisplayNetName" 'boolean t)
```

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

© 2023 Cadence Design Systems, Inc.

crossSectionDisplayUnusedLayer

```
graphic crossSectionDisplayUnusedLayer boolean { t | nil }
```

Description

(Layout MXL Only) Specifies whether the unused layers are listed in the cross section viewer. Unused layers are the layers that are defined in the technology file but not used in the cross section of a layout.

The default value is `nil`.

GUI Equivalent

Command:	*Tools – Cross Section –* `F3`
Field:	*Display – Unused Layer*

Examples

```
envGetVal("graphic" "crossSectionDisplayUnusedLayer")
envSetVal("graphic" "crossSectionDisplayUnusedLayer" 'boolean t)
```

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

© 2023 Cadence Design Systems, Inc.

crossSectionOrientationHorizontal

```
graphic crossSectionOrientationHorizontal boolean { t | nil }
```

Description

(Layout MXL Only) Specifies whether the orientation of the cutline is horizontal. The area underneath a cutline is displayed in the cross section viewer. When set to `nil`, the vertical orientation is used for the cutline.

The default value is `t`.

GUI Equivalent

Command:	*Tools – Cross Section –* `F3`
Field:	*Orientation*

Examples

```
envGetVal("graphic" "crossSectionOrientationHorizontal")
envSetVal("graphic" "crossSectionOrientationHorizontal" 'boolean nil)
```

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

© 2023 Cadence Design Systems, Inc.

crossSectionSizeCutLine

```
graphic crossSectionSizeCutLine int cutLineSize
```

Description

(Layout MXL Only) Specifies the size of the cutline. The area underneath the cutline is displayed in the cross section viewer.

The default value is `200`.

If the value of the environment variable is less than `100` or does not fit into the canvas, the size of the cutline is not modified.

GUI Equivalent

Command:	*Tools – Cross Section –* `F3`
Field:	*Size*

Examples

```
envGetVal("graphic" "crossSectionSizeCutLine")
envSetVal("graphic" "crossSectionSizeCutLine" 'int 300)
```

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

© 2023 Cadence Design Systems, Inc.

crossSectionSyncWithPaletteVisibility

```
graphic crossSectionSyncWithPaletteVisibility boolean { t | nil }
```

Description

(Layout MXL Only) Synchronizes the display of shapes and short markers, in the cross section viewer, with the palette visibility settings. The default value is `t`.

When set to `nil`, the shapes and short markers are displayed even if their visibility is disabled in the palette.

GUI Equivalent

Command:	*Tools – Cross Section*
Field:	*Display – Sync with Palette Visibility*

Examples

```
envGetVal("graphic" "crossSectionSyncWithPaletteVisibility")
envSetVal("graphic" "crossSectionSyncWithPaletteVisibility" 'boolean nil)
```

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

© 2023 Cadence Design Systems, Inc.

crossSectionViewModeAssistant

```
graphic crossSectionViewModeAssistant boolean { t | nil }
```

Description

(Layout MXL Only) Specifies whether the cross section of a layout is displayed in the Cross Section assistant.

The default value is `t`. If the value is set to `nil`, the cross section is displayed in a floating window that follows the mouse pointer.

GUI Equivalent

Command:	*Tools – Cross Section –* `F3`
Field:	*View Mode – Assistant*

Examples

```
envGetVal("graphic" "crossSectionViewModeAssistant")
envSetVal("graphic" "crossSectionViewModeAssistant" 'boolean nil)
```

Related Topics

Cross Section Viewer

Specifying Cross Section Viewer Settings

Cross Section Viewer Form

© 2023 Cadence Design Systems, Inc.

defaultNewCellName

```
graphic defaultNewCellName string cellName
```

Description

Specifies the default name of the cell for the *Cell* field in the New File form, accessed by choosing *File – New*.

The default is none.

GUI Equivalent

Command:	*File – New*
Field:	*Cell*

Examples

```
envGetVal("graphic" "defaultNewCellName")
envSetVal("graphic" "defaultNewCellName" 'string "MyCell")
```

Related Topics

New File form

© 2023 Cadence Design Systems, Inc.

defaultNewLibName

```
graphic defaultNewLibName string "libraryName"
```

Description

Specifies the default name of the library for the *Library* field in the New File form, accessed by choosing *File – New*.

The default is none.

GUI Equivalent

Command:	*File – New*
Field:	*Library*

Examples

```
envGetVal("graphic" "defaultNewLibName")
envSetVal("graphic" "defaultNewLibName" 'string "MyLib")
```

Related Topics

New File form

© 2023 Cadence Design Systems, Inc.

defaultNewViewName

```
graphic defaultNewViewName string "any_valid_view_name"
```

Description

Specifies the default name of the view for the *View* field in the New File form, accessed by choosing *File – New*.

The default is none.

GUI Equivalent

Command:	*File – New*
Field:	*View*

Examples

```
envGetVal("graphic" "defaultNewViewName")
envSetVal("graphic" "defaultNewViewName" 'string "schematic")
```

Related Topics

New File form

© 2023 Cadence Design Systems, Inc.

defaultOpenCellName

```
graphic defaultOpenCellName string "any_cell_name"
```

Description

Specifies the default name of the cell for the *Cell* field in the *Open File* form, accessed by choosing *File – Open*.

The default is none.

GUI Equivalent

Command:	*File – Open*
Field:	*Cell*

Examples

```
envGetVal("graphic" "defaultOpenCellName")
envSetVal("graphic" "defaultOpenCellName" 'string "MyCell")
```

Related Topics

Open File Form

© 2023 Cadence Design Systems, Inc.

defaultOpenLibName

```
graphic defaultOpenLibName string "any_library_name"
```

Description

Specifies the default name of the library for the *Library* field in the *Open File* form, accessed by choosing *File – Open*.

The default is none.

GUI Equivalent

Command:	*File – Open*
Field:	*Library*

Examples

```
envGetVal("graphic" "defaultOpenLibName")
envSetVal("graphic" "defaultOpenLibName" 'string "MyLib")
```

Related Topics

Open File Form

© 2023 Cadence Design Systems, Inc.

defaultOpenViewName

```
graphic defaultOpenViewName string "any_valid_view_name"
```

Description

Specifies the default name of the view for the *View* field in the *Open File* form, accessed by choosing *File – Open*.

The default is none.

GUI Equivalent

Command:	*File – Open*
Field:	*View*

Examples

```
envGetVal("graphic" "defaultOpenViewName")
envSetVal("graphic" "defaultOpenViewName" 'string "layout")
```

Related Topics

Open File Form

© 2023 Cadence Design Systems, Inc.

defaultToolName

```
graphic defaultToolName string "any_valid_tool_name"
```

Description

Specifies the default tool name in the *Application – Open with* field in the New File form, accessed by choosing *File – New*.

The default is none.

GUI Equivalent

Command:	*File – New*
Field:	*Application – Open with*

Examples

```
envGetVal("graphic" "defaultToolName")
envSetVal("graphic" "defaultToolName" 'string "Layout XL")
```

Related Topics

New File form

© 2023 Cadence Design Systems, Inc.

defCreatePinConnModel

```
layout defCreatePinConnModel cyclic { "strong" | "weak" }
```

Description

Determines the default connectivity model while creating pins.

The default is `strong`.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Connectivity*

Examples

```
envGetVal("layout" "defCreatePinConnModel")
envSetVal("layout" "defCreatePinConnModel" 'cyclic "weak")
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

delDetachedBlockages

```
layout delDetachedBlockages boolean { t | nil }
```

Description

Deletes blockages which would otherwise remain after an instance containing blockages is flattened.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Hierarchy – Flatten*
Field:	*Detached blockages*

Examples

```
envGetVal("layout" "delDetachedBlockages")
envSetVal("layout" "delDetachedBlockages" 'boolean t)
```

Related Topics

Flatten Form

© 2023 Cadence Design Systems, Inc.

deleteFluidShapes

```
layout deleteFluidShapes boolean { t | nil }
```

Description

Deletes the fluid shapes when flattening Pcell instances.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "deleteFluidShapes")
envSetVal("layout" "deleteFluidShapes" 'boolean t)
```

Related Topics

Delete Form

© 2023 Cadence Design Systems, Inc.

deleteTarget

```
layout deleteTarget cyclic { "unknown" | "selected" | "net" | "pathSegment" }
```

Description

Checks which selected objects are deleted when you use the *Edit – Delete* command.

The default is `selected`.

GUI Equivalent

Command:	*Edit – Delete*
Field:	*Net Interconnect, Selected, Segment of a Path*

Examples

```
envGetVal("layout" "deleteTarget")
envSetVal("layout" "deleteTarget" 'cyclic "unknown")
envSetVal("layout" "deleteTarget" 'cyclic "pathSegment")
```

Related Topics

Delete Form

© 2023 Cadence Design Systems, Inc.

descendEIPPopupSelectionWindow

```
layout descendEIPPopupSelectionWindow cyclic { "off" | "shortform" | "fullform" |
     "highlightedobject" }
```

Description

Lists potential overlapping target objects under the selected object when you invoke the *Edit in Place* or *Descend* command. The valid values are:

- `off`: Default. Details of overlapping target objects are not displayed.
- `shortform`: Lists the overlapping target objects and destination cellviews.
- `fullform`: Lists the overlapping target objects, destination cellviews, and hierarchy levels.
- `highlightedobject`: Lists the overlapping targets objects, destination cellviews, and hierarchy levels only for the selected object.

GUI Equivalent

None

Examples

```
envGetVal("layout" "descendEIPPopupSelectionWindow")
envSetVal("layout" "descendEIPPopupSelectionWindow" 'cyclic "highlightedobject")
```

Related Topics

Hierarchical Designs

© 2023 Cadence Design Systems, Inc.

descendPromptForView

```
layout descendPromptForView boolean { t | nil }
```

Description

Determines whether to open the *View Name* field in the Descend form when you click an instance.

In *View Name* field of the Descend form, you can choose a different view to open.

If there is only one view of the master cell, Descend does not prompt you for a view name.

The default view is the view name of the selected object, when the default value `nil` is used.

GUI Equivalent

Command:	*Edit – Hierarchy – Descend Read*
Field:	*Prompt For View Name*

Examples

```
envGetVal("layout" "descendPromptForView")
envSetVal("layout" "descendPromptForView" 'boolean t)
```

Related Topics

Descend Options Form

© 2023 Cadence Design Systems, Inc.

dimGridLines

```
layout dimGridLines boolean { t | nil }
graphic dimGridLines boolean { t | nil }
```

Description

Checks if the grid lines appear dimmed on the canvas.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Dim Grid Lines*

Examples

```
envGetVal("layout" "dimGridLines")
envSetVal("layout" "dimGridLines" 'boolean t)
envGetVal("graphic" "dimGridLines")
envSetVal("graphic" "dimGridLines" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

dimmingAutoEnabled

```
layout dimmingAutoEnabled boolean { t | nil }
graphic dimmingAutoEnabled boolean { t | nil }
```

Description

Determines whether or not to automatically dim the design when you select an object or when the design contains highlight sets or probes.

The default is `nil` for graphic, and `t` for layout.

GUI Equivalent

Command:	*Options – Display*
Field:	*Automatic Dimming*

Examples

```
envGetVal("layout" "dimmingAutoEnabled")
envSetVal("layout" "dimmingAutoEnabled" 'boolean nil)
envGetVal("graphic" "dimmingAutoEnabled")
envSetVal("graphic" "dimmingAutoEnabled" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

dimmingInGray

```
graphic dimmingInGray boolean { t | nil }
```

Description

Dims a design in gray. This help you to easily follow highlighted nets when tracing nets in a hierarchy.

The default value is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "dimmingInGray")
envSetVal("graphic" "dimmingInGray" 'boolean t)
```

This environment variable can also be set for a specific window. For example:

```
window = hiGetCurrentWindow()
window~> dimmingInGray = t
```

Related Topics

dimGridLines

© 2023 Cadence Design Systems, Inc.

dimmingIntensity

```
layout dimmingIntensity int luminosity
graphic dimmingIntensity int luminosity
```

Description

Determines the luminosity of the dimmed objects.

When set to `0`, nothing is dimmed. When set to `100`, dimmed objects are not visible.

The default is `50`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Dim Intensity*

Examples

```
envGetVal("layout" "dimmingIntensity")
envSetVal("layout" "dimmingIntensity" 'int 70)
envGetVal("graphic" "dimmingIntensity")
envSetVal("graphic" "dimmingIntensity" 'int 70)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

dimmingOn

```
layout dimmingOn boolean { t | nil }
graphic dimmingOn boolean { t | nil }
```

Description

Specifies whether you can change the luminosity of the objects in the cellview.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Enable Dimming*

Examples

```
envGetVal("layout" "dimmingOn")
envSetVal("layout" "dimmingOn" 'boolean t)
envGetVal("graphic" "dimmingOn")
envSetVal("graphic" "dimmingOn" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

dimmingScope

```
layout dimmingScope string { "none" | "all" | "outside" | "eipSurround" }
graphic dimmingScope string { "none" | "all" | "outside" | "eipSurround" }
```

Description

Determines how objects in the cellview are dimmed.

- `none`: Default.No dimming is applied.
- `outside`: Dims everything but the instance, figure group, or any object you are editing in place.
- `all`: The design is completely dimmed as soon as you select this option. If you select an object and the dimming is set to 100, only the selection box is displayed.
- `eipSurround`: Dims all the objects surrounding the EIP hierarchy.

GUI Equivalent

Command:	*Options – Display*
Field:	*Scope*

Examples

```
envGetVal("layout" "dimmingScope")
envSetVal("layout" "dimmingScope" 'string "eipSurround")
envGetVal("graphic" "dimmingScope")
envSetVal("graphic" "dimmingScope" 'string "all")
envSetVal("graphic" "dimmingScope" 'string "outside")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

disableLayerFuncSnapToMfgGrid

```
layout disableLayerFuncSnapToMfggrid boolean { t | nil }
```

Description

Specifies that the `leLayer*`, `leMergeShapes` and `leSizeShape` functions do not put data on grid.

The default is `t`. When set to `nil`, it forces the `le` functions and the corresponding functions from the menus to put data results on grid.

Note: The `dbLayer*` functions do not put date results back onto the manufacturing grid and are unaffected by the environment variable.

GUI Equivalent

None

Examples

```
envGetVal("layout" "disableLayerFuncSnapToMfggrid")
envSetVal("layout" "disableLayerFuncSnapToMfggrid" 'boolean nil)
```

Related Topics

mfgGridResolution

© 2023 Cadence Design Systems, Inc.

disableMissingLayerPopUp

```
graphic disableMissingLayerPopUp boolean { t | nil }
```

Description

Disables the missing layer pop-up window.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "disableMissingLayerPopUp")
envSetVal("graphic" "disableMissingLayerPopUp" 'boolean t)
```

Related Topics

Layer Generation

© 2023 Cadence Design Systems, Inc.

disableRoundPathStyle

```
layout disableRoundPathStyle boolean { t | nil }
```

Description

Checks if the round path style option is disabled from the *End Type* field of the Create Path form.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "disableRoundPathStyle")
envSetVal("layout" "disableRoundPathStyle" 'boolean t)
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

discardCVEditOnWindowClose

```
graphic discardCVEditOnWindowClose boolean { t | nil }
```

Description

Controls whether the changes done in the cellview are saved when you click *No* in the *Save Changes* dialog box.

The default is `t`. When set to `nil`, the changes done in the cellview are preserved in the memory when you click *No* in the *Save Changes* dialog box.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "discardCVEditOnWindowClose")
envSetVal("graphic" "discardCVEditOnWindowClose" 'boolean nil)
```

Related Topics

Interactive Snapping

© 2023 Cadence Design Systems, Inc.

displayFigGroupName

```
layout displayFigGroupName boolean { t | nil }
```

Description

Determines whether to display only the group name for a figure group or not.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Group Name Only*

Examples

```
envGetVal("layout" "displayFigGroupName")
envSetVal("layout" "displayFigGroupName" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayFigGroupShapes

```
graphic displayFigGroupShapes boolean { t | nil }
```

Description

Determines whether to display the group shapes or only the group name and the bounding box for a figure group.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "displayFigGroupShapes")
envSetVal("graphic" "displayFigGroupShapes" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayInstanceOrientation

```
layout displayInstanceOrientation boolean { t | nil }
graphic displayInstanceOrientation boolean { t | nil }
```

Description

Determines whether to display instance orientation in the cellview or not.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Instance Orientation*

Examples

```
envGetVal("layout" "displayInstanceOrientation")
envSetVal("layout" "displayInstanceOrientation" 'boolean t)
envGetVal("graphic" "displayInstanceOrientation")
envSetVal("graphic" "displayInstanceOrientation" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayMissingMasterMarkers

```
graphic displayMissingMasterMarkers boolean { t | nil }
```

Description

Lets you enable or disable the display of missing master markers.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "displayMissingMasterMarkers")
envSetVal("graphic" "displayMissingMasterMarkers" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayOptionsFile

```
layout displayOptionsFile string "fileName"
```

Description

Specifies the file to which you want to store or from which you want to load the settings.

The default is "`~/.cdsenv`".

GUI Equivalent

Command:	*Options – Display*
Field:	*File*

Examples

```
envGetVal("layout" "displayOptionsFile")
envSetVal("layout" "displayOptionsFile" 'string "~/.cdsenv")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayPCellOrientation

```
graphic displayPCellOrientation boolean { t | nil }
```

Description

Displays the orientation for Pcells.

The default is `t`.

Note: The environment variable, `displayInstanceOrientation`, must be enabled for the `displayPCellOrientation` environment variable to work.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "displayPCellOrientation ")
envSetVal("graphic" "displayPCellOrientation " 'boolean nil)
```

Related Topics

displayInstanceOrientation

© 2023 Cadence Design Systems, Inc.

displayPinNames

```
layout displayPinNames boolean { t | nil }
graphic displayPinNames boolean { t | nil }
```

Description

Determines whether to display pin name text displays or labels in the cellview or not.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Pin Names*

Examples

```
envGetVal("layout" "displayPinNames")
envSetVal("layout" "displayPinNames" 'boolean nil)
envGetVal("graphic" "displayPinNames")
envSetVal("graphic" "displayPinNames" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayPRBoundaryAtCurrentLevelOnly

```
graphic displayPRBoundaryAtCurrentLevelOnly boolean { t | nil }
layout displayPRBoundaryAtCurrentLevelOnly boolean { t | nil }
```

Description

Determines whether to display the PR boundary at the current level only or not.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Current Level PR Boundary*

Examples

```
envGetVal("graphic" "displayPRBoundaryAtCurrentLevelOnly")
envSetVal("graphic" "displayPRBoundaryAtCurrentLevelOnly" 'boolean t)
envGetVal("layout" "displayPRBoundaryAtCurrentLevelOnly")
envSetVal("layout" "displayPRBoundaryAtCurrentLevelOnly" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayPRHighestResolution

```
layout displayPRHighestResolution boolean { t | nil }
graphic displayPRHighestResolution boolean { t | nil }
```

Description

Determines whether to use the filter size for objects that is bigger or equal to the filter size used for shapes.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Highest Resolution for P&R Objects*

Examples

```
envGetVal("layout" "displayPRHighestResolution")
envSetVal("layout" "displayPRHighestResolution" 'boolean t)
envGetVal("graphic" "displayPRHighestResolution")
envSetVal("graphic" "displayPRHighestResolution" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayRefPoint

```
layout displayRefPoint boolean { t | nil }
```

Description

Specifies if the reference point is displayed in the cellview or not.

The default is `nil`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Display Reference Point*

Examples

```
envGetVal("layout" "displayRefPoint")
envSetVal("layout" "displayRefPoint" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

displayResolution

```
graphic displayResolution string { "Low" | "Medium" | "High" | "Very High" }
layout displayResolution string { "Low" | "Medium" | "High" | "Very High" }
```

Description

Lets you set the display resolution in the design canvas.

The default is `Medium`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Resolution*

Examples

```
envGetVal("graphic" "displayResolution")
envSetVal{"graphic" "displayResolution" 'string "Low")
envGetVal("layout" "displayResolution")
envSetVal{"layout" "displayResolution" 'string "Very High")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayRowName

```
layout displayRowName boolean { t | nil }
graphic displayRowName boolean { t | nil }
```

Description

Displays the names of placed rows.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Row Name*

Examples

```
envGetVal("layout" "displayRowName")
envSetVal("layout" "displayRowName" 'boolean t)
envGetVal("graphic" "displayRowName")
envSetVal("graphic" "displayRowName" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayRowSite

```
layout displayRowSite boolean { t | nil }
graphic displayRowSite boolean { t | nil }
```

Description

Displays the position of row sites, which are represented by a cross marker on the row.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Display Row Site*

Examples

```
envGetVal("layout" "displayRowSite")
envSetVal("layout" "displayRowSite" 'boolean t)
envGetVal("graphic" "displayRowSite")
envSetVal("graphic" "displayRowSite" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayRulerDistanceAtEnd

```
layout displayRulerDistanceAtEnd boolean { t | nil }
```

Description

Displays the ruler length at the end of the ruler during the drag operation when you create a single- or multi-segment ruler. For an auto ruler, the length is displayed at the center of the ruler segment during the drag operation.

The default value is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "displayRulerDistanceAtEnd")
envSetVal("layout" "displayRulerDistanceAtEnd" 'boolean t)
```

Related Topics

Rulers

© 2023 Cadence Design Systems, Inc.

displayRulerInHierarchy

```
graphic displayRulerInHierarchy boolean { t | nil }
```

Description

Specifies whether saved rulers from lower hierarchy levels of a cellview are displayed when you increase the display level.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "displayRulerInHierarchy")
envSetVal("graphic" "displayRulerInHierarchy" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayRulerMiddlePoint

```
graphic displayRulerMiddlePoint boolean { t | nil }
```

Description

Displays a tick and the measurement value in the middle point of a ruler. You can extend the tick from both sides of the ruler.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "displayRulerMiddlePoint")
envSetVal("graphic" "displayRulerMiddlePoint" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displaySchDashText

```
graphic displaySchDashText boolean { t | nil }
```

Description

Specifies how to display text or labels.

The default is `nil`, which means that the text or labels always display, even when the window is zoomed out. When set to `t`, text or labels are displayed as small lines when the schematic or symbol is zoomed out.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "displaySchDashText")
envSetVal("graphic" "displaySchDashText" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displaySelectionHighlightWhenDimmed

```
layout displaySelectionHighlightWhenDimmed boolean { t | nil }
graphic displaySelectionHighlightWhenDimmed boolean { t | nil }
```

Description

Specifies the display of selected objects in their original colors when the design or other objects in the design are set to be dimmed.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*True Color Selection only*

Examples

```
envGetVal("layout" "displaySelectionHighlightWhenDimmed")
envSetVal("layout" "displaySelectionHighlightWhenDimmed" 'boolean nil)
envGetVal("graphic" "displaySelectionHighlightWhenDimmed")
envSetVal("graphic" "displaySelectionHighlightWhenDimmed" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displaySigDigits

```
layout displaySigDigits int numberOfDecimalDigits
```

Description

Controls the number of decimal digits that display during the dynamic display of measurements of shapes. Enter an integer between `0` and `10`.

The default is `4`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "displaySigDigits")
envSetVal("layout" "displaySigDigits" 'int 7)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayStartLevel

```
layout displayStartLevel int startLevel
```

Description

Specifies the start level for viewing the hierarchy in the Set Area View Level form. Specify any integer between `0` and `32`.

The default is `0`.

GUI Equivalent

Command:	*View – Area Display – Set*
Field:	*Display Levels – From*

Examples

```
envGetVal("layout" "displayStartLevel")
envSetVal("layout" "displayStartLevel" 'int 15)
```

Related Topics

Display Options Form

Set Area View Level Form

© 2023 Cadence Design Systems, Inc.

displayStopLevel

```
layout displayStopLevel int stopLevel
```

Description

Specifies the stop level for viewing the hierarchy in the Set Area View Level form. Specify any integer between `0` and `32`.

The default is `0`.

GUI Equivalent

Command:	*View – Area Display – Set*
Field:	*Display Levels – To*

Examples

```
envGetVal("layout" "displayStopLevel")
envSetVal("layout" "displayStopLevel" 'int 20)
```

Related Topics

Set Area View Level Form

© 2023 Cadence Design Systems, Inc.

displayStretchHandles

```
layout displayStretchHandles boolean { t | nil }
graphic displayStretchHandles boolean { t | nil }
```

Description

Specifies whether stretch handles are displayed for stretchable Pcells.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Stretch Handles*

Examples

```
envGetVal("layout" "displayStretchHandles")
envSetVal("layout" "displayStretchHandles" 'boolean nil)
envGetVal("graphic" "displayStretchHandles")
envSetVal("graphic" "displayStretchHandles" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

displayViaShapes

```
layout displayViaShapes boolean { t | nil }
graphic displayViaShapes boolean { t | nil }
```

Description

Displays via attributes.

The default is `nil`, which means that the vias are displayed as the bounding box of each via. When set to `t`, via attributes are graphically displayed, resulting in detailed images. For large designs, setting via shapes to `t` increases the amount of time it takes to render a design.

GUI Equivalent

Command:	*Options – Display*
Field:	*Via Shapes*

Examples

```
envGetVal("layout" "displayViaShapes")
envSetVal("layout" "displayViaShapes" 'boolean t)
envGetVal("graphic" "displayViaShapes")
envSetVal("graphic" "displayViaShapes" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

doNotShowChopPRBDialog

```
layout doNotShowChopPRBDialog boolean { t | nil }
```

Description

Lets you select if you want to display the warning dialog box when the *Chop* command is run outside the PR boundary or when there is no PR boundary in the design.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "doNotShowChopPRBDialog")
envSetVal("layout" "doNotShowChopPRBDialog" 'boolean t)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

dotsOn

```
layout dotsOn boolean { t | nil }
graphic dotsOn boolean { t | nil }
```

Description

Determines whether to display diamond for dot shapes and markers at via origins in the cellview. Enabling `dotsOn` can affect the redraw performance.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Dot Pins*

Examples

```
envGetVal("layout" "dotsOn")
envSetVal("layout" "dotsOn" 'boolean nil)
envGetVal("graphic" "dotsOn")
envSetVal("graphic" "dotsOn" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

dotStyleMajorGrid

```
layout dotStyleMajorGrid boolean { t | nil }
graphic dotStyleMajorGrid boolean { t | nil }
```

Description

Dims the dots of the major grid.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Dim Major Dots*

Examples

```
envGetVal("layout" "dotStyleMajorGrid")
envSetVal("layout" "dotStyleMajorGrid" 'boolean t)
envGetVal("graphic" "dotStyleMajorGrid")
envSetVal("graphic" "dotStyleMajorGrid" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

dragDisplayNetName

```
graphic dragDisplayNetName boolean { t | nil }
```

Description

Displays the net name of an object when you drag the object during the move, copy, or stretch operation.

The default is `nil`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Drag net name*

Examples

```
envGetVal("graphic" "dragDisplayNetName")
envSetVal("graphic" "dragDisplayNetName" 'boolean t)
```

Related Topics

Net Name Display Options Form

© 2023 Cadence Design Systems, Inc.

drawDottedGridOn

```
layout drawDottedGridOn boolean { t | nil }
graphic drawDottedGridOn boolean { t | nil }
```

Description

Determines whether the grid is displayed as dots or lines, when `drawGridOn` is set to `t`.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Grid Controls – Type – dots* and *lines*

Examples

```
envGetVal("layout" "drawDottedGridOn")
envSetVal("layout" "drawDottedGridOn" 'boolean nil)
envGetVal("graphic" "drawDottedGridOn")
envSetVal("graphic" "drawDottedGridOn" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

drawEngine

```
graphic drawEngine int { 0 | 1 }
```

Description

Determines whether or not all data in a cellview is rendered.

The default is `0`, which means that all data is drawn including that which will be obscured by data on overlying layers.

When set to `1`, which means that only data which will ultimately be viewable is drawn. This function works best on large designs with closely placed data. It is slower on small designs, as the pruning of overlapping data takes longer than redrawing a small amount of data.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "drawEngine")
envSetVal("graphic" "drawEngine" 'int 1)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

drawGridOn

```
layout drawGridOn boolean { t | nil }
graphic drawGridOn boolean { t | nil }
```

Description

Determines whether the grid is displayed.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Grid Controls – Type – none*

Examples

```
envGetVal("layout" "drawGridOn")
envSetVal("layout" "drawGridOn" 'boolean nil)
envGetVal("graphic" "drawGridOn")
envSetVal("graphic" "drawGridOn" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

drawInstancePinNames

```
layout drawInstancePinNames boolean { t | nil }
graphic drawInstancePinNames boolean { t | nil }
```

Description

Determines whether pin names are displayed in instances.

The layout default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Instance Pin Names*

Examples

```
envGetVal("layout" "drawInstancePinNames")
envSetVal("layout" "drawInstancePinNames" 'boolean nil)
envGetVal("graphic" "drawInstancePinNames")
envSetVal("graphic" "drawInstancePinNames" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

drawInstancePins

```
layout drawInstancePins boolean { t | nil }
graphic drawInstancePins boolean { t | nil }
```

Description

Controls the display of the pins that are one level more than the stop display level. Enabling `drawInstancePins` can affect the redraw performance.

The layout default is `nil`, the graphic default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Instance Pins*

Examples

```
envGetVal("layout" "drawInstancePins")
envSetVal("layout" "drawInstancePins" 'boolean t)
envGetVal("graphic" "drawInstancePins")
envSetVal("graphic" "drawInstancePins" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

drawLabelsSurround

```
graphic drawLabelsSurround boolean { t | nil }
```

Description

Draws a halo around label and text displays.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "drawLabelsSurround")
envSetVal("graphic" "drawLabelsSurround" 'boolean t)
```

Related Topics

Halo Editor Form

© 2023 Cadence Design Systems, Inc.

drawPRBoundary

```
graphic drawPRBoundary boolean { t | nil }
```

Description

Determines whether the place and route boundary of instances is displayed, or the instance bounding box is displayed.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "drawPRBoundary")
envSetVal("graphic" "drawPRBoundary" 'boolean nil)
```

Related Topics

instanceDrawingMode

© 2023 Cadence Design Systems, Inc.

drawSmallLabelOrigin

```
graphic drawSmallLabelOrigin boolean { t | nil }
```

Description

Determines whether the small labels should be displayed in the design.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "drawSmallLabelOrigin")
envSetVal("graphic" "drawSmallLabelOrigin" 'boolean nil)
```

Related Topics

Label Generation for Overlapping Shapes

© 2023 Cadence Design Systems, Inc.

drawSurroundingOn

```
layout drawSurroundingOn boolean { t | nil }
graphic drawSurroundingOn boolean { t | nil }
```

Description

Determines whether the surrounding design is displayed when you edit in place.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*EIP Surround*

Examples

```
envGetVal("layout" "drawSurroundingOn")
envSetVal("layout" "drawSurroundingOn" 'boolean nil)
envGetVal("graphic" "drawSurroundingOn")
envSetVal("graphic" "drawSurroundingOn" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

drawTraversedInstBbox

```
layout drawTraversedInstBbox boolean { t | nil }
graphic drawTraversedInstBbox boolean { t | nil }
```

Description

Draws a bounding box around an instance that traverses to a size smaller than the `fitIntoClusterBoundary`. The traversing usually occurs when you zoom out of the design.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Traversed Instance BBox*

Examples

```
envGetVal("layout" "drawTraversedInstBbox")
envSetVal("layout" "drawTraversedInstBbox" 'boolean t)
envGetVal("graphic" "drawTraversedInstBbox")
envSetVal("graphic" "drawTraversedInstBbox" 'boolean t)
```

Related Topics

fitIntoClusterBoundary

Display Options Form

© 2023 Cadence Design Systems, Inc.

drawTraversedInstName

```
layout drawTraversedInstName boolean { t | nil }
graphic drawTraversedInstName boolean { t | nil }
```

Description

Displays the name of the instance for the traversed instance.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Traversed Instance Name*

Examples

```
envGetVal("layout" "drawTraversedInstName")
envSetVal("layout" "drawTraversedInstName" 'boolean t)
envGetVal("graphic" "drawTraversedInstName")
envSetVal("graphic" "drawTraversedInstName" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

drdEditSmartSnapAllowedWidthSnap

```
layout drdEditSmartSnapAllowedWidthSnap boolean { t | nil }
```

Description

Determines whether discrete dimension is enabled.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "drdEditSmartSnapAllowedWidthSnap")
envSetVal("layout" "drdEditSmartSnapAllowedWidthSnap" 'boolean nil)
```

Related Topics

Global Grid Anchoring

© 2023 Cadence Design Systems, Inc.

drfPath

```
graphic drfPath string path
```

Description

Specifies the path to the Display Resource File.

The default is none.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "drfPath")
envSetVal("graphic" "drfPath" 'string "~/.display.drf")
```

Related Topics

Display Resource File

© 2023 Cadence Design Systems, Inc.

dynamicHilightOn

```
layout dynamicHilightOn boolean { t | nil }
graphic dynamicHilightOn boolean { t | nil }
```

Description

Determines whether dynamic highlighting is on.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Dynamic Highlight*

Examples

```
envGetVal("layout" "dynamicHilightOn")
envSetVal("layout" "dynamicHilightOn" 'boolean t)
envGetVal("graphic" "dynamicHilightOn")
envSetVal("graphic" "dynamicHilightOn" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

dynamicMeasurementBackgroundColor

```
layout dynamicMeasurementBackgroundColor string color
```

Description

Lets you select the background color for the dynamic measurement labels.

The default is `black`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "dynamicMeasurementBackgroundColor")
envSetVal("layout" "dynamicMeasurementBackgroundColor" 'string "white")
```

Related Topics

Enabling Information Balloons and Dynamic Measurements

© 2023 Cadence Design Systems, Inc.

dynamicMeasurementDrawBackground

```
layout dynamicMeasurementDrawBackground boolean { t | nil }
```

Description

Turns off the background color.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "dynamicMeasurementDrawBackground")
envSetVal("layout" "dynamicMeasurementDrawBackground" 'boolean nil)
```

Related Topics

Enabling Information Balloons and Dynamic Measurements

© 2023 Cadence Design Systems, Inc.

dynamicMeasurementDisplay

```
layout dynamicMeasurementDisplay boolean { t | nil }
```

Description

Lets you specify the dynamic display measurement between the pointer and the closest edges.

The default is `nil`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Dynamic Measurement*

Examples

```
envGetVal("layout" "dynamicMeasurementDisplay")
envSetVal("layout" "dynamicMeasurementDisplay" 'boolean t)
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

dynamicMeasurementEdge

```
layout dynamicMeasurementEdge boolean { t | nil }
```

Description

Lets you dynamically display measurement between the two vertices of an edge, the edge closest to the pointer.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "dynamicMeasurementEdge")
envSetVal("layout" "dynamicMeasurementEdge" 'boolean t)
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

dynamicMeasurementFontSize

```
layout dynamicMeasurementFontSize int fontSize
```

Description

Specifies the font size for dynamic measurement labels.

The default is `12`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "dynamicMeasurementFontSize")
envSetVal("layout" "dynamicMeasurementFontSize" 'int 14)
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

dynamicMeasurementFromCursor

```
layout dynamicMeasurementFromCursor boolean { t | nil }
```

Description

Displays measurement between the pointer and closest edge. It also displays the distance between the closest edges.

The default is `nil`. The dynamic measurement displayed is the distance between closest edges.

GUI Equivalent

None

Examples

```
envGetVal("layout" "dynamicMeasurementFromCursor")
envSetVal("layout" "dynamicMeasurementFromCursor" 'boolean t)
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

dynamicMeasurementLayerScope

```
layout dynamicMeasurementLayerScope string "layerScope"
```

Description

Lets you select the layers to be considered for dynamic measurement. The options are `All`, `Active`, `Edge`, `Palette`, or `Selectable layers`.

The default is `All`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Layer Scope*

Examples

```
envGetVal("layout" "dynamicMeasurementLayerScope")
envSetVal("layout" "dynamicMeasurementLayerScope" 'string "Active")
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

dynamicMeasurementTextColor

```
layout dynamicMeasurementTextColor string { "y0" | "y1" | "y2" | "y3" | "y4" | "y5"
     | "y6" | "y7" | "y8" | "y9" }
```

Description

Lets you set and dynamically change the color of the measurement information.

The default is `y0`.

GUI Equivalent

Command:	*Options – Dynamic Display – Dynamic Measurement*
Field:	*Color*

Examples

```
envGetVal("layout" "dynamicMeasurementTextColor")
envSetVal("layout" "dynamicMeasurementTextColor" 'string "y1")
```

Related Topics

Dynamic Display Form

© 2023 Cadence Design Systems, Inc.

dynamicProbe

```
layout dynamicProbe boolean { t | nil }
```

Description

Probes any object dynamically over which the mouse pointer hovers.

The default is `t`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Dynamic Highlight*

Examples

```
envGetVal("layout" "dynamicProbe")
envSetVal("layout" "dynamicProbe" 'boolean nil)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

editInstanceBoundaryMode

```
layout editInstanceBoundaryMode cyclic { "BBox" | "prBoundary" | "snapBoundary" }
```

Description

Specifies whether the *Flip* and *Rotate* commands use the instance bounding box (`BBox`), the place and route boundary (`prBoundary`), or the snap boundary (`snapBoundary`) as the instance border when operating on instances and mosaics.

The default is `BBox`. For example, if you have a standard cell with a large label that extends well beyond the PR boundary, the bounding box of that cell will be much larger than its place and route boundary. Set this environment variable to `prBoundary` to have *Flip* and *Rotate* ignore the bounding box and consider only the PR boundary when operating on the cell.

Note: If you set this environment variable to `prBoundary` or `snapBoundary` and the instance or mosaic does not have the corresponding boundary, then `BBox` is used instead.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Instance Editing Mode*

Examples

```
envGetVal("layout" "editInstanceBoundaryMode")
envSetVal("layout" "editInstanceBoundaryMode" 'cyclic "prBoundary")
envSetVal("layout" "editInstanceBoundaryMode" 'cyclic "snapBoundary")
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

editorOptionsFile

```
layout editorOptionsFile string "fileName"
```

Description

Specifies the file to which you want to store or from which you want to load the settings.

The default is "`~/.cdsenv`".

GUI Equivalent

Command:	*Options – Editor*
Field:	*File*

Examples

```
envGetVal("layout" "editorOptionsFile")
envSetVal("layout" "editorOptionsFile" 'string "~/.cdsenv")
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

eipHighlightColor

```
layout eipHighlightColor string haloColor
```

Description

Specifies the color of the halo around a figure group and instance.

The default is `magenta`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "eipHighlightColor")
envSetVal("layout" "eipHighlightColor" 'string "yellow")
```

Related Topics

Selection Protection Highlight Options Form

© 2023 Cadence Design Systems, Inc.

enableAlignToolBarReferenceObject

```
layout enableAlignToolBarReferenceObject boolean { t | nil }
```

Description

Lets you enable the reference object on the *Align* toolbar.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "enableAlignToolBarReferenceObject")
envSetVal("layout" "enableAlignToolBarReferenceObject" 'boolean nil)
```

Related Topics

Aligning Objects by Using the Align Toolbar

© 2023 Cadence Design Systems, Inc.

enableCopyAndPaste

```
layout enableCopyAndPaste boolean { t | nil }
```

Description

Stores the selected objects and the reference point automatically in the buffer after the reference point for the copy command has been specified when set to `t` in the `Copy` command. You can use the `Paste` command to paste the objects stored in the buffer.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "enableCopyAndPaste")
envSetVal("layout" "enableCopyAndPaste" 'boolean t)
```

Related Topics

Object Copying

© 2023 Cadence Design Systems, Inc.

enableEIPEnhancements

```
layout enableEIPEnhancements boolean { t | nil }
```

Description

Lets you enable the *Edit In Place* enhancements when set to on.

- When editing in place in a hierarchy, clicking a shape at the top level performs return to top.
- For figures inside a group inside an instance, by default, the *Edit In Place* target is the figure (descend into instance and descend into the group). You can use the `Space` key to cycle through the instance and group or just the instance.
- When you return from instance and group, you can specify the options in the Group Return form.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "enableEIPEnhancements")
envSetVal("layout" "enableEIPEnhancements" 'boolean nil)
```

Related Topics

Hierarchical Designs

Group Return Form

© 2023 Cadence Design Systems, Inc.

enableLayoutIntegrityCheck

```
layoutXL enableLayoutIntegrityCheck boolean { t | nil }
```

Description

Lets you enable the *Layout Integrity Check* command.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layoutXL" "enableLayoutIntegrityCheck")
envSetVal("layoutXL" "enableLayoutIntegrityCheck" 'boolean nil)
```

Related Topics

Layout Integrity Check Form

© 2023 Cadence Design Systems, Inc.

exceptOverlapSnapMode

```
layout exceptOverlapSnapMode cyclic { "full" | "partial" }
```

Description

Controls the except overlap snap mode to be set to `full` or `partial`.

The default is `full`. When set to `partial`, the partial mode behavior is the same as full mode.

GUI Equivalent

None

Examples

```
envGetVal("layout" "exceptOverlapSnapMode")
envSetVal("layout" "exceptOverlapSnapMode" 'cyclic "partial")
```

Related Topics

Layer Shapes Exempted from Snapping by Using exceptOverlap

© 2023 Cadence Design Systems, Inc.

expandPinBusExpression

```
layout expandPinBusExpression boolean { t | nil }
```

Description

Expands the pin bus expression.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "expandPinBusExpression")
envSetVal("layout" "expandPinBusExpression" 'boolean t)
```

Related Topics

Placing a Pin Array

© 2023 Cadence Design Systems, Inc.

extendPathLengthCheck

```
layout extendPathLengthCheck boolean { t | nil }
```

Description

Checks paths or path segments created with a segment of less than half of the path width and a warning is issued.

The default is `nil`, which means that no check is made and no warning is issued.

GUI Equivalent

None

Examples

```
envGetVal("layout" "extendPathLengthCheck")
envSetVal("layout" "extendPathLengthCheck" 'boolean t)
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

extractAfterGenerateAll

```
layout extractAfterGenerateAll boolean { t | nil }
```

Description

Runs connectivity extraction as part of the *Generate All From Source* command. Switch on this environment variable to see the incomplete nets in the design immediately after layout generation has run.

The default is `t`.

Note: This option has no effect on either the *Generate Selected From Source* or *Update Components And Nets* commands. Connectivity extraction for these commands is controlled using the `extractEnabled` environment variable.

GUI Equivalent

Command:	*Connectivity – Generate – All From Source – Generate*
Field:	*Extract Connectivity after Generation*

Examples

```
envGetVal("layout" "extractAfterGenerateAll")
envSetVal("layout" "extractAfterGenerateAll" 'boolean nil)
```

Related Topics

extractEnabled

© 2023 Cadence Design Systems, Inc.

extractCellviewIllegalConnectionLimit

```
layout extractCellviewIllegalConnectionLimit int numberOfConnectionViolations
```

Description

Specifies the maximum number of illegal hierarchical connection violations generated by the extractor for a particular cellview.

The default is `1000`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Limits*
Field:	*Maximum number of illegal connections: By cellview*

Examples

```
envGetVal("layout" "extractCellviewIllegalConnectionLimit")
envSetVal("layout" "extractCellviewIllegalConnectionLimit" 'int 750)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractCellviewInvalidConnectionLimit

```
layout extractCellviewInvalidConnectionLimit int
     numberOfInvalidConnectionsCellview
```

Description

Specifies the maximum number of invalid connection markers that can be generated by the extractor for a particular cellview.

The default is `1000`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Limits*
Field:	*Maximum number of invalid connections*

Examples

```
envGetVal("layout" "extractCellviewInvalidConnectionLimit")
envSetVal("layout" "extractCellviewInvalidConnectionLimit" 'int 750)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractCellviewOpenViolationLimit

```
layout extractCellviewOpenViolationLimit int openViolations
```

Description

Specifies the maximum number of open violations generated by the extractor for a particular cellview. Limiting the number can improve performance when you are editing either a large design or a net with a large number of instance terminals and terminals in an unrouted design.

The default is `5000`.



Important

This environment variable is considered only when `extractEnabled` and `extractVerifyOpenViolations` are switched on.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Limits*
Field:	*Maximum number of opens: By cellview*

Examples

```
envGetVal("layout" "extractCellviewOpenViolationLimit")
envSetVal("layout" "extractCellviewOpenViolationLimit" 'int 3000)
```

Related Topics

extractEnabled

extractVerifyOpenViolations

© 2023 Cadence Design Systems, Inc.

extractCellviewShortViolationLimit

```
layout extractCellviewShortViolationLimit int numberOfShortViolations
```

Description

Specifies the maximum number of short violations generated by the extractor for a particular cellview.

The default is `1000`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Limits*
Field:	*Maximum number of shorts: By cellview*

Examples

```
envGetVal("layout" "extractCellviewShortViolationLimit")
envSetVal("layout" "extractCellviewShortViolationLimit" 'int 750)
```

Related Topics

Connectivity Extraction in Layout XL

© 2023 Cadence Design Systems, Inc.

extractCellviewUnnecessaryPseudoParallelConnectionLimit

```
layoutXL extractCellviewUnnecessaryPseudoParallelConnectionLimit int
     numberOfViolations
```

Description

Specifies the maximum number of unnecessary pseudo-parallel connection violations generated by the extractor for a particular cellview.

The default is `1000`.

GUI Equivalent

Command:	*Options - Connectivity - Connectivity - Limits*
Field:	*Maximum number of unnecessary pseudo-parallel connections*

Examples

```
envGetVal("layoutXL" "extractCellviewUnnecessaryPseudoParallelConnectionLimit")
envSetVal("layoutXL" "extractCellviewUnnecessaryPseudoParallelConnectionLimit"
'int 500)
```

Related Topics

Limits tab

© 2023 Cadence Design Systems, Inc.

extractConnectUnassignedInstTerms

```
layout extractConnectUnassignedInstTerms boolean { t | nil }
```

Description

Ensures that an instance terminal is automatically connected to the net assigned to the overlapping shape.

The default is `t`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Controls*
Field:	*Connect unassigned instance pins*

Examples

```
envGetVal("layout" "extractConnectUnassignedInstTerms")
envSetVal("layout" "extractConnectUnassignedInstTerms" 'boolean nil)
```

Related Topics

Connectivity Extraction in Layout XL

© 2023 Cadence Design Systems, Inc.

extractConsiderAllCutLayersInVias

```
layout extractConsiderAllCutLayersInVias boolean { t | nil }
```

Description

Considers all cut layers inside vias as extractable, even if the cut layer is not listed in the valid layers constraint.

The default is `t`.



Important

Cadence recommends that you list all the extractable layers in the `validLayers` constraint. If there are no extractable layers defined in the constraint group, Layout XL is launched with the connectivity extractor disabled.

GUI Equivalent

None

Examples

```
envGetVal("layout" "extractConsiderAllCutLayersInVias")
envSetVal("layout" "extractConsiderAllCutLayersInVias" 'boolean nil)
```

Related Topics

validLayers

viaDef

© 2023 Cadence Design Systems, Inc.

extractCheckShortedTerminalsInPcells

```
layout extractCheckShortedTerminalsInPcells boolean { t | nil }
```

Description

Runs checks to detect cases where two terminals are shorted inside a Pcell when the corresponding instance terminals are assigned to different nets. In this case, a short marker is created between a level-1 pinFig and an overlapping hierarchical shape assigned to the net of a different terminal.

This option is available only if the Extract through pcells option in the Connectivity form is enabled and the `extractStopLevel` environment variable is greater than or equal to 1 because it involves a hierarchical shape.

The default is `t`.

GUI Equivalent

Command:	*Options - Connectivity - Connectivity*
Field:	*Check shorted terminals in pcells*

Examples

```
envGetVal("layout" "extractConsiderAllCutLayersInVias")
envSetVal("layout" "extractConsiderAllCutLayersInVias" 'boolean nil)
```

Related Topics

extractStopLevel

Connectivity Form

Hierarchical Extraction

© 2023 Cadence Design Systems, Inc.

extractDeepOffPinConnectionsToLevel

`layout extractDeepOffPinConnectionsToLevel int` *`integer`*

Description

Enables the connectivity extractor to extract off-pin connections present below the `extractStopLevel` limit. It enables the detection of shorts and opens for off-pin connections with better performance than `extractStopLevel>=1` but with some limitations. This option only extracts off-pin connections between a top-level shape and an assigned hierarchical shape. Additionally, it does not check for illegal hierarchical connections.

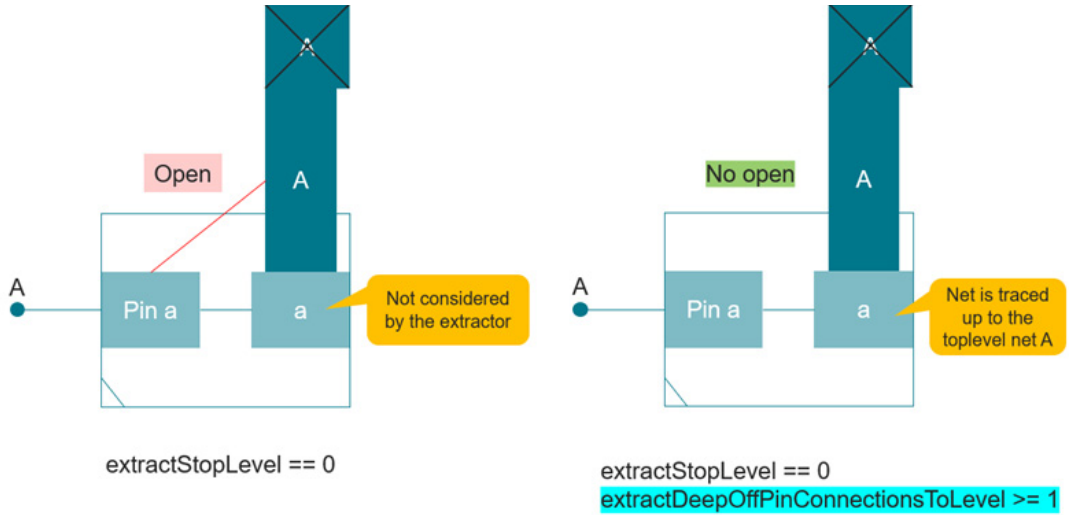
The default is 0, which means that by default this feature is disabled. It is active only when the specified value of `extractDeepOffPinConnectionsToLevel` is larger than the `extractStopLevel` value.

GUI Equivalent

None

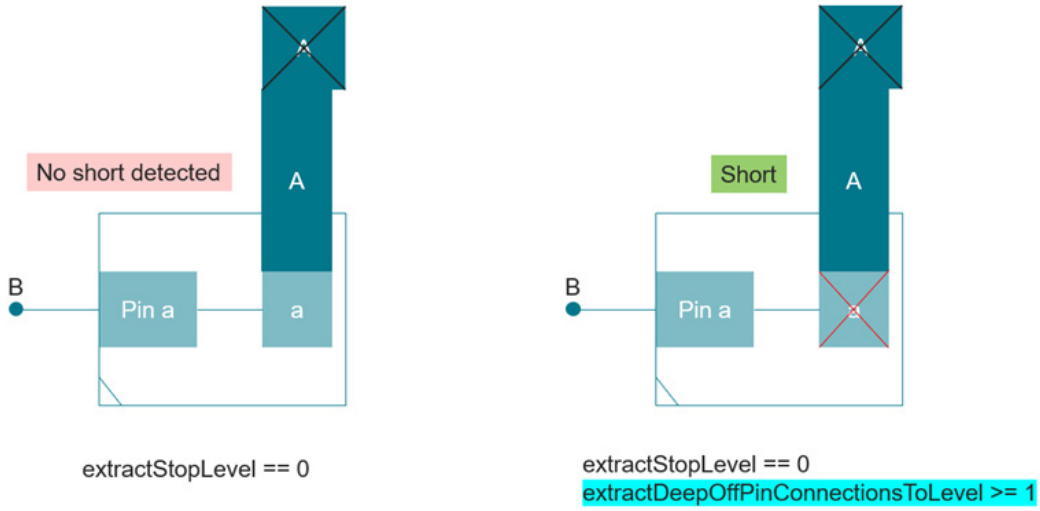
Additional Information

Case 1: The extractor does not extract the off-pin connection and hence reports an open with `extractStopLevel==0`. When `extractDeepOffPinConnectionsToLevel` >=1, the extractor extracts off-pin connection and does not report an open.



© 2023 Cadence Design Systems, Inc.

Case 2: The extractor does not extract the off-pin connection and hence does not report short when `extractStopLevel==0`. When `extractDeepOffPinConnectionsToLevel >=1`, the extractor extract the off-pin connection and reports a short.



Examples

```
envGetVal("layout" "extractDeepOffPinConnectionsToLevel")
envSetVal("layout" "extractDeepOffPinConnectionsToLevel" 'int 1)
```

Related Topics

extractStopLevel

© 2023 Cadence Design Systems, Inc.

extractEnabled

```
layout extractEnabled boolean { t | nil }
```

Description

Lets you specify the connectivity extraction during interactive editing, meaning that the connectivity model is updated automatically as you edit your design.

The default is `t`. When set to `nil`, the connectivity model is not updated during interactive editing and no new short or open violation markers are generated in the layout window.



Tip

When the incremental extractor is switched off, you can still run connectivity extraction at any time using the *Connectivity – Extract Layout* command.

GUI Equivalent

Command:	*Options – Connectivity– Connectivity – Controls*
Field:	*Update connectivity information when design is modified*

Examples

```
envGetVal("layout" "extractEnabled")
envSetVal("layout" "extractEnabled" 'boolean nil)
```

Related Topics

Connectivity Extraction in Layout XL

© 2023 Cadence Design Systems, Inc.

extractEnableVirtualConnections

```
layout extractEnableVirtualConnections boolean { t | nil}
```

Description

Enables connectivity extraction from virtual-connect labels that are either attached or stamped to a shape.

The default is `t`, which means that sets of physically connected shapes that are on the same net and tagged by a label comprising the net name followed by a virtual-connect separator are considered as connected.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Controls*
Field:	*Enable virtual connections*

Examples

```
envGetVal("layout" "extractEnableVirtualConnections")
envSetVal("layout" "extractEnableVirtualConnections" 'boolean nil)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractHierExtractSearchDepth

```
layout extractHierExtractSearchDepth 'int { 1 | 2 ... 32 }
```

Description

Specifies how many levels down the hierarchy to search for cells to extract during hierarchical extraction. Specify an integer value between `1` and `32`.

The default is `1`.

GUI Equivalent

Command:	*Connectivity – Extract Layout*
Field:	*Search Depth*

Examples

```
envGetVal("layout" "extractHierExtractSearchDepth")
envSetVal("layout" "extractHierExtractSearchDepth" 'int 3)
```

Related Topics

Extract Layout form

© 2023 Cadence Design Systems, Inc.

extractIgnoreColorLockState

```
layout extractIgnoreColorLockState boolean { t | nil }
```

Description

Specifies whether the lock state of shapes should be ignored during extraction.

The default is `nil`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Control*
Field:	*Ignore color lock state*

Examples

```
envGetVal("layout" "extractIgnoreColorLockState")
envSetVal("layout" "extractIgnoreColorLockState" 'boolean t)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractKeepShapeAssignment

```
layout extractKeepShapeAssignment boolean { t | nil }
```

Description

Ensures that the connectivity of a design is preserved by retaining the assignment of shapes on a net.

The default is `nil`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Controls*
Field:	*Keep shape assignment*

Examples

```
envGetVal("layout" "extractKeepShapeAssignment")
envSetVal("layout" "extractKeepShapeAssignment" 'boolean  t)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractNetInvalidConnectionLimit

```
layout extractNetInvalidConnectionLimit int numberOfInvalidConnectionsNet
```

Description

Specifies the maximum number of invalid connection markers that can be generated by the extractor for a particular net.

The default is `100`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Limits*
Field:	*Maximum number of invalid connections*

Examples

```
envGetVal("layout" "extractNetInvalidConnectionLimit")
```

```
envSetVal("layout" "extractNetInvalidConnectionLimit" 'int 750)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractNetOpenViolationLimit

`layout extractNetOpenViolationLimit int` *`numberOfOpenViolations`*

Description

Specifies the maximum number of open violations generated by the extractor for a particular net. This improves interactive performance when you are editing either a large design or a net with a large number of instance terminals and terminals in an unrouted design.

The default is `100`.



Important

This environment variable is considered only when `extractEnabled` and `extractVerifyOpenViolations` are switched on.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Limits*
Field:	*Maximum number of opens: By net*

Examples

```
envGetVal("layout" "extractNetOpenViolationLimit")
envSetVal("layout" "extractNetOpenViolationLimit" 'int 70)
```

Related Topics

extractEnabled

extractVerifyOpenViolations

© 2023 Cadence Design Systems, Inc.

extractNetShortViolationLimit

`layout extractNetShortViolationLimit int` *`numberOfShortViolations`*

Description

Specifies the maximum number of short violations generated by the extractor for a particular net.

The default is `100`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Limits*
Field:	*Maximum number of shorts: By net*

Examples

```
envGetVal("layout" "extractNetShortViolationLimit")
envSetVal("layout" "extractNetShortViolationLimit" 'int 75)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractPromoteAllStopShapes

```
layout extractPromoteAllStopShapes boolean { t | nil }
```

Description

Specifies that stop shapes at `extractStopLevel + 1` are visible to the extractor.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "extractPromoteAllStopShapes")
envSetVal("layout" "extractPromoteAllStopShapes" 'boolean t)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractReinitOnStopLevelChange

```
layout extractReinitOnStopLevelChange boolean { t | nil }
```

Description

Specifies if the extractor needs to be reinitialized when the extract stop level is changed.

The default is `nil`.

GUI Equivalents

None

Examples

```
envGetVal("layout" "extractReinitOnStopLevelChange")
envSetVal("layout" "extractReinitOnStopLevelChange" 'boolean t)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractRemoveStickyProperty

```
layout extractRemoveStickyProperty boolean { t | nil }
```

Description

Removes the lxStickyNet property on shapes that are connected to a connectivity source, such as a pin, instance terminal, or a label.

The default is `t`.

GUI Equivalent

Command:	*Options – LayoutXL – Connectivity (tab)*
Field:	*Remove lxStickyNet property on non-floating shapes*

Examples

```
envGetVal("layout" "extractRemoveStickyProperty")
envSetVal("layout" "extractRemoveStickyProperty" 'boolean t)
```

Related Topics

lxStickyNet

Connectivity

© 2023 Cadence Design Systems, Inc.

extractRunOnInit

```
layout extractRunOnInit boolean { t | nil }
```

Description

Runs connectivity extraction automatically when you launch Layout XL or GXL from Layout L.

The default is `nil`, which means that connectivity extraction is not run automatically when you launch either application.



Important

Enabling this environment variable means that connectivity extraction is run even if it not required and adversely impacts the performance of Layout XL during initialization. Cadence recommends that you do not enable this environment variable by default, but that you use the *Connectivity – Extract Layout* command when you know it is required.

GUI Equivalent

None

Examples

```
envGetVal("layout" "extractRunOnInit")
envSetVal("layout" "extractRunOnInit" 'boolean t)
```

Related Topics

Connectivity

© 2023 Cadence Design Systems, Inc.

extractStopLevel

```
layout extractStopLevel int { 0 | 1 | 2 | ... | 32 }
```

Description

Specifies how much of the design hierarchy the extractor considers when extracting the design. The default is `0`, which means that the extractor extracts only:

- Top-level objects. Shapes, vias, and instances are used to implement pins.
- Instance pins of top-level instances. These instance pins represent the interface of top-level instances to the lower levels in the design and are themselves considered to be at the top level.

Increasing the extractStopLevel increases the visibility of the extractor to shapes inside the hierarchy. For example:

extractStopLevel	Means...
1	extractStopLevel 0 + objects at level 1 + instance pins at level 2
2	extractStopLevel 1 + objects at level 2 + instance pins at level 3

Whatever the `extractStopLevel`, the extractor only extracts the top level - it only changes the connectivity of objects or creates markers at the top level.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Controls*
Field:	*Extract connectivity to level*

Examples

```
envGetVal("layout" "extractStopLevel")
envSetVal("layout" "extractStopLevel" 'int 20)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractVerifyAnalogSigType

```
layout extractVerifyAnalogSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as `Analog`. These markers can be viewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – Analog*

Examples

```
envGetVal("layout" "extractVerifyAnalogSigType")
envSetVal("layout" "extractVerifyAnalogSigType" 'boolean nil)
```

Related Topics

Annotation Browser

© 2023 Cadence Design Systems, Inc.

extractVerifyBigMPP

```
layout extractVerifyBigMPP boolean { t | nil }
```

Description

Specifies that the connectivity extractor is to extract through large multipart paths. A large multipart path is one that contains more than 1000 elements. These are typically used to implement physical-only structures such as isolation rings, and therefore do not need to be extracted.

The default is `nil`, which means that the extractor ignores them.



Caution

Setting this environment variable to `t` ***can adversely affect the performance of the extractor.***

GUI Equivalent

None

Examples

```
envGetVal("layout" "extractVerifyBigMPP")
envSetVal("layout" "extractVerifyBigMPP" 'boolean t)
```

Related Topics

Connectivity Extraction in Layout XL

© 2023 Cadence Design Systems, Inc.

extractVerifyClockSigType

```
layout extractVerifyClockSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as *Clock*. These markers can be viewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – Clock*

Examples

```
envGetVal("layout" "extractVerifyClockSigType")
envSetVal("layout" "extractVerifyClockSigType" 'boolean nil)
```

Related Topics

Annotation Browser

© 2023 Cadence Design Systems, Inc.

extractVerifyGroundSigType

```
layout extractVerifyGroundSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as *Ground*. These markers can be reviewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – Ground*

Examples

```
envGetVal("layout" "extractVerifyGroundSigType")
envSetVal("layout" "extractVerifyGroundSigType" 'boolean nil)
```

Related Topics

Annotation Browser

© 2023 Cadence Design Systems, Inc.

extractVerifyIllegalHierConnections

```
layout extractVerifyIllegalHierConnections boolean { t | nil }
```

Description

Reports invalid hierarchical connections in the current cellview.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Illegal hierarchical connections*

Examples

```
envGetVal("layout" "extractVerifyIllegalHierConnections")
envSetVal("layout" "extractVerifyIllegalHierConnections" 'boolean nil)
```

Related Topics

Annotation Browser

© 2023 Cadence Design Systems, Inc.

extractVerifyIllegalHierConnectionsIgnoreUnassignedHierShapes

```
layout extractVerifyIllegalHierConnectionsIgnoreUnassignedHierShapes boolean { t |
     nil }
```

Description

Ignores unassigned hierarchical shapes when checking illegal hierarchical connections.

The default is `nil`.

GUI Equivalents

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Ignore unassigned hierarchical shapes*

Examples

```
envGetVal("layout"
"extractVerifyIllegalHierConnectionsIgnoreUnassignedHierShapes")
envSetVal("layout"
"extractVerifyIllegalHierConnectionsIgnoreUnassignedHierShapes" 'boolean t)
```

Related Topics

Annotation Browser

© 2023 Cadence Design Systems, Inc.

extractVerifyIllegalOverlapsForStoppedColoredShapes

```
layout extractVerifyIllegalOverlapsForStoppedColoredShapes boolean { t | nil }
```

Description

Reports invalid overlap violations for color-dependent stopping of shapes.

The default is `nil`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Illegal overlaps for stopped colored shapes*

Examples

```
envGetVal("layout" "extractVerifyIllegalOverlapsForStoppedColoredShapes")
envSetVal("layout" "extractVerifyIllegalOverlapsForStoppedColoredShapes" 'boolean
t)
```

Related Topics

Annotation Browser

© 2023 Cadence Design Systems, Inc.

extractVerifyIncompleteMustConnectAllPins

```
layout extractVerifyIncompleteMustConnectAllPins boolean { t | nil }
```

Description

Controls if incomplete `mustConnectAllPins` must be verified or not.

The default is `t`.

GUI Equivalent

None.

Examples

```
envGetVal("layout" "extractVerifyIncompleteMustConnectAllPins")
envSetVal("layout" "extractVerifyIncompleteMustConnectAllPins" 'boolean nil)
```

Related Topics

mustConnectAllPins

© 2023 Cadence Design Systems, Inc.

extractVerifyIncNetsWithShapesOnNonRoutingLayers

```
layout extractVerifyIncNetsWithShapesOnNonRoutingLayers boolean { t | nil }
```

Description

Specifies that the connectivity extractor should create open markers between shapes on non-routing layers. Non-routing layers are typically well and oxide layers.

The default is `t`, which means that open markers are drawn between shapes on non-routing layers.

When set to `nil`, the extractor does not create open markers between shapes on non-routing layers. This is typically used to prevent open markers from being created for bulk instance terminals with pin shapes on non-routing layers, and allow the signal router to focus on open markers on the routing layers.

Important

This environment variable is considered only when `extractVerifyOpenViolations` is switched on.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Shapes on non-routing layers*

Examples

```
envGetVal("layout" "extractVerifyIncNetsWithShapesOnNonRoutingLayers")
envSetVal("layout" "extractVerifyIncNetsWithShapesOnNonRoutingLayers" 'boolean
nil)
```

Related Topics

extractVerifyOpenViolations

© 2023 Cadence Design Systems, Inc.

extractVerifyInvalidConnections

```
layout extractVerifyInvalidConnections boolean { t | nil }
```

Description

Controls if invalid connections must be verified or not.

The default is `t`.

GUI Equivalent

None.

Examples

```
envGetVal("layout" "extractVerifyInvalidConnections")
envSetVal("layout" "extractVerifyInvalidConnections" 'boolean nil)
```

Related Topics

Connectivity Extraction in Layout Extraction

© 2023 Cadence Design Systems, Inc.

extractVerifyMustConnectMarkers

```
layout extractVerifyMustConnectMarkers boolean { t | nil }
```

Description

Displays the must-connect violation markers generated by the connectivity extractor.

When set to default value `t`, the extractor shows top-level must-connect violations in the layout window but when set to `nil`, it does not.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Controls*
Field:	*Verify must-connect violations*

Examples

```
envGetVal("layout" "extractVerifyMustConnectMarkers")
envSetVal("layout" "extractVerifyMustConnectMarkers" 'boolean nil)
```

Related Topics

Connectivity Extraction in Layout XL

© 2023 Cadence Design Systems, Inc.

extractVerifyOpenViolations

```
layout extractVerifyOpenViolations boolean { t | nil }
```

Description

Toggles the verification and display of open violations in the design.

This environment variable is considered only when `extractVerifyOpenViolations` is switched on.



Tip

To improve interactive performance for large designs or unrouted designs containing nets with a large number of instance terminals and terminals, use the `extractCellviewOpenViolationLimit` and `extractNetOpenViolationLimit` environment variables. They are considered only when `extractVerifyOpenViolations` is set to `t`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Controls*
Field:	*Verify open violations*

Examples

```
envGetVal("layout" "extractVerifyOpenViolations")
envSetVal("layout" "extractVerifyOpenViolations" 'boolean nil)
```

Related Topics

extractCellviewOpenViolationLimit

extractNetOpenViolationLimit

© 2023 Cadence Design Systems, Inc.

extractVerifyOpticalPortRadiusMatch

```
layout extractVerifyOpticalPortRadiusMatch boolean { t | nil }
```

Description

Checks if the radius of the overlapping optical ports match to form a valid connection.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "extractVerifyOpticalPortRadiusMatch")
envSetVal("layout" "extractVerifyOpticalPortRadiusMatch" 'boolean t)
```

Related Topics

Connectivity Extraction in Layout XL

© 2023 Cadence Design Systems, Inc.

extractVerifyPowerSigType

```
layout extractVerifyPowerSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as *Power*. These markers can be viewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – Power*

Examples

```
envGetVal("layout" "extractVerifyPowerSigType")
envSetVal("layout" "extractVerifyPowerSigType" 'boolean nil)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

extractVerifyResetSigType

```
layout extractVerifyResetSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as *Reset*. These markers can be viewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – Reset*

Examples

```
envGetVal("layout" "extractVerifyResetSigType")
envSetVal("layout" "extractVerifyResetSigType" 'boolean nil)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

extractVerifyScanSigType

```
layout extractVerifyScanSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as *Scan*. These markers can be viewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – Scan*

Examples

```
envGetVal("layout" "extractVerifyScanSigType")
envSetVal("layout" "extractVerifyScanSigType" 'boolean nil)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

extractVerifySignalSigType

```
layout extractVerifySignalSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as *Signal*. These markers can be viewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – Signal*

Examples

```
envGetVal("layout" "extractVerifySignalSigType")
envSetVal("layout" "extractVerifySignalSigType" 'boolean nil)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

extractVerifyTieHighSigType

```
layout extractVerifyTieHighSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as *TieHigh*. These markers can be viewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – TieHigh*

Examples

```
envGetVal("layout" "extractVerifyTieHighSigType")
envSetVal("layout" "extractVerifyTieHighSigType" 'boolean nil)
```

Related Topics

Annotation Browser

© 2023 Cadence Design Systems, Inc.

extractVerifyTieLowSigType

```
layout extractVerifyTieLowSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as *TieLow*. These markers can be viewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – TieLow*

Examples

```
envGetVal("layout" "extractVerifyTieLowSigType")
envSetVal("layout" "extractVerifyTieLowSigType" 'boolean nil)
```

Related Topics

Annotation Browser

© 2023 Cadence Design Systems, Inc.

extractVerifyTieOffSigType

```
layout extractVerifyTieOffSigType boolean { t | nil }
```

Description

Creates markers for nets that have their *Signal type* set as *TieOff*. These markers can be viewed in the Annotation Browser.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Connectivity – Annotations*
Field:	*Signal types – TieOff*

Examples

```
envGetVal("layout" "extractVerifyTieOffSigType")
envSetVal("layout" "extractVerifyTieOffSigType" 'boolean nil)
```

Related Topics

Annotation Browser

© 2023 Cadence Design Systems, Inc.

extractVerifyWellSubstrateConnections

```
layout extractVerifyWellSubstrateConnections boolean { t | nil }
```

Description

Enables or disables the verification of the substrate in terms of open and short violations.

For example, no open markers are created between two disjoint substrate vias on the same net because the vias short to each other through the substrate. Or, a substrate short marker is created if a substrate via is created on a given net and placed over the substrate while the substrate is already polarized on another net by another substrate via.

The default is `nil`. The tool does not support deep well layers.

GUI Equivalent

None.

Examples

```
envGetVal("layout" "extractVerifyWellSubstrateConnections")
envSetVal("layout" "extractVerifyWellSubstrateConnections" 'boolean t)
```

Related Topics

Inherited Connections

© 2023 Cadence Design Systems, Inc.

extractVerifyUnimplementedInstTerms

```
layout extractVerifyUnimplementedInstTerms boolean { t | nil }
```

Description

Specifies that the connectivity extractor should consider instance terminals with pin shapes that are not extractable including instance terminals with no pin shapes at all.

The default is `t`. When set to `nil`, the extractor and the functions it serves ignore any instance terminal which has no pin shape or which has a pin shape on a layer that is not extractable. This allows Layout XL to extract based only on the extract layers defined in the technology file.



Important

This environment variable is considered only when `extractVerifyOpenViolations` is switched on.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Controls*
Field:	*Verify unimplemented instance terminal*

Examples

```
envGetVal("layout" "extractVerifyUnimplementedInstTerms")
envSetVal("layout" "extractVerifyUnimplementedInstTerms" 'boolean nil)
```

Related Topics

extractVerifyOpenViolations

© 2023 Cadence Design Systems, Inc.

extractVerifyUnnecessaryPseudoParallelConnections

```
layoutXL extractVerifyUnnecessaryPseudoParallelConnections boolean { t | nil }
```

Description

Reports the overlap between two sub-nets of a pseudo-parallel net.

The default is `nil`.

GUI Equivalent

Command:	*Options - Connectivity - Connectivity - Annotation*
Field:	*Unnecessary pseudo-parallel connections*

Examples

```
envGetVal("layoutXL" "extractVerifyUnnecessaryPseudoParallelConnections")
envSetVal("layoutXL" "extractVerifyUnnecessaryPseudoParallelConnections" 'boolean
t)
```

Related Topics

Annotation Browser tabs

© 2023 Cadence Design Systems, Inc.

extractVerifyWeakMarkers

```
layout extractVerifyWeakMarkers boolean { t | nil }
```

Description

Displays the weak-connect violation markers generated by the connectivity extractor.

The default is `t`. The extractor shows top-level weak-connect violations in the layout window.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Controls*
Field:	*Verify weak-connect violations*

Examples

```
envGetVal("layout" "extractVerifyWeakMarkers")
envSetVal("layout" "extractVerifyWeakMarkers" 'boolean nil)
```

Related Topics

Connectivity Form

© 2023 Cadence Design Systems, Inc.

extractVirtualConnectionSeparator

```
layout extractVirtualConnectionSeparator string "virtualConnectSeparator"
```

Description

Specifies the separator that the connectivity extractor uses to identify virtual-connect labels.

By default, the virtual-connect separator is colon (:). But, you can change the separator, as appropriate, using the *Use (identifier) to define nets as virtually connected* field in the Layout XL Options form.

The default is `(":")`.

GUI Equivalent

Command:	*Options – Connectivity – Connectivity – Controls*
Field:	*Use (identifier) to define nets as virtually connected*

Examples

```
envGetVal("layout" "extractVirtualConnectionSeparator")
envSetVal("layout" "extractVirtualConnectionSeparator" 'string ":")
```

Related Topics

Enabling Virtual Connections

© 2023 Cadence Design Systems, Inc.

filteredNetName

```
layout.ava filteredNetName string "NetName"
```

Description

Specifies the net to be used when creating vias.

The default is `""`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – Connectivity*
Field:	*Nets*

Examples

```
envGetVal("layout.ava" "filteredNetName")
envSetVal("layout.ava" "filteredNetName" 'string "net1")
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

filterSizeForResolution

```
graphic filterSizeForResolution string {"(High highVal)" | "(Medium medVal)"| "(Low
     lowVal)" }
```

Description

Enables you to customize the level of details, filter size applied to resolutions High, and/or Medium, and/or Low.

By default, this environment variable is defined as below:

```
graphic filterSizeForResolution 'string "(High 2.0) (Medium 6.0) (Low
15.0)"
```

The table below lists the different filter sizes used and the `lowVal`, `MedVal`, and `HighVal` values defined in the `filterSizeForResolution` environment variable.

Filter Size	Low	Medium	High	Very High
Shape filterSize	`lowVal` (def 15.0)	`medVal` (def 6.0)	`highVal` (def 2.0)	`0.0`
Highest Resolution for P&R Objects option deselected	`15.0`	`6.0`	`6.0`	`1.0`
Highest Resolution for P&R Objects option selected	`15.0`	`6.0`	`highVal`	`0.0`

GUI Equivalent

Command:	*Options – Display*
Field:	*Resolution*

Examples

```
envGetVal("graphic" "filterSizeForResolution")
envSetVal{"graphic" "filterSizeForResolution" 'string {(High 3.0) | (Medium 7.0)|
(Low 18.0)}
```


© 2023 Cadence Design Systems, Inc.

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

findReplaceReportFileName

```
layout findReplaceReportFileName string "fileName"
```

Description

Specifies the file name to save the detailed report for find and replace operations.

The default is "`vlsSearchReport.csv`".

GUI Equivalent

Command:	*Tools – Find/Replace*
Field:	*Save Report To*

Examples

```
envGetVal("layout" "findReplaceReportFileName")
envSetVal("layout" "findReplaceReportFileName" 'string "file1")
```

Related Topics

Find/Replace Form

© 2023 Cadence Design Systems, Inc.

fitIntoClusterBoundary

```
layout fitIntoClusterBoundary boolean { t | nil }
```

Description

Places all objects in a cluster into the cluster boundary in Layout XL.

The default is `nil`.

GUI Equivalent

Command:	*Create – P&R Objects – Cluster Boundary*
Field:	*Fit Into Boundary*

Examples

```
envGetVal("layout" "fitIntoClusterBoundary")
envSetVal("layout" "fitIntoClusterBoundary" 'boolean t)
```

Related Topics

Create Cluster Boundary Form

© 2023 Cadence Design Systems, Inc.

fixedWidthPaths

```
layout fixedWidthPaths boolean { t | nil }
```

Description

Specifies whether to override the technology file and use a fixed width by default when creating a path.

The default is `nil`.

GUI Equivalent

Command:	*Create – Shape – Path*
Field:	*Fixed Width*

Examples

```
envGetVal("layout" "fixedWidthPaths")
envSetVal("layout" "fixedWidthPaths" 'boolean t)
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

flattenExcludePcellListFile

```
layout flattenExcludePcellListFile string "pathAndFileName"
```

Description

Specifies the name of the file that contains the list of Pcells that should not be flattened. This file has a list of library, cell and view of Pcells. In case the file is not found, the Pcells are flattened and a warning message is displayed in the CIW.

- If `flattenPCells` is set to `nil`, no Pcells are flattened and the no warning message is displayed in the CIW.
- If `flattenPCells` is set to `t`, all Pcells except those specified in the file are flattened. A warning message is displayed for the Pcells specified in the file.

GUI Equivalent

None.

Examples

```
envGetVal("layout" "flattenExcludePcellListFile")
envSetVal("layout" "flattenExcludePcellListFile" 'string "/project1/technology10/
technology10.excludePcellListFile")
```

Related Topics

flattenPCells

© 2023 Cadence Design Systems, Inc.

flattenMode

```
layout flattenMode cyclic { "one level" | "displayed levels" | "user level" }
```

Description

Specifies the default number of levels of hierarchy of a cell to flatten in `leHiFlatten()` or `leFlattenInst()`.

The default is `one level`.

GUI Equivalent

Command:	*Edit – Hierarchy – Flatten*
Field:	*Hierarchy Level*

Examples

```
envGetVal("layout" "flattenMode")
envSetVal("layout" "flattenMode" 'cyclic "displayed levels")
envSetVal("layout" "flattenMode" 'cyclic "user level")
```

Related Topics

Flatten Form

leHiFlatten

leFlattenInst

© 2023 Cadence Design Systems, Inc.

flattenPCells

```
layout flattenPCells boolean { t | nil }
```

Description

Specifies whether Pcells are flattened by default or by `leHiFlatten()` or by `leFlattenInst()`.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Hierarchy – Flatten*
Field:	*Pcells*

Examples

```
envGetVal("layout" "flattenPCells")
envSetVal("layout" "flattenPCells" 'boolean t)
```

Related Topics

Flatten Form

leHiFlatten

leFlattenInst

© 2023 Cadence Design Systems, Inc.

flattenVias

```
layout flattenVias boolean { t | nil }
```

Description

Specifies whether geometries in the selected vias are flattened.

The default is `t`.

GUI Equivalent

Command:	*Edit – Hierarchy – Flatten*
Field:	*Vias*

Examples

```
envGetVal("layout" "flattenVias")
envSetVal("layout" "flattenVias" 'boolean nil)
```

Related Topics

Flatten Form

© 2023 Cadence Design Systems, Inc.

flipAroundCombinedCenter

```
layout flipAroundCombinedCenter boolean { t | nil }
```

Description

Lets you flip the multiple selected objects around the specified reference point of the imaginary bounding box around the objects.

The default is `t`.

GUI Equivalent

Command:	*Options – Editor*
	Edit – Flip – Flip
	Edit – Flip – Flip Vertical
	Edit – Flip – Flip Horizontal
Field:	*Flip Around Combined BBox*

Examples

```
envGetVal("layout" "flipAroundCombinedCenter")
envSetVal("layout" "flipAroundCombinedCenter" 'boolean nil)
```

Related Topics

Flip Form

Flip Vertical Form

Flip Horizontal Form

Layout Editor Options Form

Object Flipping

© 2023 Cadence Design Systems, Inc.

flipReference

```
layout flipReference cyclic { "LowerLeft" | "CenterLeft" | "UpperLeft" |
     "LowerCenter" | "CenterCenter" | "UpperCenter" | "LowerRight" | "CenterRight"
     | "UpperRight" | "Custom" }
```

Description

Sets the reference point around which an object is flipped vertically or horizontally.

- `LowerLeft`: Flips an object around the lower-left point of its bounding box.
- `CenterLeft`: Flips an object around the center-left point of its bounding box.
- `UpperLeft`: Flips an object around the upper-left point of its bounding box.
- `LowerCenter`: Flips an object around the lower-center point of its bounding box.
- `CenterCenter`: Default. Flips an object around the center point of its bounding box.
- `UpperCenter`: Flips an object around the upper-center point of its bounding box.
- `LowerRight`: Flips an object around the lower-right point of its bounding box.
- `CenterRight`: Flips an object around the center-right point of its bounding box.
- `UpperRight`: Flips an object around the upper-right point of its bounding box.
- `Custom`: Flips an object around the specified X and Y coordinates. This option is available in the Flip form.

GUI Equivalent

Command:	*Edit – Flip – Flip*
	Edit – Flip – Flip Vertical
	Edit – Flip – Flip Horizontal
Field:	*Reference*

Examples

```
envGetVal("layout" "flipReference")
envSetVal("layout" "flipReference" 'cyclic "LowerLeft")
envSetVal("layout" "flipReference" 'cyclic "Custom")
```


© 2023 Cadence Design Systems, Inc.

Related Topics

Flip Form

Flip Vertical Form

Flip Horizontal Form

© 2023 Cadence Design Systems, Inc.

followRoutingDirection

```
layout followRoutingDirection boolean { t | nil }
```

Description

Specifies whether the *Convert To PathSeg* command follows the routing direction.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "followRoutingDirection")
envSetVal("layout" "followRoutingDirection" 'boolean nil)
```

Related Topics

Converting Polygons to Paths

© 2023 Cadence Design Systems, Inc.

fullSelectionMode

```
graphic fullSelectionMode string { "enclosed" | "crossed" | "enclosedAndCrossed" }
```

Description

Specifies whether the full selection mode is set to crossed, enclosed or enclosed and crossed.

- `crossed`: Crossed Figures fully selects objects you partially enclose with the area selection box, and does not select fully enclosed objects.
- `enclosed`: Default. Enclosed Figures fully selects objects you completely enclose with the area selection box.
- `enclosedAndCrossed`: Enclosed and Crossed Figures fully selects objects you completely or partially enclose with the area selection box.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Full Mode*

Examples

```
envGetVal("graphic" "fullSelectionMode")
envSetVal("graphic" "fullSelectionMode" 'string "crossed")
envSetVal("graphic" "fullSelectionMode" 'string "enclosedAndCrossed")
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

ghostOfDraggedObjects

```
layout ghostOfDraggedObjects boolean { t | nil }
graphic ghostOfDraggedObjects boolean { t | nil }
```

Description

Allows viewing the ghost of the original object when it is being moved or stretched.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Dragged Object Ghost*

Examples

```
envGetVal("layout" "ghostOfDraggedObjects")
envSetVal("layout" "ghostOfDraggedObjects" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

glyphTransparency

```
layout glyphTransparency int transparencyLevel
graphic glyphTransparency int transparencyLevel
```

Description

Lets you set the transparency level of the glyph. You can choose a value from `0` to `100`, where `100` makes the glyph fully transparent.

The default is `0`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Glyph Transparency*

Examples

```
envGetVal("graphic" "glyphTransparency")
envSetVal("graphic" "glyphTransparency" 'int 70)
envGetVal("layout" "glyphTransparency")
envSetVal("layout" "glyphTransparency" 'int 35)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

gravityApertureInPixels

```
layout gravityApertureInPixels int snapDistance
```

Description

Specifies the aperture value when the snap distance between the pointer and an object `gravityDistance` is set to `Medium`.

The default is `20`.

Note: The `gravityApertureInPixels` environment variable is applied only when the value of the `gravityDistance` environment variable set to `Medium`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "gravityApertureInPixels")
envSetVal("layout" "gravityApertureInPixels" 'int 15)
```

Related Topics

gravityDistance

© 2023 Cadence Design Systems, Inc.

gravityDistance

```
layout gravityDistance string { "Small" | "Medium" | "Large" | "Very Large" }
```

Description

Specifies the snap distance between the pointer and an object. You can choose from `Small`, `Medium`, `Large`, or `Very Large`.

The default is `Medium`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Distance*

Examples

```
envGetVal("layout" "gravityDistance")
envSetVal("layout" "gravityDistance" 'string "Small")
envSetVal("layout" "gravityDistance" 'string "Very Large")
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

gravityEditCellViewOnly

```
layout gravityEditCellViewOnly boolean { t | nil }
```

Description

Implies gravity works within current cellview only.

The default is `t`. When set to `nil`, it implies gravity also works outside the current cellview. This environment variable is useful in *Edit In Place* mode, when the current cellview is not the same as the window cellview.

GUI Equivalent

None

Examples

```
envGetVal("layout" "gravityEditCellViewOnly")
envSetVal("layout" "gravityEditCellViewOnly" 'boolean nil)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

gravityOn

```
layout gravityOn boolean { t | nil }
```

Description

Specifies whether gravity mode is on. Gravity mode causes the pointer to snap to objects in the cellview.

The default is `nil`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Gravity On*

Examples

```
envGetVal("layout" "gravityOn")
envSetVal("layout" "gravityOn" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

gravitySelectableOnly

```
layout gravitySelectableOnly boolean { t | nil }
```

Description

Sets the drawing pointer to snap only to the selected objects in the cellview.

The default is `nil`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Selectable Objects Only*

Examples

```
envGetVal("layout" "gravitySelectableOnly")
envSetVal("layout" "gravitySelectableOnly" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

gravityType

```
layout gravityType string { "pin" | "edge" | "midpoint" | "nexus" | "vertex" | "end"
     | "centerline" | "origin" | "snapPatternTrack" }
```

Description

Sets the drawing pointer to snap to the specified target type.

By default, the environment variable is set to the following gravity types: `pin edge midpoint nexus vertex end centerline origin snapPatternTrack`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "gravityType")
envSetVal("layout" "gravityType" 'string midpoint)
envSetVal("layout" "gravityType" 'string snapPatternTrack)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

gravityUse5PixelApertureWhenOff

```
layout gravityUse5PixelApertureWhenOff boolean { t | nil }
```

Description

Implies gravity remains active within a small aperture of five pixels even when it is turned off.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "gravityUse5PixelApertureWhenOff")
envSetVal("layout" "gravityUse5PixelApertureWhenOff" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

grEnclosedBy

```
layout grEnclosedBy float initializeValue
```

Description

Initializes the *Enclose by* field on the *Guard Ring* form.

The default is `0.0`. You can replace the default value with another integer point number by updating this environment variable in your `.cdsenv` file. You can change the value for the *Enclose by* field by typing in the field on the Guard Ring form. The software verifies that `minSpacing` is correct. It also verifies that it is minimum DRC compliant, based on process and context. If necessary, the software overrides the user value in the form.

GUI Equivalent

Command:	*Create – MPP Guard Ring*
Field:	*Enclose by*

Examples

```
envGetVal("layout" "grEnclosedBy")
envSetVal("layout" "grEnclosedBy" 'float 2.0)
```

Related Topics

Guard Ring Form

© 2023 Cadence Design Systems, Inc.

gridMultiple

```
layout gridMultiple int numberOfGridLines
graphic gridMultiple int numberOfGridLines
```

Description

Specifies the number of minor grid lines or dots between major grid lines or dots.

The default is `5`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Major Spacing*

Examples

```
envGetVal("layout" "gridMultiple")
envSetVal("layout" "gridMultiple" 'int 7)
envGetVal("graphic" "gridMultiple")
envSetVal("graphic" "gridMultiple" 'int 7)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

gridSpacing

```
layout gridSpacing float numberOfGridLines
graphic gridSpacing float numberOfGridLines
```

Description

Specifies the number between each minor grid line or dot.

The default is `1.0` user units.

GUI Equivalent

Command:	*Options – Display*
Field:	*Minor Spacing*

Examples

```
envGetVal("layout" "gridSpacing")
envSetVal("layout" "gridSpacing" 'float 2.0)
envGetVal("graphic" "gridSpacing")
envSetVal("graphic" "gridSpacing" 'float 2.0)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

groupArrayDragMode

```
layout groupArrayDragMode cyclic { "outline" | "full" | "auto" }
```

Description

Sets the drag mode used while stretching a group array.

- `outline`: Displays only the outline of a group array during the stretch or move operation. The actual geometries in the group array are not displayed.
- `full`: Displays the preview of all the geometries in group array cells. A full preview might impact the performance in case of group arrays containing a very large number of geometries.
- `auto`: Displays the preview of the geometries in group array cells up to the limit that the system determines. This helps to prevent any performance issues due to large geometries.

The default value is `outline`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "groupArrayDragMode")
envSetVal("layout" "groupArrayDragMode" 'cyclic "full")
envSetVal("layout" "GroupArrayDragMode" 'cyclic "auto")
```

Related Topics

lxGroupArrayCreatePreset

groupArrayMaxDragFigs

© 2023 Cadence Design Systems, Inc.

groupArrayXSpacing

```
layout groupArrayXSpacing boolean float spacingValue
```

Description

Sets the spacing value used as the distance between cells on the X-axis when creating a locked group array by using the Copy form. This value is read when the `copyGroupArrayLockXY` environment variable is `t`.

The default value is `1`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "groupArrayXSpacing")
envSetVal("layout" "groupArrayXSpacing" 'float 5.0)
```

Related Topics

copyGroupArrayLockXY

groupArrayYSpacing

Group Arrays

Locked Group Arrays

© 2023 Cadence Design Systems, Inc.

groupArrayYSpacing

```
layout groupArrayYSpacing float spacingValue
```

Description

Sets the spacing value used as the distance between cells on the Y-axis when creating a locked group array by using the Copy form. This value is read when the `copyGroupArrayLockXY` environment variable is `t`.

The default value is `1`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "groupArrayYSpacing")
envSetVal("layout" "groupArrayYSpacing" 'float 5.0)
```

Related Topics

copyGroupArrayLockXY

groupArrayXSpacing

Group Arrays

Locked Group Arrays

© 2023 Cadence Design Systems, Inc.

highlightSelProtectedItems

```
graphic highlightSelProtectedItems boolean { t | nil }
```

Description

Determines whether protected items should be highlighted or not.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "highlightSelProtectedItems")
envSetVal("graphic" "highlightSelProtectedItems" 'boolean t)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

haloBottomOffset

`layout haloBottomOffset` *`numberOfUserUnits`*

Description

Specifies the number of user units the placement halo is extended from the bottom of the instance bBox.

The default is `0.0` user units.

GUI Equivalent

Command:	*Create – Instance – Define Halo* and *Create – P&R Objects – P&R Boundary – Define Halo*
Field:	*Offsets: Bottom*

Examples

```
envGetVal("layout" "haloBottomOffset")
envSetVal("layout" "haloBottomOffset" 'float 2.0)
```

Related Topics

Halo Editor Form

© 2023 Cadence Design Systems, Inc.

haloLeftOffset

```
layout haloLeftOffset float numberOfUserUnits
```

Description

Specifies the number of user units the placement halo is extended from the left of the instance bBox.

The default is `0.0` user units.

GUI Equivalent

Command:	*Create – Instance – Define Halo* and *Create – P&R Objects – P&R Boundary – Define Halo*
Field:	*Offsets: Left*

Examples

```
envGetVal("layout" "haloLeftOffset")
envSetVal("layout" "haloLeftOffset" 'float 2.0)
```

Related Topics

Halo Editor Form

© 2023 Cadence Design Systems, Inc.

haloRightOffset

`layout haloRightOffset float` *`numberOfUserUnits`*

Description

Specifies the number of user units the placement halo is extended from the right of the instance bBox.

The default is `0.0` user units.

GUI Equivalent

Command:	*Create – Instance – Define Halo* and *Create – P&R Objects – P&R Boundary – Define Halo*
Field:	*Offsets: Right*

Examples

```
envGetVal("layout" "haloRightOffset")
envSetVal("layout" "haloRightOffset" 'float 2.0)
```

Related Topics

Halo Editor Form

© 2023 Cadence Design Systems, Inc.

haloTopOffset

```
layout haloTopOffset float number0fUserUnits
```

Description

Specifies the number of user units the placement halo is extended from the top of the instance bBox.

The default is `0.0` user units.

GUI Equivalent

Command:	*Create – Instance – Define Halo* and *Create – P&R Objects – P&R Boundary – Define Halo*
Field:	*Offsets: Top*

Examples

```
envGetVal("layout" "haloTopOffset")
envSetVal("layout" "haloTopOffset" 'float 2.0)
```

Related Topics

Halo Editor Form

© 2023 Cadence Design Systems, Inc.

highlightAlignedEdges

```
layout highlightAlignedEdges boolean { t | nil }
```

Description

Checks if the *Highlight Aligned Edges On* field is set to on.

The default is `nil`.

GUI Equivalent

Command:	*Options – Dynamic Display*
Field:	*Highlight Aligned Edges On*

Examples

```
envGetVal("layout" "highlightAlignedEdges")
envSetVal("layout" "highlightAlignedEdges" 'boolean t)
```

Related Topics

Dynamic Display Form

© 2023 Cadence Design Systems, Inc.

highlightAlignedEdgesColor

```
layout highlightAlignedEdgesColor string "color"
```

Description

Specifies the color of the highlight displayed on the edges of aligned shapes.

The default is `green`.

GUI Equivalent

Command:	*Options – Dynamic Display – Highlight Aligned Edges*
Field:	*Color*

Examples

```
envGetVal("layout" "highlightAlignedEdgesColor")
envSetVal("layout" "highlightAlignedEdgesColor" 'string "blue")
```

Related Topics

Halo Editor Form

© 2023 Cadence Design Systems, Inc.

highlightAlignedEdgesWidth

```
layout highlightAlignedEdgesWidth int highlightWidth
```

Description

Specifies width of the highlight that is displayed in the edges of the shapes aligned.

The default is `3`.

GUI Equivalent

Command:	*Options – Dynamic Display – Highlight Aligned Edges*
Field:	*Width*

Examples

```
envGetVal("layout" "highlightAlignedEdgesWidth")
envSetVal("layout" "highlightAlignedEdgesWidth" 'int 4)
```

Related Topics

Dynamic Display Form

© 2023 Cadence Design Systems, Inc.

iconsOn

```
layout iconsOn boolean { t | nil }
graphic iconsOn boolean { t | nil }
```

Description

Determines whether to display the elements in an unexpanded array as rectangles with the symbol for the master cell for each element. The variable applies to an instance or contact array. In the Display Options form, set the display start and stop levels to `0`.

When set to `t`, selecting *Show Name Of instance* displays the instance name and selecting *Show Name Of master* displays the master name.

The layout default is `nil`, the graphic default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Array Icons*

Examples

```
envGetVal("layout" "iconsOn")
envSetVal("layout" "iconsOn" 'boolean t)
envGetVal("graphic" "iconsOn")
envSetVal("graphic" "iconsOn" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

ignoreBounce

```
layout ignoreBounce boolean { t | nil }
```

Description

Determines whether you use the gravity offset bounce values from the Layout Editor Options form or the `minSpacing` value from the design rules in the technology file.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "ignoreBounce")
envSetVal("layout" "ignoreBounce" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

ignoreParallelMinWidthForSquarePinResize

```
layout ignoreParallelMinWidthForSquarePinResize boolean { t | nil }
```

Description

Determines whether to ignore the `minWidth` constraint along the direction perpendicular to the WSP track for square pins.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "ignoreParallelMinWidthForSquarePinResize")
envSetVal("layout" "ignoreParallelMinWidthForSquarePinResize" 'boolean t)
```

Related Topics

minWidth

© 2023 Cadence Design Systems, Inc.

ignorePCellSnapPropertyForDevice

```
layout ignorePCellSnapPropertyForDevice boolean { t | nil }
```

Description

Determines whether to ignore the properties, `leSnapGridHorizontal`, `leSnapGridVertical`, `leSnapPatternSnappingHorizontalGrid`, and `leSna[pPatternSnappingVerticalGrid`, set on a device or the device master. These properties cause device snapping in abstract mode. Ignoring these properties cause device snapping in detail mode.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "ignorePCellSnapPropertyForDevice")
envSetVal("layout" "ignorePCellSnapPropertyForDevice" 'boolean t)
```

Related Topics

© 2023 Cadence Design Systems, Inc.

instanceDrawingMode

```
graphic instanceDrawingMode string { "BBox" | "prBoundary" | "snapBoundary" }
layout instanceDrawingMode string { "BBox" | "prBoundary" | "snapBoundary" }
```

Description

Checks if the boundary of instances is displayed as the instance bounding box, the place and route boundary, or the snap boundary.

The default is `BBox`.

This environment variable replaces the `drawPRBoundary` environment variable.

GUI Equivalent

Command:	*Options – Display*
Field:	*Instance Drawing Mode*

Examples

```
envGetVal("graphic" "instanceDrawingMode")
envSetVal("graphic" "instanceDrawingMode" 'string "prBoundary")
envGetVal("layout" "instanceDrawingMode")
envSetVal("layout" "instanceDrawingMode" 'string "snapBoundary")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

instanceSelectionMode

```
layout instanceSelectionMode cyclic { "ignoreLPP" | "useLPP" }
```

Description

Controls the selection of instances in the layout editor. You can set the environment variable to one of the following values:

- `ignoreLPP`: Default. Ignores the *Selectability* of the LPP instance drawing.
- `useLPP`: The instance is selectable on the canvas if the LPP instance drawing has *Selectability* enabled in the Layers pane.

GUI Equivalent

None

Examples

```
envGetVal("layout" "instanceSelectionMode")
envSetVal("layout" "instanceSelectionMode" 'cyclic "useLPP")
```

Related Topics

Instance Selectability

Palette Panels

© 2023 Cadence Design Systems, Inc.

instanceVisibilityMode

```
layout instanceVisibilityMode cyclic { "ignoreLPP" | "useLPP" }
```

Description

Controls the visibility of instances in the layout editor. You can set the environment variable to one of the following values:

- `ignoreLPP`: Ignores the *Visibility* of the LPP instance drawing.
- `useLPP`: Default. The instance is visible on the canvas if the LPP instance drawing has *Visibility* enabled in the *Layers* pane of the Palette Assistant.

GUI Equivalent

None

Examples

```
envGetVal("layout" "instanceVisibilityMode")
envSetVal("layout" "instanceVisibilityMode" 'cyclic "ignoreLPP")
```

Related Topics

Palette Panels

© 2023 Cadence Design Systems, Inc.

instCellName

```
layout instCellName string "cellName"
```

Description

Specifies the default name of the cell for `leHiCreateInst`.

The default is none.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Cell*

Examples

```
envGetVal("layout" "instCellName")
envSetVal("layout" "instCellName" 'string "MyCell1")
```

Related Topics

Create Instance Form

leHiCreateInst

© 2023 Cadence Design Systems, Inc.

instColumns

`layout instColumns int` *`numberOfColumns`*

Description

Specifies the number of columns for an arrayed instance.

The default is `1`.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Mosaic: Columns*

Examples

```
envGetVal("layout" "instColumns")
envSetVal("layout" "instColumns" 'int 4)
```

Related Topics

Create Instance Form

© 2023 Cadence Design Systems, Inc.

instDelX

`layout instDelX float` *`columnDistance`*

Description

Specifies the distance between the columns of an arrayed instance or contact measured from edge to edge. This field is grayed out if the number of columns specified is `1`.

The default is `0`.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Mosaic: Delta X*

Examples

```
envGetVal("layout" "instDelX")
envSetVal("layout" "instDelX" 'float 1.0)
```

Related Topics

Create Instance Form

© 2023 Cadence Design Systems, Inc.

instDelY

```
layout instDelY float rowDistance
```

Description

Specifies the distance between the rows of an arrayed instance or contact measured from edge to edge. This field is grayed out if the number of columns specified is `1`.

The default is `0`.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Mosaic: Delta Y*

Examples

```
envGetVal("layout" "instDelY")
envSetVal("layout" "instDelY" 'float 1.0)
```

Related Topics

Create Instance Form

© 2023 Cadence Design Systems, Inc.

instHalosEnabled

```
layout instHalosEnabled boolean { t | nil }
```

Description

Checks if a halo is defined when an instance is created.

The default is `t`.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Halo*

Examples

```
envGetVal("layout" "instHalosEnabled")
envSetVal("layout" "instHalosEnabled" 'boolean nil)
```

Related Topics

Create Instance Form

© 2023 Cadence Design Systems, Inc.

instLabel

```
layout instLabel string { "instance" | "master" | "both" }
graphic instLabel string { "instance" | "master" | "both" }
```

Description

Determines whether the instance name, the master name, or both the instance and master names are to be displayed.

The default is `master`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Show Name Of*

Examples

```
envGetVal("layout" "instLabel")
envSetVal("layout" "instLabel" 'string "instance")
envGetVal("graphic" "instLabel")
envSetVal("graphic" "instLabel" 'string "both")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

instLibName

```
layout instLibName string "libraryName"
```

Description

Specifies the default name of the library for `leHiCreateInst`.

The default is none.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Library*

Examples

```
envGetVal("layout" "instLibName")
envSetVal("layout" "instLibName" 'string "MyLib")
```

Related Topics

Create Instance Form

leHiCreateInst

© 2023 Cadence Design Systems, Inc.

instMosaicCreation

```
layout instMosaicCreation boolean { t | nil }
```

Description

Creates an instance array.

The default is `nil`.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Create as Mosaic*

Examples

```
envGetVal("layout" "instMosaicCreation")
envSetVal("layout" "instMosaicCreation" 'boolean t)
```

Related Topics

Create Instance Form

© 2023 Cadence Design Systems, Inc.

instName

```
layout instName string "cellName"
```

Description

Specifies one or more names for the instances you create. You can specify as many names as you want (separated by spaces). As each name is used, it is removed from the list.

If all names are used, the function automatically generates a name based on the current master's `instNamePrefix` property and puts it in the form field.

The default is none.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Names*

Examples

```
envGetVal("layout" "instName")
envSetVal("layout" "instName" 'string "Inst1")
```

Related Topics

Create Instance Form

© 2023 Cadence Design Systems, Inc.

instPhysicalOnly

```
layout instPhysicalOnly boolean { t | nil }
```

Description

Creates an instance in such a way such that it exists only in the block domain and not in the module domain.

The default is `nil`.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Physical Only*

Examples

```
envGetVal("layout" "instPhysicalOnly")
envSetVal("layout" "instPhysicalOnly" 'boolean t)
```

Related Topics

Create Instance Form

© 2023 Cadence Design Systems, Inc.

instRows

```
layout instRows int numberOfRows
```

Description

Specifies the number of rows for an arrayed instance.

The default is `1`.

GUI Equivalent

Command:	*Create – Instance*
Field:	*Mosaic: Rows*

Examples

```
envGetVal("layout" "instRows")
envSetVal("layout" "instRows" 'int 7)
```

Related Topics

Create Instance Form

© 2023 Cadence Design Systems, Inc.

instViewName

```
layout instViewName string { "layout" | "schematic" | "symbolic" }
```

Description

Specifies the name of the view for `leHiCreateInst`.

The default is `layout`.

GUI Equivalent

Command:	*Create – Instance*
Field:	*View*

Examples

```
envGetVal("layout" "instViewName")
envSetVal("layout" "instViewName" 'string "schematic")
envSetVal("layout" "instViewName" 'string "symbolic")
```

Related Topics

Create Instance Form

leHiCreateInst

© 2023 Cadence Design Systems, Inc.

keepCopiedInsideGroup

```
layout keepCopiedInsideGroup boolean { t | nil }
```

Description

Adds the copied figure group members of different groups to their respective source figure groups, when the `transparentGroup` environment variable is set to `t`.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "keepCopiedInsideGroup")
envSetVal("layout" "keepCopiedInsideGroup" 'boolean t)
```

Related Topics

transparentGroup

Display Options Form

© 2023 Cadence Design Systems, Inc.

keepFirstTermName

```
layout keepFirstTermName boolean { t | nil }
```

Description

Specifies if the first terminal name is not removed on the Create Pin form when the pin is created.

The default is `nil`.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Keep First Name*

Examples

```
envGetVal("layout" "keepFirstTermName")
envSetVal("layout" "keepFirstTermName" 'boolean t)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

keepGuardRingEndsConnected

```
layout keepGuardRingEndsConnected boolean { t | nil }
```

Description

Controls whether a guard ring with touching ends will stay connected or will result in opening up the guard ring during the *Stretch* command.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Stretch*
Field:	*Keep guard ring ends connected*

Examples

```
envGetVal("layout" "keepGuardRingEndsConnected")
envSetVal("layout" "keepGuardRingEndsConnected" 'boolean t)
```

Related Topics

Stretch Form

© 2023 Cadence Design Systems, Inc.

keepWiresConnectedToNonWires

```
layout keepWiresConnectedToNonWires boolean { t | nil }
```

Description

Specifies if the connection with shapes, such as pin figures, rectangles, polygons, and MPP, coincident with the selected vertex during the stretch operation should be maintained.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Stretch*
Field:	*Keep wires connected to shapes*

Examples

```
envGetVal("layout" "keepWiresConnectedToNonWires")
envSetVal("layout" "keepWiresConnectedToNonWires" 'boolean t)
```

Related Topics

Stretch Form

© 2023 Cadence Design Systems, Inc.

labelAttach

```
layout labelAttach boolean { t | nil }
```

Description

Specifies whether the system prompts you to attach labels to an object.

The default is `t`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Attach*

Examples

```
envGetVal("layout" "labelAttach")
envSetVal("layout" "labelAttach" 'boolean nil)
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelAutoConn

```
layout labelAutoConn boolean { t | nil }
```

Description

Specifies whether the method for creating labels is automatic using the connectivity information of objects to be labeled.

The default is `t`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Mode – Auto – From Interconnect*

Examples

```
envGetVal("layout" "labelAutoConn")
envSetVal("layout" "labelAutoConn" 'boolean nil)
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelAutoInst

```
layout labelAutoInst boolean { t | nil }
```

Description

Specifies whether the method for creating labels is automatic using instances.

The default is `t`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Mode – Auto – From Instance*

Examples

```
envGetVal("layout" "labelAutoInst")
envSetVal("layout" "labelAutoInst" 'boolean nil)
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelDrafting

```
layout labelDrafting boolean { t | nil }
```

Description

Specifies whether labels are restricted to rotate no more than `90` degrees.

The default is `t`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Drafting*

Examples

```
envGetVal("layout" "labelDrafting")
envSetVal("layout" "labelDrafting" 'boolean nil)
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelExpand

```
layout labelExpand boolean { t | nil }
```

Description

Specifies whether or not the specified label string or pattern is to be expanded to separate label names in the sequence.

The default is `t` in *Manual* mode.

GUI Equivalent

Command:	*Create – Label*
Field:	*Expand*

Examples

```
envGetVal("layout" "labelExpand")
envSetVal("layout" "labelExpand" 'boolean nil)
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelFontStyle

```
layout labelFontStyle cyclic { "stick" | "gothic" | "roman" | "math" | "euroStyle"
     | "swedish" | "script" | "fixed" | "milspec" }
```

Description

Specifies the default font of labels.

The default is `stick`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Font*

Examples

```
envGetVal("layout" "labelFontStyle")
envSetVal("layout" "labelFontStyle" 'cyclic "script")
envSetVal("layout" "labelFontStyle" 'cyclic "milspec")
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelHeight

```
layout labelHeight float heightOfLabels
```

Description

Specifies the height of labels in user units.

The default is `1.0` user units.

GUI Equivalent

Command:	*Create – Label*
Field:	*Height*

Examples

```
envGetVal("layout" "labelHeight")
envSetVal("layout" "labelHeight" 'float 3.0)
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelJustify

```
layout labelJustify cyclic { "lowerLeft" | "centerLeft" | "upperLeft" |
     "lowerCenter" | "centerCenter" | "upperCenter" | "lowerRight" | "centerRight"
     | "upperRight" }
```

Description

Specifies the default justification of a label with respect to its origin.

The default is `centerCenter`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Justification*

Examples

```
envGetVal("layout" "labelJustify")
envSetVal("layout" "labelJustify" 'cyclic "lowerLeft")
envSetVal("layout" "labelJustify" 'cyclic "centerRight")
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelKeepLabel

```
layout labelKeepLabel boolean { t | nil }
```

Description

Specifies whether or not the specified label string or pattern should be retained in the *Label (Pattern)* field after all the generated labels have been placed.

The default is `t` in *Manual* mode.

GUI Equivalent

Command:	*Create – Label*
Field:	*Keep Label*

Examples

```
envGetVal("layout" "labelKeepLabel")
envSetVal("layout" "labelKeepLabel" 'boolean nil)
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelLPP

```
layout labelLPP cyclic { "auto" | "currentLayer" | "sameLayer" | "selectLayer" }
```

Description

Specifies the layer or purpose to be used for creating labels.This determines the layer to place the label from the `stampLabelLayers` section of the technology file.

- `currentLayer`: Default. Uses the layer/purpose set as the current entry layer in the *Layers* assistant.
- `sameLayer`: Uses the layer of the object to be labeled.
- `selectLayer`: Specifies the layer/purpose from the drop-down list in the *Create Label* form.

GUI Equivalent

Command:	*Create – Label*
Field:	*Label Layer/Purpose*

Examples

```
envGetVal("layout" "labelLPP")
envSetVal("layout" "labelLPP" 'cyclic "sameLayer")
envSetVal("layout" "labelLPP" 'cyclic "selectLayer")
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelMode

```
layout labelMode cyclic { "manual" | "auto" }
```

Description

Determines the method for creating labels - manually, using the specified label text or pattern or automatically, using connectivity information or instances or both.

The default is `manual`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Mode*

Examples

```
envGetVal("layout" "labelMode")
envSetVal("layout" "labelMode" 'cyclic "auto")
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelOverbar

```
layout labelOverbar boolean { t | nil }
```

Description

Determines how text strings containing underscore characters are displayed.

The default is `nil`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Overbar*

Examples

```
envGetVal("layout" "labelOverbar")
envSetVal("layout" "labelOverbar" 'boolean t)
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelPurpose

```
layout labelPurpose string purposeName
```

Description

Lets you fix the purpose for labels in the *Use same layer as shape, select purpose* field of the Create Label form. If you are trying to fix an invalid value for the purpose, then the previously set purpose value will be retained.

The default is `drawing`.

GUI Equivalent

Command:	*Create – Label*
Field:	*Use same layer as shape, select purpose*

Examples

```
envGetVal("layout" "labelPurpose")
envSetVal("layout" "labelPurpose" 'string "drawing")
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

labelText

```
layout labelText string "any_string_or_pattern"
```

Description

Specifies the label text or the pattern to use for generating label names in *Manual* mode.

If no text is specified, the command does not proceed and prompts you to specify a label text.

GUI Equivalent

Command:	*Create – Label*
Field:	*Label (Pattern)*

Examples

```
envGetVal("layout" "labelText")
envSetVal("layout" "labelText" 'string "A<0:3> A<<0:3>> A<1:4>-B<0:1> A<0:6:2>")
```

Related Topics

Create Label Form

© 2023 Cadence Design Systems, Inc.

leCheckMissingLayers

```
graphic leCheckMissingLayers boolean { t | nil }
```

Description

Determines if checking is done to identify layers not defined in the technology file.

The default is `nil`. When set to `t`, checking is done on `maskLayout` cell views only.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "leCheckMissingLayers")
envSetVal("graphic" "leCheckMissingLayers" 'boolean t)
```

Related Topics

maskLayoutDefaultApp

© 2023 Cadence Design Systems, Inc.

leCheckMissingMasters

```
graphic leCheckMissingMasters boolean { t | nil }
```

Description

Determines if unbound masters are checked while opening a design.

The default is `nil`. When set to `t`, or if the cellview is not maskLayout, the check is done.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "leCheckMissingMasters")
envSetVal("graphic" "leCheckMissingMasters" 'boolean t)
```

Related Topics

maskLayoutDefaultApp

© 2023 Cadence Design Systems, Inc.

leChopPreservePathSeg

```
layout leChopPreservePathSeg boolean { t | nil }
```

Description

Determines if pathsegs are preserved from being chopped.

In all modes for the *Chop* command, the pathSeg is not converted to polygon. In line and wire modes of the *Chop* command, a path or pathSeg is be chopped but the resultant geometry is a path or pathSeg only when the default value `t` is used.

When set to `nil`, in all modes of the *Chop* command pathSeg is converted to polygon. In line mode, a path or pathSeg is chopped and chop shapes are exactly according to the line drawn.

In wire mode, if the environment variable is set to `t` or `nil`, a path or pathSeg remains a path or pathSeg after *Chop* command.

GUI Equivalent

None

Examples

```
envGetVal("layout" "leChopPreservePathSeg")
envSetVal("layout" "leChopPreservePathSeg" 'boolean nil)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

leDispNetExpr

```
layout leDispNetExpr boolean { t | nil }
graphic leDispNetExpr boolean { t | nil }
```

Description

Checks if the net expression of a terminal is displayed in the text label of pins.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Net Expressions*

Examples

```
envGetVal("layout" "leDispNetExpr")
envSetVal("layout" "leDispNetExpr" 'boolean nil)
envGetVal("graphic" "leDispNetExpr")
envSetVal("graphic" "leDispNetExpr" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

leWindowBBox

```
layout leWindowBBox string screenCoordinates
```

Description

Specifies the default location and size of a layout window.

The default is `((120 185) (1020 875))`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "leWindowBBox")
envSetVal("layout" "leWindowBBox" 'string "xyz")
```

Related Topics

Layout XL Desktop

© 2023 Cadence Design Systems, Inc.

lockAngles

```
layout lockAngles boolean { t | nil }
```

Description

Specifies that angles of partially selected shapes are not modified when the shapes are stretched. This option has no effect on instances or arrays.

The default is `t`.

GUI Equivalent

Command:	*Edit – Stretch*
Field:	*Lock angles*

Examples

```
envGetVal("layout" "lockAngles")
envSetVal("layout" "lockAngles" 'boolean nil)
```

Related Topics

Stretch Form

© 2023 Cadence Design Systems, Inc.

lppVisibilityMode

```
layout lppVisibilityMode cyclic { "checkValidity" | "donotCheckValidity" }
```

Description

Controls the visibility of a layer purpose pair (LPP) based on its visibility and validity.

- `checkValidity`: Displays only the shapes on valid and visible LPPs in the design window.
- `donotCheckValidity`: Default. Displays shapes on all visible LPPs in the design window, irrespective of the validity of the LPPs.

GUI Equivalent

Command:	*Options – Display*
Field:	*Set LPP Visibility Mode*

Examples

```
envGetVal("layout" "lppVisibilityMode")
envSetVal("layout" "lppVisibilityMode" 'cyclic "checkValidity")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

magAnchorPosition

```
layout magAnchorPosition cyclic { "top left corner" | "top right corner" | "bottom
     left corner" | "bottom right corner" }
```

Description

Specifies where to position the magnifier with respect to the pointer position.

By default, the magnifier appears at the top left corner of the design display area with respect to the pointer.

GUI Equivalent

Command:	*Options – Magnifier*
Field:	*Anchored at*

Examples

```
envGetVal("layout" "magAnchorPosition")
envSetVal("layout" "magAnchorPosition" 'cyclic "top right corner")
envSetVal("layout" "magAnchorPosition" 'cyclic "bottom left corner")
```

Related Topics

Magnifier Options Form

© 2023 Cadence Design Systems, Inc.

magConfined

```
layout magConfined boolean { t | nil }
```

Description

Keeps the magnifier within the design display area and always displays it completely.

The default is `t`. When set to `nil`, the magnifier might not be visible completely if it is close to the edge of the design display canvas.

GUI Equivalent

Command:	*Options – Magnifier*
Field:	*Keep Within Window*

Examples

```
envGetVal("layout" "magConfined")
envSetVal("layout" "magConfined" 'boolean nil)
```

Related Topics

Magnifier Options Form

© 2023 Cadence Design Systems, Inc.

magCursorOffset

```
layout magCursorOffset int magnifierDistance
```

Description

Specifies the distance of the magnifier from the pointer position. You can specify any value from `0` through `100`.

The default is `15`.

GUI Equivalent

Command:	*Options – Magnifier*
Field:	*offset by*

Examples

```
envGetVal("layout" "magCursorOffset")
envSetVal("layout" "magCursorOffset" 'int 35)
```

Related Topics

Magnifier Options Form

© 2023 Cadence Design Systems, Inc.

magDisplayMode

```
layout magDisplayMode cyclic { "Anchored at Cursor Position" | "Anchored at" | "Not
     Anchored" }
```

Description

Specifies the type of anchoring to use for the magnifier.

The default is `Not Anchored`.

GUI Equivalent

Command:	*Options – Magnifier*
Field:	*Anchored at Cursor Position, Anchored at, Not Anchored*

Examples

```
envGetVal("layout" "magDisplayMode")
envSetVal("layout" "magDisplayMode" 'cyclic "Anchored at Cursor Position")
envSetVal("layout" "magDisplayMode" 'cyclic "Anchored at")
```

Related Topics

Magnifier Options Form

© 2023 Cadence Design Systems, Inc.

magFloatPosition

```
layout magFloatPosition cyclic { "Center" | "Position Magnfier" }
```

Description

Lets you customize the position of the magnifier.

- `Center`: The center of the magnifier coincides the pointer position.
- `Position Magnifier`: Default. You can specify the `magCursorOffset` after setting the `magOffsetCorner`.

GUI Equivalent

Command:	*Options – Magnifier*
Field:	*Not Anchored: Center, Position Magnifier*

Examples

```
envGetVal("layout" "magFloatPosition")
envSetVal("layout" "magFloatPosition" 'cyclic "Center")
```

Related Topics

magCursorOffset

magOffsetCorner

Magnifier Options Form

© 2023 Cadence Design Systems, Inc.

magOffsetCorner

```
layout magOffsetCorner cyclic { "top left" | "top right" | "bottom right" | "bottom
     left" }
```

Description

Specifies the magnifier position with respect to the pointer position.

The default is `top left`.

GUI Equivalent

Command:	*Options – Magnifier*
Field:	*Position Magnifier*

Examples

```
envGetVal("layout" "magOffsetCorner")
envSetVal("layout" "magOffsetCorner" 'cyclic "top right")
envSetVal("layout" "magOffsetCorner" 'cyclic "bottom left")
```

Related Topics

Magnifier Options Form

© 2023 Cadence Design Systems, Inc.

magScale

```
layout magScale float zoomingFactor
```

Description

Changes the zooming factor of the magnifier. You can set a value from `1` to `1000`.

The default is `4.0`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "magScale")
envSetVal("layout" "magScale" 'float 10.0)
```

Related Topics

Magnifying the Design Area

© 2023 Cadence Design Systems, Inc.

magSize

```
layout magSize int magnifierSize
```

Description

Specifies the size of the magnifier. You can set any value from `50` through `250`.

The default is `200`.

GUI Equivalent

Command:	*Options – Magnifier*
Field:	*Magnifier Size*

Examples

```
envGetVal("layout" "magSize")
envSetVal("layout" "magSize" 'int 100)
```

Related Topics

Magnifier Options Form

© 2023 Cadence Design Systems, Inc.

maintainConnections

```
layout maintainConnections boolean { t | nil }
```

Description

Checks if automatic re-connection mode for pin-path and pin-pathseg connections is active.

The default is `nil`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Maintain Connections*

Examples

```
envGetVal("layout" "maintainConnections")
envSetVal("layout" "maintainConnections" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

makeCellCellType

```
layout makeCellCellType cyclic { "none" | "block" | "blockRing" | "cover" |
     "coverBump" | "pad" | "padSpacer" | "core" | "coreSpacer" | "coreAntenna" |
     "corner" | "softMacro" | "via" | "blockBlackBox" | "padAreaIO" }
```

Description

Specifies the type of cell you want to create.

The default is `none`.

GUI Equivalent

Command:	*Edit – Hierarchy – Make Cell*
Field:	*Type*

Examples

```
envGetVal("layout" "makeCellCellType")
envSetVal("layout" "makeCellCellType" 'cyclic "corner")
envSetVal("layout" "makeCellCellType" 'cyclic "padAreaIO")
```

Related Topics

Make Cell Form

Cell Types

© 2023 Cadence Design Systems, Inc.

makeCellGridAlignment

```
layout makeCellGridAlignment cyclic { "none" | "snapOrigin" | "pushGrid" }
```

Description

Makes adjustments to the global grids in the new cell being created. It adjusts the global grids to ensure that on-grid geometries do not become off-grid after a Make Cell operation.

Valid values are:

- `none`: (Default) No grid adjustment is performed.
- `snapOrigin`: Snaps the origin of the made cell to the nearest intersection of all grid points. If the origin point for a Make Cell operation is "`Custom`", the origin of the made cell is not updated.
- `pushGrid`: Uses the grid offset mechanism to ensure that the grids computed in the new, made cell are parallel to that of the design from which the Make Cell operation was invoked. The origin of the new, made cell is not updated in this case.

GUI Equivalent

None

Examples

```
envGetVal("layout" "makeCellGridAlignment")
envSetVal("layout" "makeCellGridAlignment" 'cyclic "snapOrigin")
envSetVal("layout" "makeCellGridAlignment" 'cyclic "pushGrid")
```

Related Topics

Make Cell Form

Cell Types

© 2023 Cadence Design Systems, Inc.

makeCellUseInstanceDrawingMode

```
layout makeCellUseInstanceDrawingMode boolean { t | nil }
```

Description

Specifies whether instance drawing mode is used to compute the bounding box of a new cell to determine its origin. For example, when the instance drawing mode is set to PR boundary and you create an instance, the coordinates of the bounding box of the PR boundary are used to determine the origin of the instance if the environment variable is set to `t`.

For example, when you create an instance with this environment variable set to `t` and the instance drawing mode set to PR boundary, the coordinates of the bounding box of the PR boundary are used to determine the origin of the new instance.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "makeCellUseInstanceDrawingMode")
envSetVal("layout" "makeCellUseInstanceDrawingMode" 'boolean t)
```

Related Topics

Making Cells

© 2023 Cadence Design Systems, Inc.

markerAccessMode

```
graphic markerAccessMode cyclic { "read" | "write" }
```

Description

Specifies the mode, read or write, for opening a cellview.

The default is `write`.

GUI Equivalent

Command:	*Verify – Markers – Find*
Field:	*Cellview Access Mode*

Examples

```
envGetVal("graphic" "markerAccessMode")
envSetVal("graphic" "markerAccessMode" 'cyclic "read")
```

Related Topics

Find Marker Form

© 2023 Cadence Design Systems, Inc.

markerDeleteAccessModeIsEdit

```
graphic markerDeleteAccessModeIsEdit boolean { t | nil }
```

Description

Lets you specify if you want to open the cellview in read or edit mode when deleting the markers.

The default is `t`, which means that the cellview is re-opened in edit mode for deleting markers. In this mode, you can additionally set `markerDeleteConfirmModeChange` to `t` to get a confirmation dialog box.

When set to `nil`, markers are deleted and the cellview remains in read mode.

GUI Equivalent

Command:	*Verify – Markers – Delete*, *Verify – Markers – Delete All*
Field:	*Cellview Access Mode*

Examples

```
envGetVal("graphic" "markerDeleteAccessModeIsEdit")
envSetVal("graphic" "markerDeleteAccessModeIsEdit" 'boolean nil)
```

Related Topics

markerDeleteConfirmModeChange

Delete All Markers Form

Delete Marker Form

© 2023 Cadence Design Systems, Inc.

markerDeleteConfirmModeChange

```
graphic markerDeleteConfirmModeChange boolean { t | nil }
```

Description

Opens a dialog box that enables you to confirm whether or not the cellview(s) are to be re-opened in edit mode when you try to delete markers on the cellview(s).

The default is `nil`. In this mode, markers are deleted and the cellviews are re-opened in edit mode with no dialog box. When set to `t`, a dialog box appears where you can specify all, none, or some cellviews to reopen in edit mode to delete markers on them. In this mode, markers are not deleted in cellviews that stay in read mode.

This environment variable has no effect if `markerDeleteAccessModeIsEdit` is set to `nil`.

GUI Equivalent

Command:	*Verify – Markers – Delete*, *Verify – Markers – Delete All*
Field:	*Confirm Access Mode Change*

Examples

```
envGetVal("graphic" "markerDeleteConfirmModeChange")
envSetVal("graphic" "markerDeleteConfirmModeChange" 'boolean t)
```

Related Topics

markerDeleteAccessModeIsEdit

Delete All Markers Form

Delete Marker Form

© 2023 Cadence Design Systems, Inc.

markerDeleteScope

```
graphic markerDeleteScope cyclic { "cellview" | "hierarchy starting from top
     cellview" | "hierarchy starting from current cellview" }
```

Description

Defines the scope of the deletion of markers. You can set the scope for deletion of markers from the current cellview, throughout the hierarchy, or throughout the hierarchy starting with the current cellview.

The default is `cellview`.

GUI Equivalent

Command:	*Verify – Markers – Delete All*
Field:	*Search Scope*

Examples

```
envGetVal("graphic" "markerDeleteScope")
envSetVal("graphic" "markerDeleteScope" 'cyclic "hierarchy starting from top
cellview")
envSetVal("graphic" "markerDeleteScope" 'cyclic "hierarchy starting from current
cellview")
```

Related Topics

Delete All Markers Form

© 2023 Cadence Design Systems, Inc.

markerDeleteSeverities

```
graphic markerDeleteSeverities string { "info" | "acknowledgedWarning" | "warning"
     | "signedOffError" | "error" | "signedOffCriticalError" | "criticalError" |
     "fatalError" }
```

Description

Defines the default severities. By default, the environment variable is set to the following marker severities: `info` `acknowledgedWarning` `warning` `signedOffError` `error` `signedOffCriticalError` `criticalError` `fatalError`.

GUI Equivalent

Command:	*Verify – Markers – Delete All*
Field:	*Severity*

Examples

```
envGetVal("graphic" "markerDeleteSeverities")
envSetVal("graphic" "markerDeleteSeverities" 'string "info")
envSetVal("graphic" "markerDeleteSeverities" 'string "signedOffCriticalError")
envSetVal("graphic" "markerDeleteSeverities" 'string "fatalError")
```

Related Topics

Delete All Markers Form

© 2023 Cadence Design Systems, Inc.

markerDeleteSources

```
graphic markerDeleteSources string { "VLS Connectivity:nil" | "Default:t" }
```

Description

Defines the state of the sources. The string to be specified in the environment variable is tool name followed by its state `t` or `nil`. The tool name and state are separated by a colon (:). Multiple tools are separated by a comma.

The `Default` key can be used to set the state of any other tool by default.

GUI Equivalent

Command:	*Verify – Markers – Delete All*
Field:	*Sources: select*

Examples

```
envGetVal("graphic" "markerDeleteSources")
envSetVal("graphic" "markerDeleteSources" 'string "VLS Connectivity:t")
```

Related Topics

Delete All Markers Form

© 2023 Cadence Design Systems, Inc.

markerDeleteSourcesMode

```
graphic markerDeleteSourcesMode cyclic { "all" | "select" }
```

Description

Defines the source mode for deletion of markers.

- `all`: Default. Signifies all markers that match the severities and scope criteria will be deleted.
- `select`: Signifies that a list of marker owners will be computed and only the sources that have been selected will be deleted.

GUI Equivalent

Command:	*Verify – Markers – Delete All*
Field:	*Sources*

Examples

```
envGetVal("graphic" "markerDeleteSourcesMode")
envSetVal("graphic" "markerDeleteSourcesMode" 'cyclic "select")
```

Related Topics

Delete All Markers Form

© 2023 Cadence Design Systems, Inc.

markerSearchBy

```
graphic markerSearchBy cyclic { "master" | "occurrences" }
```

Description

Specifies which markers will display, markers in the master cell only or all occurrences of markers.

The default is `occurrences`.

GUI Equivalent

Command:	*Verify – Markers – Find*
Form Field:	*Find by*

Examples

```
envGetVal("graphic" "markerSearchBy")
envSetVal("graphic" "markerSearchBy" 'cyclic "master")
```

Related Topics

Find Marker Form

© 2023 Cadence Design Systems, Inc.

markerSearchScope

```
graphic markerSearchScope cyclic { "cellview" | "hierarchy starting from top
     cellview" | "hierarchy starting from current cellview" | "library" }
```

Description

Specifies what part of the design hierarchy you want to search.

The default is `cellview`.

GUI Equivalent

Command:	*Verify – Markers – Find*
Field:	*Search Scope*

Examples

```
envGetVal("graphic" "markerSearchScope")
envSetVal("graphic" "markerSearchScope" 'cyclic "hierarchy starting from current
cellview")
envSetVal("graphic" "markerSearchScope" 'cyclic "library")
```

Related Topics

Find Marker Form

© 2023 Cadence Design Systems, Inc.

markerSeverity

```
graphic markerSeverity string { "fatalError" | "criticalError" |
     "signedOffCriticalError" | "error" | "signedOffError" | "warning" |
     "acknowledgedWarning" | "info" | "annotation" }
```

Description

Specifies the types of warnings and errors to be searched for and highlighted in a design.

Replace *list* with one or more of the following values, separated by white space: `fatalError`, `criticalError`, `signedOffCriticalError`, `error`, `signedOffError`, `warning`, `acknowledgedWarning`, `info`, `annotation`.

- `all` or `both`: Displays all types of markers.
- `none`: Switches off the marker display.

The default is `none`.

GUI Equivalent

Command:	*Verify – Markers – Find*
Field:	*Severity*

Examples

```
envGetVal("graphic" "markerSeverity")
envSetVal("graphic" "markerSeverity" 'string "info fatalError warning")
envSetVal("graphic" "markerSeverity" 'string "none")
envSetVal("graphic" "markerSeverity" 'string "annotation")
```

Related Topics

Find Marker Form

© 2023 Cadence Design Systems, Inc.

markerZoom

```
graphic markerZoom boolean { t | nil }
```

Description

Determines whether to zoom to the next or previous marker while using the schematic editor's Find Marker form. Does not apply to the layout editor.

The default is `nil`.

GUI Equivalent

Command:	*Verify – Markers – Find*
Field:	*Zoom To Markers*

Examples

```
envGetVal("graphic" "markerZoom")
envSetVal("graphic" "markerZoom" 'boolean t)
```

Related Topics

Find Marker Form

© 2023 Cadence Design Systems, Inc.

markerZoomScale

```
layout markerZoomScale int objectDistance
graphic markerZoomScale int objectDistance
```

Description

Enlarges the position of the marker in the window by zooming in on the specified marker; when off, displays the marker but does not zoom in.

The layout default is `1`, the graphic default is `4`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "markerZoomScale")
envSetVal("layout" "markerZoomScale" 'int 2)
envGetVal("graphic" "markerZoomScale")
envSetVal("graphic" "markerZoomScale" 'int 3)
```

Related Topics

Find Marker Form

© 2023 Cadence Design Systems, Inc.

markNetAutoColorCycle

```
layout markNetAutoColorCycle boolean { t | nil }
```

Description

Checks if *Mark Net* command marks the nets by cycling automatically through layers `y0` to `y9` using the colors and styles of those layers.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Auto Highlight Color*

Examples

```
envGetVal("layout" "markNetAutoColorCycle")
envSetVal("layout" "markNetAutoColorCycle" 'boolean nil)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetConnectedLayer

```
layout markNetConnectedLayer boolean { t | nil }
```

Description

Determines whether the settings defined for the connected layers in the *Connected Layers* tab are used for tracing a net.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Customized Connected Layers*

Examples

```
envGetVal("layout" "markNetConnectedLayer")
envSetVal("layout" "markNetConnectedLayer" 'boolean nil)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetExcludePurposes

```
layout markNetExcludePurposes boolean { t | nil }
```

Description

Determines whether to exclude the specified purposes from tracing. The default purpose values that are listed are *annotate*, *boundary*, *error*, *flight*, *label*, and *warning*. You can delete any of the default purpose values.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Exclude Purposes*

Examples

```
envGetVal("layout" "markNetExcludePurposes")
envSetVal("layout" "markNetExcludePurposes" 'boolean nil)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetHilightObj

```
layout markNetHilightObj boolean { t | nil }
```

Description

Sets the *Mark Net* command to display the traced connectivity after every 10,000 shapes are processed.

The default is `nil`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Highlight Incrementally*

Examples

```
envGetVal("layout" "markNetHilightObj")
envSetVal("layout" "markNetHilightObj" 'boolean t)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetNumThreads

```
layout markNetNumThreads int threadMode
```

Description

Controls the chaser thread mode of the *Mark Net* command.

- If you set this environment variable to an integer less than or equal to `0`, the *Mark Net* command runs in auto mode.
- If you set the value to `1`, the *Mark Net* command runs in single thread mode.
- If you set the value to an integer greater than `1`, the *Mark Net* command runs in custom thread mode.

The default is `0`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "markNetNumThreads")
envSetVal("layout" "markNetNumThreads" 'int 5)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetOptionsFile

```
layout markNetOptionsFile string "fileName"
```

Description

Specifies the path to the `markNetOptions` file to be read when a new Virtuoso session is started. You can specify the absolute file path or only the filename. The file path can be specified using Unix shell environment variables such as $HOME and $PWD; for example, `$HOME/myMarkNetOptions` or `$PWD/myMarkNetOptions`.

If you specify only the filename, the file is looked up using the Cadence Search File (CSF) mechanism.

The default is "".

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Save/Load custom settings for Via, Connected and Stop Layers*

Examples

```
envGetVal("layout" "markNetOptionsFile")
envSetVal("layout" "markNetOptionsFile" 'string "$HOME/file1")
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetRange

```
layout markNetRange cyclic { "Top to bottom" | "Current to bottom" | "Top to
     current" | "Top to stop level" | "Current to stop level" | "User defined" }
```

Description

Sets the hierarchical range that the *Mark Net* command uses to determine which nets will highlight and which will not.

- `Top to bottom`: Default. Makes available all nets in the entire hierarchy of the cellview for marking.
- `Current to bottom`: Makes available the nets in the hierarchy from the current level of hierarchy to the Display Level To field in the Display Options form (stop level) for marking.
- `Top to current`: Makes available the nets in the hierarchy from the top level to the current level of hierarchy for marking. `Top to stop level` makes available the nets in the hierarchy from the top level to the level defined as the stop level.
- `Current to stop level`: Makes available the nets in the hierarchy from the current level to the level defined as the stop level.
- `User defined`: Makes available the nets in a numerical hierarchical range for marking in conjunction with `markNetStartLevel` and `markNetStopLevel`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Hierarchy Range*

Examples

```
envGetVal("layout" "markNetRange")
envSetVal("layout" "markNetRange" 'cyclic "Current to bottom")
envSetVal("layout" "markNetRange" 'cyclic "User defined")
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetSaveViaInfo

```
layout markNetSaveViaInfo boolean { t | nil }
```

Description

Lets you specify the via shapes to be saved in the cellview created by running the *Connectivity – Nets – Save All Mark Nets* command.

The default is `nil`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Retain Via Info for Save MarkNets*

Examples

```
envGetVal("layout" "markNetSaveViaInfo")
envSetVal("layout" "markNetSaveViaInfo" 'boolean t)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetSelectOverlapLayer

```
layout markNetSelectOverlapLayer boolean { t | nil }
```

Description

Lets you select the LPP from where the *Mark Net* command starts tracing the net when multiple overlapping shapes are selected.

The default is `nil`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Select From Overlaps*

Examples

```
envGetVal("layout" "markNetSelectOverlapLayer")
envSetVal("layout" "markNetSelectOverlapLayer" 'boolean t)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetStartLevel

```
layout markNetStartLevel int startLevel
```

Description

Sets the start level of the hierarchical range the *Mark Net* command uses to determine which nets will highlight and which will not. Specify any integer from `0` to `32`.

The default is `0`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Start*

Examples

```
envGetVal("layout" "markNetStartLevel")
envSetVal("layout" "markNetStartLevel" 'int 20)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetStopLayer

```
layout markNetStopLayer boolean { t | nil }
```

Description

Determines whether the settings defined for the stop layers in the *Stop Layers* tab are used for tracing a net.

The default is `t`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Enable Stop Layers*

Examples

```
envGetVal("layout" "markNetStopLayer")
envSetVal("layout" "markNetStopLayer" 'boolean nil)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetStopLevel

```
layout markNetStopLevel int stopLevel
```

Description

Sets the stop level of the hierarchical range the *Mark Net* command uses to determine which nets will highlight and which will not. Specify any integer from 0 to 32.

The default is `0`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Stop*

Examples

```
envGetVal("layout" "markNetStopLevel")
envSetVal("layout" "markNetStopLevel" 'int 20)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetThickline

```
layout markNetThickline boolean { t | nil }
```

Description

Checks whether the traced nets are displayed in bold.

The default is `nil`.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Thick Line*

Examples

```
envGetVal("layout" "markNetThickline")
envSetVal("layout" "markNetThickline" 'boolean t)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

markNetViaLayer

```
layout markNetViaLayer boolean { t | nil }
```

Description

Determines whether the settings defined for the via layers on the *Via Layers* tab are used for tracing a net.

The default is `t`. When set to `nil`, the via layers defined in the technology file are used.

GUI Equivalent

Command:	*Connectivity – Nets – Mark – Options*
Field:	*Customized Via Layers*

Examples

```
envGetVal("layout" "markNetViaLayer")
envSetVal("layout" "markNetViaLayer" 'boolean nil)
```

Related Topics

MarkNet Options Form

© 2023 Cadence Design Systems, Inc.

maskLayoutDefaultApp

```
graphic maskLayoutDefaultApp string "registeredApplication"
```

Description

Automates the launching of a default application for cellviews of type `maskLayout`.

This environment variable is ignored when launching Layout XL from a schematic window. In this case, the layout view always opens in Layout XL. This environment variable setting is also ignored when descending into or editing-in-place in the same view type. To apply this setting, the environment variable `saveViewSubType` must be set to `nil`, otherwise the Layout cellview automatically opens in the same application in which it was last saved.

GUI Equivalent

Command:	*File – Set Default Application*
Field:	*Set default application for all layout cellviews to*

Examples

```
envGetVal("graphic" "maskLayoutDefaultApp")
envSetVal("graphic" "maskLayoutDefaultApp" 'string "Layout XL")
```

Related Topics

saveViewSubType

Set Default Application Form

© 2023 Cadence Design Systems, Inc.

maxAutoStoreSelectionObj

```
graphic maxAutoStoreSelectionObj int numberofSelectedElements
```

Description

Sets the maximum number of selected elements with a selection set for the *Max number of selected elements within a set* field on the Selection Options form, accessed by choosing *Options – Selection*. If the number of elements in the selection exceeds this number, the selection set is not kept.

The default is 100.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Max number of selected elements within a set*

Examples

```
envGetVal("graphic" "maxAutoStoreSelectionObj")
envSetVal("graphic" "maxAutoStoreSelectionObj" 'int 80)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

maxAutoStoreSelectionSet

```
graphic maxAutoStoreSelectionSet int numberOfStoredSets
```

Description

Sets the maximum number of stored sets for the *Max number of Stored sets* field on the Selection Options form, accessed by choosing *Options – Selection*.

The default is `10`.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Max number of Stored sets*

Examples

```
envGetVal("graphic" "maxAutoStoreSelectionSet")
envSetVal("graphic" "maxAutoStoreSelectionSet" 'int 7)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

maxPathSegInSpineAreaSelection

```
graphic maxPathSegInSpineAreaSelection int numberOfPathsegs
```

Description

Controls the maximum number of pathsegs up to which spine can be computed. If the number of pathsegs exceeds the environment variable setting, the spine, comprising the pathsegs, is not computed. Setting this variable appropriately helps prevent performance issues.

The default is `10000`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "maxPathSegInSpineAreaSelection")
envSetVal("graphic" "maxPathSegInSpineAreaSelection" 'int 9000)
```

Related Topics

Object Conversion

© 2023 Cadence Design Systems, Inc.

measurementDisplayOn

```
layout measurementDisplayOn boolean { t | nil }
```

Description

Checks whether the *Measurement Display On* filed is set to on.

The default is `nil`.

GUI Equivalent

Command:	*Options – Dynamic Display*
Field:	*Measurement Display On*

Examples

```
envGetVal("layout" "measurementDisplayOn")
envSetVal("layout" "measurementDisplayOn" 'boolean t)
```

Related Topics

Dynamic Display Form

© 2023 Cadence Design Systems, Inc.

measurementTextHeight

```
layout measurementTextHeight int { 4.0 |5.0 | 6.0| ... | 50.0 }
```

Description

Specifies the height of the text displayed during dynamic measurement. Valid values are any integer between `4` and `50`.

The default is `10`.

GUI Equivalent

Command:	*Options – Dynamic Display – Dynamic Measurement*
Field:	*Text Height (pixels)*

Examples

```
envGetVal("layout" "measurementTextHeight")
envSetVal("layout" "measurementTextHeight" 'int 20)
```

Related Topics

Dynamic Display Form

© 2023 Cadence Design Systems, Inc.

mergeEnable

```
layout mergeEnable boolean { t | nil }
graphic mergeEnable boolean { t | nil }
```

Description

Makes the display more readable by removing all the extra edges in the design.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Merge Shapes*

Examples

```
envGetVal("layout" "mergeEnable")
envSetVal("layout" "mergeEnable" 'boolean t)
envGetVal("graphic" "mergeEnable")
envSetVal("graphic" "mergeEnable" 'boolean t)
```

Related Topics

Dynamic Selection Assistant Options Form

© 2023 Cadence Design Systems, Inc.

mergeScope

```
layout mergeScope cyclic { "pcells" | "except current" | "all" }
graphic mergeScope cyclic { "pcells" | "except current" | "all" }
```

Description

Specifies the hierarchies for which the *Enable Merge* command applies.

- `pcells`: Default. All the Pcells in the hierarchy including the current cellview.
- `except current`: All the hierarchies except the current cellview.
- `all`: All the hierarchies including the current cellview.

GUI Equivalent

Command:	*Options – Display*
Field:	*Scope*

Examples

```
envGetVal("layout" "mergeScope")
envSetVal("layout" "mergeScope" 'cyclic "except current")
envGetVal("graphic" "mergeScope")
envSetVal("graphic" "mergeScope" 'cyclic "all")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

mergeVias

```
graphic mergeVias boolean { t | nil }
```

Description

Skips via shapes during the *Merge* operation.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "mergeVias")
envSetVal("graphic" "mergeVias" 'boolean nil)
```

Related Topics

Object Merging

© 2023 Cadence Design Systems, Inc.

mergeXStippleShapes

```
graphic mergeXStippleShapes boolean { t | nil }
```

Description

Lets you merge the layers with X stipple.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "mergeXStippleShapes")
envSetVal("graphic" "mergeXStippleShapes" 'boolean t)
```

Related Topics

Object Merging

© 2023 Cadence Design Systems, Inc.

meshNBThread

```
graphic meshNBThread int numberOfThreads
```

Description

Controls the maximum number of threads activated when the display is in high resolution. When set to an integer less than or equal to `0`, the display in high resolution runs in auto mode.

The default is `5`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "meshNBThread")
envSetVal("graphic " "meshNBThread" 'int 10)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

minInstSizeInPixelsForStretchHandles

`layout minInstSizeInPixelsForStretchHandles int` *`instanceSizeInPixels`*

Description

Zooms the stretch handle display. When you set this environment variable to any integer value between 1 to 500, stretch handles are displayed only when the instances cross that size in pixels.

The default is `0`. When set to `0`, stretch handles are displayed for all instances.

GUI Equivalent

None

Examples

```
envGetVal("layout" "minInstSizeInPixelsForStretchHandles")
envSetVal("layout" "minInstSizeInPixelsForStretchHandles" 'int 100)
```

Related topics

Stretchable Parameterized Cells

displayStretchHandles

© 2023 Cadence Design Systems, Inc.

modalCommands

```
layout modalCommands boolean { t | nil }
```

Description

Specifies if commands are modal (automatically repeat).

The default is `t`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Repeat Commands*

Examples

```
envGetVal("layout" "modalCommands")
envSetVal("layout" "modalCommands" 'boolean nil)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

modCornerRadius

```
layout modCornerRadius float radiusValue
```

Description

Specifies the default modify-corner-radius value.

The default is `1.0` user units.

GUI Equivalent

Command:	*Edit – Advanced – Modify Corner*
Field:	*Radius*

Examples

```
envGetVal("layout" "modCornerRadius")
envSetVal("layout" "modCornerRadius" 'float 2.0)
```

Related Topics

Modify Corner Form

© 2023 Cadence Design Systems, Inc.

modCornerType

```
layout modCornerType cyclic { "radial" | "chamfer" }
```

Description

Specifies the default modify-corner type.

The default is `radial`.

GUI Equivalent

Command:	*Edit – Advanced – Modify Corner*
Field:	*Type of Corner*

Examples

```
envGetVal("layout" "modCornerType")
envSetVal("layout" "modCornerType" 'cyclic "chamfer")
```

Related Topics

Modify Corner Form

© 2023 Cadence Design Systems, Inc.

moveConnectedInstancesAndWires

```
layout moveConnectedInstancesAndWires boolean { t | nil }
```

Description

(Layout EXL Only) Specifies whether the instances and wires connected to an instance are also moved or stretched to maintain connectivity when you move the instance.

The default is `nil`, which means that connected instances and wires are not moved.

GUI Equivalent

Command:	*Edit – Stretch*
Field:	*Move Connected Instances And Wires*

Examples

```
envGetVal("layout" "moveConnectedInstancesAndWires")
envSetVal("layout" "moveConnectedInstancesAndWires" 'boolean t)
```

Related Topics

Stretch Form

© 2023 Cadence Design Systems, Inc.

mppASCIIFileName

```
layout mppASCIIFileName string { "cellTechLibName.mpp.templates.il" |
     "your_ASCII_file_name.il" }
```

Description

Specifies the default file name or path and file name displayed in the *ASCII File* field in the Save Multipart Path Templates form and the *Template File* field in the Load Multipart Path Template form.

The default is `cellTechLibName.mpp.templates.il`.

GUI Equivalent

Command:	*Create – Multipart Path – Load Template* or *Create – Multipart Path – Save Template*
Field:	*Template File* or *ASCII File*

Examples

```
envGetVal("layout" "mppASCIIFileName")
envSetVal("layout" "mppASCIIFileName" 'string "cellTechLibName.mpp.templates.il")
```

Related Topics

Magnifier Options Form

Save Multipart Path Templates Form

© 2023 Cadence Design Systems, Inc.

mppTemplate

```
layout mppTemplate string { "New" |
     "the_name_of_any_MPP_template_defined_in_your_technology_file" }
```

Description

Specifies the default for the *MPP Template* field in the Create Multipart Path form. You can choose *New* or the name of a specific MPP template you created.

The default is `New`.

GUI Equivalent

Command:	*Create – Multipart Path*
Field:	*MPP Template*

Examples

```
envGetVal("layout" "mppTemplate")
envSetVal("layout" "mppTemplate" 'string "cellTechLibName.mpp.templates.il")
```

Related Topics

Create Multipart Path Form

© 2023 Cadence Design Systems, Inc.

multiMfgResolutionSnap

```
layout multiMfgResolutionSnap boolean { t | nil }
```

Description

(Virtuoso Photonics Option) Checks if the objects are snapped to higher manufacturing grids instead of the base manufacturing grid during creation, stretching, moving, and copying of instances, shapes, pins, and vias. Pins and shapes are snapped to the manufacturing grid defined for the layer in the technology file. Instances or vias are snapped to the least common multiple value of manufacturing grids of all the layers in the instance or via.

The default is `nil`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Layer Manufacturing Grid*

Examples

```
envGetVal("layout" "multiMfgResolutionSnap")
envSetVal("layout" "multiMfgResolutionSnap" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

mustConnectAllPins

```
layout mustConnectAllPins boolean { t | nil }
```

Description

Checks if the `LEF58_MUSTJOINALLPORTS` property is applied to all newly created pin terminals in the layout, which implies that the router must connect all pins of the selected terminal. This setting is applicable only when *Connectivity* in the Create Pin form is set to *Weak*.

The default is `nil`, in which case the property is not applied.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Must Connect All Pins*

Examples

```
envGetVal("layout" "mustConnectAllPins")
envSetVal("layout" "mustConnectAllPins" 'boolean t)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

multiThreadInfo

```
graphic multiThreadInfo boolean { t | nil }
```

Description

Displays information messages about the number of threads running when redrawing the canvas. The number of threads varies according to the display resolution.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "multiThreadInfo")
envSetVal("graphic" "multiThreadInfo" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

muteUnstretchableOption

```
layout muteUnstretchableOption boolean { t | nil }
```

Description

Lets you control whether to display a dialog box when you attempt to stretch multiple figures and only some of the figures in the selected set can be legally stretched.

The default is `t`, which means that no form appears, and you are only able to stretch the legally stretchable figures; figures that cannot legally be stretched do not stretch.

When set to `nil`, a form opens to let you choose whether to stretch only the legally stretchable figures within the selected set or to cancel the stretch command. You can partially select three rectangles, and your stretch would cause one rectangle to have zero area, a dialog box appears, stating that one shape in the selected set will not be stretched. You can click *OK* to stretch the two stretchable rectangles and leave the unstretchable rectangle unstretched, or you can cancel the command.

GUI Equivalent

None

Examples

```
envGetVal("layout" "muteUnstretchableOption")
envSetVal("layout" "muteUnstretchableOption" 'boolean nil)
```

Related Topics

Object Stretching

© 2023 Cadence Design Systems, Inc.

netBasedSelectionNetName

```
layout netBasedSelectionNetName string "the_name_of_the_net"
graphic netBasedSelectionNetName string "the_name_of_the_net"
```

Description

Specifies the name of the nets for which the shapes, vias, and instances can be selected. You can specify the name of a single net or multiple nets separated by a space.

GUI Equivalent

None

Examples

```
envGetVal("layout" "netBasedSelectionNetName")
envSetVal("layout" "netBasedSelectionNetName" 'string "net1 net2")
envGetVal("graphic" "netBasedSelectionNetName")
envSetVal("graphic" "netBasedSelectionNetName" 'string "net1 net2")
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

netBasedSelectionOn

```
layout netBasedSelectionOn boolean { t | nil }
graphic netBasedSelectionOn boolean { t | nil }
```

Description

Checks if the *Net Based Selection* option is enabled.

The default is `nil`.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Net Based Selection*

Examples

```
envGetVal("layout" "netBasedSelectionOn")
envSetVal("layout" "netBasedSelectionOn" 'boolean t)
envGetVal("graphic" "netBasedSelectionOn")
envSetVal("graphic" "netBasedSelectionOn" 'boolean t)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

netBasedSelectionReverse

```
layout netBasedSelectionReverse boolean { t | nil }
```

```
graphic netBasedSelectionReverse boolean { t | nil }
```

Description

Determines whether all assigned shapes, vias, and instances with a net other than that specified in the text field can be selected.

The default is `nil`.

GUI Equivalent

Command:	*Windows – Toolbars – Net Based Selection*
Field:	*Reverse*

Examples

```
envGetVal("layout" "netBasedSelectionReverse")
envSetVal("layout" "netBasedSelectionReverse" 'boolean t)
envGetVal("graphic" "netBasedSelectionReverse")
envSetVal("graphic" "netBasedSelectionReverse" 'boolean t)
```

Related Topics

Ways to Select Shapes Based on Nets

© 2023 Cadence Design Systems, Inc.

netExprDisplay

```
graphic netExprDisplay string { "value only" | "value (default)" | "value
     [property]" | "value [expression]" | "expression only" }
```

Description

Specifies how to display inherited connections values. The value is what the connection resolves to; if it is overridden at a higher level in the hierarchy, the full path to a local signal might be displayed.

The default is `expression only`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Net Expressions*

Examples

```
envGetVal("graphic" "netExprDisplay")
envSetVal("graphic" "netExprDisplay" 'string "value only")
envSetVal("graphic" "netExprDisplay" 'string "value (default)")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

netName

```
layout netName string net<i>Name</i>
```

Description

Specifies the name of the net with which you want to associate the object being created.

GUI Equivalent

Command:	Occurs in multiple forms
Field:	*Net Name*

Examples

```
envGetVal("layout" "netName")
envSetVal("layout" "netName" 'string "net1")
```

Related Topics

Create Rectangle Form

© 2023 Cadence Design Systems, Inc.

netNameDisplayDrawBackground

```
graphic netNameDisplayDrawBackground boolean { t | nil }
```

Description

Displays net name labels with a black background.

The default is `t`. When set to `nil`, net name labels are displayed without a background.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Draw Background - Net Name*

Examples

```
envGetVal("graphic" "netNameDisplayDrawBackground")
envSetVal("graphic" "netNameDisplayDrawBackground" 'boolean nil)
```

Related Topics

Net Name Display Options form

Net Name Display

© 2023 Cadence Design Systems, Inc.

netNameDisplayDrawOnTop

```
graphic netNameDisplayDrawOnTop boolean { t | nil }
```

Description

Displays net name labels on top of the shapes in the design.

The default is `nil`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Draw Labels on Top- Net Name*

Examples

```
envGetVal("graphic" "netNameDisplayDrawOnTop")
envSetVal("graphic" "netNameDisplayDrawOnTop" 'boolean t)
```

Related Topics

Net Name Display Options

Net Name Display

© 2023 Cadence Design Systems, Inc.

netNameDisplayEnabled

```
graphic netNameDisplayEnabled boolean { t | nil }
```

Description

Displays net name labels on layout shapes that are big enough to display a label.

The default is `t`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Enable net name display- Net Name*

Examples

```
envGetVal("graphic" "netNameDisplayEnabled")
envSetVal("graphic" "netNameDisplayEnabled" 'boolean nil)
```

Related Topics

Net Name Display Options

Net Name Display

© 2023 Cadence Design Systems, Inc.

netNameDisplayFilter

```
graphic netNameDisplayFilter boolean { t | nil }
```

Description

Controls the selective display of net name labels. This means depending on the filter mode selected, the specified nets have their name labels either shown or hidden in the layout canvas.

The default is `nil`, which means all nets have their labels displayed.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*filter net names*

Examples

```
envGetVal("graphic" "netNameDisplayFilter")
envSetVal("graphic" "netNameDisplayFilter" 'boolean t)
```

Related Topics

Net Name Filtering

Net Name Display

© 2023 Cadence Design Systems, Inc.

netNameDisplayFilterDefaultList

```
graphic netNameDisplayFilterDefaultList string "net1 net2 net3"
```

Description

Adds a default list of net names to be used for name filtering, if no information is currently stored in the design.

The default is `""`. This means, by default, no filter list is available.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Filter net names*

Examples

```
envGetVal("graphic" "netNameDisplayFilterDefaultList")
envSetVal("graphic" "netNameDisplayFilterDefaultList" 'string "vdd vss")
```

Related Topics

Net Name Filtering

Net Name Display

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayFilterLayers

```
graphic netNameDisplayFilterLayers boolean { t | nil }
```

Description

Controls the selective display of layers for which name of the net is not displayed.

The default is `nil`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Hide Layers*

Examples

```
envGetVal("graphic" "netNameDisplayFilterLayers")
envSetVal("graphic" "netNameDisplayFilterLayers" 'boolean t)
```

Related Topics

Net Name Filtering

Net Name Display

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayFilterMode

```
graphic netNameDisplayFilterMode cyclic { "Show" | "Hide" }
```

Description

Controls whether the net name labels for the specified nets are displayed or hidden.

The default is `Hide`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Filter Mode*

Examples

```
envGetVal("graphic" "netNameDisplayFilterMode")
envSetVal("graphic" "netNameDisplayFilterMode" 'cyclic "Show")
```

Related Topics

Net Name Filtering

Net Name Display

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayFontStyle

```
graphic netNameDisplayFontStyle string { "euroStyle" | "gothic" | "math" | "roman"
     | "script" | "stick" | "fixed" | "swedish" | "raster" | "milSpec" | "serif" |
     "sansSerif" | "mono" | "serifItalic" | "serifBold" | "serifBoldItalic" |
     "sansBold" | "sansLight" | "nil" }
```

Description

Sets the font style to use for displaying the net name label.

The default is `"stick"`.

Note: The font style that you set using the environment variable applies only for the current session.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "netNameDisplayFontStyle")
envSetVal("graphic" "netNameDisplayFontStyle" 'string "euroStyle")
envSetVal("graphic" "netNameDisplayFontStyle" 'string "gothic")
```

Related Topics

Net Name Display

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayMaxLabelSize

```
graphic netNameDisplayMaxLabelSize int sizeOfLabel
```

Description

Specifies the size of the label on the net. You can use the slider to select the size.

The default is `15`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Label size slider*

Examples

```
envGetVal("graphic" "netNameDisplayMaxLabelSize")
envSetVal("graphic" "netNameDisplayMaxLabelSize" 'int 30)
```

Related Topics

Net Name Filtering

Net Name Display

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayMaxLabelSizeFull

```
graphic netNameDisplayMaxLabelSizeFull boolean { t | nil }
```

Description

Controls whether the label created is of maximum size such that it covers the net.

The default is `nil`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Full size*

Examples

```
envGetVal("graphic" "netNameDisplayMaxLabelSizeFull")
envSetVal("graphic" "netNameDisplayMaxLabelSizeFull" 'boolean t)
```

Related Topics

Net Name Filtering

Net Name Display

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayNetShow

```
graphic netNameDisplayNetShow boolean { t | nil }
```

Description

Displays the net name labels on shapes.

The default is `t`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Net Name: Show*

Examples

```
envGetVal("graphic" "netNameDisplayNetShow")
envSetVal("graphic" "netNameDisplayNetShow" 'boolean nil)
```

Related Topics

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayScope

```
graphic netNameDisplayScope cyclic { "edit" | "full" }
```

Description

Displays the net name in full hierarchy or only in the current edit level.

The default is `edit`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Hierarchy Range*

Examples

```
envGetVal("graphic" "netNameDisplayScope")
envSetVal("graphic" "netNameDisplayScope" 'cyclic "full")
```

Related Topics

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplaySkipInstances

```
graphic netNameDisplaySkipInstances boolean { t | nil }
```

Description

Prevents the net names from being displayed on the instance pins.

The default is `nil`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Net Name: On Instance pins*

Examples

```
envGetVal("graphic" "netNameDisplaySkipInstances")
envSetVal("graphic" "netNameDisplaySkipInstances" 'boolean t)
```

Related Topics

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplaySkipPcells

```
graphic netNameDisplaySkipPcells boolean { t | nil }
```

Description

Prevents the net names from being displayed on the Pcell pins.

The default is `nil`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Net Name: On Pcell pins*

Examples

```
envGetVal("graphic" "netNameDisplaySkipPcells")
envSetVal("graphic" "netNameDisplaySkipPcells" 'boolean t)
```

Related Topics

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplaySkipLayers

```
graphic netNameDisplaySkipLayers string "layerName"
```

Description

Specifies the name of the layers for which the net name is not displayed.

The default is `""`, which means that by default, no filter list is available.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Layer Names*

Examples

```
envGetVal("graphic" "netNameDisplaySkipLayers")
envSetVal("graphic" "netNameDisplaySkipLayers" 'string "ndiff pdiff")
envSetVal("graphic" "netNameDisplaySkipLayers" 'string "")
```

Related Topics

Net Name Display

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplaySkipVias

```
graphic netNameDisplaySkipVias boolean { t | nil }
```

Description

Prevents the net name labels from being displayed on vias.

The default is `nil`, which means by default, net name labels are displayed on vias.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Skip labels on vias*

Examples

```
envGetVal("graphic" "netNameDisplaySkipVias")
envSetVal("graphic" "netNameDisplaySkipVias" 'boolean t)
```

Related Topics

Net Name Display

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayTermShow

```
graphic netNameDisplayTermShow boolean { t | nil }
```

Description

Displays terminal names on the shapes.

The default is `t`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Terminal Name: Show*

Examples

```
envGetVal("graphic" "netNameDisplayTermShow")
envSetVal("graphic" "netNameDisplayTermShow" 'boolean nil)
```

Related Topics

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayTermSkipInstances

```
graphic netNameDisplayTermSkipInstances boolean { t | nil }
```

Description

Prevents the terminal names from being displayed on the instance pins.

The default is `nil`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Terminal Name: On Instance pins*

Examples

```
envGetVal("graphic" "netNameDisplayTermSkipInstances")
envSetVal("graphic" "netNameDisplayTermSkipInstances" 'boolean t)
```

Related Topics

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayTermSkipPcells

```
graphic netNameDisplayTermSkipPcells boolean { t | nil }
```

Description

Prevents the terminal names from being displayed on Pcell pins.

The default is `nil`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Terminal Name: On Pcell pins*

Examples

```
envGetVal("graphic" "netNameDisplayTermSkipPcells")
envSetVal("graphic" "netNameDisplayTermSkipPcells" 'boolean t)
```

Related Topics

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayUserColor

```
graphic netNameDisplayUserColor string { "white" | "yellow" | "anyColorFromPalette"
     }
```

Description

Sets the font color to use for the net name label. You can choose from any of the colors in the palette.

The default is `"white"`.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Color – User*

Examples

```
envGetVal("graphic" "netNameDisplayUserColor")
envSetVal("graphic" "netNameDisplayUserColor" 'string "magenta")
envSetVal("graphic" "netNameDisplayUserColor" 'string "cyan")
```

Related Topics

Net Name Display

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netNameDisplayUseShapeColor

```
graphic netNameDisplayUseShapeColor boolean { t | nil }
```

Description

Sets the font color for the net name label to that of the shape with which the label is associated.

The default is `nil`, which means that all the net name labels are drawn using the font color that is either specified using the `netNameDisplayUserColor` environment variable or selected from the *Color – User* color palette in the Net Name Display Options form.

GUI Equivalent

Command:	*Options – Net Name Display*
Field:	*Color – Same as Shape*
	Color – User

Examples

```
envGetVal("graphic" "netNameDisplayUseShapeColor")
envSetVal("graphic" "netNameDisplayUseShapeColor" 'boolean t)
```

Related Topics

Net Name Display

netNameDisplayUserColor

Net Name Display Options

© 2023 Cadence Design Systems, Inc.

netsOn

```
layout netsOn boolean { t | nil }
graphic netsOn boolean { t | nil }
```

Description

Verifies if the nets are highlighted or not. Enabling `netsOn` can affect the redraw performance.

The layout default is `nil`, the graphic default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Nets*

Examples

```
envGetVal("layout" "netsOn")
envSetVal("layout" "netsOn" 'boolean t)
envGetVal("graphic" "netsOn")
envSetVal("graphic" "netsOn" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

nPtsToTaper

```
layout nPtsToTaper int numberOfSegments
```

Description

Specifies the default number of segments in the taper. This is only used for exponentially tapered lines.

The default is `20`.

GUI Equivalent

Command:	*Create – Microwave – Taper*
Field:	*Resolution*

Examples

```
envGetVal("layout" "nPtsToTaper")
envSetVal("layout" "nPtsToTaper" 'int 30)
```

Related Topics

Transmission Line Taper Form

© 2023 Cadence Design Systems, Inc.

numLevels

```
layout numLevels int numberOfLevels
```

Description

Specifies the number of levels to copy in `leYankFigs`.

The default is `32`.

GUI Equivalent

Command:	*Edit – Basic – Yank*
Field:	*Yank Levels*

Examples

```
envGetVal("layout" "numLevels")
envSetVal("layout" "numLevels" 'int 10)
```

Related Topics

Yank Form

© 2023 Cadence Design Systems, Inc.

numSides

```
layout numSides int numberOfSides
```

Description

Specifies the number of sides used to convert a conic to a polygon.

The default is `20`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Conic Sides*

Examples

```
envGetVal("layout" "numSides")
envSetVal("layout" "numSides" 'int 100)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

openSavedCellViewInWindow

```
layout openSavedCellViewInWindow boolean { t | nil }
```

Description

Lets you specify if you want to open a saved copy of the design.

The default is `nil`.

GUI Equivalent

Command:	*File – Save a Copy*
Field:	*Open Saved Copy*

Examples

```
envGetVal("layout" "openSavedCellViewInWindow")
envSetVal("layout" "openSavedCellViewInWindow" 'boolean t)
```

Related Topics

Save a Copy Form

© 2023 Cadence Design Systems, Inc.

openToStopLevel

```
layout openToStopLevel boolean { t | nil }
graphic openToStopLevel boolean { t | nil }
```

Description

Determines when the cellview data is loaded during rendering.

The default is `t`. All data is loaded to the window stop level (corresponding to the `displayStopLevel` variable), then rendered. All data is loaded, even data too small to see.

When set to `nil`, data loads on demand during rendering, to the `displayResolution` limit. This can slightly slow initial redraw, zooms or pans, but will dramatically reduce the amount of data loaded, since data too small to see is not rendered.

GUI Equivalent

Command:	*Options – Display*
Field:	*Open To Stop Level*

Examples

```
envGetVal("layout" "openToStopLevel")
envSetVal("layout" "openToStopLevel" 'boolean nil)
envGetVal("graphic" "openToStopLevel")
envSetVal("graphic" "openToStopLevel" 'boolean nil)
```

Related Topics

Display Options Form

displayResolution

© 2023 Cadence Design Systems, Inc.

orientation

```
layout orientation cyclic { "R0" | "R90" | "R180" | "R270" | "MY" | "MYR90" | "MX"
     | "MXR90" }
```

Description

Specifies the incremental orientation with which objects are placed during creation or editing, with an initial setting of `R0`. The default is `R0`.

This environment variable corresponds to the *Rotate*, *Sideways*, and *Upside Down* buttons on any form. It resets to `R0` after each usage, so you can use it only to set an orientation bindkey. Each time you press the bindkey the value specified is added to whichever one of the eight legal orientations the object is currently in (`R0`, `R90`, `R180`, `R270`, `MY`, `MYR90`, `MX`, or `MXR90`). Therefore, this variable is only used as a relative value.

GUI Equivalent

None

Examples

```
envGetVal("layout" "orientation")
envSetVal("layout" "orientation" 'cyclic "R90")
envSetVal("layout" "orientation" 'cyclic "MXR90")
```

Related Topics

resetOrientation

© 2023 Cadence Design Systems, Inc.

originMarkersOn

```
layout originMarkersOn boolean { t | nil }
graphic originMarkersOn boolean { t | nil }
```

Description

Determines whether the cell origins are displayed when viewing the bounding boxes of cell instances. The origins appear as plus signs.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Instance Origins*

Examples

```
envGetVal("layout" "originMarkersOn")
envSetVal("layout" "originMarkersOn" 'boolean nil)
envGetVal("graphic" "originMarkersOn")
envSetVal("graphic" "originMarkersOn" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

overrideFontName

```
layout overrideFontName string "font_name"
graphic overrideFontName string "font_name"
```

Description

Specifies fonts that override the default fonts.

Supported font choices are `euroStyle`, `gothic`, `math`, `roman`, `script`, `stick`, `fixed`, `swedish`, `raster`, `milSpec`, `serif`, `sansSerif`, `mono`, `serifItalic`, `serifBold`, `serifBoldItalic`, `sansBold`, and `sansLight`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "overrideFontName")
envSetVal("layout" "overrideFontName" 'string "(stick sansLight) (gothic raster)"
)
envGetVal("graphic" "overrideFontName")
envSetVal("graphic" "overrideFontName" 'string "(stick sansLight) (gothic raster)"
)
```

In the above example, the second pair of font replaces the first one.

Related Topics

useOverrideFontName

© 2023 Cadence Design Systems, Inc.

packetDialogBoxes

```
graphic packetDialogBoxes boolean { t | nil }
```

Description

Checks if a message box appears when a packet assigned to a layer in the technology file is not defined in the `display.drf` file. The message box lists the undefined packets.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "packetDialogBoxes")
envSetVal("graphic" "packetDialogBoxes" 'boolean nil)
```

Related Topics

Layout Editor Default Setup

© 2023 Cadence Design Systems, Inc.

padLabelLPP

```
layout padLabelLPP string { "lpp" }
```

Description

Specifies the layer-purpose pair on which the *Pad Opening Info* command draws pad labels.

The default is `"text drawing"`.

GUI Equivalent

Command:	*Tools – Pad Opening Info; Mode: Generate Labels*
Field:	*Label Control: Label Options: Layer/Purpose*

Examples

```
envGetVal("layout" "padLabelLPP")
envSetVal("layout" "padLabelLPP" 'string "edgeLayer drawing Via5 drawing Metal7
drawing NWell drawing")
```

Related Topics

Pad Opening Info Form

© 2023 Cadence Design Systems, Inc.

partialSelect

```
layout partialSelect boolean { t | nil }
graphic partialSelect boolean { t | nil }
```

Description

Controls whether you can select vertices or line segments of an object.

The layout default is `nil`, the graphic default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "partialSelect")
envSetVal("layout" "partialSelect" 'boolean t)
envGetVal("graphic" "partialSelect")
envSetVal("graphic" "partialSelect" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

partialSelectProximityMaxPixel

`graphic partialSelectProximityMaxPixel int` *`distanceBetweenVertexAndCursor`*

Description

Overrides the current shape selection behavior. Setting this variable enables you to select the entire shape by clicking anywhere on the center line or by selecting an edge, when partial select is on and you are zoomed in at the shape end.

The default is `60`. To override current selection behavior, set this value to `5`.

If you want to select a vertex, the distance between the vertex and the cursor should be less than the value you set here.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "partialSelectProximityMaxPixel")
envSetVal("graphic" "partialSelectProximityMaxPixel" 'int 50)
```


© 2023 Cadence Design Systems, Inc.

partialSelectionMode

```
graphic partialSelectionMode string { "vertexAndEdge" | "enclosedEdge" |
     "crossedEdge" | "enclosedAndCrossed" }
```

Description

Specifies whether the partial selection mode is set to vertex and edge, enclosed edge or crossed edge, or enclosed and crossed.

- `vertexAndEdge`: Default. Fully selects all objects you completely enclose with the area selection box, vertices of any objects you partially select with the area selection box, and edges of any objects you enclose with the area selection box.
- `enclosedEdge`: Fully selects all objects you completely enclose with the area selection box, and the edges of any objects you partially enclose with the area selection box.
- `crossedEdge`: Selects the edges of any objects you enclose with the area selection box, and does not select fully enclosed objects.
- `enclosedAndCrosse`: Fully selects all objects you completely enclose with the area selection box, the vertices of any objects you partially enclose with the area selection box, and the edges of any objects you enclose with the area selection box.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Partial Mode*

Examples

```
envGetVal("graphic" "partialSelectionMode")
envSetVal("graphic" "partialSelectionMode" 'string "enclosedEdge")
envSetVal("graphic" "partialSelectionMode" 'string "crossedEdge")
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

pasteRef

```
layout pasteRef cyclic { "origin" | "lowerLeft" | "lowerCenter" | "lowerRight" |
     "centerLeft" | "centerCenter" | "centerRight" | "upperLeft" | "upperCenter" |
     "upperRight" | "user specified" }
```

Description

Specifies the reference point of the bounding box when pasting. You can select one of the ten reference points when pasting.

The default is `user specified`.

GUI Equivalent

Command:	*Edit – Basic – Paste*
Field:	*Reference Points*

Examples

```
envGetVal("layout" "pasteRef")
envSetVal("layout" "pasteRef" 'cyclic "origin")
envSetVal("layout" "pasteRef" 'cyclic "centerLeft")
```

Related Topics

Paste Form

© 2023 Cadence Design Systems, Inc.

pathBeginExt

`layout pathBeginExt float` *`pathExtension`*

Description

Specifies the beginning extension of paths with variable end type.

The default is `0.0` user units.

GUI Equivalent

Command:	*Create – Shape – Path* and *Create – Wiring –Wire*
Field:	*Begin Extension* and *Begin*

Examples

```
envGetVal("layout" "pathBeginExt")
envSetVal("layout" "pathBeginExt" 'float 1.0)
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

pathCL

```
layout pathCL cyclic { "yes" | "no" | "only" }
graphic pathCL string { "yes" | "no" | "only" }
```

Description

Lets you specify if you want to see path borders and center lines, path borders only, or path center lines only.

The default is `yes`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Path Display*

Examples

```
envGetVal("layout" "pathCL")
envSetVal("layout" "pathCL" 'cyclic "no")
envGetVal("graphic" "pathCL")
envSetVal("graphic" "pathCL" 'string "only")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

pathEndExt

`layout pathEndExt float` *`pathExtension`*

Description

Specifies the end extension of paths with variable end type.

The default is `0.0` user units.

GUI Equivalent

Command:	*Create – Shape – Path* and *Create – Wiring –Wire*
Field:	*End Extension* and *End*

Examples

```
envGetVal("layout" "pathEndExt")
envSetVal("layout" "pathEndExt" 'float 1.0)
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

pathJustify

```
layout pathJustify cyclic { "left" | "center" | "right" }
```

Description

Specifies the justification of paths.

The default is `center`.

GUI Equivalent

Command:	*Create – Shape – Path*
Field:	*Justification*

Examples

```
envGetVal("layout" "pathJustify")
envSetVal("layout" "pathJustify" 'cyclic "right")
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

pathOffset

```
layout pathOffset float pathOffset
```

Description

Specifies the offset of a path center-line from the drawn coordinates.

The default is `0.0` user units.

GUI Equivalent

Command:	*Create – Shape – Path*
Field:	*Offset*

Examples

```
envGetVal("layout" "pathOffset")
envSetVal("layout" "pathOffset" 'float 1.0)
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

pathStyle

```
layout pathStyle cyclic { "truncateExtend" | "extendExtend" | "roundRound" |
     "varExtendExtend" }
```

Description

(Virtuoso RF Option) Specifies the default end type for paths.

The default is `truncateExtend`.

GUI Equivalent

Command:	*Create – Shape – Path* and *Create – Wiring –Wire*
Field:	*End Type* and *End Style*

Examples

```
envGetVal("layout" "pathStyle")
envSetVal("layout" "pathStyle" 'cyclic "extendExtend")
envSetVal("layout" "pathStyle" 'cyclic "varExtendExtend")
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

pathWidth

```
layout pathWidth float width
graphic pathWidth float width
```

Description

Specifies the default width of paths.

The layout default is `0.6` user units, the graphic default is `5.0` user units.

GUI Equivalent

Command:	*Create – Shape – Path*
Field:	*Width*

Examples

```
envGetVal("layout" "pathWidth")
envSetVal("layout" "pathWidth" 'float 2.0)
envGetVal("graphic" "pathWidth")
envSetVal("graphic" "pathWidth" 'float 2.0)
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

pinAccessDir

```
layout pinAccessDir string { "pinAccessDirections" }
```

Description

Specifies the default access directions of a pin.

The default is `list ("left" "right" "top" "bottom")`.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Access Direction*

Examples

```
envGetVal("layout" "pinAccessDir")
envSetVal("layout" "pinAccessDir" 'string "left right")
```

Related Topics

Pin Creation

© 2023 Cadence Design Systems, Inc.

pinDefaultObjectType

```
layout pinDefaultObjectType cyclic ( "Label" | "Text Display" | "Both" )
```

Description

Specifies the default object type while creating pins from objects.

The default is `Both`.

GUI Equivalent

Command:	*Tools – Create Pins From Labels*
Field:	*Object Type*

Related Topics

Pin Creation

© 2023 Cadence Design Systems, Inc.

pinLabelPlacementMode

```
layout pinLabelPlacementMode cyclic { "manual" | "auto" }
```

Description

Determines whether the mode to create labels for pins is manual or auto.

The default is `manual`.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Create Label*

Examples

```
envGetVal("layout" "pinLabelPlacementMode")
envSetVal("layout" "pinLabelPlacementMode" 'cyclic "auto")
```

Related Topics

Pin Creation

© 2023 Cadence Design Systems, Inc.

pinLayerPurpose

```
layout pinLayerPurpose string { "any_valid_layer_purpose" | "" }
```

Description

Specifies the layer purpose on which a shape pin is placed. You can specify any valid layer purpose, such as `"pin"` or `"drawing"`.

The default is " ".

GUI Equivalent

None

Examples

```
envGetVal("layout" "pinLayerPurpose")
envSetVal("layout" "pinLayerPurpose" 'string "pin")
```

Related Topics

Pin Creation

© 2023 Cadence Design Systems, Inc.

pinLPP

```
layout pinLPP cyclic { "sameLayer" | "currentLayer" }
```

Description

Determines the layer purpose pair to be used for creating auto pins.

The default is `sameLayer`.

GUI Equivalent

Command:	*Create – Pin; Mode: Auto*
Field:	*Pin Layer/Purpose*

Examples

```
envGetVal("layout" "pinLPP")
envSetVal("layout" "pinLPP" 'cyclic "sameLayer")
```

Related Topics

Pin Creation

© 2023 Cadence Design Systems, Inc.

pinMode

```
layout pinMode cyclic { "Manual" | "Auto" }
```

Description

Determines the mode for creating pins.

The default is `Manual`.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Mode*

Examples

```
envGetVal("layout" "pinMode")
envSetVal("layout" "pinMode" 'cyclic "Auto")
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

pinSelectionMode

```
layout pinSelectionMode cyclic { "ignoreLPP" | "useLPPIncludeShapes" |
     "useLPPExcludeShapes" }
```

Description

Determines the selectability of pins and shapes.

- `ignoreLPP`: Pins are selectable if the *Pin* check box in the *Objects* assistant is selected.
- `useLPPIncludeShapes`: Default. Instance pins are selectable if the *Pin* check box in the *Object* pane is selected and the LPP of the shape corresponding to the instance pin is set to selectable in the *Layer* pane.
- `useLPPExcludeShapes`: Instance pins are selectable only if the *Pin* check box in the *Object* pane is selected. The LPP of the shape corresponding to the instance pin is not considered.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pinSelectionMode")
envSetVal("layout" "pinSelectionMode" 'cyclic "ignoreLPP")
envSetVal("layout" "pinSelectionMode" 'cyclic "useLPPExcludeShapes")
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

pinShape

```
layout pinShape cyclic { "rectangle" | "polygon" | "auto pin" | "dot" | "circle" |
     "instance" | "via" | "existing object"}
```

Description

Specifies the default type of pin shape.

The default is `rectangle`.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Pin Shape*

Examples

```
envGetVal("layout" "pinShape")
envSetVal("layout" "pinShape" 'cyclic "polygon")
envSetVal("layout" "pinShape" 'cyclic "auto pin")
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

pinTextCreateAsLabel

```
layout pinTextCreateAsLabel boolean { t | nil }
```

Description

Specifies that a pin is created as a label or a text display.

The default is `nil`.

GUI Equivalent

Command:	*Create – Pin – Create Label – Options – Set Pin Label Text Style*
Field:	*Text Options: Create As Label*

Examples

```
envGetVal("layout" "pinTextCreateAsLabel")
envSetVal("layout" "pinTextCreateAsLabel" 'boolean t)
```

Related Topics

Set Pin Label Text Style Form

© 2023 Cadence Design Systems, Inc.

pinTextLayer

`layout pinTextLayer string "`*`any_valid_layer`*`"`

Description

Specifies the layer of the layer-purpose pair used for pin text displays and on the Pin Name Display form. These are the labels created for pin names.

The default is `text`.

GUI Equivalent

Command:	*Create – Pin – Option*
Field:	*Layer Name*

Examples

```
envGetVal("layout" "pinTextLayer")
envSetVal("layout" "pinTextLayer" 'string "via1")
```

Related Topics

Set Pin Label Text Style Form

© 2023 Cadence Design Systems, Inc.

pinTextLayerNames

```
layout pinTextLayerNames string { "Pimp" | "Poly" | "any_other_validLayer_name" }
```

Description

Specifies the layer names used to filter the *Layer Names* for the Set Pin Label Text Style form.

The default is `""`.

Note: If the layer names to be used for the filter are not specified, or none of the specified layer names match a valid layer name, all the valid layers are used.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pinTextLayerNames")
envSetVal("layout" "pinTextLayerNames" 'string "prBoundary")
envSetVal("layout" "pinTextLayerNames" 'string "Resdum")
```

Related Topics

Set Pin Label Text Style Form

Set Pin Label Text Style

© 2023 Cadence Design Systems, Inc.

pinTextPurpose

```
layout pinTextPurpose string "any_valid_purpose"
```

Description

Specifies the purpose of the layer-purpose pair used for pin text displays and on the Pin Name Display form. These are the labels created for pin names.

The default is `drawing`.

GUI Equivalent

Command:	*Create – Pin – Option*
Field:	*Layer Purpose*

Examples

```
envGetVal("layout" "pinTextPurpose")
envSetVal("layout" "pinTextPurpose" 'string "drawing")
```

Related Topics

Set Pin Label Text Style Form

© 2023 Cadence Design Systems, Inc.

pinTextPurposeNames

```
layout pinTextPurposeNames string { "drawing" | "any_other_validPurpose_name" }
```

Description

Specifies the layer purpose names used to filter the *Layer Purpose* for the Set Pin Label Text Style Form form.

The default is `""`.

If the layer purpose names to be used for the filter are not specified, or none of the specified layer purpose names match a valid layer purpose, all the valid layer purposes are used.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pinTextPurposeNames")
envSetVal("layout" "pinTextPurposeNames" 'string "drawing")
```

Related Topics

Set Pin Label Text Style Form

© 2023 Cadence Design Systems, Inc.

pinTextSamePurpose

```
layout pinTextSamePurpose boolean { t | nil }
```

Description

Specifies whether the pin name should have the same purpose as the pin.

The default is `nil`.

GUI Equivalent

Command:	*Create – Pin – Create Label – Options – Set Pin Label Text Style*
Field:	*Layer Purpose: Same As Pin*

Examples

```
envGetVal("layout" "pinTextSamePurpose")
envSetVal("layout" "pinTextSamePurpose" 'boolean t)
```

Related Topics

Set Pin Label Text Style Form

© 2023 Cadence Design Systems, Inc.

pinTextSameLayer

```
layout pinTextSameLayer boolean { t | nil }
```

Description

Determines whether to create the pin name label on the same layer/purpose as the pin being created or create it on the layer specified by `pinTextLayer` and the purpose specified by `pinTextPurpose`. The pin name is created on the layer specified by `pinTextLayer` and the purpose specified by `pinTextPurpose` when the default value `nil` is used.

When set to `t`, it creates the pin name label on the same layer/purpose as the pin being created.

GUI Equivalent

Command:	*Create – Pin – Create Label – Options – Set Pin Label Text Style*
Field:	*Layer Name: Same As Pin*

Examples

```
envGetVal("layout" "pinTextSameLayer")
envSetVal("layout" "pinTextSameLayer" 'boolean t)
```

Related Topics

Set Pin Label Text Style Form

© 2023 Cadence Design Systems, Inc.

pinTextOrientation

```
layout pinTextOrientation cyclic { "RO" | "R90" | "R180" | "R270" | "MY" | "MYR90"
     | "MX" | "MXR90" }
```

Description

Sets the orientation of the pin name on the Set Pin Label Text Style form.

The valid values are `R0`, `R90`, `R180`, `R270`, `MX`, `MY`, `MXR90`, and `MYR90`.

The default is `R0`.

GUI Equivalent

Command:	*Create – Pin – Create Label – Options – Set Pin Label Text Style*
Field:	*Orientation*

Examples

```
envGetVal("layout" "pinTextOrientation")
envSetVal("layout" "pinTextOrientation" 'cyclic "R180")
envSetVal("layout" "pinTextOrientation" 'cyclic "MYR90")
```

Related Topics

Set Pin Label Text Style Form

© 2023 Cadence Design Systems, Inc.

pinWidth

```
layout pinWidth float pinWidth
```

Description

Specifies the default pin width. The default is as defined in the technology file.

GUI Equivalent

Command:	*Create – Pin; Mode: Auto*
Field:	*Width*

Examples

```
envGetVal("layout" "pinWidth")
envSetVal("layout" "pinWidth" 'float 2.0)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

pinXPitch

`layout pinXPitch float` *`arrayDistance`*

Description

Specifies the horizontal distance between the center points of pins in an array.

The default is `0.0` user units.

GUI Equivalent

Command:	*Create – Pin*
Field:	*X Pitch*

Examples

```
envGetVal("layout" "pinXPitch")
envSetVal("layout" "pinXPitch" 'float 1.0)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

pinYPitch

```
layout pinYPitch float arrayDistance
```

Description

Specifies the vertical distance between the center points of pins in an array.

The default is `0.0` user units.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Y Pitch*

Examples

```
envGetVal("layout" "pinYPitch")
envSetVal("layout" "pinYPitch" 'float 1.0)
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

pointSelectAperture

```
graphic pointSelectAperture int numberOfPixels
```

Description

Sets the number of pixels for *aperture* for the *Point Selection Mode* field on the Selection Options form, accessed by choosing *Options – Selection*. The aperture defines a box around the point inferred by a mouse click to consolidate unintentional pointer movements between two clicks due to a jittery hand. Two clicks within the aperture are treated as a double click the same point.

The default is `5` pixels.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Point Selection Controls: User Aperture Mode*

Examples

```
envGetVal("graphic" "pointSelectAperture")
envSetVal("graphic" "pointSelectAperture" 'int 7)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

pointSelectCycleMode

```
graphic pointSelectCycleMode string { "toggle" | "cycle" }
```

Description

Sets the *Overlap Mode* cyclic field to *Toggle* or *Cycle* in the *Point Selection Controls* section on the Selection Options form, accessed by choosing *Options – Selection*.

The default is `toggle`.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Overlap Mode: Toggle* or *Overlap Mode: Cycle*

Examples

```
envGetVal("graphic" "pointSelectCycleMode")
envSetVal("graphic" "pointSelectCycleMode" 'string "cycle")
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

polygonArc

```
layout polygonArc boolean { t | nil }
```

Description

Indicates that the next three points you enter in the polygon define an arc. Resets to `nil` when you enter the third point.

The default is `nil`.

GUI Equivalent

Command:	*Create – Shape – Polygon*
Field:	*Create Arc*

Examples

```
envGetVal("layout" "polygonArc")
envSetVal("layout" "polygonArc" 'boolean t)
```

Related Topics

Create Polygon Form

© 2023 Cadence Design Systems, Inc.

preservePins

```
layout preservePins boolean { t | nil }
```

Description

Specifies whether pin information is preserved when an instance or mosaic is flattened.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Hierarchy – Flatten*
Field:	*Preserve: Pins*

Examples

```
envGetVal("layout" "preservePins")
envSetVal("layout" "preservePins" 'boolean t)
```

Related Topics

Flatten Form

© 2023 Cadence Design Systems, Inc.

preservePinFigs

```
layout preservePinFigs boolean { t | nil }
```

Description

Specifies whether the figure of the pin is preserved when the instance is flattened.

The figure moves up to the top level without any connectivity information when the default value `t` is used. When set to `nil`, no figure is preserved and there is no connectivity.

GUI Equivalent

Command:	*Edit – Hierarchy – Flatten*
Field:	*Preserve: Pin geometries*

Examples

```
envGetVal("layout" "preservePinFigs")
envSetVal("layout" "preservePinFigs" 'boolean nil)
```

Related Topics

Flatten Form

© 2023 Cadence Design Systems, Inc.

preserveRodObjects

```
layout preserveRodObjects boolean { t | nil }
```

Description

Specifies whether ROD properties are preserved when an instance is flattened.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Hierarchy – Flatten*
Field:	*Preserve: ROD objects*

Examples

```
envGetVal("layout" "preserveRodObjects")
envSetVal("layout" "preserveRodObjects" 'boolean t)
```

Related Topics

Flatten Form

© 2023 Cadence Design Systems, Inc.

preserveTerminalContacts

```
layout preserveTerminalContacts boolean { t | nil }
```

Description

Preserves the terminal contacts will be when you abut devices in Layout XL.

The default is `nil`.



Tip

To preserve properties on a single instance, create a boolean property called `preserveTerminalContacts` with a value of `t` on the instance in question. This lets you preserve properties (and therefore contacts) on selected instances while dropping them for all other instances.

GUI Equivalent

None

Examples

```
envGetVal("layout" "preserveTerminalContacts")
envSetVal("layout" "preserveTerminalContacts" 'boolean t)
```

Related Topics

Flatten Form

© 2023 Cadence Design Systems, Inc.

preserveTermName

```
layout preserveTermName boolean { t | nil }
```

Description

Specifies whether the terminal name of the pin is preserved when the instance is flattened.

When set to `t`, the terminal name of the pin is preserved. When the default value `nil` is used, the instance name is added as a prefix to the terminal name of the pin when the instance is flattened.

GUI Equivalent

Command:	*Edit – Hierarchy – Flatten*
Field:	*Preserve: Term name*

Examples

```
envGetVal("layout" "preserveTermName")
envSetVal("layout" "preserveTermName" 'boolean t)
```

Related Topics

Flatten Form

© 2023 Cadence Design Systems, Inc.

preventDeletionOfLogicalInstance

```
layout preventDeletionOfLogicalInstance boolean { t | nil }
```

Description

Prevents the deletion of non-physical-only instances.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Delete*
Field:	*Preserve Non-Physical-Only Instances*

Examples

```
envGetVal("layout" "preventDeletionOfLogicalInstance")
envSetVal("layout" "preventDeletionOfLogicalInstance" 'boolean t)
```

Related Topics

Delete Form

© 2023 Cadence Design Systems, Inc.

probeCycleBool

```
graphic probeCycleBool boolean { t | nil }
```

Description

Specifies whether to cycle the probe layer-purpose pairs. If cycling, the current layer-purpose pair for the probe is picked from `probeHiliteLppList`.

The default is `t`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Cycle* check box in the *Display Layer* section

Examples

```
envGetVal("graphic" "probeCycleBool")
envSetVal("graphic" "probeCycleBool" 'boolean nil)
```

Related Topics

probeHiliteLppList

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeFilledInstance

```
graphic probeFilledInstance boolean { t | nil }
```

Description

Probes the content of the master symbol if the inherited connection exists in the hierarchy.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "probeFilledInstance")
envSetVal("graphic" "probeFilledInstance" 'boolean t)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeFilledInstanceParam

`graphic probeFilledInstanceParam int` *`integer`*

Description

Sets the transparency level when probing. The value `0` signifies that probe is fully transparent and the value `1000` signifies that probe is not transparent.

The default is `500`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "probeFilledInstanceParam")
envSetVal("graphic" "probeFilledInstanceParam" 'int 100)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeHighlight

```
graphic probeHighlight boolean { t | nil }
```

Description

Determines whether the probes will be drawn with probe highlight.

The default is `t`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Highlight Color*

Examples

```
envGetVal("graphic" "probeHighlight")
envSetVal("graphic" "probeHighlight" 'boolean nil)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeHiliteLppList

```
graphic probeHiliteLppList string "lppList"
```

Description

Lets you specify the layer-purpose list to be used for probe.

The default value is `hilite drawing hilite drawing1 hilite drawing2 hilite drawing3 hilite drawing4 hilite drawing5 hilite drawing6 hilite drawing7 hilite drawing8 hilite drawing9`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	LPP drop-down list in the *Display Layer* section

Examples

```
envGetVal("graphic" "probeHiliteLppList")
envSetVal("graphic" "probeHiliteLppList" 'string "x y z")
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeHiliteLppString

```
graphic probeHiliteLppString string "lpp"
```

Description

Specifies the layer-purpose pair to be used by probe if probeCycleBool is set to `nil`.

If probeCycleBool is set to `t`, then `probeHiliteLppString` stores the current layer-purpose pair used by the probe.

GUI Equivalent

Command:	*Options – Highlight*
Field:	LPP drop-down list in the *Display Layer* section

Examples

```
envGetVal("graphic" "probeHiliteLppString")
envSetVal("graphic" "probeHiliteLppString" 'string "x y z")
```

Related Topics

probeCycleBool

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeInstDisplayRadio

```
graphic probeInstDisplayRadio string { "instancesOnly" | "instancesAndNet" }
layout probeInstDisplayRadio string { "instancesOnly" | "instancesAndNet" }
```

Description

Specifies whether the nets associated with probed or selected instances are also highlighted.

The default is `instancesOnly`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*From Selected Objects (Pins, Instances, Shapes)*

Examples

```
envGetVal("graphic" "probeInstDisplayRadio")
envSetVal("graphic" "probeInstDisplayRadio" 'string "instancesAndNet")
envGetVal("layout" "probeInstDisplayRadio")
envSetVal("layout" "probeInstDisplayRadio" 'string "instancesAndNet")
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeNetDisplayCheckInstance

```
graphic probeNetDisplayCheckInstance boolean { t | nil }
```

Description

Specifies that instances on probed nets are highlighted.

The default is `nil`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Instances*

Examples

```
envGetVal("graphic" "probeNetDisplayCheckInstance")
envSetVal("graphic" "probeNetDisplayCheckInstance" 'boolean t)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeNetDisplayCheckPins

```
graphic probeNetDisplayCheckPins boolean { t | nil }
```

Description

Specifies that pins on probed nets are also highlighted.

The default is `t`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Pins*

Examples

```
envGetVal("graphic" "probeNetDisplayCheckPins")
envSetVal("graphic" "probeNetDisplayCheckPins" 'boolean nil)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeNetDisplayFlightLines

```
graphic probeNetDisplayFlightLines string { "physical" | "logical" }
```

Description

Specifies whether the probe flight lines show physical or logical connectivity.

- `physical`: Default. Specifies that all shapes and opens on the probed net are highlighted
- `logical`: Specifies that segments linking all pins on the probed net are highlighted.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Flightlines*

Examples

```
envGetVal("graphic" "probeNetDisplayFlightLines")
envSetVal("graphic" "probeNetDisplayFlightLines" 'string "logical")
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeNetDisplayHierarchyLevel

```
layout probeNetDisplayHierarchyLevel int numberOfHierarchyLevels
graphic probeNetDisplayHierarchyLevel int numberOfHierarchyLevels
```

Description

Sets the number of hierarchy levels to display when probing.

The default is `0`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Hierarchy Depth*

Examples

```
envGetVal("layout" "probeNetDisplayHierarchyLevel")
envSetVal("layout" "probeNetDisplayHierarchyLevel" 'int 2)
envGetVal("graphic" "probeNetDisplayHierarchyLevel")
envSetVal("graphic" "probeNetDisplayHierarchyLevel" 'int 2)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeNetNameFilterByName

```
graphic probeNetNameFilterByName boolean { t | nil }
```

Description

Lets you specify the selective display of probes depending on their net names.

The default is `nil`, which implies all probes are displayed.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Filter net names*

Examples

```
envGetVal("graphic" "probeNetNameFilterByName")
envSetVal("graphic" "probeNetNameFilterByName" 'boolean t)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeNetNameFilterByNameDefaultList

```
graphic probeNetNameFilterByNameDefaultList string "net1 net2 net3"
```

Description

Adds a default list of net names to be used for probe filtering by name, if no information is currently stored in the design.

The default is `" "`, which implies no filter list is available.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Net Names*

Examples

```
envGetVal("graphic" "probeNetNameFilterByNameDefaultList")
envSetVal("graphic" "probeNetNameFilterByNameDefaultList" 'string "netA netB")
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeNetNameFilterByNameMode

```
graphic probeNetNameFilterByNameMode cyclic { "Hide" | "Show" }
```

Description

Controls whether the probes that have the specified nets are displayed or hidden.

The default is `"Hide"`, this implies if the probe net is in the filtering list, it will be hidden.

GUI Equivalent

Command:	*Options – Highlight*
Form Field:	*Filter Mode*

Examples

```
envGetVal("graphic" "probeNetNameFilterByNameMode")
envSetVal("graphic" "probeNetNameFilterByNameMode" 'cyclic "Show"))
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeNetStopAtDisplayLevel

```
graphic probeNetStopAtDisplayLevel boolean { t | nil }
layout probeNetStopAtDisplayLevel boolean { t | nil }
```

Description

Specifies whether to stop the drawing of probed net at the display level.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "probeNetStopAtDisplayLevel")
envSetVal("graphic" "probeNetStopAtDisplayLevel" 'boolean nil
envGetVal("layout" "probeNetStopAtDisplayLevel")
envSetVal("layout" "probeNetStopAtDisplayLevel" 'boolean nil)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probePromptBusBit

```
graphic probePromptBusBit boolean { t | nil }
```

Description

Specifies whether to display the prompt that lets you select the specific bit of a bus when the `geAddNetProbe()` function is called.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "probePromptBusBit")
envSetVal("graphic" "probePromptBusBit" 'boolean t)
```

Related Topics

geAddNetProbe

© 2023 Cadence Design Systems, Inc.

probeShapeDisplay

```
graphic probeShapeDisplay string { "shapesOnly" | "shapesAndNet" }
```

Description

Specifies whether the nets associated with probed shapes are also highlighted.

The default is `"shapesOnly"`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*From Selected Objects (Pins, Instances, Shapes)*

Examples

```
envGetVal("graphic" "probeShapeDisplay")
envSetVal("graphic" "probeShapeDisplay" 'string "shapesAndNet")
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeTermDisplayRadio

```
graphic probeTermDisplayRadio string { "pinOnly" | "pinsAndNet" }
```

Description

Specifies whether the nets associated with probed pins are also highlighted.

The default is `"pinOnly"`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*From Selected Objects (Pins, Instances, Shapes)*

Examples

```
envGetVal("graphic" "probeTermDisplayRadio")
envSetVal("graphic" "probeTermDisplayRadio" 'string "pinsAndNet")
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

probeVisibleLpp

```
graphic probeVisibleLpp boolean { t | nil }
```

Description

Specifies that only shapes and pins on the visible layer-purpose pairs are highlighted.

The default is `nil`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Visible Shapes Only*

Examples

```
envGetVal("graphic" "probeVisibleLpp")
envSetVal("graphic" "probeVisibleLpp" 'boolean t)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

propEditHideTopSection

```
layout propEditHideTopSection boolean { t | nil }
```

Description

Filters the top section of the Edit Properties form.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "propEditHideTopSection")
envSetVal("layout" "propEditHideTopSection" 'boolean t)
```

Related Topics

Edit Properties Form

© 2023 Cadence Design Systems, Inc.

propEditDefaultSelectAll

```
layout propEditDefaultSelectAll cyclic { "Never" | "Always" | "Invocation" |
     "Retain" }
```

Description

Sets the default selection behavior of the selection tree when you invoke the Edit Properties form.

- `Never`: Shows only a single object as selected in the selection tree when you invoke the form.
- `Always`: Shows all objects as selected in the selection tree when you invoke the form. Selection changes on the canvas do not affect the selection behavior in the tree; all objects are always shown as selected in the tree unless you change the selection in the tree. For example, when you invoke the form with three objects selected on the canvas, all three objects appear as selected in the tree. Now when you select an additional object on the canvas, the tree lists four objects and all objects are marked as selected.
- `Invocation`: Shows all objects as selected in the selection tree when you invoke the form. Upon selection changes on the canvas, the tree shows only a single object as selected. For example, when you invoke the form with three objects selected on the canvas, all three objects appear as selected in the tree. When you select one more object on the canvas, the tree lists four objects, with only one object marked as selected.
- `Retain`: Maintains the last selection behavior of the selection tree when you invoke the form.

The default is `Never`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "propEditDefaultSelectAll")
envSetVal("layout" "propEditDefaultSelectAll" 'cyclic "Always")
```

Related Topics

Edit Properties Form

© 2023 Cadence Design Systems, Inc.

propEditorHeight

```
layout propEditorHeight int height
```

Description

Specifies the height of the Property Editor form. Enter any integer greater than zero.

The default is `320`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "propEditorHeight")
envSetVal("layout" "propEditorHeight" 'int 100)
```

Related Topics

Property Editor Assistant

© 2023 Cadence Design Systems, Inc.

propEditorWidth

```
layout propEditorWidth int width
```

Description

Specifies the width of the Property Editor form. Enter any integer greater than zero.

The default is `575`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "propEditorWidth")
envSetVal("layout" "propEditorWidth" 'int 200)
```

Related Topics

Property Editor Assistant

© 2023 Cadence Design Systems, Inc.

propertyEditorAutoResize

```
layout propertyEditorAutoResize boolean { t | nil }
```

Description

Resizes the *Property Editor* automatically.

The default is `t`. When set to `nil`, the *Property Editor* can be resized. It remains fixed for all object types in the current session.

GUI Equivalent

None

Examples

```
envGetVal("layout" "propertyEditorAutoResize")
envSetVal("layout" "propertyEditorAutoResize" 'boolean t)
```

Related Topics

Property Editor Assistant

© 2023 Cadence Design Systems, Inc.

propEditStretchableTree

```
layout propEditStretchableTree boolean { t | nil }
```

Description

Allows the selection tree in the Property Editor to be stretchable.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "propEditStretchableTree")
envSetVal("layout" "propEditStretchableTree" 'boolean t)
```

Related Topics

Property Editor Assistant

© 2023 Cadence Design Systems, Inc.

protectedObjHighlightHaloLevel

```
graphic protectedObjHighlightHaloLevel string { "over" | "under" }
```

Description

Determines the placement of the halo relative to the object boundary box.

The default is `under`.

GUI Equivalent

Command:	*Edit – Select – Selection Protection Options*
Field:	*Level*

Examples

```
envGetVal("graphic" "protectedObjHighlightHaloLevel")
envSetVal("graphic" "protectedObjHighlightHaloLevel" 'string "over")
```

Related Topics

Selection Protection Highlight Options Form

© 2023 Cadence Design Systems, Inc.

protectedObjHighlightHaloMode

```
graphic protectedObjHighlightHaloMode string { "normal" | "fadeout" | "glow" |
     "plain" | "random" }
```

Description

Determines the placement of the halo relative to the object boundary box.

The default is `fadeout`.

GUI Equivalent

Command:	*Edit – Select – Selection Protection Options*
Field:	*Halo Mode*

Examples

```
envGetVal("graphic" "protectedObjHighlightHaloMode")
envSetVal("graphic" "protectedObjHighlightHaloMode" 'string "plain")
envSetVal("graphic" "protectedObjHighlightHaloMode" 'string "random")
```

Related Topics

Selection Protection Highlight Options Form

© 2023 Cadence Design Systems, Inc.

protectedObjHighlightHaloTrans

```
graphic protectedObjHighlightHaloTrans int visibility
```

Description

Controls the visibility of the halo. The visibility of the halo is the maximum at the lowest transparency setting.

The default is `40`.

GUI Equivalent

Command:	*Edit – Select – Selection Protection Options*
Field:	*Transparency*

Examples

```
envGetVal("graphic" "protectedObjHighlightHaloTrans")
envSetVal("graphic" "protectedObjHighlightHaloTrans" 'int 60)
```

Related Topics

Selection Protection Highlight Options Form

© 2023 Cadence Design Systems, Inc.

protectedObjHighlightHaloWidth

```
graphic protectedObjHighlightHaloWidth string { "thick" | "normal" | "thin" }
```

Description

Specifies the thickness of the halo on a protected object.

The default is `normal`.

GUI Equivalent

Command:	*Edit – Select – Selection Protection Options*
Field:	*Width*

Examples

```
envGetVal("graphic" "protectedObjHighlightHaloWidth")
envSetVal("graphic" "protectedObjHighlightHaloWidth" 'string "thick")
envSetVal("graphic" "protectedObjHighlightHaloWidth" 'string "thin")
```

Related Topics

Selection Protection Highlight Options Form

© 2023 Cadence Design Systems, Inc.

pteAllLayersScope

```
layout pteAllLayersScope cyclic { "lswLayers" | "techfileLayers" | "nil" }
```

Description

Defines the scope of layer-purpose pairs to be displayed in the *Layers* panel.

- `lswLayers`: Default. Only the layer-purpose pairs defined in the LSW information file are displayed in the *Layers* panel.
- `techfileLayers`: Only the valid layer-purpose pairs defined in the technology file are displayed in the *Layers* panel.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Scope of 'All Layers' Layer Set*

Examples

```
envGetVal("layout" "pteAllLayersScope")
envSetVal("layout" "pteAllLayersScope" 'cyclic "techfileLayers")
envSetVal("layout" "pteAllLayersScope" 'cyclic "nil")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteApplyParentState

```
layout pteApplyParentState boolean { t | nil }
```

Description

Toggles between the synchronized and desynchronized states of parent and child objects in the *Objects* and *Grids* panels.

The default is `t`, which is the synchronized mode. In this mode, when the visibility or selectability of a parent object is changed, the visibility and selectability of child objects is reset to coincide with that of the parent object.

To switch to the desynchronized mode, set the environment variable to `nil`.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Visibility and selectability reset by parent*
	Sync visibility/selectability by parent

Examples

```
envGetVal("layout" "pteApplyParentState")
envSetVal("layout" "pteApplyParentState" 'boolean nil)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteAutoRedrawMode

```
layout pteAutoRedrawMode cyclic { "currentWindow" | "noWindow" | "allWindows" }
```

Description

Determines the scope of the auto redraw feature.

- `currentWindow`: Default. The current window canvas is updated in real time whenever a change is made to a Palette panel.
- `noWindow`: The auto redraw feature is turned off. As a result, manual redraw is needed to update all windows.
- `allWindows`: All synchronized window canvases are updated in real time, whenever a change is made to a Palette panel.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Auto Redraw*

Examples

```
envGetVal("layout" "pteAutoRedrawMode")
envSetVal("layout" "pteAutoRedrawMode" 'cyclic "noWindow")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteCheckableColumnClickMode

```
layout pteCheckableColumnClickMode cyclic { "turnOnOff" | "sort" }
```

Description

Determines the behavior when the Validity (*V*), Selectability (*S*), or Membership (*m*) columns in the *Layers* panel is clicked.

- `turnOnOff`: Default.The statuses toggle between the on and off modes.
- `sort`: The layers are sorted based on their validity, selectability, or membership statuses.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Checkable Column Click*
	Header Column Click

Examples

```
envGetVal("layout" "pteCheckableColumnClickMode")
envSetVal("layout" "pteCheckableColumnClickMode" 'cyclic "sort")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteCollapseLayersInViewBy

```
layout pteCollapseLayersInViewBy boolean { t | nil }
```

Description

Collapses the *Layers* panel when you select the *View By* options.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteCollapseLayersInViewBy")
envSetVal("layout" "pteCollapseLayersInViewBy" 'boolean t)
```

Related Topics

Viewing Layer-Purpose Pairs in a Tree Structure

© 2023 Cadence Design Systems, Inc.

pteColoredLppSortOrder

```
layout pteColoredLppSortOrder string orderOfLPP
```

Description

Specifies the order in which colored layer-purpose pairs are listed within each master layer-purpose pair grouping.

The default value is `""`. You can use a space, a comma (`,`), a semicolon (`;`), a hyphen (`-`), a forward slash (`/`), a period (`.`), or an underscore (_) to separate the values in *`string_value`*.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteColoredLppSortOrder")
envSetVal("layout" "pteColoredLppSortOrder" 'string "metal1 drawing")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteColorMixedSort

```
layout pteColorMixedSort boolean { t | nil }
```

Description

Lets you enable or disable the *Color mixed sort* feature.

The default is `nil`, which causes layer-purpose pairs to be sorted into two distinct groups, colored and uncolored.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Color mixed sort*

Examples

```
envGetVal("layout" "pteColorMixedSort")
envSetVal("layout" "pteColorMixedSort" 'boolean t)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteCompactMPT

```
layout pteCompactMPT boolean { t | nil }
```

Description

Controls the setting of the *Compact MPT mode* check box available in the Palette Options form.

The default is `nil`. When set to `t`, *MPT Support* opens in compact mode in a new layout window.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Compact MPT mode*

Examples

```
envGetVal("layout" "pteCompactMPT")
envSetVal("layout" "pteCompactMPT" 'boolean t)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteDeferRedrawOnMouseMove

```
layout pteDeferRedrawOnMouseMove boolean { t | nil }
```

Description

Stops canvas redraw when there is MMB and mouse movement on the *Layers* panel. Only one redraw is done when the mouse is released

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteDeferRedrawOnMouseMove")
envSetVal("layout" "pteDeferRedrawOnMouseMove" 'boolean t)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteDisplayActiveLppInForeground

```
layout pteDisplayActiveLppInForeground boolean { t | nil }
```

Description

Allows you to select or deselect the *Display Active LPP in Foreground* check box.

The default is `nil`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Display Active LPP in Foreground*

Examples

```
envGetVal("layout" "pteDisplayActiveLppInForeground")
envSetVal("layout" "pteDisplayActiveLppInForeground" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

pteEnableMouseBindings

```
layout pteEnableMouseBindings boolean { t | nil }
```

Description

Turns on or off the selectability of a layer-purpose pair when a right-click is performed in the *Layers* panel.

The default is `t`. When set to `nil`, it causes a right-click in the *Layers* panel to display the context-sensitive Palette menu.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteEnableMouseBindings")
envSetVal("layout" "pteEnableMouseBindings" 'boolean nil)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteFindModeOn

```
layout pteFindModeOn boolean { t | nil }
```

Description

Toggles between the *Filter* and *Find* search modes.The *Filter* mode is switched on.

The default is `nil`. When set to `t`, the *Find* mode is switched on.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Search Mode*

Examples

```
envGetVal("layout" "pteFindModeOn")
envSetVal("layout" "pteFindModeOn" 'boolean t)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteIncludeCutLayers

```
layout pteIncludeCutLayers boolean { t | nil }
```

Description

Controls the inclusion and exclusion of cut layers as routing layers. Cut layers are included as routing layers.

The default is `t`. When set to `nil`, cut layers are excluded from routing layers.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteIncludeCutLayers")
envSetVal("layout" "pteIncludeCutLayers" 'boolean nil)
```

Related Topics

Layers Panel

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteInfraDefValueObjGrids

```
layout pteInfraDefValueObjGrids string "any_string"
```

Description

Sets the visibility and selectability of the specified objects and grids. Consider the following while setting the environment variable:

- Use an underscore to specify objects with multi-word names. For example, `Snap Patterns` must be listed as `Snap_Patterns`.
- Provide a semicolon-separated list to specify multiple objects and grids.
- Specify the first value to set the visibility of an object or grid; specify the second value to set the selectability.

  For example, `Pins t nil` means that the visibility of pins is enabled but selectability is disabled. `Shapes nil` means that the visibility of shapes is disabled, which implies that the selectability of shapes is automatically disabled.

This environment variable does not take effect when typed in the CIW in your current Virtuoso session. Exit the session and then start a new session after setting the environment variable in the `.cdsinit` or `.cdsenv` file.

The default value is `""`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteInfraDefValueObjGrids")
envSetVal("layout" "pteInfraDefValueObjGrids" 'string "Snap_Patterns nil")
envSetVal("layout" "pteInfraDefValueObjGrids" 'string "Pins t nil;Label t
nil;Shapes nil;Instances t nil;P&R_Boundary t nil;Area_Boundary t t;Snap_Patterns
nil")
```

Related Topics

Palette Panels

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteInfraObjSelFollowVis

```
layout pteInfraObjSelFollowVis string "listOfObjects"
```

Description

Specifies the list of objects for which selectability follows visibility.

The default value is `""`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteInfraObjSelFollowVis")
envSetVal("layout" "pteInfraObjSelFollowVis" 'string "Snap_Patterns")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteInfraWirebondProfileColors

```
layout pteInfraWirebondProfileColors string wirebondProfileColors
```

Description

(Virtuoso RF Option) Specifies the list of colors in which wirebond profiles are displayed on the canvas.

The default value is `""`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteInfraWirebondProfileColors")
envSetVal("layout" "pteInfraWirebondProfileColors" 'string "blue magenta brown")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteInverseColorScheme

```
layout pteInverseColorScheme boolean { t | nil }
```

Description

Lets you inverse the color scheme of the Palette panels.

The default is `nil`.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Inverse color scheme*

Examples

```
envGetVal("layout" "pteInverseColorScheme")
envSetVal("layout" "pteInverseColorScheme" 'boolean t)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteLayerSetActivationMode

```
layout pteLayerSetActivationMode cyclic { "filterOnly" | "filterAndApply" |
     "filterVisSelectability" }
```

Description

Determines the default action when one or more layer sets are selected.

- `filterOnly`: Default. Displays all layer-purpose pairs present in the selected layer sets. The visibility and selectability status of the layer-purpose pairs remains unchanged.
- `filterAndApply`: Displays all layer-purpose pairs present in the selected layer sets. The visibility and selectability status of layer-purpose pairs is reset to that defined in the layer set files.
- `filterVisSelectability`: Displays all layer-purpose pairs present in the selected layer sets. The visibility and selectability of all layer-purpose pairs is turned on.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Layer Set Activation*

Examples

```
envGetVal("layout" "pteLayerSetActivationMode")
envSetVal("layout" "pteLayerSetActivationMode" 'cyclic "filterAndApply")
envSetVal("layout" "pteLayerSetActivationMode" 'cyclic "filterVisSelectability")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteLayerSetManagerLeftClickMode

```
layout pteLayerSetManagerLeftClickMode cyclic { "activateLayerSet" |
     "forceVisibilityAndSelectabilityOnly" }
```

Description

Determines the default left mouse button click behavior for layer sets.

- `activateLayerSet`: Default. The selected layer is activated. The visibility and selectability statuses for the selected layer are automatically turned on.
- `forceVisibilityAndSelectabilityOnly`: The selected layer set is activated and the visibility and selectability statuses of the constituent layers is forcibly turned on. The visibility and selectability statuses of all other layers are forcibly turned off.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Layer Set Manager Left Click*

Examples

```
envGetVal("layout" "pteLayerSetManagerLeftClickMode")
envSetVal("layout" "pteLayerSetManagerLeftClickMode" 'cyclic
"forceVisibilityAndSelectabilityOnly")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteLayersSelectabilityDepth

`layout pteLayersSelectabilityDepth int` *`selectabilityDepth`*

Description

Sets the selectability depth for layers. The depth value determines the number of layer-purpose pairs for which the selectability attribute is affected when you hold down `Ctrl` or both `Ctrl` and `Shift` and right-click a layer-purpose pair in the *Layers* panel.

The default is `1`.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Selectability Depth For Layers*
	Depth for Selectability

Examples

```
envGetVal("layout" "pteLayersSelectabilityDepth")
envSetVal("layout" "pteLayersSelectabilityDepth" 'int 3)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteLayersVisibilityDepth

`layout pteLayersVisibilityDepth int` *`visibilityDepth`*

Description

Sets the visibility depth for layers. The depth value determines the number of layer-purpose pairs for which the visibility attribute is affected when you hold down `Ctrl` or both `Ctrl` and `Shift` and click in the *Layers* panel a layer-purpose pair with the middle mouse button.

The default is `1`.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Visibility Depth For Layers*
	Depth for Visibility

Examples

```
envGetVal("layout" "pteLayersVisibilityDepth")
envSetVal("layout" "pteLayersVisibilityDepth" 'int 3)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteLinkColorVisibilityToShapeVisibility

```
layout pteLinkColorVisibilityToShapeVisibility boolean { t | nil }
```

Description

Controls the visibility of colored shapes. Colored shapes become invisible if you deselect the Visibility (*V*) check box corresponding to the layer-purpose pairs on which they are drawn.

The default is `t`. When set to `nil`, only the colored outline is hidden; the shapes remain visible.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Shape visibility by color visibility*

Examples

```
envGetVal("layout" "pteLinkColorVisibilityToShapeVisibility")
envSetVal("layout" "pteLinkColorVisibilityToShapeVisibility" 'boolean nil)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteLppGroup

```
layout pteLppGroup string purposeNames
```

Description

Lets you group layers by the specified purposes and view the *LPP Group* option in the *View By* menu.

GUI Equivalent

Command:	*Palette Menu (Layers)*
Field:	*View By – LPP Group*

Examples

```
envGetVal("layout" "pteLppGroup")
envSetVal("layout" "pteLppGroup" 'string "drawing label")
```

Related Topics

Layers Panel

© 2023 Cadence Design Systems, Inc.

pteLppsDisplayOrderControl

```
layout pteLppsDisplayOrderControl cyclic { "palette" | "techfile" }
```

Description

Allows you to select or deselect the *Palette LPPs Display Order* check box.

- `techfile`: Default. The check box is deselected.
- `palette`: The check box is selected. You can change the display order of layer-purpose pairs in the canvas by dragging up and down the layer-purpose pairs listed in the Palette.

Note: Palette must be set to *Multi-select* mode.

GUI Equivalent

Command:	*Options - Layout Editor Options*
	Options – Editor
Field:	*Palette LPPs Display Order*

Examples

```
envGetVal("layout" "pteLppsDisplayOrderControl")
envSetVal("layout" "pteLppsDisplayOrderControl" 'cyclic "palette")
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

pteLSManagerAsAssistant

```
layout pteLSManagerAsAssistant boolean { t | nil }
```

Description

Allows you to enable the layer set manager as an assistant. The default is `nil`.

The shell environment variable `CDS_PALETTE_TYPE` should be set to `MultiAssistant` for this environment variable to work, as shown below:

```
setenv CDS_PALETTE_TYPE MultiAssistant
```

The environment variable should be added to `.cdsinit` or `.cdsenv` files.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteLSManagerAsAssistant")
envSetVal("layout" "pteLSManagerAsAssistant" 'boolean t)
```

Related Topics

Setting the Palette Container

© 2023 Cadence Design Systems, Inc.

pteLSManagerRespectFilters

```
layout pteLSManagerRespectFilters boolean { t | nil }
```

Description

Turns on and off the visibility or selectability status using the *+* and – buttons in the *Layer Set Manager*, which are used for the layers present in a layer set, work with respect to the filter settings defined on the *Scope* and *Filter* toolbar.

The *Scope* toolbar provides three options for filtering layers, namely, *Valid*, *Used*, and *Routing*. The *Filter* toolbar enables you to define search criteria for filtering layers.

For example, if you select the *Valid* check box on the *Scope* toolbar, and click the *+* button in the *Layer Set Manager* to turn on the visibility of a layer set, then visibility is turned on for only the valid layers of the layer set. It does not affect the visibility setting of the invalid layers.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteLSManagerRespectFilters")
envSetVal("layout" "pteLSManagerRespectFilters" 'boolean t)
```

Related Topics

Layers Panel

© 2023 Cadence Design Systems, Inc.

pteMPTAwareSelectionTrigger

```
layout pteMPTAwareSelectionTrigger boolean { t | nil }
```

Description

Sends the color and lock information of layer-purpose pairs to user-registered functions when full MPT support is enabled in the Palette assistant.

The default value is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteMPTAwareSelectionTrigger")
envSetVal("layout" "pteMPTAwareSelectionTrigger" 'boolean t)
```

Related Topics

MPT Support

Layers Panel

© 2023 Cadence Design Systems, Inc.

pteMPTMode

```
layout pteMPTMode boolean { t | nil }
```

Description

Determines if the Palette MPT feature is turned on or off at the time of opening a new layout window.The Palette MPT feature is disabled by default.

The default is `nil`. When set to `t`, the *MPT* check box is displayed on the *Scope* toolbar and the Color (*C*) column is displayed in the *Layers* panel when you open a new layout window.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteMPTMode")
envSetVal("layout" "pteMPTMode" 'boolean t)
```

Related Topics

MPT Support

Layers Panel

© 2023 Cadence Design Systems, Inc.

pteMPTScope

```
layout pteMPTScope boolean { t | nil }
```

Description

Controls the setting of the *MPT* check box at the time of opening a new layout window and is evaluated only if set to `t`.

The default is `nil`, which means that the MPT check box is deselected and all layer-purpose pairs are listed in the in the *Layers* panel.

When set to `t`, the *MPT* check box is selected and only the layer-purpose pairs with more than one color are listed in the *Layers* panel.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteMPTScope")
envSetVal("layout" "pteMPTScope" 'boolean t)
```

Related Topics

pteMPTMode

© 2023 Cadence Design Systems, Inc.

pteMovableTools

```
layout pteMovableTools boolean { t | nil }
```

Description

Checks if the toolbars can be moved in the workspace.

The default is `nil`.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Movable tools*

Examples

```
envGetVal("layout" "pteMovableTools")
envSetVal("layout" "pteMovableTools" 'boolean t)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteMultiSelectionMode

```
layout pteMultiSelectionMode boolean { t | nil }
```

Description

Toggles between the single-selection and multi-selection mode.

The default is `nil`, which means that single-section mode is turned on. When set to `t`, multi-selection mode is turned on.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Multi-select mode*

Examples

```
envGetVal("layout" "pteMultiSelectionMode")
envSetVal("layout" "pteMultiSelectionMode" 'boolean t)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteProhibitToggleOnMouseMove

```
layout pteProhibitToggleOnMouseMove boolean { t | nil }
```

Description

Sets the visibility and selectability of LPPs same as to match that of the starting LPP when you move the middle mouse button on the Layers panel in the Palette assistant.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteProhibitToggleOnMouseMove")
envSetVal("layout" "pteProhibitToggleOnMouseMove" 'boolean t)
```

Related Topics

Layers Panel

© 2023 Cadence Design Systems, Inc.

pteRetainFilterSettings

```
layout pteRetainFilterSettings boolean { t | nil }
```

Description

Synchronizes the setting of the *Scope* filters, namely *Valid*, *Used*, *Routing*, and *MPT*, across synchronized layout windows. The default is `nil`.

For example, if you apply the *Routing* filter by selecting the *Routing* check box in one layout window, it is automatically applied to all open layout windows, and also to those that are opened subsequently, as long as the windows stay synchronized.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteRetainFilterSettings")
envSetVal("layout" "pteRetainFilterSettings" 'boolean t)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteRoutingOnlyScope

```
layout pteRoutingOnlyScope boolean { t | nil }
```

Description

Displays only routing layers are in the *Layers* panel.

The default is `nil`.

GUI Equivalent

Command:	*Layers Panel*
Field:	*Routing*

Examples

```
envGetVal("layout" "pteRoutingOnlyScope")
envSetVal("layout" "pteRoutingOnlyScope" 'boolean t)
```

Related Topics

Layers Panel

© 2023 Cadence Design Systems, Inc.

pteShowColorControlsForMPT

```
layout pteShowColorControlsForMPT boolean { t | nil }
```

Description

Displays the two check boxes that control the visibility of colored shapes in compact mode.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteShowColorControlsForMPT")
envSetVal("layout" "pteShowColorControlsForMPT" 'boolean t)
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteShowColoredLSManager

```
layout pteShowColoredLSManager boolean { t | nil }
```

Description

Displays colored buttons for layer sets to help you quickly identify layer sets in the *Layer Set Manager* in the *Layers* panel of the Palette assistant.

By default, the active layer set is highlighted in black. For any inactive layer set, a colored button is shown if a layer-purpose pair is defined for that layer set by using the `pteSetLayerSetColoredLpp` SKILL function, in which case the color of the button is picked from the specified layer-purpose pair.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteShowColoredLSManager")
envSetVal("layout" "pteShowColoredLSManager" 'boolean t)
```

Related Topics

Layer Set Manager

pteSetLayerSetColoredLpp

© 2023 Cadence Design Systems, Inc.

pteShowLSManager

```
layout pteShowLSManager boolean { t | nil }
```

Description

Displays the *Layer Set Manager* toolbar in the *Layers* panel, overriding workspace settings.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteShowLSManager")
envSetVal("layout" "pteShowLSManager" 'boolean t)
```

Related Topics

Layer Set Manager

Palette Panels

© 2023 Cadence Design Systems, Inc.

pteShowMPTScope

```
layout pteShowMPTScope boolean { t | nil }
```

Description

Displays the *MPT* check box on the *Scope* toolbar when you enable *MPT Support*.

The default is `t`. When set to `nil`, it hides the *MPT* check box

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteShowMPTScope")
envSetVal("layout" "pteShowMPTScope" 'boolean nil)
```

Related Topics

MPT Support

Layers Panel

© 2023 Cadence Design Systems, Inc.

pteShowUsedSystemLpps

```
layout pteShowUsedSystemLpps boolean { t | nil }
```

Description

Displays the layers containing all types of design objects when the *Used* filter is enabled in the Layers panel.

The default is `nil`, which implies that, by default, the *Layers* panel displays only the layers containing design objects of type shape. When set to `t`, the *Used* filter is not dynamic and needs to be reapplied, for example while switching to another tab or window, editing in place, or descending into a hierarchy.

GUI Equivalent

None

Examples

```
envGetVal("layout" "pteShowUsedSystemLpps")
envSetVal("layout" "pteShowUsedSystemLpps" 'boolean t)
```

Related Topics

Layers Panel

© 2023 Cadence Design Systems, Inc.

pteUsedLayersEipScope

```
layout pteUsedLayersEipScope cyclic { "currentToBottom" | "topToBottom" }
```

Description

Lets you specify the layers to display in the Layers panel when *Edit In Place* mode is active. If different technology files are associated with cellviews at different hierarchical levels in the design, only the layers defined in the technology file associated with the cellview being edited in place are displayed in the Layers panel.

- `currentToBottom`: Default. Displays layers used in the design at the current hierarchy level and below. This is the default option.
- `topToBottom`: Displays all layers used in the design.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	*Scope of Used Layers in EIP*

Examples

```
envGetVal("layout" "pteUsedLayersEipScope")
envSetVal("layout" "pteUsedLayersEipScope" 'cyclic "topToBottom")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

pteWindowSynchronizationMode

```
layout pteWindowSynchronizationMode cyclic { "independent" | "synchronized" }
```

Description

Sets the synchronization mode for new windows to either independent or synchronized.

The default is `synchronized`.

GUI Equivalent

Command:	*Palette Menu – Options*
Field:	Window Synchronization
	Keep windows synchronized

Examples

```
envGetVal("layout" "pteWindowSynchronizationMode")
envSetVal("layout" "pteWindowSynchronizationMode" 'cyclic "independent")
```

Related Topics

Palette Options Form

© 2023 Cadence Design Systems, Inc.

QAHoldTimeMilliSec

```
layout QAHoldTimeMilliSec int time
```

Description

Lets you specify the time (in milliseconds) required to hold the pointer over an object. While using the *Quick Align* command, if the pointer remains stationary over an object for longer than the specified time, the potential points, edges, and centerlines in the object are highlighted.

The default is `0`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "QAHoldTimeMilliSec")
envSetVal("layout" "QAHoldTimeMilliSec" 'int 100)
```

Related Topics

Object Alignment

© 2023 Cadence Design Systems, Inc.

QAMaxPixelsPerMilliSec

```
layout QAMaxPixelsPerMilliSec int time
```

Description

Defines the maximum velocity with which the pointer should move between potential targets for the *Quick Align* command to be able to snap to them. If the pointer moves with the velocity higher than the value defined in the `QAMaxPixelsPerMilliSec` environment variable, the *Quick Align* command does not snap to those targets.

GUI Equivalent

None

Examples

```
envGetVal("layout" "QAMaxPixelsPerMilliSec")
envSetVal("layout" "QAMaxPixelsPerMilliSec" 'int 10)
```

Related Topics

Object Alignment

© 2023 Cadence Design Systems, Inc.

quickAlignMode

```
layout quickAlignMode cyclic { "MoveOrStretch" | "Copy" }
```

Description

Specifies the mode to be used for alignment.

- `MoveOrStretch`: Default. The object is moved or stretched during alignment.
- `Copy`: The object is copied during alignment.

GUI Equivalent

Command:	*Edit – Quick Align*
Field:	*Move/Stretch, Copy*
	Copy reference

Examples

```
envGetVal("layout" "quickAlignMode")
envSetVal("layout" "quickAlignMode" 'cyclic "Copy")
```

Related Topics

Quick Align Form

© 2023 Cadence Design Systems, Inc.

quickAlignSameLayerTarget

```
layout quickAlignSameLayerTarget boolean { t | nil }
```

Description

Snaps the pointer to only those targets that are on the same layer as the object with the reference point, edge, or centerline.

The default is `nil`, which means that the pointer snaps to edges, centerlines, and points on a target object on any layer if they are within the aperture distance from the pointer.

GUI Equivalent

Command:	*Edit – Quick Align*
Field:	*On Same Layer*

Examples

```
envGetVal("layout" "quickAlignSameLayerTarget")
envSetVal("layout" "quickAlignSameLayerTarget" 'boolean t)
```

Related Topics

Quick Align Form

© 2023 Cadence Design Systems, Inc.

quickAlignSnapTarget

```
layout quickAlignSnapTarget cyclic { "Edges and Points" | "Edges" | "Points" }
```

Description

Specifies the method to enable the pointer to the reference set and target object.

- `Edges and Points`: Default. Snaps the pointer to edges, centerlines, vertices, nexus, and end points in the reference set and target object.
- `Edges`: Snaps the pointer to only the edges and centerlines in the reference set and target object.
- `Points`: Snaps the pointer to vertices, midpoint of edges, nexus, end points, and origin point of objects in reference set and target.

GUI Equivalent

Command:	*Edit – Quick Align*
Field:	*Snap Target*

Examples

```
envGetVal("layout" "quickAlignSnapTarget")
envSetVal("layout" "quickAlignSnapTarget" 'cyclic "Edges")
```

Related Topics

Quick Align Form

© 2023 Cadence Design Systems, Inc.

quickAlignSpacingType

```
layout quickAlignSpacingType cyclic { "none" | "min" | "user" }
```

Description

Specifies the spacing between the reference edge, centerline, or point and the target edge, centerline, or point after the alignment.

- `none`: Default. No spacing is applied.
- `min`: The `minSpacing` value of the reference object layer is applied.
- `user`: A user-specified spacing value is applied. Specify the value by using the environment variable `quickAlignUserSpacing`.

GUI Equivalent

Command:	*Edit – Quick Align*
Field:	*No Spacing, Min Spacing, User Spacing*

Examples

```
envGetVal("layout" "quickAlignSpacingType")
envSetVal("layout" "quickAlignSpacingType" 'cyclic "user")
```

Related Topics

Quick Align Form

quickAlignUserSpacing

© 2023 Cadence Design Systems, Inc.

quickAlignUserSpacing

```
layout quickAlignUserSpacing float spacingValue
```

Description

Specifies the user-defined spacing value between the reference edge, centerline, or point and the target edge, centerline, or point after the alignment.You can specify an integer.

The default is `0`.

GUI Equivalent

Command:	*Edit – Quick Align*
Field:	*User Spacing*

Examples

```
envGetVal("layout" "quickAlignUserSpacing")
envSetVal("layout" "quickAlignUserSpacing" 'float 1.0)
```

Related Topics

Quick Align Form

© 2023 Cadence Design Systems, Inc.

realColorSelection

```
graphic realColorSelection boolean { t | nil }
```

Description

Specifies the display of selected objects in their original colors when the design or other objects in the design are set to be dimmed.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*True Color Selection only*

Examples

```
envGetVal("graphic" "realColorSelection")
envSetVal("graphic" "realColorSelection" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

realignLabelToPinCenter

```
layout realignLabelToPinCenter boolean { t | nil }
```

Description

Allows you to realign the label to the center of the pin.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "realignLabelToPinCenter")
envSetVal("layout" "realignLabelToPinCenter" 'boolean t)
```

Related Topics

Quick Align Form

© 2023 Cadence Design Systems, Inc.

rectangularWSPPinAsSquare

```
layout rectangularWSPPinAsSquare boolean { t | nil }
```

Description

Lets you specify if you want to create a square pin.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "rectangularWSPPinAsSquare")
envSetVal("layout" "rectangularWSPPinAsSquare" 'boolean t)
```

Related Topics

Pin Creation

Pin Snapping to Width Spacing Patterns

© 2023 Cadence Design Systems, Inc.

rectBaseMode

```
layout rectBaseMode cyclic { "Bottom left" | "Bottom right" | "Top left" | "Top
     right" | "Center" }
```

Description

Specifies the base creation point of rectangle that you create: `Bottom left`, `Bottom right`, `Top left`, `Top right`, `Center`.

The default is `Bottom left`.

GUI Equivalent

Command:	*Create – Shape – Rectangle*
Field:	*Justification*

Examples

```
envGetVal("layout" "rectBaseMode")
envSetVal("layout" "rectBaseMode" 'cyclic "Top right")
envSetVal("layout" "rectBaseMode" 'cyclic "Center")
```

Related Topics

Create Rectangle Form

© 2023 Cadence Design Systems, Inc.

rectHeight

```
layout rectHeight float height
```

Description

Specifies the height of the rectangle.

The default is `0.0`.

GUI Equivalent

Command:	*Create – Shape – Rectangle*
Field:	*Size: Height*

Examples

```
envGetVal("layout" "rectHeight")
envSetVal("layout" "rectHeight" 'float 3.0)
```

Related Topics

Create Rectangle Form

© 2023 Cadence Design Systems, Inc.

rectOnSnapPatternGrid

```
layout rectOnSnapPatternGrid boolean { t | nil }
```

Description

Enables the first point of the rectangle to snap correctly to the appropriate track enclosure line. This implies when you create a rectangle downwards in the cellview, the first point snaps to the upper track enclosure line. When you create a rectangle upwards in the cellview, the first point snaps to lower track enclosure line. To specify the second point correctly, you are guided by the DRD discrete dimension markers. The discrete dimension markers appear only if discrete dimension is enabled and the `allowedWidthRanges` constraint is specified in the technology file.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "rectOnSnapPatternGrid")
envSetVal("layout" "rectOnSnapPatternGrid" 'boolean t)
```

Related Topics

Create Rectangle Form

© 2023 Cadence Design Systems, Inc.

rectSmartSnapEnable

```
layout rectSmartSnapEnable boolean { t | nil }
```

Description

Specifies whether the rectangle is to be created with smart snapping.

The default is `nil`.

GUI Equivalent

Command:	*Create – Shape – Rectangle*
Field:	*Enable smart snapping*

Examples

```
envGetVal("layout" "rectSmartSnapEnable")
envSetVal("layout" "rectSmartSnapEnable" 'boolean t)
```

Related Topics

Create Rectangle Form

© 2023 Cadence Design Systems, Inc.

rectSmartSnapOnActiveLayer

```
layout rectSmartSnapOnActiveLayer boolean { t | nil }
```

Description

Specifies whether the rectangle is to be created with smart snapping on current active layer only.

The default is `nil`.

GUI Equivalent

Command:	*Create – Shape – Rectangle*
Field:	*Active layer only*

Examples

```
envGetVal("layout" "rectSmartSnapOnActiveLayer")
envSetVal("layout" "rectSmartSnapOnActiveLayer" 'boolean t)
```

Related Topics

Create Rectangle Form

© 2023 Cadence Design Systems, Inc.

rectWidth

```
layout rectWidth float width
```

Description

Specifies the width of the rectangle.

The default is `0.0`.

GUI Equivalent

Command:	*Create – Shape – Rectangle*
Field:	*Size: Width*

Examples

```
envGetVal("layout" "rectWidth")
envSetVal("layout" "rectWidth" 'float 6.0)
```

Related Topics

Create Rectangle Form

© 2023 Cadence Design Systems, Inc.

recursionCheck

```
layout recursionCheck boolean { t | nil }
```

Description

Specifies whether to check for recursive hierarchy when creating instances.

The default is `t`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Recursion Check*

Examples

```
envGetVal("layout" "recursionCheck")
envSetVal("layout" "recursionCheck" 'boolean nil)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

remasterInstCheckTerms

```
layout remasterInstCheckTerms boolean { t | nil }
```

Description

Checks the terminals of cellviews in the specified libraries for matching or a subset of matching terminals.

The default is `nil`.

GUI Equivalent

Command:	*Tools – Remaster Instances*
Field:	*Check Terminals*

Examples

```
envGetVal("layout" "remasterInstCheckTerms")
envSetVal("layout" "remasterInstCheckTerms" 'boolean t)
```

Related Topics

Remaster Instances Form

© 2023 Cadence Design Systems, Inc.

remasterInstSearchCellText

```
layout remasterInstSearchCellText string { "any_cell_name }
```

Description

Specifies the name of the cell that contains the instances you want to use to re-master the instances specified by `remasterInstUpdateCellText`.

GUI Equivalent

Command:	*Tools – Remaster Instances*
Field:	*Search for: CellName*

Examples

```
envGetVal("layout" "remasterInstSearchCellText")
envSetVal("layout" "remasterInstSearchCellText" 'string "MyCell")
```

Related Topics

Remaster Instances Form

© 2023 Cadence Design Systems, Inc.

remasterInstSearchLibText

```
layout remasterInstSearchLibText string { "any_library_name" }
```

Description

Specifies the name of the library which contains the cellviews you want to use to re-master the instances specified by `remasterInstUpdateLibText`.

GUI Equivalent

Command:	*Tools – Remaster Instances*
Field:	*Search for: ViewName*

Examples

```
envGetVal("layout" "remasterInstSearchViewText")
envSetVal("layout" "remasterInstSearchViewText" 'string "MyLib")
```

Related Topics

Remaster Instances Form

© 2023 Cadence Design Systems, Inc.

remasterInstSearchViewText

```
layout remasterInstSearchViewText string { "any_view_name" }
```

Description

Specifies the name of the view you want to use to re-master the instances specified by `remasterInstUpdateViewText`.

GUI Equivalent

Command:	*Tools – Remaster Instances*
Field:	*Search for: ViewName*

Examples

```
envGetVal("layout" "remasterInstSearchViewText")
envSetVal("layout" "remasterInstSearchViewText" 'string "MyView")
```

Related Topics

Remaster Instances Form

© 2023 Cadence Design Systems, Inc.

remasterInstUpdateCellText

```
layout remasterInstUpdateCellText string { "any_cell_name" }
```

Description

Specifies the name of the cellview that you want updated with instances specified by `remasterInstSearchLibText`.

GUI Equivalent

Command:	*Tools – Remaster Instances*
Field:	*Update to: CellName*

Examples

```
envGetVal("layout" "remasterInstUpdateCellText")
envSetVal("layout" "remasterInstUpdateCellText" 'string "MyCell")
```

Related Topics

Remaster Instances Form

© 2023 Cadence Design Systems, Inc.

remasterInstUpdateLibText

```
layout remasterInstUpdateLibText string { "any_library_name" }
```

Description

Specifies the name of the library that you want updated with instances specified by `remasterInstSearchCellText`.

GUI Equivalent

Command:	*Tools – Remaster Instances*
Field:	*Update to: Library*

Examples

```
envGetVal("layout" "remasterInstUpdateLibText")
envSetVal("layout" "remasterInstUpdateLibText" 'string "MyLib")
```

Related Topics

Remaster Instances Form

© 2023 Cadence Design Systems, Inc.

remasterInstUpdateViewText

```
layout remasterInstUpdateViewText string { "any_cell_name" }
```

Description

Specifies the name of the view that you want updated with instances specified by `remasterInstSearchViewText`.

GUI Equivalent

Command:	*Tools – Remaster Instances*
Field:	*Update to: ViewName*

Examples

```
envGetVal("layout" "remasterInstUpdateViewText")
envSetVal("layout" "remasterInstUpdateViewText" 'string "MyCell")
```

Related Topics

Remaster Instances Form

© 2023 Cadence Design Systems, Inc.

removeChopShape

```
layout removeChopShape boolean { t | nil }
```

Description

Specifies whether the area cut by `leHiChop` or `leChopShape` should be removed.

The default is `t`.

GUI Equivalent

Command:	*Edit – Basic – Chop*
Field:	*Remove chop*

Examples

```
envGetVal("layout" "removeChopShape")
envSetVal("layout" "removeChopShape" 'boolean nil)
```

Related Topics

Chop Form

leHiChop

leChopShape

© 2023 Cadence Design Systems, Inc.

removeEmptyNetTerm

```
layout removeEmptyNetTerm boolean { t | nil }
```

Description

Specifies whether the empty net and terminal should be removed with the deleted pin.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "removeEmptyNetTerm")
envSetVal("layout" "removeEmptyNetTerm" 'boolean nil)
```

Related Topics

Object Deletion

Pin Creation

© 2023 Cadence Design Systems, Inc.

removeTerminals

```
layout removeTerminals cyclic { "never" | "always" | "noInstTermsOnNet" }
```

Description

Controls the deletion of the terminal associated with a pin when the pin is deleted.

- `never`: Default. The terminal will not be deleted.
- `always`: The terminal will be deleted if it has no pin figure.
- `noInstTermsOnNet`: The terminal will be deleted if there is no instTerm and pin figure on the terminal.

GUI Equivalent

None

Examples

```
envGetVal("layout" "removeTerminals")
envSetVal("layout" "removeTerminals" 'cyclic "always")
envSetVal("layout" "removeTerminals" 'cyclic "noInstTermsOnNet")
```

Related Topics

Object Deletion

Pin Creation

© 2023 Cadence Design Systems, Inc.

replaceFigs

```
layout replaceFigs boolean { t | nil }
```

Description

Specifies whether selected figures are replaced with a new cell created from the selected figures. This setting is used by `leHiMakeCell` and `leMakeCell`.

The default is `t`.

GUI Equivalent

Command	*Edit – Hierarchy – Make Cell*
Field	*Replace selected figures*

Examples

```
envGetVal("layout" "replaceFigs")
envSetVal("layout" "replaceFigs" 'boolean nil)
```

Related Topics

Make Cell Form

leHiMakeCell

leMakeCell

© 2023 Cadence Design Systems, Inc.

resetOrientation

```
layout resetOrientation boolean { t | nil }
```

Description

Specifies whether the *Move* or *Copy* commands or SKILL code using `leHiMove` or `leHiCopy` resets orientation to `0` prior to the move or copy, or retains the specified orientation as set by the `orientation` environment variable. Terminating the *Move* or *Copy* command always resets `orientation` to `0`.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "resetOrientation")
envSetVal("layout" "resetOrientation" 'boolean nil)
```

Related Topics

Copy Form

Move Form

leHiMove

leHiCopy

© 2023 Cadence Design Systems, Inc.

reshapeType

```
layout reshapeType cyclic { "unknown" | "line" | "rectangle" }
```

Description

Specifies the default shape used by `leHiReShape`.

The default is `rectangle`.

GUI Equivalent

Command:	*Edit – Advanced – Reshape*
Field:	*Mode*

Examples

```
envGetVal("layout" "reshapeType")
envSetVal("layout" "reshapeType" 'cyclic "unknown")
envSetVal("layout" "reshapeType" 'cyclic "line")
```

Related Topics

Reshape Form

© 2023 Cadence Design Systems, Inc.

retainSelectionSet

```
layout retainSelectionSet boolean { t | nil }
```

Description

Specifies that the shapes selection set is retained for the *Move* and *Stretch* commands when the *Repeat Commands* option is selected in the Layout Editor Options form.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "retainSelectionSet")
envSetVal("layout" "retainSelectionSet" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

reuseExistingFrozenMaster

```
layout reuseExistingFrozenMaster boolean { t | nil }
```

Description

Uses the existing frozen Pcell master, if available, when converting a Pcell instance to a non-Pcell instance.

The default is `nil`.

GUI Equivalent

Command:	Instance shortcut menu – *Freeze Pcell*
Field:	*Use Existing Master*

Examples

```
envGetVal("layout" "reuseExistingFrozenMaster")
envSetVal("layout" "reuseExistingFrozenMaster" 'boolean t)
```

Related Topics

Freeze Pcell Instance Form

Freezing and Unfreezing Pcell Instances

© 2023 Cadence Design Systems, Inc.

rodAutoName

```
layout rodAutoName string { "pin" | "rectangle" | "polygon" | "path" }
```

Description

Specifies that rectangles, polygons and/or shape pins be created as ROD objects. These shapes are given either system- or user-assigned names.

The default is "".

GUI Equivalent

Command:	*Create – Pin, Create – Shape – Rectangle, Create – Shape – Polygon, Create – Shape – Path, Create – Wiring – Wire*
Field:	*Create as ROD object*

Examples

```
envGetVal("layout" "rodAutoName")
envSetVal("layout" "rodAutoName" 'string "pin")
envSetVal("layout" "rodAutoName" 'string "rectangle")
envSetVal("layout" "rodAutoName" 'string "rectangle pin polygon")
```

Related Topics

Create Pin Form

Create Rectangle Form

Create Polygon Form

Create Path Form

© 2023 Cadence Design Systems, Inc.

rotateAroundCombinedCenter

```
layout rotateAroundCombinedCenter boolean { t | nil }
```

Description

Enables multiple selected objects to be rotated around the specified reference point of the imaginary bounding box around the objects.

The default is `t`.

GUI Equivalent

Command:	*Options – Editor*
	Edit – Rotate – Rotate Left
	Edit – Rotate – Rotate Right
Field:	*Rotate Around Combined BBox*

Examples

```
envGetVal("layout" "rotateAroundCombinedCenter")
envSetVal("layout" "rotateAroundCombinedCenter" 'boolean nil)
```

Related Topics

Rotate Left Form

Rotate Right Form

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

rotateReference

```
layout rotateReference cyclic { "LowerLeft" | "CenterLeft" | "UpperLeft" |
     "LowerCenter" | "CenterCenter" | "UpperCenter" | "LowerRight" | "CenterRight"
     | "UpperRight" | "Custom" }
```

Description

Sets the reference point around which an object is rotated.

- `LowerLeft`: Rotates an object around the lower-left point of its bounding box.
- `CenterLeft`: Rotates an object around the center-left point of its bounding box.
- `UpperLeft`: Rotates an object around the upper-left point of its bounding box.
- `LowerCenter`: Rotates an object around the lower-center point of its bounding box.
- `CenterCenter`: Default. Rotates an object around the center point of its bounding box.
- `UpperCenter`: Rotates an object around the upper-center point of its bounding box.
- `LowerRight`: Rotates an object around the lower-right point of its bounding box.
- `CenterRight`: Rotates an object around the center-right point of its bounding box.
- `UpperRight`: Rotates an object around the upper-right point of its bounding box.
- `Custom`: Rotates an object around the specified X and Y coordinates. This option is available in the Rotate form.

GUI Equivalent

Command:	*Edit – Rotate – Rotate*
	Edit – Rotate – Rotate Left
	Edit – Rotate – Rotate Right
Field:	*Reference*

Examples

```
envGetVal("layout" "rotateReference")
envSetVal("layout" "rotateReference" 'cyclic "LowerLeft")
envSetVal("layout" "rotateReference" 'cyclic "Custom")
```


© 2023 Cadence Design Systems, Inc.

Related Topics

Rotate Form

Rotate Left Form

Rotate Right Form

© 2023 Cadence Design Systems, Inc.

rotateInstParam

```
layout rotateInstParam string "rotationParameter"
```

Description

Enables you to specify the name of the parameter to be used to control the rotation of the Pcell.

The default is `rotation`.

GUI Equivalent

Command:	*Edit – Properties– Parameter*
Field:	*Rotation*

Examples

```
envGetVal("layout" "rotateInstParam")
envSetVal("layout" "rotateInstParam" 'string "rotation")
```

Related Topics

Edit Properties Form

© 2023 Cadence Design Systems, Inc.

rowAbutFirstUnit

```
layout rowAbutFirstUnit boolean { t | nil }
```

Description

Sets the *Options: Abut The First Row* check box in the *Definition* section of the *Create – P&R Objects – Row* form to abut the first and second rows.

The default is `nil`.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Options: Abut The First Row*

Examples

```
envGetVal("layout" "rowAbutFirstUnit")
envSetVal("layout" "rowAbutFirstUnit" 'boolean t)
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowAbutNumberOfUnits

`layout rowAbutNumberOfUnits int` *`dataEntryValue`*

Description

Sets the value of the data entry field for the *Abut* option in the *Definition* section of the *Create – P&R Objects – Row* form to specify how many rows you want to abut. The valid values are any integer greater than zero.

The default value is `2`.

To specify a value for this data entry field, the *Abut* option must be set to `t`. You can set the *Abut* option with the `rowAbutUnit` environment variable.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Options: Abut: rows*

Examples

```
envGetVal("layout" "rowAbutNumberOfUnits")
envSetVal("layout" "rowAbutNumberOfUnits" 'int 10)
```

Related Topics

rowAbutUnit

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowAbutUnit

```
layout rowAbutUnit boolean { t | nil }
```

Description

Sets the *Options: Abut* check box in the *Definition* section of the *Create – P&R Objects – Row* form to abut a specified number of rows.

The default is `nil`. When set to `t`, you can set the number of rows for the data entry field next to the *Abut* option with the `rowAbutNumberOfUnits` environment variable.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Options: Abut*

Examples

```
envGetVal("layout" "rowAbutUnit")
envSetVal("layout" "rowAbutUnit" 'boolean t)
```

Related Topics

rowAbutNumberOfUnits

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowBaseName

```
layout rowBaseName string "BaseNameField"
```

Description

Sets the *Base Name* field in the *Definition* section of the *Create – P&R Objects – Row* form to assign a unique name for each row object you create. The software adds an underscore followed by a number which is greater than or equal to 1. As you create new sites, each site is increment by 1. For example, if your base name is ROW_1, the name assigned to the first site is ROW_1_1, the next is ROW_1_2, the next is ROW_1_3, and so on. If you do not specify a base name, the software assigns the name row_number, which increments by 1 each time you add a site and rows.

The default value is `""`.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Base Name*

Examples

```
envGetVal("layout" "rowBaseName")
envSetVal("layout" "rowBaseName" 'string "row1")
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowCreateMethod

```
layout rowCreateMethod cyclic { "assisted" | "define" | "partition" | "rectilinear"
     }
```

Description

Determines the create method for rows.The assisted mode will not take into account any areas or boundaries. You can create rows that use a rectangular, rectilinear, or a partitioned area to determine the number and placement of the rows

The default is `assisted`.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Create Method*

Examples

```
envGetVal("layout" "rowCreateMethod")
envSetVal("layout" "rowCreateMethod" 'cyclic "define")
envSetVal("layout" "rowCreateMethod" 'cyclic "rectilinear")
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowChannelWidth

```
layout rowChannelWidth float userUnits
```

Description

Sets the *Options: Channel Width* data entry field in the *Definition* section of the *Create – P&R Objects – Row* form to specify the space between the rows you create.

If you specify to abut rows with `rowAbutUnit` in the form, *Channel Width* is overridden.

The default is `0.0`.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Options: Channel Width*

Examples

```
envGetVal("layout" "rowChannelWidth")
envSetVal("layout" "rowChannelWidth" 'float 1.0)
```

Related Topics

rowAbutUnit

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowFirstUnitOrient

```
layout rowFirstUnitOrient string { "Horizontal" | "Vertical" }
```

Description

Sets the *Row Orientation* radio button in the *Definition* section of the *Create – P&R Objects – Row* form.

The valid values are `Horizontal` and `Vertical`. The default is `Horizontal`.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Row Orientation*

Examples

```
envGetVal("layout" "rowFirstUnitOrient")
envSetVal("layout" "rowFirstUnitOrient" 'string "Vertical")
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowFlipFirstUnit

```
layout rowFlipFirstUnit boolean { t | nil }
```

Description

Sets the *Options: Flip The First Row* check box in the *Definition* section of the *Create – P&R Objects – Row* form to flip the first row in the site.

The default is `nil`.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Options: Flip The First Row*

Examples

```
envGetVal("layout" "rowFlipFirstUnit")
envSetVal("layout" "rowFlipFirstUnit" 'boolean t)
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowFlipUnit

```
layout rowFlipUnit boolean { t | nil }
```

Description

Sets the *Options: Flip Every Other Row* check box in the *Definition* section of the *Create – P&R Objects – Row* form to flip every other row in the site. The valid values are `t` and `nil`.

The default is `nil`.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Options: Flip Every Other Row*

Examples

```
envGetVal("layout" "rowFlipUnit")
envSetVal("layout" "rowFlipUnit" 'boolean t)
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowNumberOfRows

```
layout rowNumberOfRows int NumberOfRows
```

Description

Sets the value of the *Number of Rows* field in the *Create – P&R Objects – Row* form to the number of sites in each row.

The default is `1`.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Number of Rows*

Examples

```
envGetVal("layout" "rowNumberOfRows")
envSetVal("layout" "rowNumberOfRows" 'int 5)
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowPartitionBottomOffset

```
layout rowPartitionBottomOffset float offset
```

Description

Sets the offset in which the rows are not allowed to come closer to the bottom of the boundary or cluster edge than the specified float number.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Offset: Bottom*

Examples

```
envGetVal("layout" "rowPartitionBottomOffset")
envSetVal("layout" "rowPartitionBottomOffset" 'float 2.0)
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowPartitionLeftOffset

```
layout rowPartitionLeftOffset float offset
```

Description

Sets the offset in which the rows are not allowed to come closer to the left side of the boundary or cluster edge than the specified float number.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Offset: Left*

Examples

```
envGetVal("layout" "rowPartitionLeftOffset")
envSetVal("layout" "rowPartitionLeftOffset" 'float 2.0)
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowPartitionRightOffset

```
layout rowPartitionRightOffset float offset
```

Description

Sets the offset in which the rows are not allowed to come closer to the right side of the boundary or cluster edge than the specified float number.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Offset: Right*

Examples

```
envGetVal("layout" "rowPartitionRightOffset")
envSetVal("layout" "rowPartitionRightOffset" 'float 2.0)
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowPartitionTopOffset

```
layout rowPartitionTopOffset float Offset
```

Description

Sets the offset in which the rows are not allowed to come closer to the top of the boundary or cluster edge than the specified float number.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Offset: Top*

Examples

```
envGetVal("layout" "rowPartitionTopOffset")
envSetVal("layout" "rowPartitionTopOffset" 'float 2.0)
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowNumberOfSites

```
layout rowNumberOfSites int numberOfSites
```

Description

Sets the value of the *Number of Sites* data entry field in the *Definition* section of the *Create – P&R Objects – Row* form to the number of sites in each row. The number of sites is any integer equal to or greater than zero.

The default is `1`.

GUI Equivalent

Command:	*Create – P&R Objects – Row*
Field:	*Number of Sites*

Examples

```
envGetVal("layout" "rowNumberOfSites")
envSetVal("layout" "rowNumberOfSites" 'int 5)
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rowSiteEnable

```
layout rowSiteEnable boolean { t | nil }
```

Description

Snaps the instances to the nearest site within a row.The instance must be either over a legal row or the distance between the instance and the legal row must be within the specified aperture in Layout Editor Options form.

The default is `nil`.

The priority of snapping is as follows:

- snap boundary: if the instance has a snap boundary, the snap boundary will snap to the row site.
- prBoundary: if the instance has a prBoundary but no snap boundary, the prBoundary will snap to the row site.
- cellview: if the instance neither has a snap boundary nor a prBoundary, then the lower left corner of the instance will snap to the row site.

Note: If you are creating or editing an instance and `rowSiteEnable` is set to `t`, `rowSiteEnable` takes precedence over `snapToGrid` for snapping of the instance.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Instance to Row*

Examples

```
envGetVal("layout" "rowSiteEnable")
envSetVal("layout" "rowSiteEnable" 'boolean t)
```

Related Topics

Layout Editor Options Form

snapToGrid

Snap Pattern Snapping for FinFET Devices in Layout Editor

© 2023 Cadence Design Systems, Inc.

rowSiteSymmetry

```
layout rowSiteSymmetry boolean { t | nil }
```

Description

Snaps an instance to a row by retaining the original orientation of the instance if the symmetry setting in the row site definition allows instance orientation.

Orientation of an instance is retained in the following cases:

- If the row site definition is symmetric in X and the instance has MY orientation.
- If the row site definition is symmetric in Y and the instance has MX orientation.
- If the row site definition is symmetric in X and Y.

The default is `nil`, which implies that an instance is snapped to a row by matching the orientation of the row.

GUI Equivalent

None

Examples

```
envGetVal("layout" "rowSiteSymmetry")
envSetVal("layout" "rowSiteSymmetry" 'boolean t)
```

Related Topics

Rows

© 2023 Cadence Design Systems, Inc.

rowUnitOrient

```
layout rowUnitOrient cyclic { "R0" | "R90" | "R180" | "R270" | "MX" | "MY" | "MXR90"
     | "MYR90" }
```

Description

Sets the *Starting Site Orientation* cyclic field in the *Definition* section of the *Create – P&R Objects – Row* form to set the orientation of the site and the first row created. All subsequent rows will follow the orientation you set in this field. The valid values are `R0`, `R90`, `R180`, `R270`, `MX`, `MY`, `MXR90`, and `MYR90`.

The default is `RO`.

GUI Equivalent

Command	*Create – P&R Objects – Row*
Field	*Starting Site Orientation*

Examples

```
envGetVal("layout" "rowUnitOrient")
envSetVal("layout" "rowUnitOrient" 'cyclic "R90")
envSetVal("layout" "rowUnitOrient" 'cyclic "MX")
envSetVal("layout" "rowUnitOrient" 'cyclic "MYR90")
```

Related Topics

Create Row Form

© 2023 Cadence Design Systems, Inc.

rulerColorPacketName

```
graphic rulerColorPacketName string "packetName"
```

Description

Enables you to specify the name of the packet which has the `Outline` attribute with the color information for the savable ruler. You can add a new packet with the required attributes or modify attributes of a packet in the `display.drf` file or through the *Display Resource Editor*.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "rulerColorPacketName")
envSetVal("graphic" "rulerColorPacketName" 'string "ref_drawing")
```

Related Topics

Display Resource Editor

© 2023 Cadence Design Systems, Inc.

rulerDisplayType

```
layout rulerDisplayType cyclic ( "Ruler" | "Distance" | "Both" )
```

Description

Specifies the display type of the ruler.

The default is `Ruler`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Type*

Examples

```
envGetVal("layout" "rulerDisplayType")
envSetVal("layout" "rulerDisplayType" 'cyclic "Distance")
envSetVal("layout" "rulerDisplayType" 'cyclic "Both")
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

rulerEdgeMode

```
layout rulerEdgeMode boolean { t | nil }
```

Description

Enables edge measurement mode for rulers.

The default is `nil`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Edge Measurement*

Examples

```
envGetVal("layout" "rulerEdgeMode")
envSetVal("layout" "rulerEdgeMode" 'boolean t)
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

rulerFontBold

```
layout rulerFontBold boolean { t | nil }
```

Description

Enables you to display the ruler labels in bold face.

The default is `nil`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Display Options: Bold*

Examples

```
envGetVal("layout" "rulerFontBold")
envSetVal("layout" "rulerFontBold" 'boolean t)
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

rulerFontSize

```
layout rulerFontSize cyclic { "Small" | "Medium" | "Large" }
graphic rulerFontSize int fontSize
```

Description

Specifies the font size for the ruler labels.

The default is `12` for int, and `"Small"` for cyclic.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Display Options: Font Size*

Examples

```
envGetVal("layout" "rulerFontSize")
envSetVal("layout" "rulerFontSize" 'cyclic "Medium")
envGetVal("graphic" "rulerFontSize")
envSetVal("graphic" "rulerFontSize" 'int 10)
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

rulerLabelBackgroundColor

```
layout rulerLabelBackgroundColor cyclic { "nil" | "black" | "gray" | "pink" | "red"
     | "green" | "forest" | "blue" | "navy" | "chocolate" }

graphic rulerLabelBackgroundColor string { "nil" | "black" | "gray" | "pink" | "red"
     | "green" | "forest" | "blue" | "navy" | "chocolate" }
```

Description

Specifies the background color of the ruler labels.

The default is `black`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Display Options: Background Color*

Examples

```
envGetVal("layout" "rulerLabelBackgroundColor")
envSetVal("layout" "rulerLabelBackgroundColor" 'cyclic "gray")
envSetVal("layout" "rulerLabelBackgroundColor" 'cyclic "chocolate")
envGetVal("graphic" "rulerLabelBackgroundColor")
envSetVal("graphic" "rulerLabelBackgroundColor" 'string "pink")
envSetVal("graphic" "rulerLabelBackgroundColor" 'string "green")
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

rulerLimitDecimals

```
graphic rulerLimitDecimals boolean { t | nil }
```

Description

Limits the number of decimal places displayed for the end-point value on a diagonal ruler. For example, the end-point value `5.830962` might appear as `5.831~`. The number of decimal places displayed is calculated proportional to the DBUPerUU property of a cellview.

The default is `nil`.

Note: If both `rulerLimitDecimals` and `displayRulerMiddlePoint` are set to `t`, `rulerLimitDecimals` also applies to the middle-point value on a diagonal ruler.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "rulerLimitDecimals")
envSetVal("graphic" "rulerLimitDecimals" 'boolean t)
```

Related Topics

displayRulerMiddlePoint

© 2023 Cadence Design Systems, Inc.

rulerSegmentModeType

```
layout rulerSegmentModeType cyclic ( "Single" | "Multiple" | "Auto" )
```

Description

Specifies the type of ruler created by the measurement.You can set the environment variable to one of the following values:

- `Single`: Creates a single-segment ruler. This is the measurement between points on the canvas.
- `Multiple`: Creates a multiple-segment ruler. In this mode, you can view the measurement between consecutive points that you specify.
- `Auto`: Creates a multiple-segment ruler. In this mode, points are automatically calculated. There are two ways in which the ruler is created in the auto mode:
  - An auto ruler is created between two specified points on the canvas.
  - You press the `Shift` key in the auto mode. In this case, an orthogonal ruler is created with all edges visible on the canvas in vertical or horizontal direction. In this case, the direction of the ruler is based on the edge closest to the pointer.

The default is `Single`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Segment Mode*

Examples

```
envGetVal("layout" "rulerSegmentModeType")
envSetVal("layout" "rulerSegmentModeType" 'cyclic "Multiple")
envSetVal("layout" "rulerSegmentModeType" 'cyclic "Auto")
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

rulerSegSnapMode

```
layout rulerSegSnapMode string { "orthogonal" | "diagonal" | "anyAngle" |
     "horizontal" | "vertical" }
```

Description

Determines how the single-segment ruler entry points snap to the grid. This environment variable takes effect when `useRulerSnapModes` is set to `t`.

The default is `orthogonal`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Snap Mode*

Examples

```
envGetVal("layout" "rulerSegSnapMode")
envSetVal("layout" "rulerSegSnapMode" 'string "diagonal")
envSetVal("layout" "rulerSegSnapMode" 'string "anyAngle")
```

Related Topics

useRulerSnapModes

rulerSnapMode

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

rulerSnapMode

```
layout rulerSnapMode string { "orthogonal" | "diagonal" | "anyAngle" | "L90XFirst"
     | "L90YFirst" }
```

Description

Determines how the multi-segment ruler entry points snap to the grid. This environment variable takes effect when `useRulerSnapModes` is set to `t`.

The default is `orthogonal`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Snap Mode*

Examples

```
envGetVal("layout" "rulerSnapMode")
envSetVal("layout" "rulerSnapMode" 'string "diagonal")
envSetVal("layout" "rulerSnapMode" 'string "L90XFirst")
```

Related Topics

useRulerSnapModes

rulerSegSnapMode

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

saveACopyLibName

```
layout saveACopyLibName string "any_library_name"
```

Description

Specifies the library name where the design is saved while saving a copy of the design. The user may edit the library name value while using the Save a Copy form and save the design to a different library.

The default value is `""`, which implies that by default the library of the current design will be used for saving a copy of the design.

GUI Equivalent

Command:	*File – Save a Copy*
Field:	*Library* cyclic field

Examples

```
envGetVal("layout" "saveACopyLibName")
envSetVal("layout" "saveACopyLibName" 'string "MyLib")
```

Related Topics

Save a Copy Form

© 2023 Cadence Design Systems, Inc.

saveFindReplaceReportToFile

```
layout saveFindReplaceReportToFile boolean { t | nil }
```

Description

Enables the *Save Report To* option for find and replace operations.

The default is `nil`.

GUI Equivalent

Command:	*Tools – Find/Replace*
Field:	*Save Report To*

Examples

```
envGetVal("layout" "saveFindReplaceReportToFile")
envSetVal("layout" "saveFindReplaceReportToFile" 'boolean t)
```

Related Topics

Find/Replace Form

findReplaceReportFileName

© 2023 Cadence Design Systems, Inc.

saveRulers

```
layout saveRulers boolean { t | nil }
```

Description

Saves the rulers in a cellview when you save the cellview.

The default is `t`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Savable Measurement*

Examples

```
envGetVal("layout" "saveRulers")
envSetVal("layout" "saveRulers" 'boolean nil)
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

schematicDefaultApp

`graphic schematicDefaultApp string "`*`registeredApplication`*`"`

Description

Automates the launching of a default application for cellviews of type `schematic`. This environment variable setting is ignored when descending into or editing-in-place in the same view type.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "schematicDefaultApp")
envSetVal("graphic" "schematicDefaultApp" 'string "VSE L")
```

Related Topics

Set Default Application Form

© 2023 Cadence Design Systems, Inc.

schematicDefaultTier

```
graphic schematicDefaultTier string { L | XL }
```

Description

Sets the *Choose Schematic/Symbol Level* field in the File – Set Default Application form to a default application to launch for cellviews of type `schematic` or `symbol`.

The default is `""`.

GUI Equivalent

Command:	In a schematic or symbol cellview, *File – Set Default Application*
Field:	*Choose Schematic/Symbol Level*

Examples

```
envGetVal("graphic" "schematicDefaultTier")
envSetVal("graphic" "schematicDefaultTier" 'string "L")
envSetVal("graphic" "schematicDefaultTier" 'string "XL")
```

Related Topics

Set Default Application Form

© 2023 Cadence Design Systems, Inc.

schematicSymbolDefaultApp

```
graphic schematicSymbolDefaultApp string "registered_application"
```

Description

Automates the launching of a default application for cellviews of type `schematicSymbol`. This environment variable setting is ignored when descending into or editing-in-place in the same view type.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "schematicSymbolDefaultApp")
envSetVal("graphic" "schematicSymbolDefaultApp" 'string "VSE L")
```

Related Topics

Set Default Application Form

© 2023 Cadence Design Systems, Inc.

scrollPercent

```
graphic scrollPercent int scrollingPercent
layout scrollPercent int scrollingPercent
```

Description

Enables you to specify the scrolling percent per window.

The default is `25`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Scroll Percent*

Examples

```
envGetVal("graphic" "scrollPercent")
envSetVal("graphic" "scrollPercent" 'int 30)
envGetVal("layout" "scrollPercent")
envSetVal("layout" "scrollPercent" 'int 30)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

segSnapMode

```
layout segSnapMode string { "orthogonal" | "diagonal" | "anyAngle" | "horizontal"
     | "vertical" }
graphic segSnapMode string { "orthogonal" | "diagonal" | "anyAngle" | "horizontal"
     | "vertical" }
```

Description

Determines how *Edit* command entry points snap to the grid.

The layout default is `orthogonal`, the graphic default is `anyAngle`.

Note: `anyAngle` is not a valid value for the *Create Wire* (XL, GXL) command.

GUI Equivalent

Command:	*Options – Display*
Field:	*Snap Modes*: *Edit*

Examples

```
envGetVal("layout" "segSnapMode")
envSetVal("layout" "segSnapMode" 'cyclic "diagonal")
envGetVal("graphic" "segSnapMode")
envSetVal("graphic" "segSnapMode" 'cyclic "orthogonal")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

selectCreatedObj

```
layout selectCreatedObj boolean { t | nil }
```

Description

Determines whether or not an object remains selected after its creation.

The default is `nil`. When set to `t`, all objects you create are selected after creation. Any object selected before starting the create command also remain selected.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Select Created Object*

Examples

```
envGetVal("graphic" "selectCreatedObj")
envSetVal("graphic" "selectCreatedObj" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

selectHilightPacket

```
graphic selectHilightPacket string "packet_name"
```

Description

Specifies the packet name to be used to display the selection highlight.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "selectHilightPacket")
envSetVal("graphic" "selectHilightPacket" 'string "myPacket")
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

selectionAperture

```
layout selectionAperture int numberOfPixels
graphic selectionAperture int numberOfPixels
```

Description

Creates a selection aperture box around the pointer with the specified number of pixels. A setting of 5 will create a square of 6 by 6 pixels, which includes the center pixel. When a point is clicked the pointer will select the closest object contained within the aperture area. The next click will select the second closest object contained within the aperture area, and so on. Only one object is selected with each click. When no more objects are contained within the aperture area that have not already been selected, the pointer will cycle back to the first object selected. and each subsequent click will repeat the selection order.

The default is `5`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "selectionAperture")
envSetVal("layout" "selectionAperture" 'int 10)
envGetVal("graphic" "selectionAperture")
envSetVal("graphic" "selectionAperture" 'int 10)
```

Related Topics

Technology File

© 2023 Cadence Design Systems, Inc.

selectMarkersAtLast

```
graphic selectMarkersAtLast boolean { t | nil }
```

Description

Selects markers at last, after the shapes even if the marker edge is closer than the shape edge or markers and shapes overlap.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "selectMarkersAtLast")
envSetVal("graphic" "selectMarkersAtLast" 'boolean t)
```

Related Topics

Markers

© 2023 Cadence Design Systems, Inc.

selectObjectLargerThanWindow

```
graphic selectObjectLargerThanWindow boolean { t | nil }
```

Description

Controls whether or not you can select an object when its size is greater than the size of the design canvas in which it is displayed.

The default is `t`.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Select objects larger than the window*

Examples

```
envGetVal("graphic" "selectObjectLargerThanWindow")
envSetVal("graphic" "selectObjectLargerThanWindow" 'boolean nil)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

selectObjectInEditCvFirst

```
graphic selectObjectInEditCvFirst boolean { t | nil }
```

Description

Controls whether you can select objects belonging to the cellview being edited in preference over the other objects, which might be on a lower layers.

The default is `nil`.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Select objects of edit cellview first*

Examples

```
envGetVal("graphic" "selectObjectInEditCvFirst")
envSetVal("graphic" "selectObjectInEditCvFirst" 'boolean t)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

selectOnlyInstWithSelectableShapes

```
layout selectOnlyInstWithSelectableShapes boolean { t | nil }
```

Description

Determines whether instances are selectable.

An instance or instances are selected when the *Instances* option is on, even if none of the shapes within them are on selectable layers when the default value `nil` is used.

When set to `t`, an instance is selected only if *Instances* is selected in the *Objects* assistant and the instance or instances contain shapes that are on selectable layers.

GUI Equivalent

None

Examples

```
envGetVal("layout" "selectOnlyInstWithSelectableShapes")
envSetVal("layout" "selectOnlyInstWithSelectableShapes" 'boolean t)
```

Related Topics

Creating Instances

© 2023 Cadence Design Systems, Inc.

selectPartialPathHighlightStyle

```
graphic selectPartialPathHighlightStyle cyclic { "centerLine" | "boundary" |
     "centerLineAndBoundary" }
```

Description

Specifies the style used for highlighting the boundary ends of a path or pathSeg in partial selection mode.

- `centerLine`: Default. Highlights only a part of the centerline of a path or pathSeg.
- `boundary`: Highlights the edge of a path or pathSeg.
- `centerLineAndBoundary`: Highlights part of the centerline as well as the edge of a path or pathSeg.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "selectPartialPathHighlightStyle")
envSetVal("graphic" "selectPartialPathHighlightStyle" 'cyclic "boundary")
envSetVal("graphic" "selectPartialPathHighlightStyle" 'cyclic
"centerLineAndBoundary")
```

Related Topics

Stretching Paths

© 2023 Cadence Design Systems, Inc.

selectProtectedItems

```
graphic selectProtectedItems boolean { t | nil }
```

Description

Determines whether protected items should be selected or not.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "selectProtectedItems")
envSetVal("graphic" "selectProtectedItems" 'boolean t)
```

Related Topics

Object Selectability Controls

© 2023 Cadence Design Systems, Inc.

selectUnderCursorMenuAlphaSorting

```
graphic selectUnderCursorMenuAlphaSorting boolean { t | nil }
```

Description

Specifies the objects listed in the *(De)Select Under Cursor* shortcut menu appear in alphabetical order.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "selectUnderCursorMenuAlphaSorting")
envSetVal("graphic" "selectUnderCursorMenuAlphaSorting" 'boolean t)
```

Related Topics

Object Selection

© 2023 Cadence Design Systems, Inc.

selectUnderCursorMenuLimit

`graphic selectUnderCursorMenuLimit int` *`ObjectsForDeseclection`*

Description

Controls the appearance of the *(De)Select Under Cursor* shortcut menu.

The default is `20`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "selectUnderCursorMenuLimit")
envSetVal("graphic" "selectUnderCursorMenuLimit" 'int 25)
```

Related Topics

Object Selection

© 2023 Cadence Design Systems, Inc.

setDxDyToZero

```
layout setDxDyToZero boolean { t | nil }
```

Description

Enables you to set the *dX* and *dY* fields to `0` on the status toolbar.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "setDxDyToZero")
envSetVal("layout" "setDxDyToZero" 'boolean nil)
```

Related Topics

Status Toolbar

© 2023 Cadence Design Systems, Inc.

setupConstraintGroup

```
layout setupConstraintGroup string "anyValidConstraintGroup"
```

Description

Specifies the default set of legal interconnect layers which Layout XL (`virtuosoDefaultExtractorSetup`) use for connectivity extraction.

A custom constraint group can have user defined names (`myRoutingLayers`). A constraint group must minimally contain valid interconnect layers, and may optionally include a list of valid vias.

GUI Equivalent

None

Examples

```
envGetVal("layout" "setupConstraintGroup")
envSetVal("layout" "setupConstraintGroup" 'string "x y z")
```

Related Topics

Layout XL Constraint Group

© 2023 Cadence Design Systems, Inc.

showFindReplaceReport

```
layout showFindReplaceReport boolean { t | nil }
```

Description

Displays a detailed report for any find and replace operation in CIW. This function is supported for all *Find*, *Replace,* and *Replace All* operations.

The default is `nil`.

GUI Equivalent

Command:	*Tools – Find/Replace*
Field:	*Show Detailed Report in CIW*

Examples

```
envGetVal("layout" "showFindReplaceReport")
envSetVal("layout" "showFindReplaceReport" 'boolean t)
```

Related Topics

Find/Replace Form

© 2023 Cadence Design Systems, Inc.

showAnglesSigDigits

`layout showAnglesSigDigits int` *`decimalPlaces`*

Description

Specifies the number of decimal places displayed for the interior and exterior angles between the edges of a shape.

The default value is `0`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "showAnglesSigDigits")
envSetVal("layout" "showAnglesSigDigits" 'int 2)
```

Related Topics

Viewing Angles

© 2023 Cadence Design Systems, Inc.

showWhatsNew

```
layout showWhatsNew string { "yes" | "no" }
```

Description

Determines whether the layout editor What's New window is displayed when a cellview window is opened.

The default is `yes`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "showWhatsNew")
envSetVal("layout" "showWhatsNew" 'string "no")
```

Related Topics

Opening a New Design

Opening an Existing Cellview

© 2023 Cadence Design Systems, Inc.

slotEdgeToSlotOffset

```
layout slotEdgeToSlotOffset float edgedistance
```

Description

Specifies the distance between the slot edge and the shape edge.

The default is `1.0`.

GUI Equivalent

Command:	*Create – Slot*, *Edit – Advanced – Slot*, *Create – Shape – Rectangle*, *Create – Shape – Path – Slot Options*
Field:	*Spacing: Edge*
	Slot Offset: Edge to Slot

Examples

```
envGetVal("layout" "slotEdgeToSlotOffset")
envSetVal("layout" "slotEdgeToSlotOffset" 'float 3.0)
```

Related Topics

Create Slot Form

Create Rectangle Form

Create Path Form

© 2023 Cadence Design Systems, Inc.

slotEnable

```
layout slotEnable boolean { t | nil }
graphic slotEnable boolean { t | nil }
```

Description

Overlaps the shape to slot with another shape that is displayed like a hole in the shape.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Slot Shapes*

Examples

```
envGetVal("layout" "slotEnable")
envSetVal("layout" "slotEnable" 'boolean t)
envGetVal("graphic" "slotEnable")
envSetVal("graphic" "slotEnable" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

slotEnableRect

```
layout slotEnableRect boolean { t | nil }
```

Description

Specifies whether the rectangle is to be created with slots.

The default is `nil`.

GUI Equivalent

Command:	*Create – Shape – Rectangle*
Field:	*Slotting: Enable*

Examples

```
envGetVal("layout" "slotEnableRect")
envSetVal("layout" "slotEnableRect" 'boolean t)
```

Related Topics

Create Rectangle Form

© 2023 Cadence Design Systems, Inc.

slotEnablePath

```
layout slotEnablePath boolean { t | nil }
```

Description

Specifies whether the path is to be created with slots.

The default is `nil`.

GUI Equivalent

Command:	*Create – Shape – Path – Path Options*
Field:	*Enable Metal Slotting*

Examples

```
envGetVal("layout" "slotEnablePath")
envSetVal("layout" "slotEnablePath" 'boolean t)
```

Related Topics

Create Path Form

© 2023 Cadence Design Systems, Inc.

slotInterSlotSpacing

```
layout slotInterSlotSpacing float slotDistance
```

Description

Specifies the distance between the slots in a row.

The default is `1.0`.

GUI Equivalent

Command:	*Create – Slot*, *Edit – Advanced – Slot*, *Create – Shape – Rectangle*, *Create – Shape – Path – Slot Options*
Field:	*Spacing: Column*
	Slot Spacing: Between Slots

Examples

```
envGetVal("layout" "slotInterSlotSpacing")
envSetVal("layout" "slotInterSlotSpacing" 'float 3.0)
```

Related Topics

Create Slot Form

Edit Synchronous Clone Form

Create Rectangle Form

Create Path Form

© 2023 Cadence Design Systems, Inc.

slotLength

```
layout slotLength float slotLength
```

Description

Specifies the length of a slot.

The default is `1.0`.

GUI Equivalent

Command:	*Create – Slot*, *Edit – Advanced – Slot*, *Create – Shape – Rectangle*, *Create – Shape – Path – Slot Options*
Field:	*Dimensions: Length*
	Slot Dimensions: Length

Examples

```
envGetVal("layout" "slotLength")
envSetVal("layout" "slotLength" 'float 3.0)
```

Related Topics

Create Slot Form

Edit Synchronous Clone Form

Create Rectangle Form

Create Path Form

© 2023 Cadence Design Systems, Inc.

slotRowSpacing

```
layout slotRowSpacing float slotRowDistance
```

Description

Specifies the distance between two slot rows.

The default is `1.0`.

GUI Equivalent

Command:	*Create – Slot*, *Edit – Advanced – Slot*, *Create – Shape – Rectangle*, *Create – Shape – Path – Slot Options*
Field:	*Spacing: Row*
	Slot Spacing: Between Slot Rows

Examples

```
envGetVal("layout" "slotRowSpacing")
envSetVal("layout" "slotRowSpacing" 'float 2.0)
```

Related Topics

Create Slot Form

Edit Synchronous Clone Form

Create Rectangle Form

Create Path Form

© 2023 Cadence Design Systems, Inc.

slotShapeMode

```
layout slotShapeMode cyclic { "Preserve" | "Polygon" | "ROD" }
```

Description

Specifies the type of the resultant shape with slots.

The default is `Preserve`.

GUI Equivalent

Command:	*Create – Slot*, *Edit – Advanced – Slot*
Field:	*Shape Type*

Examples

```
envGetVal("layout" "slotShapeMode")
envSetVal("layout" "slotShapeMode" 'cyclic "Polygon")
envSetVal("layout" "slotShapeMode" 'cyclic "ROD")
```

Related Topics

Create Slot Form

Edit Synchronous Clone Form

© 2023 Cadence Design Systems, Inc.

slotSpecifyArea

```
layout slotSpecifyArea boolean ( t | nil )
```

Description

Selects the *Width/Area* option.

The default is `nil`.

GUI Equivalent

Command:	*Create – Slot*, *Edit – Advanced – Slot*, *Create – Shape – Rectangle*, *Create – Shape – Path – Slot Options*
Field:	*Specify width, area*
	Slot Dimensions: Specify: Width/Area

Examples

```
envGetVal("layout" "slotSpecifyArea")
envSetVal("layout" "slotSpecifyArea" 'boolean t)
```

Related Topics

Create Slot Form

Edit Synchronous Clone Form

Create Rectangle Form

Create Path Form

© 2023 Cadence Design Systems, Inc.

slotStaggerOffset

```
layout slotStaggerOffset float rowspacing
```

Description

Specifies the spacing between the edges of slots in alternate rows.

The default is `0.0`.

GUI Equivalent

Command:	*Create – Slot*, *Edit – Advanced – Slot*, *Create – Shape – Path – Slot Options*
Field:	*Stagger*
	Slot Offset: Stagger

Examples

```
envGetVal("layout" "slotStaggerOffset")
envSetVal("layout" "slotStaggerOffset" 'float 1.0)
```

Related Topics

Create Slot Form

Edit Synchronous Clone Form

Create Rectangle Form

Create Path Form

© 2023 Cadence Design Systems, Inc.

slotWarnWhenMoreThanRows

```
layout slotWarnWhenMoreThanRows int numberOfRows
```

Description

Displays a warning message in the CIW if you try to create a rectangle or a path with more slot rows than specified by this environment variable.

You can create a rectangle or a path with more slot rows than the number specified, but the slots do not appear in the drag set while creating the rectangle or path if the slot rows exceed the specified number.

The default is `1000`.

GUI Equivalent

Command:	*Create – Slot*, *Edit – Advanced – Slot*, *Create – Shape – Rectangle*, *Create – Shape – Path*
Field:	NA

Examples

```
envGetVal("layout" "slotWarnWhenMoreThanRows")
envSetVal("layout" "slotWarnWhenMoreThanRows" 'int 1500)
```

Related Topics

Create Slot Form

© 2023 Cadence Design Systems, Inc.

slotWidth

```
layout slotWidth float slotWidth
```

Description

Specifies the width of a slot.

The default is `1.0`.

GUI Equivalent

Command:	*Create – Slot*, *Edit – Advanced – Slot*, *Create – Shape – Rectangle*, *Create – Shape – Path – Slot Options*
Field:	*Dimensions: Width*
	Slot Dimensions: Width

Examples

```
envGetVal("layout" "slotWidth")
envSetVal("layout" "slotWidth" 'float 2.0)
```

Related Topics

Create Slot Form

Edit Synchronous Clone Form

Create Path Form

© 2023 Cadence Design Systems, Inc.

snapDeviceInDetailMode

```
layout snapDeviceInDetailMode boolean { t | nil }
```

Description

Snaps devices that do not have any property override, in detail mode. Also, detail mode snapping in horizontal direction is performed using shapes on `poly` layer only.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapDeviceInDetailMode")
envSetVal("layout" "snapDeviceInDetailMode" 'boolean nil)
```

Related Topics

Interactive Snapping

© 2023 Cadence Design Systems, Inc.

snapMode

```
layout snapMode string { "orthogonal" | "diagonal" | "anyAngle" | "L90XFirst" |
     "L90YFirst" }
graphic snapMode string { "orthogonal" | "diagonal" | "anyAngle" | "L90XFirst" |
     "L90YFirst" }
```

Description

Determines how *Create* command entry points snap to the grid.

The layout default is `orthogonal`, the graphic default is `anyAngle`.

Note: `anyAngle` is not a valid value for the *Create Wire* (XL, GXL) command.

GUI Equivalent

Command:	*Options – Display*
Field:	*Snap Modes: Create*

Examples

```
envGetVal("layout" "snapMode")
envSetVal("layout" "snapMode" 'string "diagonal")
envGetVal("graphic" "snapMode")
envSetVal("graphic" "snapMode" 'string "orthogonal")
envSetVal("graphic" "snapMode" 'string "L90YFirst")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

snapPatternDisplayTrackIndex

```
graphic snapPatternDisplayTrackIndex boolean { t | nil }
```

Description

Enables the `trackIndex` display packets set for specific track numbers to be used for display, based on the snap periods defined for a snap pattern definition.

The default value is `t`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "snapPatternDisplayTrackIndex")
envSetVal("graphic" "snapPatternDisplayTrackIndex" 'boolean nil)
```

This environment variable can also set be for a specific window as required. For example:

```
window = hiGetCurrentWindow()
window~>snapPatternDisplayTrackIndex = nil
```

Related Topics

Snap Pattern Display Based on Track Index

Changing Graphic Environment Variables That Are Cellview Properties

© 2023 Cadence Design Systems, Inc.

snapPatternDrawingStyle

```
layout snapPatternDrawingStyle cyclic { "full" | "tracks" | "boundary" | "periods"
     | "periodsAndTracks" }
```

Description

Determines whether you see the boundary, the tracks, the fins, and periods or the boundary and the tracks, or just the boundary for snap patterns, or periods, or periods and tracks for width spacing patterns.

The default is `full`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Snap Pattern Display*

Examples

```
envGetVal("layout" "snapPatternDrawingStyle")
envSetVal("layout" "snapPatternDrawingStyle" 'cyclic "tracks")
envSetVal("layout" "snapPatternDrawingStyle" 'cyclic "periodsAndTracks")
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

snapPatternFastDetailModeSnapping

```
layout snapPatternFastDetailModeSnapping boolean { t | nil }
```

Description

Prioritizes the snapping of grids at the higher level of hierarchy and snapping is done level by level by level.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapPatternFastDetailModeSnapping")
envSetVal("layout" "snapPatternFastDetailModeSnapping" 'boolean t)
```

Related Topics

Interactive Snapping

© 2023 Cadence Design Systems, Inc.

snapPatternGlobalGridAnchoring

```
graphic snapPatternGlobalGridAnchoring boolean { t | nil }
```

Description

Determines whether the snap pattern global grid anchoring is enabled.

This environment variable applies only to newly created cell views. Once a cellview is created, the setting of this environment variable is retained. To view or edit the setting after the cellview is created, you can use the functions, dbSetGlobalGridOffsetReferenceType and dbGetGlobalGridOffsetReferenceType.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "snapPatternGlobalGridAnchoring")
envSetVal("graphic" "snapPatternGlobalGridAnchoring" 'boolean t)
envSetVal("graphic" "snapPatternGlobalGridAnchoring" 'boolean nil)
```

Related Topics

dbSetGlobalGridOffsetReferenceType

dbGetGlobalGridOffsetReferenceType

Global Grid Anchoring

© 2023 Cadence Design Systems, Inc.

snapPatternIndexSnappingLayers

```
layout snapPatternIndexSnappingLayers string { "" | "t" | "layerNames" }
```

Description

Specifies whether the snap period-based snapping is enabled, and if enabled, the order of layers to be used for snapping. You can set any of the following values:

- `""`: Disables the snap period-based snapping. Instances can be snapped to any track of a grid.
- `t`: Enables the snap-period based snapping to be performed by using the default order of layers specified in the technology file. This is the default value.
- *`layerNames`*: Enables the snap period-based snapping to be performed by using the order of layers that you specify.

When snap period-based snapping is enabled, snapping is performed based on the snapGridPeriod constraint defined for a grid and the leSnapPatternSnappingHorizontalGridDetail or leSnapPatternSnappingVerticalGridDetail property defined for instances or the master cellview.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapPatternIndexSnappingLayers")
envSetVal("layout" "snapPatternIndexSnappingLayers" 'string "Active")
envSetVal("layout" "snapPatternIndexSnappingLayers" 'string "")
```

Related Topics

Detail Mode Snapping to Specific Tracks of a Global Grid

© 2023 Cadence Design Systems, Inc.

snapPatternSnapping

```
layout snapPatternSnapping boolean { t | nil }
```

Description

Determines whether the snap pattern snapping is enabled.

The default is `t`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Snap Pattern Snapping*

Examples

```
envGetVal("layout" "snapPatternSnapping")
envSetVal("layout" "snapPatternSnapping" 'boolean nil)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

snapPatternSnappingDepth

```
layout snapPatternSnappingDepth int snapDepth
```

Description

Lets you specify the snap depth for snap pattern snapping.

The default is `32`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapPatternSnappingDepth")
envSetVal("layout" "snapPatternSnappingDepth" 'int 20)
```

Related Topics

Interactive Snapping

© 2023 Cadence Design Systems, Inc.

snapPatternSnappingDepthMethod

```
layout snapPatternSnappingDepthMethod cyclic { "DisplayPlusOne" | "UserDefined" }
```

Description

Determines the method to be used to define the snap depth for snap pattern snapping.

The value display stop level + 1 is used as the snap depth when the default value `DisplayPlusOne` is used.

When set to `UserDefined`, the snap depth specified by the environment variable `snapPatternSnappingDepth` is used.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapPatternSnappingDepthMethod")
envSetVal("layout" "snapPatternSnappingDepthMethod" 'cyclic "UserDefined")
```

Related Topics

snapPatternSnappingDepth

© 2023 Cadence Design Systems, Inc.

snapPatternSnappingMode

```
layout snapPatternSnappingMode cyclic { "detail" | "abstract" }
```

Description

Determines whether the instance placement mode is `detail` or `abstract`.

The default is `detail`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapPatternSnappingMode")
envSetVal("layout" "snapPatternSnappingMode" 'cyclic "abstract")
```

Related Topics

Abstract Mode Placement

© 2023 Cadence Design Systems, Inc.

snapPatternSnapToXYGridFirst

```
layout snapPatternSnapToXYGridFirst boolean { t | nil }
```

Description

Enables snapping to the snap pattern grids closest to the X and Y snap spacing values.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapPatternSnapToXYGridFirst")
envSetVal("layout" "snapPatternSnapToXYGridFirst" 'boolean t)
```

Related Topics

Interactive Snapping

© 2023 Cadence Design Systems, Inc.

snapRectToWSPGrid

```
layout snapRectToWSPGrid boolean { t | nil }
```

Description

Snaps rectangles to the width spacing pattern grid.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapRectToWSPGrid")
envSetVal("layout" "snapRectToWSPGrid" 'boolean t)
```

Related Topics

Interactive Snapping

© 2023 Cadence Design Systems, Inc.

snapResizeWireObjects

```
layout snapResizeWireObjects boolean { t | nil }
```

Description

Snaps, resizes, and colors all pathsegs, vias, and pins in the selection set to the active width spacing pattern during the *Move*, *Copy*, and *Flip* commands.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapResizeWireObjects")
envSetVal("layout" "snapResizeWireObjects" 'boolean t)
```

Related Topics

Move Form

Copy Form

Flip Form

snapResizeWireObjectsMaxDragLimit

snapResizeWireObjectsMaxLimit

© 2023 Cadence Design Systems, Inc.

snapResizeWireObjectsMaxDragLimit

```
layout snapResizeWireObjectsMaxDragLimit int dragLimit
```

Description

Specifies the number of wire objects from the selection set that can be shown as being snapped and resized interactively during the drag operation in *Move* and *Copy* commands. The specified value is effective only when `snapResizeWireObjects` is enabled.

The default value is `100`. Setting a larger value could lead to poor drag performance.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapResizeWireObjectsMaxDragLimit")
envSetVal("layout" "snapResizeWireObjectsMaxDragLimit" 'int 80)
```

Related Topics

snapResizeWireObjects

© 2023 Cadence Design Systems, Inc.

snapResizeWireObjectsMaxLimit

```
layout snapResizeWireObjectsMaxLimit int maxLimit
```

Description

Specifies the maximum number of wire objects from the selection set that can be snapped and resized during the *Move* and *Copy* command operations. The specified value is effective only when `snapResizeWireObjects` is enabled.

The default value is `5000`. Setting a larger value could lead to poor performance.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapResizeWireObjectsMaxLimit")
envSetVal("layout" "snapResizeWireObjectsMaxLimit" 'int 2000)
```

Related Topics

snapResizeWireObjects

© 2023 Cadence Design Systems, Inc.

snapStdCellInDetailMode

```
layout snapStdCellInDetailMode boolean { t | nil }
```

Description

Snaps a standard cell in detail mode.

The default is `nil`, which snaps a standard cell in abstract mode if it has a PR boundary.

A standard cell is a cell whose component type is `STDCELL`, `BOUNDARYCELL`, `STDSUBCONT`, or `FILLER`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapStdCellInDetailMode")
envSetVal("layout" "snapStdCellInDetailMode" 'boolean t)
```

Related Topics

Instance Placement Modes

© 2023 Cadence Design Systems, Inc.

snapToGrid

```
layout snapToGrid boolean { t | nil }
graphic snapToGrid boolean { t | nil }
```

Description

Snaps the top-level and soft block pins to a grid. It also snaps the place and route (PR) boundary of the top-level and soft block instances to a grid. The snapping of pins and PR boundary is dependent on the block type, and the snapping works when the display level is greater than zero.

If the block is an analog block, the PR boundary and pin edges are snapped to the manufacturing grid. For a digital block, the PR boundary is snapped to the placement grid and the pin centers are snapped to the routing grid. The snapping is supported for both virtual and non-virtual pins. Pins with placement status `unplaced`, `fixed`, `locked`, or `unknown` are not snapped even if they are touching the boundary.

The default value is `nil`.

GUI Equivalent

Command:	*Edit – Chop*
	Edit – Copy
	Edit – Move
	Edit – Stretch
	Edit – Advanced – Reshape
Field:	*Snap – Pins and boundary to grid*

Examples

```
envGetVal("layout" "snapToGrid")
envSetVal("layout" "snapToGrid" 'boolean t)
envGetVal("graphic" "snapToGrid")
envSetVal("graphic" "snapToGrid" 'boolean t)
```

Related Topics

Chop Form

© 2023 Cadence Design Systems, Inc.

Copy Form

Move Form

Stretch Form

Reshape Form

© 2023 Cadence Design Systems, Inc.

snapToMfgGrid

```
layout snapToMfgGrid boolean { t | nil }
```

Description

Snaps the converted polygon vertices to mfgGrid.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapToMfgGrid")
envSetVal("layout" "snapToMfgGrid" 'boolean nil)
```

Related Topics

mfgGridResolution

© 2023 Cadence Design Systems, Inc.

snapUsingInstanceAsReference

```
layout snapUsingInstanceAsReference boolean { t | nil }
```

Description

Selects an instance as the reference object for snapping when the selection set has combination of wires and instances.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "snapUsingInstanceAsReference")
envSetVal("layout" "snapUsingInstanceAsReference" 'boolean t)
```

Related Topics

Object Snapping

© 2023 Cadence Design Systems, Inc.

softFence

```
layout softFence boolean { t | nil }
```

Description

Determines whether or not to create the area boundary as a soft fence.

The default is `nil`.

GUI Equivalent

Command:	*Create – P&R Objects – Area Boundary*
Field:	*Soft Fence*

Examples

```
envGetVal("layout" "softFence")
envSetVal("layout" "softFence" 'boolean t)
```

Related Topics

Create Area Boundary Form

© 2023 Cadence Design Systems, Inc.

spineSelection

```
graphic spineSelection boolean { t | nil }
```

Description

Determines whether any additional contiguous pathsegs are selected, when a pathseg is selected.

The default is `t`.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Selection Controls: Spine*

Examples

```
envGetVal("graphic" "spineSelection")
envSetVal("graphic" "spineSelection" 'boolean nil)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

startLevel

```
layout startLevel int highestLevel
graphic startLevel int highestLevel
```

Description

Specifies the number of the highest level in the hierarchy visible. Enter any integer from 0 to 32.

The default is `0`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Display Levels: Start*

Examples

```
envGetVal("layout" "startLevel")
envSetVal("layout" "startLevel" 'int 10)
envGetVal("graphic" "startLevel")
envSetVal("graphic" "startLevel" 'int 10)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

statusBarXYdXdYLength

```
layout statusBarXYdXdYLength int width
```

Description

Controls the width of the *X*, *Y*, *dX*, and *dY* fields on the status toolbar.

The default is `10`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "statusBarXYdXdYLength")
envSetVal("layout" "statusBarXYdXdYLength" 'int 5)
```

Related Topics

Status Toolbar

© 2023 Cadence Design Systems, Inc.

statusToolbarFields

```
layout statusToolbarFields string { "(SelectorInformation)" }
```

Description

Specifies the information that you want to display on the status toolbar. By using selectors, you can display information such as the cellview size, count of selected objects, and values of environment variables, shell environment variables, and window variables.

The format of the string is a series of selectors, where each selector is listed in parenthesis. Use the following format to define *SelectorInformation* for each selector in the series.

(*selector tooltip style_format*)

For each selector, you can optionally specify a tooltip and the style for displaying the information. You can also use expressions for conditional formatting of the information. For information about specifying tooltips, display style, and conditional formatting for selectors, see Status Toolbar Customization.

The following table describes the selectors that you can use to display information on the status toolbar.

Selector	Description
X	Displays the X coordinate of the mouse pointer.
Y	Displays the Y coordinate of the mouse pointer.
dX	Displays the dX coordinate of the mouse pointer from the last reference point.
dY	Displays the dY coordinate of the mouse pointer from the last reference point.
EXYDXDY	Displays the editable group of X, Y, dX, and dY coordinates.
Cmd	Displays the name of the currently active command such as Move or Copy.
CVSize	Displays the width and height of the cellview.
Dist	Displays the distance from the last reference.
DRDNotify	Displays whether DRD Notify mode is on or off.
DRDEnforce	Displays whether DRD Enforce mode is on or off.
DRDPost-Edit	Displays whether DRD Post-Edit mode is on or off.

© 2023 Cadence Design Systems, Inc.

Selector	Description
`Mode`	Displays the selection mode as full (F) or partial (P).
`PreSelSelObjInfo`	Displays the selection or pre-selection information about objects.
`Select`	Displays the count of selected objects.
`SelC`	Displays the count of selectable objects under the cursor in the format *`current/total`*. For example, 2/4 implies second of four overlapping objects.
`SelI`	Displays the selection count of instances.
`SelO`	Displays the selection count of objects.
`SelN`	Displays the selection count of nets.
`T`	Displays a fixed text string on the toolbar.
`Tpt`	Displays the status of transparent group mode as `t` or blank.
`E`	Displays the value of the specified environment variable. You need to specify the data type as per the variable.
`S`	Displays the value of the specified SKILL variable. You need to specify the data type as per the variable.
`SH`	Displays the value of the specified shell variable.
`W`	Displays the value of the specified window variable. You need to specify the data type as per the variable.

The default is `""`, which means that the predefined set of fields is displayed on the toolbar.

GUI Equivalent

None

Examples

```
envGetVal("layout" "statusToolbarFields")
```

The following command displays X and Y coordinates on the status toolbar:

```
envSetVal("layout" "statusToolbarFields" 'string "(X) (Y)")
```


© 2023 Cadence Design Systems, Inc.

The following command displays X and Y coordinates, with `X` as the tooltip for the X coordinate and `Y Coordinate` as the tool tip for the Y coordinate.

```
envSetVal("layout" "statusToolbarFields" 'string "(X t) (Y \"Y Coordinate\")")
```

The following command displays the size of the cellview, editable X, Y, dX, and dY fields, name of the current command, selection count, count of selectable objects under the cursor, and selection information:

```
envSetVal("layout" "statusToolbarFields" 'string "(CVSize \"Boundary size\") \\|
(EXYDXDY) \\| (Cmd t (V == \"\" \"Idle\") (V != \"\" \"<b>%s</b>\")) \\|  (Select
t (V==\"0\" \"s: %s\")(V!=\"0\" \"<span style=\\\"color:red;font-
weight:bold\\\">s: %s</span>\")) \\| (SelC \"Object under cursor\" \"C/T: %s\") \\|
(PreSelSelObjInfo \"Selected object info\")")
```

Related Topics

Status Information

© 2023 Cadence Design Systems, Inc.

stopLevel

```
layout stopLevel int lowestLevel
graphic stopLevel int lowestLevel
```

Description

Specifies the number of the lowest level in the hierarchy visible. Enter any integer from 0 to 32.

The default is `0`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Display Levels: Stop*

Examples

```
envGetVal("layout" "stopLevel")
envSetVal("layout" "stopLevel" 'int 10)
envGetVal("graphic" "stopLevel")
envSetVal("graphic" "stopLevel" 'int 10)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

stretchPcellApplyToName

```
layout stretchPcellApplyToName boolean { t | nil }
```

Description

Determines whether the *Reference Dimension* and *Minimum and Maximum* fields apply to Name only or to the whole expression for Pcells, in the Stretch in X and Stretch in Y forms.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "stretchPcellApplyToName")
envSetVal("layout" "stretchPcellApplyToName" 'boolean nil)
```

Related Topics

Stretch Form

© 2023 Cadence Design Systems, Inc.

stretchHandlesLayer

```
graphic stretchHandlesLayer string "any_valid_layer-pupose_pair"
```

Description

Specifies the layer that stretch handles are displayed on for stretchable Pcells.

The default is `y0 drawing`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "stretchHandlesLayer")
envSetVal("graphic" "stretchHandlesLayer" 'string "metal1 drawing")
```

Related Topics

Stretch Form

© 2023 Cadence Design Systems, Inc.

stretchViaMode

```
layout stretchViaMode cyclic { "Stretch Row/Col" | "Stretch Metal" }
```

Description

Specifies the mode of stretching of the via.

The default is `Stretch Metal`.

GUI Equivalent

Command:	*Edit – Stretch*
Field:	*Special Objects: Via Mode*

Examples

```
envGetVal("layout" "stretchViaMode")
envSetVal("layout" "stretchViaMode" 'cyclic "Stretch Row/Col")
```

Related Topics

Stretch Form

© 2023 Cadence Design Systems, Inc.

supressFreezePcellUI

```
layout supressFreezePcellUI boolean { t | nil }
```

Description

Freezes the Pcell instance without displaying the Freeze Pcell Instance Form. The default values for library, cell, and view are used for the freeze operation.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "supressFreezePcellUI")
envSetVal("layout" "supressFreezePcellUI" 'boolean t)
```

Related Topics

Freeze Pcell Instance Form

© 2023 Cadence Design Systems, Inc.

tapConnectedFigsOnly

```
layout tapConnectedFigsOnly boolean { t | nil }
```

Description

Ignores the figures that are not associated with a net during tapping of an object.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "tapConnectedFigsOnly")
envSetVal("layout" "tapConnectedFigsOnly" 'boolean t)
```

Related Topics

Object Tapping

© 2023 Cadence Design Systems, Inc.

taperStyle

```
layout taperStyle cyclic { "unknown" | "linear" | "exponential" }
```

Description

Specifies the default type of taper

The default is `linear`.

GUI Equivalent

Command:	*Create – Microwave– Taper*
Field:	*Taper Style*

Examples

```
envGetVal("layout" "taperStyle")
envSetVal("layout" "taperStyle" 'cyclic "unknown")
envSetVal("layout" "taperStyle" 'cyclic "exponential")
```

Related Topics

Transmission Line Taper Form

© 2023 Cadence Design Systems, Inc.

tapLayerFilter

`layout tapLayerFilter string "`*`list_of_any_valid_layers`*`"`

Description

Specifies the filter layers to be tapped. The layers names must be separated by a space.and enclosed in quotation marks; for example, `"metal1 metal2"`

The default is `""`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "tapLayerFilter")
envSetVal("layout" "tapLayerFilter" 'string "metal1 metal2")
```

Related Topics

Find Marker Form

© 2023 Cadence Design Systems, Inc.

tapPurposeList

```
layout tapPurposeList string "list_of_any_valid_layer_purposes"
```

Description

Prioritizes the order in which tap cycles through layers. When you set Wire Auto Tap or Shape Auto Tap to on, and *Select from Overlaps* to off, the tap purpose list is used to prioritize the order in which layers are cycled through during tap. When Wire Auto Tap and Shape Auto Tap are set to off and *Select from Overlaps* is on, the *Choose object to tap from* form appears, so the Tap Purpose List is not considered.

The default is `drawing`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Tap Purpose List*

Examples

```
envGetVal("layout" "tapPurposeList")
envSetVal("layout" "tapPurposeList" 'string "metal1 drawing")
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

tapType

```
layout tapType string { "layer" | "net" | "attribute" }
```

Description

Sets what information is obtained from tapping.

- `layer`. Default. The tap command sets the entry layer to the layer of the object on which you have tapped.
- `net`: The tap command captures the net names of rectangles, polygons and paths.
- `attribute`: The tap command captures the path net name, beginning and ending extension in user units, the path end type, and the path width, for trls, bends and tapers the tap command captures the bend factor (maximum width before they are chamfered), taper style (linear or exponential), number of points to taper in user units, the bend style (bend chamfer or radial), the width of a chamfered bend in a transmission line as a multiple of its standard width in user units, the number of segments in a radial bend in a transmission line as a function of the number of segments in a 180-degree radial bend, the radius of a radial bend in a transmission line as a multiple of the transmission-line width in user units, the width of the first end of a transmission line bend in user units, the width of the second end of a transmission line bend, and the width of the second end of a transmission line bend in user units.

Tapping of nets is only supported in the layout editor XL and GXL applications.

GUI Equivalent

Command:	*Tools – Tap*
Field:	*Layer, Net, Attributes*

Examples

```
envGetVal("layout" "tapType")
envSetVal("layout" "tapType" 'string "net")
envSetVal("layout" "tapType" 'string "attribute")
```

Related Topics

Tap Form

© 2023 Cadence Design Systems, Inc.

targetColor

```
layout targetColor string "color"
```

Description

Controls the color of the snap target highlight while creating a ruler.

The default is `yellow`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Snap Target Color*

Examples

```
envGetVal("layout" "targetColor")
envSetVal("layout" "targetColor" 'string "white pink red green")
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

targetLineStyle

```
layout targetLineStyle string "lineStyle"
```

Description

Controls the style of the snap target highlight while creating a ruler.

Line style options include `dashed`, `dots`, `dashDot`, `shortDash`, `doubleDash`, `thickLine`, and `thickDash`.

The default is `thickLine`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Snap Target Line Style*

Examples

```
envGetVal("layout" "targetLineStyle")
envSetVal("layout" "targetLineStyle" 'string "dashed")
envSetVal("layout" "targetLineStyle" 'string "solid")
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

targetWidth

`layout targetWidth float` *`snapTargetWidth`*

Description

Controls the width of the snap target highlight while creating a ruler. The value must be a integer point number.

The default is `3.0`.

GUI Equivalent

Command:	*Tools – Create Measurement*
Field:	*Snap Target Width*

Examples

```
envGetVal("layout" "targetWidth")
envSetVal("layout" "targetWidth" 'float 5.0)
```

Related Topics

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

termDirection

```
layout termDirection cyclic { "input" | "output" | "inputOutput" | "switch" |
     "jumper" | "unused" | "tristate" }
```

Description

Specifies the default I/O type of a pin.

The default is `inputOutput`.

GUI Equivalent

Command:	*Create – Pin*
Field:	*I/O Type*

Examples

```
envGetVal("layout" "termDirection")
envSetVal("layout" "termDirection" 'cyclic "switch")
envSetVal("layout" "termDirection" 'cyclic "jumper")
envSetVal("layout" "termDirection" 'cyclic "tristate")
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

termName

```
layout termName string "any_terminal_name"
```

Description

Specifies one or more names for terminals on which pins are created. You can specify as many names as you want (separated by spaces). As each name is used, it is removed from the list. If all names are used, the field in the Pin form becomes empty, and you cannot create any more pins until you specify a terminal name.

The default is `none`.

GUI Equivalent

Command:	*Create – Pin*
Field:	*Terminal Names*

Examples

```
envGetVal("layout" "termName")
envSetVal("layout" "termName" 'string "term1")
```

Related Topics

Create Pin Form

© 2023 Cadence Design Systems, Inc.

textJustificationOn

```
layout textJustificationOn boolean { t | nil }
graphic textJustificationOn boolean { t | nil }
```

Description

Determines whether the origins of labels are displayed. The origins appear as plus signs.

The default is `t`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Label Origins*

Examples

```
envGetVal("layout" "textJustificationOn")
envSetVal("layout" "textJustificationOn" 'boolean nil)
envGetVal("graphic" "textJustificationOn")
envSetVal("graphic" "textJustificationOn" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

tinyInstDetail

```
layout tinyInstDetail boolean { t | nil }
graphic tinyInstDetail boolean { t | nil }
```

Description

Determines whether to display all the details of an instance, even those smaller than the size set in the `displayResolution` environment variable.

The cellview redraws faster when the default value `nil` is used.

GUI Equivalent

None

Examples

```
envGetVal("layout" "tinyInstDetail")
envSetVal("layout" "tinyInstDetail" 'boolean t)
envGetVal("graphic" "tinyInstDetail")
envSetVal("graphic" "tinyInstDetail" 'boolean t)
```

Related Topics

displayResolution

© 2023 Cadence Design Systems, Inc.

transientRulerColorPacketName

```
graphic transientRulerColorPacketName string "packetName"
```

Description

Enables you to specify the name of the packet which has the `Outline` attribute with the color information for the non-savable ruler.

You can add a new packet with the required attributes or modify attributes of a packet in the `display.drf` file or through the *Display Resource Editor*.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "transientRulerColorPacketName")
envSetVal("graphic" "transientRulerColorPacketName" 'string "ref_drawing")
```

Related Topics

Display Resource Editor

© 2023 Cadence Design Systems, Inc.

transparentGroup

```
layout transparentGroup boolean { t | nil }
```

Description

Allows editing of groups without EIP into group. The variable setting is applicable only for the current window.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Transparent Group*

Examples

```
envGetVal("layout" "transparentGroup")
envSetVal("layout" "transparentGroup" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

trimEnable

```
layout trimEnable boolean { t | nil }
graphic trimEnable boolean { t | nil }
```

Description

Hides the layer that is cut using the `Trim` command.

The default is `nil`.

GUI Equivalent

Command:	*Options – Display*
Field:	*Trim Shapes*

Examples

```
envGetVal("layout" "trimEnable")
envSetVal("layout" "trimEnable" 'boolean t)
envGetVal("graphic" "trimEnable")
envSetVal("graphic" "trimEnable" 'boolean t)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

trueColorProbe

```
graphic trueColorProbe boolean { t | nil }
```

Description

Determines the drawing mode for the next probe.

The default is `nil`.

GUI Equivalent

Command:	*Options – Highlight*
Field:	*Drawing Mode*

Examples

```
envGetVal("graphic" "trueColorProbe")
envSetVal("graphic" "trueColorProbe" 'boolean t)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

trlBendFac

```
layout trlBendFac float maximumWidth
```

Description

Specifies the maximum width (as a multiple of its standard width) that standard bends in a transmission line can achieve before they are automatically chamfered. This attribute is used only for bend, style bend.

The default is `1.0`.

GUI Equivalent

Command:	*Create – Microwave – Trl, Create – Microwave – Bend*
Field:	*Bend Factor*

Examples

```
envGetVal("layout" "trlBendFac")
envSetVal("layout" "trlBendFac" 'float 3.0)
```

Related Topics

Transmission Line Form

Transmission Line Bend Form

© 2023 Cadence Design Systems, Inc.

trlBendStyle

```
layout trlBendStyle cyclic { "bend" | "chamfer" | "radial" }
```

Description

Specifies the types of bends used in creating a transmission line.

The default is `bend`.

GUI Equivalent

Command:	*Create – Microwave – Trl, Create – Microwave – Bend*
Field:	*Bend Style*

Examples

```
envGetVal("layout" "trlBendStyle")
envSetVal("layout" "trlBendStyle" 'cyclic "chamfer")
envSetVal("layout" "trlBendStyle" 'cyclic "radial")
```

Related Topics

Transmission Line Form

Transmission Line Bend Form

© 2023 Cadence Design Systems, Inc.

trlChamFac

```
layout trlChamFac float width
```

Description

Specifies the width of a chamfered bend in a transmission line as a multiple of its standard width. This attribute is used only for bend, style chamfer.

The default is `0.6`.

GUI Equivalent

Command:	*Create – Microwave – Trl, Create – Microwave – Bend*
Field:	*Chamfer Factor*

Examples

```
envGetVal("layout" "trlChamFac")
envSetVal("layout" "trlChamFac" 'float 0.8)
```

Related Topics

Transmission Line Form

Transmission Line Bend Form

© 2023 Cadence Design Systems, Inc.

trlNPtsToPI

```
layout trlNPtsToPI int numberOfSegments
```

Description

Specifies the number of segments in a radial bend in a transmission line as a function of the number of segments in a 180-degree radial bend.

The default is `20`.

GUI Equivalent

Command:	*Create – Microwave – Trl, Create – Microwave – Bend, Create – Microwave – Taper*
Field:	*Resolution*

Examples

```
envGetVal("layout" "trlNPtsToPI")
envSetVal("layout" "trlNPtsToPI" 'int 30)
```

Related Topics

Transmission Line Form

Transmission Line Bend Form

Transmission Line Taper Form

© 2023 Cadence Design Systems, Inc.

trlRadFac

```
layout trlRadFac float radius
```

Description

Specifies the radius of a radial bend in a transmission line as a multiple of the transmission-line width. This attribute is used only for bend-style radial.

The default is `3.0` user units.

GUI Equivalent

Command:	*Create – Microwave – Trl, Create – Microwave – Bend*
Field:	*Radius Factor*

Examples

```
envGetVal("layout" "trlRadFac")
envSetVal("layout" "trlRadFac" 'float 5.0)
```

Related Topics

Transmission Line Form

Transmission Line Bend Form

Transmission Line Taper Form

© 2023 Cadence Design Systems, Inc.

trlWidth1

```
layout trlWidth1 float width
```

Description

Specifies the width of the first end of a transmission line bend.

The default is `20` user units.

GUI Equivalent

Command:	*Create – Microwave – Trl, Create – Microwave – Bend, Create – Microwave – Taper*
Field:	*Width*
	Side 1 Width

Examples

```
envGetVal("layout" "trlWidth1")
envSetVal("layout" "trlWidth1" 'float 30.0)
```

Related Topics

Transmission Line Form

Transmission Line Bend Form

Transmission Line Taper Form

© 2023 Cadence Design Systems, Inc.

trlWidth2

```
layout trlWidth2 float width
```

Description

Specifies the width of the second end of a transmission line bend.

The default is `20` user units.

GUI Equivalent

Command:	*Create – Microwave – Bend, Create – Microwave – Taper*
Field:	*Side 2 Width*

Examples

```
envGetVal("layout" "trlWidth2")
envSetVal("layout" "trlWidth2" 'float 30.0)
```

Related Topics

Transmission Line Bend Form

Transmission Line Taper Form

© 2023 Cadence Design Systems, Inc.

turnInfixOffWhenSmartSnapping

```
layout turnInfixOffWhenSmartSnapping boolean { t | nil }
```

Description

Controls the infix mode when commands that use smart snapping are active.

The default is `t`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Turn Off Infix in Smart Snapping*

Examples

```
envGetVal("layout" "turnInfixOffWhenSmartSnapping")
envSetVal("layout" "turnInfixOffWhenSmartSnapping" 'boolean nil)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

undoRedoHistoryLimit

```
layout undoRedoHistoryLimit int numberOfCommands
```

Description

Specifies the number of commands displayed in the drop-down menu of *Undo* and *Redo* commands. Valid range is `5` to `20`.

The default is `10`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "undoRedoHistoryLimit")
envSetVal("layout" "undoRedoHistoryLimit" 'int 20)
```

Related Topics

Basic Command Functions in Layout XL

© 2023 Cadence Design Systems, Inc.

updateLabelLPPWithPin

```
layout updateLabelLPPWithPin boolean { t | nil }
```

Description

Specifies that whenever the LPP of a pin is changed to another LPP, the label attached to that pin changes to the new LPP.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "updateLabelLPPWithPin")
envSetVal("layout" "updateLabelLPPWithPin" 'boolean t)
```

Related Topics

Pin Creation

© 2023 Cadence Design Systems, Inc.

updatePCellIncrement

`layout updatePCellIncrement float` *`everyGridSnap`*

Description

Specifies how often the system updates Pcell parameters and regenerates a Pcell during a stretch operation. Grid snap is as defined by the technology file variable `mfgGridResolution`.

The default is `-1.0`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "updatePCellIncrement")
envSetVal("layout" "updatePCellIncrement" 'float -2.0)
```

Related Topics

Stretch Form

© 2023 Cadence Design Systems, Inc.

useAlphaBlending

```
graphic useAlphaBlending boolean { t | nil }
```

Description

Determines whether alpha blending is enabled.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "useAlphaBlending")
envSetVal("graphic" "useAlphaBlending" 'boolean nil)
```

Related Topics

Controlling Transparency of Tracks With Alpha Blending

© 2023 Cadence Design Systems, Inc.

useDynamicHilightPacket

```
graphic useDynamicHilightPacket boolean { t | nil }
layout useDynamicHilightPacket boolean { t | nil }
```

Description

Creates the dynamic highlight display packet and uses it to draw dynamic highlights in the design.

The default is `t` for layout, `nil` for others.

When set to `nil`, the dynamic highlight is drawn with a non-configurable thin dashed outline.

When set to `t`, and you do not have the dynamicHilight display packet defined in the display.drf file, then the packet is created internally on the first dynamically highlighted figure. However, this packet is not saved automatically in display.drf. Therefore, if you close Virtuoso at this stage, display.drf is not modified, and next time, you open Virtuoso and then DRE, the dynamicHilight display packet will not be displayed.

However, if you open DRE, save display.drf, and then, close Virtuoso, the next time you will open Virtuoso and DRE, the dynamicHilight display packet will be present, even if environment variable is set to `nil`.

GUI Equivalent

None

Examples

```
envGetVal("graphic" "useDynamicHilightPacket")
envSetVal("graphic" "useDynamicHilightPacket" 'boolean t)
envGetVal("layout" "useDynamicHilightPacket")
envSetVal("layout" "useDynamicHilightPacket" 'boolean nil)
```

Related Topics

Highlight Options Form

© 2023 Cadence Design Systems, Inc.

useFlipReference

```
layout useFlipReference boolean { t | nil }
```

Description

Activates the options in the Flip form for specifying the reference and the direction used for flipping an object.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Flip – Flip*
Form Field:	*Specify reference and direction*

Examples

```
envGetVal("layout" "useFlipReference")
envSetVal("layout" "useFlipReference" 'boolean t)
```

Related Topics

Flip Form

Flipping Objects

© 2023 Cadence Design Systems, Inc.

useOverrideFontName

```
layout useOverrideFontName boolean { t | nil }
graphic useOverrideFontName boolean { t | nil }
```

Description

Determines whether to override the default font name for a window.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "useOverrideFontName")
envSetVal("layout" "useOverrideFontName" 'boolean t)
envGetVal("graphic" "useOverrideFontName")
envSetVal("graphic" "useOverrideFontName" 'boolean t)
```

Related Topics

overrideFontName

© 2023 Cadence Design Systems, Inc.

useRelativeXY

```
layout useRelativeXY boolean { t | nil }
```

Description

Indicates that the points in the *XY* field are the points relative to the last entered points of the command (delta X and Y).

The default is `nil`.

GUI Equivalent

Command:	*Tools – Enter Points*
Field:	*Use Relative XY*

Examples

```
envGetVal("layout" "useRelativeXY")
envSetVal("layout" "useRelativeXY" 'boolean t)
```

Related Topics

Enter Points Form

© 2023 Cadence Design Systems, Inc.

useRotateReference

```
layout useRotateReference boolean { t | nil }
```

Description

Activates the options in the Rotate form for specifying the reference settings used for rotating an object.

The default is `nil`.

GUI Equivalent

Command:	*Edit – Rotate – Rotate*
Form Field:	*Specify reference and angle*

Examples

```
envGetVal("layout" "useRotateReference")
envSetVal("layout" "useRotateReference" 'boolean t)
```

Related Topics

Rotate Form

Object Chopping

© 2023 Cadence Design Systems, Inc.

useRulerSnapModes

```
layout useRulerSnapModes boolean { t | nil }
```

Description

Defines whether snap modes specific to the ruler command are used when you invoke the command.

When set to `t`, ruler snap modes set in the `rulerSnapMode` and `rulerSegSnapMode` environment variables are used, and window snap modes are restored when you exit the ruler command. When set to `nil`, window snap modes are used for the ruler.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "useRulerSnapModes")
envSetVal("layout" "useRulerSnapModes" 'boolean t)
```

Related Topics

rulerSnapMode

rulerSegSnapMode

Create Measurement Form

© 2023 Cadence Design Systems, Inc.

useTapPurposeList

```
layout useTapPurposeList boolean { t | nil }
```

Description

Prioritizes the order in which tap cycles through layers.

When you set *Wire Auto Tap* or *Shape Auto Tap* to on, and *Select from Overlaps* to off, the tap purpose list is used to prioritize the order in which layers are cycled through during tap.

When *Wire Auto Tap* and *Shape Auto Tap* are set to off and *Select from Overlaps* is on, the *Choose object to tap from* form appears, so the Tap Purpose List is not considered.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "useTapPurposeList")
envSetVal("layout" "useTapPurposeList" 'boolean nil)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

useTrueBBox

```
layout useTrueBBox boolean { t | nil }
graphic useTrueBBox boolean { t | nil }
```

Description

Specifies how instance outlines at the display stop level are determined for display.

When set to `t`, one more level of hierarchy opens. If the instance's master contains a shape on the layer purpose "`instance`" "`drawing`", its bounding box displays as the outline. If the instance contains no shape on the layer purpose "`instance`" "`drawing`", the instance's master bounding box displays. Opening the master ensures an accurate display of the outline, but can affect loading and display time.

When set to `nil`, the current instance bounding box displays without opening any additional hierarchy. This enables large designs to open and display much faster.

GUI Equivalent

Command:	*Options – Display*
Field:	*Display Controls: Use True BBox*

Examples

```
envGetVal("layout" "useTrueBBox")
envSetVal("layout" "useTrueBBox" 'boolean t)
envGetVal("graphic" "useTrueBBox")
envSetVal("graphic" "useTrueBBox" 'boolean nil)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

validLppFilterOn

```
layout validLppFilterOn boolean { t | nil }
```

Description

Specifies that only valid layers displayed in the *Layers* assistant are populated in the layer field of the *Property Editor*, *Move*, *Copy*, and *Label* forms.

The default is `nil`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Use Layer Palette LPPs Only*

Examples

```
envGetVal("layout" "validLppFilterOn")
envSetVal("layout" "validLppFilterOn" 'boolean t)
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

viaAlignCuts

```
layout viaAlignCuts boolean { t | nil }
```

Description

Specifies the alignment of cuts between different tiles of a polygonal overlap.

The default is `t`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Align Cuts*

Examples

```
envGetVal("layout" "viaAlignCuts")
envSetVal("layout" "viaAlignCuts" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAlignCuts

```
layout.ava viaAlignCuts boolean { t | nil }
```

Description

Specifies the alignment of cuts between different tiles of a polygonal overlap.

When this option is enabled, cuts between different tiles of a polygonal overlap are aligned. When this option is disabled, the polygonal overlap is decomposed into a set of largest tiles. Vias are created independently in each tile, which may result in cuts not being aligned between different tiles.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant Options*
Field:	*Align cuts*

Examples

```
envGetVal("layout.ava" "viaAlignCuts")
envSetVal("layout.ava" "viaAlignCuts" 'boolean t)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaAllowOverridePurposes

```
layout viaAllowOverridePurposes boolean { t | nil }
```

Description

Specifies that the purpose defined in cdsGenViaDef or cdsViaDef in the technology file is used and cannot be changed by the user or by the viaEngine when set to `nil`.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaAllowOverridePurposes")
envSetVal("layout" "viaAllowOverridePurposes" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAnnotateOverlapsWithNoValidSolution

```
layout.ava viaAnnotateOverlapsWithNoValidSolution boolean { t | nil }
```

Description

Creates annotations on overlaps for which no via is generated.

The default is `t`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – Annotations*
Field:	*Overlaps with no solution*

Examples

```
envGetVal("layout.ava" "viaAnnotateOverlapsWithNoValidSolution")
envSetVal("layout.ava" "viaAnnotateOverlapsWithNoValidSolution" 'boolean nil)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAnnotateViasNotRetained

```
layout.ava viaAnnotateViasNotRetained boolean { t | nil }
```

Description

Creates annotations on overlaps on which vias have not been created because they add a metal or create DRC violations.

The default is `t`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – Annotations*
Field:	*Vias not retained*

Examples

```
envGetVal("layout.ava" "viaAnnotateViasNotRetained")
envSetVal("layout.ava" "viaAnnotateViasNotRetained" 'boolean nil)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAnnotateViasWithAddedMetal

```
layout.ava viaAnnotateViasWithAddedMetal boolean { t | nil }
```

Description

Creates annotations on vias that add metal to the design.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – Annotations*
Field:	*Vias adding metal*

Examples

```
envGetVal("layout.ava" "viaAnnotateViasWithAddedMetal")
envSetVal("layout.ava" "viaAnnotateViasWithAddedMetal" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAnnotateViasWithNoAddedMetal

```
layout.ava viaAnnotateViasWithNoAddedMetal boolean { t | nil }
```

Description

Creates annotations on vias that do not add metal to the design.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – Annotations*
Field:	*Vias not adding metal*

Examples

```
envGetVal("layout.ava" "viaAnnotateViasWithNoAddedMetal")
envSetVal("layout.ava" "viaAnnotateViasWithNoAddedMetal" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAnnotateViasWithSingleCut

```
layout.ava viaAnnotateViasWithSingleCut boolean { t | nil }
```

Description

Creates annotations on overlaps for which no via is generated.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – Annotations*
Field:	*Overlaps with no solution*

Examples

```
envGetVal("layout.ava" "viaAnnotateViasWithSingleCut")
envSetVal("layout.ava" "viaAnnotateViasWithSingleCut" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAttemptToUseBridgeVias

```
layout.ava viaAttemptToUseBridgeVias boolean { t | nil }
```

Description

Places a bridge via according to the topology. If a bridge via is invalid, a basic via is created.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Optimization*
Field:	*Attempt to use bridge vias*

Examples

```
envGetVal("layout.ava" "viaAttemptToUseBridgeVias")
envSetVal("layout.ava" "viaAttemptToUseBridgeVias" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAutoFromArea

```
layout viaAutoFromArea boolean { t | nil }
```

Description

Specifies the area, rectangular or polygonal, in which the layout editor will place the via(s) in all overlapping regions.

The default is `t`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*From Area*

Examples

```
envGetVal("layout" "viaAutoFromArea")
envSetVal("layout" "viaAutoFromArea" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAutoOnlyOnTopLayers

```
layout viaAutoOnlyOnTopLayers boolean { t | nil }
```

Description

Specifies that you can create vias using top two layers only.

The default is `nil`.

GUI Equivalent

Command:	*Create – Via, Options* button in the *Create Via* form
Field:	*Create Vias Using Top Two Layers Only*

Examples

```
envGetVal("layout" "viaAutoOnlyOnTopLayers")
envSetVal("layout" "viaAutoOnlyOnTopLayers" 'boolean t)
```

Related Topics

Create Via Options Form

© 2023 Cadence Design Systems, Inc.

viaAutoOnlyOnTopLayers

```
layout.ava viaAutoOnlyOnTopLayers boolean { t | nil }
```

Description

Lets you create vias only on top two layers.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant Options*
Field:	*Create vias using top 2 layers*

Examples

```
envGetVal("layout.ava" "viaAutoOnlyOnTopLayers")
envSetVal("layout.ava" "viaAutoOnlyOnTopLayers" 'boolean t)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaAutoViaAlignment

```
layout viaAutoViaAlignment cyclic {"auto" | "top-left" | "top-center" | "top-right"
     | "center-left" | "center-center" | "center-right" | "bottom-left" | "bottom-
     center" | "bottom-right" | "edge-align" | "two-edge-align" | "two-edge-align-
     center"}
```

Description

Specifies the alignment of the via.

The default value is `auto`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Via Alignment*

Examples

```
envGetVal("layout" "viaAutoViaAlignment")
envSetVal("layout" "viaAutoViaAlignment" 'cyclic "top-left")
envSetVal("layout" "viaAutoViaAlignment" 'cyclic "center-left")
envSetVal("layout" "viaAutoViaAlignment" 'cyclic "bottom-right")
envSetVal("layout" "viaAutoViaAlignment" 'cyclic "two-edge-align")
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAutoViaAlignment

```
layout.ava viaAutoViaAlignment cyclic {"auto" | "top-left" | "top-center" | "top-
     right" | "center-left" | "center-center" | "center-right" | "bottom-left" |
     "bottom-center" | "bottom-right" | "edge-align" | "two-edge-align" | "two-
     edge-align-center"}
```

Description

Specifies the alignment of the via.

The default value is `auto`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides*
Field:	*Alignment*

Examples

```
envGetVal("layout.ava" "viaAutoViaAlignment")
envSetVal("layout.ava" "viaAutoViaAlignment" 'cyclic "top-left")
envSetVal("layout.ava" "viaAutoViaAlignment" 'cyclic "center-left")
envSetVal("layout.ava" "viaAutoViaAlignment" 'cyclic "bottom-right")
envSetVal("layout.ava" "viaAutoViaAlignment" 'cyclic "two-edge-align")
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAutoViaCutArrayDirection

```
layout viaAutoViaCutArrayDirection cyclic {"auto" | "horizontal" | "vertical"}
```

Description

Specifies the preferred direction for the cut box as either `auto`, `horizontal`, or `vertical`.

The default is `auto`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Cut BBox Orientation*

Examples

```
envGetVal("layout" "viaAutoViaCutArrayDirection")
envSetVal("layout" "viaAutoViaCutArrayDirection" 'cyclic "horizontal")
envSetVal("layout" "viaAutoViaCutArrayDirection" 'cyclic "vertical")
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAutoViaCutArrayDirection

```
layout.ava viaAutoViaCutArrayDirection cyclic { "auto" | "horizontal" | "vertical"
     }
```

Description

Specifies the preferred direction for the cut box as either `auto`, `horizontal`, or `vertical`.

The default is `auto`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides*
Field:	*Cut Array Dir*

Examples

```
envGetVal("layout.ava" "viaAutoViaCutArrayDirection")
envSetVal("layout.ava" "viaAutoViaCutArrayDirection" 'cyclic "horizontal")
envSetVal("layout.ava" "viaAutoViaCutArrayDirection" 'cyclic "vertical")
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAutoViaEnableMaxStackDepth

```
layout viaAutoViaEnableMaxStackDepth boolean { t | nil }
```

Description

Specifies the maximum stack depth.

The default is `nil`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Max Stack Depth*

Examples

```
envGetVal("layout" "viaAutoViaEnableMaxStackDepth")
envSetVal("layout" "viaAutoViaEnableMaxStackDepth" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAutoViaEnclosureDirection

```
layout viaAutoViaEnclosureDirection cyclic {"XY" | "wire" | "minArea" |
     "horizontal" | "vertical"}
```

Description

Specifies the preferred enclosure direction for the via engine in auto mode as either `XY`, `wire`, `minArea`, `horizontal`, or `vertical`. This environment variable defines the way enclosures are set on the via when there are `minOppositeExtension` rules in the technology file.

- `XY`: Specifies the X and Y coordinates of the via in the left right and top bottom direction. For example, `(minOppExtension "Metal1" "Via1" (0.0 0.03))`. This rule means that the Metal1 enclosure around Via1 should be 0.0 in one direction and 0.03 in the other (opposite) direction.
- `wire`: The wire associated to the via is queried and if found, its direction is used to compute the enclosure direction, the higher enclosure value in the wire direction. For example, if the via Metal1-Metal2 connect two vertical pathseg, the wire direction is vertical, thus 0.03 will be applied in Y (top/bottom) and 0 in X (left/right) for both the layers. In case the via connect two orthogonal pathsegs, the direction of the Metal1 pathseg is used to determine the enclosure direction for the rule, (`minOppExtension "Metal1" "Via1" (0.0 0.03))`, and the Metal2 pathseg for the rule, `( minOppExtension "Metal2" "Via1" (0.0 0.03))`.
- `minArea`: Default. The enclosure direction is chosen to minimize the via area for each layer. Depending on the size of the `cutBox` , the enclosure will be applied in one direction or the other. For example, for a 1x10 via (big via in X, small in Y) the `0.03` in `( minOppExtension "Metal1" "Via1" (0.0 0.03) )` will be applied in X because it leads to a smaller area of `Metal1` than setting the `0.03` in Y.
- `horizontal`: The higher enclosure value in the horizontal direction. In the example, `( minOppExtension "Metal1" "Via1" (0.0 0.03) )`, `0.03` is applied in left/right coordinates and `0` is applied in top/bottom coordinates.
- `vertical`: The higher enclosure value in the vertical direction. In the example, `( minOppExtension "Metal1" "Via1" (0.0 0.03) )`, `0.03` is applied in top/bottom coordinates and 0 is applied in left/right coordinates.

GUI Equivalent

None

© 2023 Cadence Design Systems, Inc.

Examples

```
envGetVal("layout" "viaAutoViaEnclosureDirection")
envSetVal("layout" "viaAutoViaEnclosureDirection" 'cyclic "XY")
envSetVal("layout" "viaAutoViaEnclosureDirection" 'cyclic "wire")
envSetVal("layout" "viaAutoViaEnclosureDirection" 'cyclic "vertical")
```

Related Topics

minOppExtension

© 2023 Cadence Design Systems, Inc.

viaAutoViaEnclosureDirection

```
layout.ava viaAutoViaEnclosureDirection cyclic { "XY" | "wire" | "minArea" |
     "horizontal" | "vertical" }
```

Description

Specifies the preferred enclosure direction for the via engine in auto mode. This environment variable defines the way enclosures are set in a via when there are `minOppositeExtension` rules in the technology file.

- `XY`: Specifies the X and Y coordinates of the via in the left right and top bottom directions. For example, `(minOppExtension "Metal1" "Via1" (0.0 0.03))`. This rule means that the Metal1 enclosure around Via1 should be 0.0 in one direction and 0.03 in the other (opposite) direction.
- `wire`: The wire associated with the via is queried and, if found, its direction is used to compute the enclosure direction, the higher enclosure value in the wire direction. For example, if the via Metal1-Metal2 connect two vertical pathseg, the wire direction is vertical, 0.03 is applied in the Y (top/bottom) direction and 0 in the X (left/right) direction for both the layers. In case the via connects two orthogonal pathsegs, the direction of the Metal1 pathseg is used to determine the enclosure direction for the rule, (`minOppExtension "Metal1" "Via1" (0.0 0.03))`, and the Metal2 pathseg for the rule, `( minOppExtension "Metal2" "Via1" (0.0 0.03))`.
- `minArea`: This is the default value. The enclosure direction is chosen to minimize the via area for each layer. Depending on the size of the `cutBox` , the enclosure is applied in one direction or the other. For example, for a 1x10 via (big via in X, small in Y) the `0.03` in `( minOppExtension "Metal1" "Via1" (0.0 0.03) )` is applied in X because it leads to a smaller area of `Metal1` than setting the `0.03` in Y.
- `horizontal`: The higher enclosure value in the horizontal direction. In the example, `( minOppExtension "Metal1" "Via1" (0.0 0.03) ), 0.03` is applied in left/right coordinates and `0` is applied in top/bottom coordinates.
- `vertical`: The higher enclosure value in the vertical direction. In the example, `( minOppExtension "Metal1" "Via1" (0.0 0.03) ), 0.03` is applied in top/bottom coordinates and 0 is applied in left/right coordinates.

© 2023 Cadence Design Systems, Inc.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides*
Field:	*Enclosure Dir*

Examples

```
envGetVal("layout.ava" "viaAutoViaEnclosureDirection")
envSetVal("layout.ava" "viaAutoViaEnclosureDirection" 'cyclic "XY")
envSetVal("layout.ava" "viaAutoViaEnclosureDirection" 'cyclic "wire")
envSetVal("layout.ava" "viaAutoViaEnclosureDirection" 'cyclic "vertical")
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAutoViaExcludeNetsName

```
layout.ava viaAutoViaExcludeNetsName string "NetName"
```

Description

Excludes specified nets when creating vias.

The default is `""`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – Connectivity*
Field:	*Exclude Nets*

Examples

```
envGetVal("layout.ava" "viaAutoViaExcludeNetsName")
envSetVal("layout.ava" "viaAutoViaExcludeNetsName" 'string "net1")
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAutoViaExcludePowerAndGroundNets

```
layout.ava viaAutoViaExcludePowerAndGroundNets boolean { t | nil }
```

Description

Excludes power and ground nets when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Nets*
Field:	*Exclude power/ground*

Examples

```
envGetVal("layout.ava" "viaAutoViaExcludePowerAndGroundNets")
envSetVal("layout.ava" "viaAutoViaExcludePowerAndGroundNets" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAutoViaHints

```
layout viaAutoViaHints boolean { t | nil }
```

Description

Displays the *Auto Via Preview Bindkeys* hint box on the canvas.

The default is `nil`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto* and *Fast Edit*
Field:	*Show Hints*

Examples

```
envGetVal("layout" "viaAutoViaHints")
envSetVal("layout" "viaAutoViaHints" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAutoViaMaxStackDepth

```
layout viaAutoViaMaxStackDepth int maximumStack
```

Description

Specifies the value of the maximum stack depth.

The default is `1`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Max Stack Depth*

Examples

```
envGetVal("layout" "viaAutoViaMaxStackDepth")
envSetVal("layout" "viaAutoViaMaxStackDepth" 'int 5)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAutoViaMinNumCuts

```
layout viaAutoViaMinNumCuts int numberOfCuts
```

Description

Specifies the minimum number of cuts to be automatically created for any overlapping area between two shapes.

The default is `1`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Min Number of Cuts*

Examples

```
envGetVal("layout" "viaAutoViaMinNumCuts")
envSetVal("layout" "viaAutoViaMinNumCuts" 'int 5)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAutoViaMinNumCuts

```
layout.ava viaAutoViaMinNumCuts int numberOfCuts
```

Description

Specifies the minimum number of cuts to be automatically created for any overlapping area between two shapes.

The default is `1`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides*
Field:	*Min Num Cuts*

Examples

```
envGetVal("layout.ava" "viaAutoViaMinNumCuts")
envSetVal("layout.ava" "viaAutoViaMinNumCuts" 'int 5)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAutoViaOnSelectedNets

```
layout.ava viaAutoViaOnSelectedNets boolean { t | nil }
```

Description

Enables you to specify the net to be used when creating vias.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout.ava" "viaAutoViaOnSelectedNets")
envSetVal("layout.ava" "viaAutoViaOnSelectedNets" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaAutoViaPreviewShow

```
layout viaAutoViaPreviewShow boolean { t | nil }
```

Description

Displays a preview of the via that will be created.

The default is `t`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto* and *Fast Edit*
Field:	*Show Preview*

Examples

```
envGetVal("layout" "viaAutoViaPreviewShow")
envSetVal("layout" "viaAutoViaPreviewShow" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaAutoViaPreviewShow

```
layout.ava viaAutoViaPreviewShow boolean { t | nil }
```

Description

Shows a preview of the via that will be created.

The default is `t`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant Options*
Field:	*Show preview*

Examples

```
envGetVal("layout.ava" "viaAutoViaPreviewShow")
envSetVal("layout.ava" "viaAutoViaPreviewShow" 'boolean nil)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaCheckBlockageTypes

```
layout viaCheckBlockageTypes boolean { t | nil }
```

Description

Specifies whether or not to create any via in case a blockage exists between the top and bottom metal of the current overlap. Only via and cut layer blockages are considered, metal layer blockages are not considered.

The default is `nil`.

GUI Equivalent

Command:	*Create – Via*
Field:	*Consider Blockages*

Examples

```
envGetVal("layout" "viaCheckBlockageTypes")
envSetVal("layout" "viaCheckBlockageTypes" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaCheckBlockageTypes

```
layout.ava viaCheckBlockageTypes boolean { t | nil }
```

Description

Does not create any via in case a blockage exists between the top and bottom metal of the current overlap.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant Options*
Field:	*Consider blockages*

Examples

```
envGetVal("layout.ava" "viaCheckBlockageTypes")
envSetVal("layout.ava" "viaCheckBlockageTypes" 'boolean t)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaCheckMinNumCut

```
layout viaCheckMinNumCut boolean { t | nil }
```

Description

Determines whether the `minNumCut` constraint for the cut layers specified in the technology file is ignored while creating a via.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaCheckMinNumCut")
envSetVal("layout" "viaCheckMinNumCut" 'boolean nil)
```

Related Topics

minNumCut

© 2023 Cadence Design Systems, Inc.

viaCheckPlateMaxWidth

```
layout viaCheckPlateMaxWidth boolean { t | nil }
```

Description

Determines whether the `maxWidth` constraint for the metal layers specified in the technology file is ignored while creating a via.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaCheckPlateMaxWidth")
envSetVal("layout" "viaCheckPlateMaxWidth" 'boolean nil)
```

Related Topics

maxWidth

© 2023 Cadence Design Systems, Inc.

viaCheckPlateMinWidth

```
layout viaCheckPlateMinWidth boolean { t | nil }
```

Description

Determines whether the `minWidth` constraint for the metal layers specified in the technology file is ignored while creating a via.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaCheckPlateMinWidth")
envSetVal("layout" "viaCheckPlateMinWidth" 'boolean nil)
```

Related Topics

minWidth

© 2023 Cadence Design Systems, Inc.

viaCheckShapeConn

```
layout viaCheckShapeConn boolean { t | nil }
```

Description

Specifies whether or not to look for connectivity information on the intersecting shapes.

The default is `nil`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Connect Shapes On The Same Net*

Examples

```
envGetVal("layout" "viaCheckShapeConn")
envSetVal("layout" "viaCheckShapeConn" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaCheckUserSet

```
layout viaCheckUserSet boolean { t | nil }
```

Description

Lets retain user specified parameters during via editing.The Property Editor considers that all via parameters are user specified. So, when you change one via parameter, the other parameters do not change, when the default value `nil` is used.

When set to `t`, the Property Editor checks which via parameters are user specified and which have values based on the design rules.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaCheckUserSet")
envSetVal("layout" "viaCheckUserSet" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaColumns

`layout viaColumns int` *`numberOfColumns`*

Description

Specifies the default number of columns of via to be cut in a via array. The *Columns* field on the Create Via form is updated based on the value you specify in this environment variable.

The default is `1`.

GUI Equivalent

Command:	*Create – Via; Mode: Single*
Field:	*Columns*

Examples

```
envGetVal("layout" "viaColumns")
envSetVal("layout" "viaColumns" 'int 10)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaCompuEnclosures

```
layout viaCompuEnclosures boolean { t | nil }
```

Description

Allows you to automatically calculate enclosures.

The default is `t`.

GUI Equivalent

Command	*Create – Via*
Field	*Enclosures: Compute*

Examples

```
envGetVal("layout" "viaCompuEnclosures")
envSetVal("layout" "viaCompuEnclosures" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaComputeFromMode

```
layout viaComputeFromMode cyclic { "Row/Col" | "Width/Height" | "Shape(s)" | "Drawn
     Area" }
```

Description

Lets you create auto vias at the intersection of shapes in the *Single* and *Stack* modes. In the *Single* and *Stack* modes: You can select a *Via Definition* and specify the *Row* and *Column* Spacing, *Width* and *Height*, *Shape(s)* or, *Drawn Area* for the via. In the *Stack* mode, you can also specify the *Start Layer* and the *End Layer* of the via stack. So you can pre-define the viaDefs that will be used for each stacked via.

The default is `Row/Col`.

GUI Equivalent

Command	*Create – Via*
Field	*Compute From*

Examples

```
envGetVal("layout" "viaComputeFromMode")
envSetVal("layout" "viaComputeFromMode" 'cyclic "Drawn Area")
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaConnectivityMode

```
layout.ava viaConnectivityMode cyclic { "OnSameNet" | "Ignored" }
```

Description

Specifies whether to check for connectivity on the same net while creating vias.

The default is `OnSameNet`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create*
Field:	*Connectivity*

Examples

```
envGetVal("layout.ava" "viaConnectivityMode")
envSetVal("layout.ava" "viaConnectivityMode" 'cyclic "Ignored")
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaConstraintGroup

```
layout viaConstraintGroup string "name_of_constraint_group"
```

Description

Specifies the constraint group from which to use valid vias for creating and editing vias.

The default is `Same as Wire`.

GUI Equivalent

Command:	*Options – Editor*
Field:	*Wire Editing: Create Via: Via*

Examples

```
envGetVal("layout" "viaConstraintGroup")
envSetVal("layout" "viaConstraintGroup" 'string "foundry")
```

Related Topics

Layout Editor Options Form

© 2023 Cadence Design Systems, Inc.

viaCreateInGroup

```
layout.ava viaCreateInGroup string "GroupName"
```

Description

Specifies the group in which a new via is to be created.

The default is `None`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – More Options*
Field:	*Create In Group*

Examples

```
envGetVal("layout.ava" "viaCreateInGroup")
envSetVal("layout.ava" "viaCreateInGroup" 'string New figGroup)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaCreateInRoute

```
layout viaCreateInRoute boolean { t | nil }
```

Description

Enables the vias to be created in a new route in the *Single* (in Auto mode), *Stack* (in Auto mode), and *Auto* via creation modes.

The default is `t`.

GUI Equivalent

Command:	*Create – Via – Options*
Field:	*Create In Route*

Examples

```
envGetVal("layout" "viaCreateInRoute")
envSetVal("layout" "viaCreateInRoute" 'boolean nil)
```

Related Topics

Create Via Options Form

© 2023 Cadence Design Systems, Inc.

viaCreateInRoute

```
layout.ava viaCreateInRoute boolean { t | nil }
```

Description

Lets you to create a via in a new route. If this option is selected, every new via is created in a new route.

The default is `t`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant Options*
Field:	*Create in route*

Examples

```
envGetVal("layout.ava" "viaCreateInRoute")
envSetVal("layout.ava" "viaCreateInRoute" 'boolean nil)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaCutClassInAuto

```
layout viaCutClassInAuto string "name_of_cut_class"
```

Description

Specifies cut class or cut class group to be used. This environment variable value overwrites the value specified in the `cutClasses` constraint defined for the cut layer of the selected viaDef in the `layerRules` section of the technology file.

The default is `Default`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Cut Class*

Examples

```
envGetVal("layout" "viaCutClassInAuto")
envSetVal("layout" "viaCutClassInAuto" 'string "Auto")
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaCutClassInAuto

`layout.ava viaCutClassInAuto string "`*`name_of_cut_class`*`"`

Description

Specifies the cut class or cut class group to be used for via creation. This environment variable value overwrites the value specified in the `cutClasses` constraint defined for the cut layer of the selected viaDef in the `layerRules` section of the technology file.

The default is `Default`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides*
Field:	*Cut Class*

Examples

```
envGetVal("layout.ava" "viaCutClassInAuto")
envSetVal("layout" "viaCutClassInAuto" 'string "Auto")
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaCutSpacingMethod

```
layout viaCutSpacingMethod cyclic { "distribute" | "minimum" }
```

Description

Allows selection of the cut spacing method:

- `distribute`: Spreads the cut spacing.
- `minimum`: Uses the minimum spacing rule constraints

The default value is `distribute`.

GUI Equivalent

Command	*Create – Via – Options*
Field	*Cut Spacing Method*

Examples

```
envGetVal("layout" "viaCutSpacingMethod")
envSetVal("layout" "viaCutSpacingMethod" 'cyclic "minimum")
```

Related Topics

Create Via Options Form

© 2023 Cadence Design Systems, Inc.

viaCutClassOrientation

```
layout viaCutClassOrientation cyclic { "SameOrientation" | "Vertical" |
     "Horizontal" | "Auto" }
```

Description

Lets you specify the preferred orientation for the cut class as `SameOrientation`, `Vertical`, `Horizontal`, or `Auto`.

The default is `Auto`.

GUI Equivalent

Command	*Create – Via; Mode: Auto*
Field	*Cut Class Orientation*

Examples

```
envGetVal("layout" "viaCutClassOrientation")
envSetVal("layout" "viaCutClassOrientation" 'cyclic "SameOrientation")
envSetVal("layout" "viaCutClassOrientation" 'cyclic "Horizontal")
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaCutClassOrientation

```
layout.ava viaCutClassOrientation cyclic {"Auto" | "Horizontal" | "Vertical" |
     "SameOrientation"}
```

Description

Lets you specify the preferred orientation for the cut class as `SameOrientation`, `Vertical`, `Horizontal`, or `Auto`.

The default is `Auto`.

GUI Equivalent

Command	*Assistants – Auto Via Assistant – Constraint Overrides*
Field	*Cut Class Orient*

Examples

```
envGetVal("layout.ava" "viaCutClassOrientation")
envSetVal("layout.ava" "viaCutClassOrientation" 'cyclic "SameOrientation")
envSetVal("layout.ava" "viaCutClassOrientation" 'cyclic "Horizontal")
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaDRCCheckHierarchyRangeMode

```
layout.ava viaDRCCheckHierarchyRangeMode cyclic { "current to stop level" |
     "current to bottom" | "current to user level" }
```

Description

Sets the levels of hierarchy to be considered for layout integrity check.

- *current to stop level*: Runs the check from the current edit cellview level to the stop level specified in the *Display Levels Stop* field in the Display Options form.
- *current to bottom*: Runs the check from the current edit cellview level to the bottom of the hierarchy.
- *current to user level*: Runs the check from the current edit cellview level to the level that you specify in the text box.

The default is `current to stop level`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto via assistant options – DRC Check*
Field:	*Hierarchy Range*

Examples

```
envGetVal("layout.ava" "viaDRCCheckHierarchyRangeMode")
envSetVal("layout.ava" "viaDRCCheckHierarchyRangeMode" 'cyclic "current to
bottom")
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaEnableAddMetal

```
layout.ava viaEnableAddMetal boolean { t | nil }
```

Description

Adds a metal layer to the via when analyzing or creating vias using Auto Via Assistant.

The default is `t`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All – Add Metal*
Field:	*Enable*

Examples

```
envGetVal("layout.ava" "viaEnableAddMetal")
envSetVal("layout.ava" "viaEnableAddMetal" 'boolean nil)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaEnableAnnotation

```
layout.ava viaEnableAnnotation boolean { t | nil }
```

Description

Enables annotations when creating vias.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – Annotations*
Field:	*Enable*

Examples

```
envGetVal("layout.ava" "viaEnableAnnotation")
envSetVal("layout.ava" "viaEnableAnnotation" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaEnableDRCCheck

```
layout.ava viaEnableDRCCheck boolean { t | nil }
```

Description

Prevents the creation of a via if it introduces DRC violations with the neighboring shapes.

The default is `t`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create – Optimization*
Field:	*DRC check*

Examples

```
envGetVal("layout.ava" "viaEnableDRCCheck")
envSetVal("layout.ava" "viaEnableDRCCheck" 'boolean nil)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaEnableUserSkillCallbacks

```
layout viaEnableUserSkillCallbacks boolean { t | nil }
```

Description

Controls the creation of a via based on SKILL callbacks. If the user callback returns `t`, the via is created. If it returns `nil`, the via is not created.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaEnableUserSkillCallbacks")
envSetVal("layout" "viaEnableUserSkillCallbacks" 'boolean nil)
```

Related Topics

Creating a Via Automatically in Single and Stack Modes

© 2023 Cadence Design Systems, Inc.

viaEnableValidPurposes

```
layout viaEnableValidPurposes boolean { t | nil }
```

Description

Lest you specify a list of valid purposes for via creation in automated modes: auto, fast edit, compute from shape: single, compute from shape: stack.

The default is `nil`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto, Fast Edit; Options*
Field:	*Valid Purposes*

Examples

```
envGetVal("layout" "viaEnableValidPurposes")
envSetVal("layout" "viaEnableValidPurposes" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaEnableValidPurposes

```
layout.ava viaEnableValidPurposes boolean { t | nil }
```

Description

Lest you specify a list of valid purposes for via creation.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant Options – Overlaps*
Field:	*Valid Purposes*

Examples

```
envGetVal("layout.ava" "viaEnableValidPurposes")
envSetVal("layout.ava" "viaEnableValidPurposes" 'boolean t)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaExtendEnclosureBeyondOverlap

```
layout viaExtendEnclosureBeyondOverlap boolean { t | nil }
```

Description

Enables the via enclosures to be extended beyond the overlap region of the intersecting shapes in the *Single* (in Auto mode), *Stack* (in Auto mode), and *Auto* via creation modes.

The default is `t`.

GUI Equivalent

Command:	*Create – Via – Options*
Field:	*Extend Enclosure Beyond Overlap*

Examples

```
envGetVal("layout" "viaExtendEnclosureBeyondOverlap")
envSetVal("layout" "viaExtendEnclosureBeyondOverlap" 'boolean nil)
```

Related Topics

Create Via Options Form

© 2023 Cadence Design Systems, Inc.

viaExtendEnclosureBeyondOverlap

```
layout.ava viaExtendEnclosureBeyondOverlap boolean { t | nil }
```

Description

Extends a via enclosure beyond the overlap region between two intersecting shapes. The shape layer that exists beyond the overlap region accommodates the extended enclosures.

The default is `t`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant Options – Enclosures*
Field:	*Extend beyond overlap*

Examples

```
envGetVal("layout.ava" "viaExtendEnclosureBeyondOverlap")
envSetVal("layout.ava" "viaExtendEnclosureBeyondOverlap" 'boolean nil)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaFastEditFromSelectedVias

```
layout viaFastEditFromSelectedVias boolean { t | nil }
```

Description

Lets you use fast edit command from a set of selected vias.

The default is `t`.

GUI Equivalent

Command:	*Create – Via; Mode: Fast Edit*
Field:	*From Selected Vias*

Examples

```
envGetVal("layout" "viaFastEditFromSelectedVias")
envSetVal("layout" "viaFastEditFromSelectedVias" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaFillEnclosureOverlap

```
layout viaFillEnclosureOverlap boolean { t | nil }
```

Description

Extends via enclosure such that it covers the complete overlap area between the two intersecting shapes.

The default is `t`.

GUI Equivalent

Command:	*Create – Via – Options*
Field:	*Fill Overlap with Enclosure*

Examples

```
envGetVal("layout" "viaFillEnclosureOverlap")
envSetVal("layout" "viaFillEnclosureOverlap" 'boolean nil)
```

Related Topics

Create Via Options Form

© 2023 Cadence Design Systems, Inc.

viaFillEnclosureOverlap

```
layout.ava viaFillEnclosureOverlap boolean { t | nil }
```

Description

Extends via enclosure such that it covers the complete overlap area between the two intersecting shapes. The bounding box of the via enclosure coincides with the bounding box of the overlap region.

The default is `t`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant Options – Enclosures*
Field:	*Fill overlap*

Examples

```
envGetVal("layout.ava" "viaFillEnclosureOverlap")
envSetVal("layout.ava" "viaFillEnclosureOverlap" 'boolean nil)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaFillpolygonalOverlap

```
layout.ava viaFillPolygonalOverlap cyclic { "Fully" | "Partially" | "Horizontally"
     | "Vertically" }
```

Description

Specifies the fill method to be used while creating a via.

The default is `Fully`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto via assistant options*
Field:	*Fill Method*

Examples

```
envGetVal("layout.ava" "viaFillPolygonalOverlap")
envSetVal("layout.ava" "viaFillPolygonalOverlap" 'cyclic "Partially")
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaFixOverlapShapeViolations

```
layout.ava viaFixOverlapShapeViolations boolean { t | nil }
```

Description

Prevents the creation of a via if it introduces DRC violations on the overlap.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Optimization*
Field:	*Fix DRC on overlap shapes*

Examples

```
envGetVal("layout.ava" "viaFixOverlapShapeViolations")
envSetVal("layout.ava" "viaFixOverlapShapeViolations" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaHideEnclosures

```
layout viaHideEnclosures boolean { t | nil }
```

Description

Toggles the display of the section in the form that sets the enclosure values in the single mode on the Create Via form. The form is expanded and displays the enclosure values.

The default is `t`. When set to `nil`, the enclosure values form is collapsed.

GUI Equivalent

Command:	*Create – Via; Mode: Single*
Field:	*Show Enclosures / Hide Enclosures*

Examples

```
envGetVal("layout" "viaHideEnclosures")
envSetVal("layout" "viaHideEnclosures" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaIgnorePurposes

```
layout viaIgnorePurposes string "purposes_to_ignore"
```

Description

Lets you specify the list of purposes to be ignored for via creation in *Auto* and *Fast Edit* modes, and when single or stack is selected in the *Compute From* field.

The default is `""`.

GUI Equivalent

Command:	*Create – Via; Mode: Auto, Fast Edit; Options*
Field:	*Ignore Purposes*

Examples

```
envGetVal("layout" "viaIgnorePurposes")
envSetVal("layout" "viaIgnorePurposes" 'string "marker blockage")
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaIgnorePurposes

```
layout.ava viaIgnorePurposes string "purposes_to_ignore"
```

Description

Specifies a list of purposes to be ignored during via creation.

The default is `""`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto Via Assistant Options*
Field:	*Ignore Purposes*

Examples

```
envGetVal("layout.ava" "viaIgnorePurposes")
envSetVal("layout.ava" "viaIgnorePurposes" 'string "marker blockage")
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaMaximizeCuts

```
layout viaMaximizeCuts boolean { t | nil }
```

Description

Lets you get a bigger cut size at the cost of more added metal outside the overlap.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaMaximizeCuts")
envSetVal("layout" "viaMaximizeCuts" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaMaximizeCuts

```
layout.ava viaMaximizeCuts boolean { t | nil }
```

Description

Lets you get a bigger cut size at the cost of more added metal outside the overlap.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Maximize cut*

Examples

```
envGetVal("layout.ava" "viaMaximizeCuts")
envSetVal("layout.ava" "viaMaximizeCuts" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaMaxStackDepth

`layout.ava viaMaxStackDepth int` *`maximumStack`*

Description

Specifies the value of the maximum stack depth.

The default is `1`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – More Options*
Field:	*Max Stack Depth*

Examples

```
envGetVal("layout.ava" "viaAutoViaMaxStackDepth")
envSetVal("layout.ava" "viaAutoViaMaxStackDepth" 'int 5)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaMode

```
layout viaMode cyclic { "Single" | "Stack" | "Auto" }
```

Description

Specifies the mode of via creation.

- `Single`: Default. Creates a via.
- `Stack`: Creates a multi-layered via.
- `Auto`: Creates a via and automatically places it on the intersection of two paths.

GUI Equivalent

Command:	*Create – Via*
Field:	*Mode*

Examples

```
envGetVal("layout" "viaMode")
envSetVal("layout" "viaMode" 'cyclic "Stack")
envSetVal("layout" "viaMode" 'cyclic "Auto")
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaMultiProcessMode

```
layout.ava viaMultiProcessMode cyclic { "distributeOnMultiCPU" | "singleCPU" }
```

Description

Specifies the processing options to be used during via analysis.

The default is `distributeOnMultiCPU`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Optimization*
Field:	*Processing*

Examples

```
envGetVal("layout.ava" "viaMultiProcessMode")
envSetVal("layout.ava" "viaMultiProcessMode" 'cyclic "singleCPU")
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaMultiProcessNumCPU

`layout.ava viaMultiProcessNumCPU int` *`numberOfCuts`*

Description

Specifies the number of CPUs to be used during via analysis.

The default is `8`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Optimization*
Field:	*Max Num CPUs*

Examples

```
envGetVal("layout.ava" "viaMultiProcessNumCPU")
envSetVal("layout.ava" "viaMultiProcessNumCPU" 'int 12)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaOverrideAddMetalConstraint

```
layout.ava viaOverrideAddMetalConstraint boolean { t | nil }
```

Description

Lets you enable the *Add Metal* constraint override option when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Add Metal*

Examples

```
envGetVal("layout.ava" "viaOverrideAddMetalConstraint")
envSetVal("layout.ava" "viaOverrideAddMetalConstraint" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaOverrideAlignmentConstraint

```
layout.ava viaOverrideAlignmentConstraint boolean { t | nil }
```

Description

Lets you enable the *Alignment* constraint override option when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Alignment*

Examples

```
envGetVal("layout.ava" "viaOverrideAlignmentConstraint")
envSetVal("layout.ava" "viaOverrideAlignmentConstraint" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaOverrideCutArrayDirConstraint

```
layout.ava viaOverrideCutArrayDirConstraint boolean { t | nil }
```

Description

Lets you enable the *Cut Array Dir* constraint override option when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Cut Array Dir*

Examples

```
envGetVal("layout.ava" "viaOverrideCutArrayDirConstraint")
envSetVal("layout.ava" "viaOverrideCutArrayDirConstraint" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaOverrideCutClassOrientConstraint

```
layout.ava viaOverrideCutClassOrientConstraint boolean { t | nil }
```

Description

Lets you enable the *Cut Class Orient* constraint override option when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Cut Class Orient*

Examples

```
envGetVal("layout.ava" "viaOverrideCutClassOrientConstraint")
envSetVal("layout.ava" "viaOverrideCutClassOrientConstraint" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaOverrideEnclosureDirConstraint

```
layout.ava viaOverrideEnclosureDirConstraint boolean { t | nil }
```

Description

Lets you enable the *Enclosure Dir* constraint override option when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Enclosure Dir*

Examples

```
envGetVal("layout.ava" "viaOverrideEnclosureDirConstraint")
envSetVal("layout.ava" "viaOverrideEnclosureDirConstraint" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaOverrideMaximizeCutsConstraint

```
layout.ava viaOverrideMaximizeCutsConstraint boolean { t | nil }
```

Description

Lets you enable the *Maximize Cut* constraint override option when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Maximize cut*

Examples

```
envGetVal("layout.ava" "viaOverrideMaximizeCutsConstraint")
envSetVal("layout.ava" "viaOverrideMaximizeCutsConstraint" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaOverrideMinNumCutsConstraint

```
layout.ava viaOverrideMinNumCutsConstraint boolean { t | nil }
```

Description

Lets you enable the *Min Num Cuts* override option when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Min Num Cuts*

Examples

```
envGetVal("layout.ava" "viaOverrideMinNumCutsConstraint")
envSetVal("layout.ava" "viaOverrideMinNumCutsConstraint" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaOverrideOffCenterViaConstraint

```
layout.ava viaOverrideOffCenterViaConstraint boolean { t | nil }
```

Description

Lets you enable the *Place via on centerline* constraint override option when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Place via on centerline*

Examples

```
envGetVal("layout.ava" "viaOverrideOffCenterViaConstraint")
envSetVal("layout.ava" "viaOverrideOffCenterViaConstraint" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaParamCalcMode

```
layout viaParamCalcMode cyclic { "minRulesAndViaDef" | "minRules" |
     "viaDefDefaults"| "minRuleEnclosures" | "viaDefEnclosures" }
```

Description

Lets you reset the values in the form to the technology database values if you want to revert from the manually updated parameter values to the defaults. When you first open the Create Via form, the via parameter values are calculated based on the Via Parameters Calculation Mode setting in the Create Via section in the Layout Editor Options Form.

- `minRulesAndViaDef`: Default. All the via parameter fields, such as the number of cut rows and columns, cut sizes, spacing, and enclosures, are reset to the minimum rule or the viaDef default values, whichever is greater.
- `minRules`: All the via parameter fields are reset to the minimum rule values. ViaDef defaults are not checked. Therefore, the calculated values can be less than the viaDef defaults. For example, if the minExtensionDistance rule is 0.05 and the default enclosure in viaDef is 0.1, 0.05 is used as the enclosure value in the Create Via form. The viaDef default enclosure value is ignored.
- `minRuleEnclosures`: Minimum rules are used to calculate only the enclosure field values. Values in the other fields are not reset.
- `viaDefDefaults`: All the via parameter fields are reset to the viaDef defaults. Minimum rules are not checked.
- `viaDefDefaultsEnclosures`: ViaDef defaults are used to calculate only the enclosure field values. Values in the other fields are not reset.

GUI Equivalent

Command:	*Create – Via; Mode: Single*
Field:	*Reset parameters to*

Examples

```
envGetVal("layout" "viaParamCalcMode")
envSetVal("layout" "viaParamCalcMode" 'cyclic "viaDefDefaults")
```


© 2023 Cadence Design Systems, Inc.

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaPreventDRCOnOverlap

```
layout viaPreventDRCOnOverlap boolean { t | nil}
```

Description

Prevents the creation of auto via on overlap if it introduces DRC violations. Auto via is not created on the overlap if DRC violations are introduced.

The default is `t`. When set to `nil`, auto via is created even if DRC violations are introduced on the overlap.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Prevent DRC Violations On The Overlap*

Examples

```
envGetVal("layout" "viaPreventDRCOnOverlap")
envSetVal("layout" "viaPreventDRCOnOverlap" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaPreventDRCWithNeighbors

```
layout viaPreventDRCWithNeighbors boolean { t | nil}
```

Description

Prevents the creation of auto via if it introduces DRC violations with neighboring shapes. Auto via is created even if DRC violations are introduced with the neighboring shapes when the default value `nil` is used.

When set to `t`, auto via is not created if DRC violations are introduced with neighboring shapes.

GUI Equivalent

Command:	*Create – Via; Mode: Auto*
Field:	*Prevent DRC Violations With Neighbors*

Examples

```
envGetVal("layout" "viaPreventDRCWithNeighbors")
envSetVal("layout" "viaPreventDRCWithNeighbors" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaPreventOffCenterVia

```
layout.ava viaPreventOffCenterVia boolean { t | nil }
```

Description

Places the via on the centerline when analyzing or creating vias using Auto Via Assistant.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides – Show All*
Field:	*Place via on centerline*

Examples

```
envGetVal("layout.ava" "viaPreventOffCenterVia")
envSetVal("layout.ava" "viaPreventOffCenterVia" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaPrintStatisticsReport

```
layout viaPrintStatisticsReport boolean { t | nil}
```

Description

Prints a statistics report for auto via in the form of a table in the CIW.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaPrintStatisticsReport")
envSetVal("layout" "viaPrintStatisticsReport" 'boolean nil)
```

Related Topics

Creating a Via Automatically in Single and Stack Modes

© 2023 Cadence Design Systems, Inc.

viaPrintStatisticsReport

```
layout.ava viaPrintStatisticsReport boolean { t | nil }
```

Description

Shows a report of the vias that are to be created.

The default is `t`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Auto via assistant options*
Field:	*Show report*

Examples

```
envGetVal("layout.ava" "viaPrintStatisticsReport")
envSetVal("layout.ava" "viaPrintStatisticsReport" 'boolean nil)
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaPrintStatisticsReportByStackDepth

```
layout viaPrintStatisticsReportByStackDepth boolean { t | nil}
```

Description

Prints a statistics report sorted by stack depth for auto via in the form of a table in the CIW. The `viaPrintStatisticsReport` should be set to `t` for this environment variable to print the statistics report.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaPrintStatisticsReportByStackDepth")
envSetVal("layout" "viaPrintStatisticsReportByStackDepth" 'boolean nil)
```

Related Topics

Creating a Via Automatically in Single and Stack Modes

© 2023 Cadence Design Systems, Inc.

viaTryWithSmallerCutClass

```
layout.ava viaTryWithSmallerCutClass boolean { t | nil }
```

Description

Places a via with a smaller cut class if a larger via is not DRC aware.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Optimization*
Field:	*Try with smaller cut classes*

Examples

```
envGetVal("layout.ava" "viaTryWithSmallerCutClass")
envSetVal("layout.ava" "viaTryWithSmallerCutClass" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaRows

```
layout viaRows int numberOfRows
```

Description

Specifies the default number of rows of via to be cut in a via array. The *Via Rows* field on the Create Via form is updated based on the value you specify in this environment variable.

The default is `1`.

GUI Equivalent

Command:	*Create – Via; Mode: Single*
Field:	*Rows*

Examples

```
envGetVal("layout" "viaRows")
envSetVal("layout" "viaRows" 'int 10)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaShowAllConstraintOverride

```
layout.ava viaShowAllConstraintOverride boolean { t | nil }
```

Description

Displays all the constraint override options for the analysis or creation of vias using Auto Via Assistant.

The default value is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Constraint Overrides*
Field:	*Show All*

Examples

```
envGetVal("layout.ava" "viaShowAllConstraintOverride")
envSetVal("layout.ava" "viaShowAllConstraintOverride" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaSelectionMode

```
layout viaSelectionMode cyclic { "unknown" | "ignoreLayers" | "useCutLayers" |
     "useNonCutLayers" }
```

Description

Controls the selection of vias. You can set the environment variable to one of the following values:

- `unknown`: Prevents the selection of vias.
- `ignoreLayers`: Default. Enables the selection of vias if the *Selectability* check box for *Vias* in the *Objects* assistant is selected.
- `useCutLayers`: Enables the selection of vias if the *Selectability* check boxes for *Vias* in the *Objects* assistant and for the respective cut layer in the *Layers* assistant are selected.
- `useNonCutLayers`: Enables the selection of vias if the *Selectability* check boxes for *Vias* in the Objects assistant are selected. Also, both top and bottom layer should be set to visible and at least one bottom or top layer should be selected in the *Layers* assistant.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaSelectionMode ")
envSetVal("layout" "viaSelectionMode " 'cyclic "useCutLayers")
envSetVal("layout" "viaSelectionMode " 'cyclic "useNonCutLayers")
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

viaStackColumns

```
layout viaStackColumns int numberOfColumns
```

Description

Specifies the number of columns in the top via of the via stack.

The default is `1`.

GUI Equivalent

Command:	*Create – Via; Mode: Stack*
Field:	*Top Columns*

Examples

```
envGetVal("layout" "viaStackColumns")
envSetVal("layout" "viaStackColumns" 'int 10)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaStackRows

```
layout viaStackRows int numberOfRows
```

Description

Specifies the number of rows in the top via of the via stack.

The default is `1`.

GUI Equivalent

Command:	*Create – Via; Mode: Stack*
Field:	*Top Rows*

Examples

```
envGetVal("layout" "viaStackRows")
envSetVal("layout" "viaStackRows" 'int 10)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaStackSelection

```
graphic viaStackSelection boolean { t | nil }
```

Description

Toggles the via stack selection mode and allows the via stack to be selected with a single click.

The default is `t`.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Via Stack*

Examples

```
envGetVal("graphic" "viaStackSelection")
envSetVal("graphic" "viaStackSelection" 'boolean nil)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

viaStackSelectionMatchingSizes

```
graphic viaStackSelectionMatchingSizes boolean { t | nil }
```

Description

Allows to select vias belonging to the same via stack from a set of multiple super-imposed via stacks that share the origin and adjacent layers.

The default is `nil`.

GUI Equivalent

Command:	*Options – Selection*
Field:	*Vias with matching sizes between common layers*

Examples

```
envGetVal("graphic" "viaStackSelectionMatchingSizes")
envSetVal("graphic" "viaStackSelectionMatchingSizes" 'boolean t)
```

Related Topics

Selection Options Form

© 2023 Cadence Design Systems, Inc.

viaUseEffectiveCuts

```
layout viaUseEffectiveCuts boolean { t | nil }
```

Description

Disables the use of the `numCuts` parameter on the `cutClass` definition to optimize the total number of cuts for a via.The via engine uses the total cut area for optimization.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaUseEffectiveCuts")
envSetVal("layout" "viaUseEffectiveCuts" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaUseEntireOverlap

```
layout viaUseEntireOverlap boolean { t | nil }
```

Description

Lets you use the entire overlap to create the via.

The default is `t`. When set to `nil`, an intersection between the user area and the overlap box is used to create the via.

GUI Equivalent

Command	*Create – Via; Mode: Auto*
Field	*Use Entire Overlap*

Examples

```
envGetVal("layout" "viaUseEntireOverlap")
envSetVal("layout" "viaUseEntireOverlap" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaUseRowColDefault

```
layout viaUseRowColDefault boolean { t | nil }
```

Description

Lets you manage how to set the values in the *Rows* and *Columns* fields in the Create Via form and lets you use the edited values in the Create Via form.

The default is `t`. When set to `nil`, you can force the rows and columns values by setting values in the environment variables, `viaRows` and `viaColumns`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaUseRowColDefault")
envSetVal("layout" "viaUseRowColDefault" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaUseEntireOverlap

```
layout.ava viaUseEntireOverlap boolean { t | nil }
```

Description

Lets you use the entire overlap to run the analysis on vias or create vias.

The default is `nil`.

GUI Equivalent

Command:	*Assistants – Auto Via Assistant – Create vias in rectangle*
Field:	*Use entire overlap*

Examples

```
envGetVal("layout.ava" "viaUseEntireOverlap")
envSetVal("layout.ava" "viaUseEntireOverlap" 'boolean t)
```

Related Topics

Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

viaValidateDRCStandalone

```
layout viaValidateDRCStandalone boolean { t | nil }
```

Description

Specifies if the *Create Via - Single* and *Create Via - Stack* commands in the Create Via form can create a via with DRC violations. In general, not all metal shapes are present when creating a standalone via, so not all design rules can be checked.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaValidateDRCStandalone")
envSetVal("layout" "viaValidateDRCStandalone" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaValidPurposes

`layout viaValidPurposes string "`*`list_of_valid_purposes`*`"`

Description

Lets you specify a list of valid purposes for via creation in automated modes: auto, fast edit, compute from shape: single, compute from shape: stack. The via is not be created even if one purpose of the overlap shapes does not belong to the list of valid purposes specified in this environment variable.

The default is `drawing pin`.

GUI Equivalent

Command	*Create – Via; Mode: Auto, Fast Edit; Options*
Field	*Valid Purposes*

Examples

```
envGetVal("layout" "viaValidPurposes")
envSetVal("layout" "viaValidPurposes" 'string "drawing")
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaValidPurposes

```
layout.ava viaValidPurposes string "any_valid_purposes"
```

Description

Lets you to specify a list of valid purposes for via creation. The via will not be created even if one purpose of the overlap shapes does not belong to the list of valid purposes specified in this field.

The default is `drawing pin`.

GUI Equivalent

Command	*Assistants – Auto Via Assistant – Auto Via Assistant Options*
Field	*Valid Purposes*

Examples

```
envGetVal("layout.ava" "viaValidPurposes")
envSetVal("layout.ava" "viaValidPurposes" 'string "drawing")
```

Related Topics

Auto Via Assistant Options Form

© 2023 Cadence Design Systems, Inc.

viaWallFromSelectedShapesEnabled

```
layout viaWallFromSelectedShapesEnabled boolean { t | nil }
```

Description

Lets you use the wall functionality from the selected shapes.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaWallFromSelectedShapesEnabled")
envSetVal("layout" "viaWallFromSelectedShapesEnabled" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaWallKeepViasEnabled

```
layout viaWallKeepViasEnabled boolean { t | nil }
```

Description

Retains the existing vias when a wall is created.

The default is `nil`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaWallKeepViasEnabled")
envSetVal("layout" "viaWallKeepViasEnabled" 'boolean t)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viaWallPreviewEnabled

```
layout viaWallPreviewEnabled boolean { t | nil }
```

Description

Lets you specify the wall capability.

The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layout" "viaWallPreviewEnabled")
envSetVal("layout" "viaWallPreviewEnabled" 'boolean nil)
```

Related Topics

Create Via Form

© 2023 Cadence Design Systems, Inc.

viewNameToTypeList

```
graphic viewNameToTypeList string "any_valid_view_name"
```

Description

Specifies the correct view name associated with the default tool name in the Create New File form.

GUI Equivalent

Command:	(CIW) File - New - Cellview
Field:	View

Examples

```
envGetVal("graphic" "viewNameToTypeList")
envSetVal("graphic" "viewNameToTypeList" 'string "schematic")
```

Related Topics

New File form

© 2023 Cadence Design Systems, Inc.

xSnapSpacing

```
layout xSnapSpacing float minimumDistance
graphic xSnapSpacing float minimumDistance
```

Description

Controls the minimum distance the pointer moves in the X direction.

The layout default is `0.1` user units, the graphic default is `1.0` user units.

GUI Equivalent

Command:	*Options – Display*
Field:	*Grid Controls*: *X Snap Spacing*

Examples

```
envGetVal("layout" "xSnapSpacing")
envSetVal("layout" "xSnapSpacing" 'float 0.3)
envGetVal("graphic" "xSnapSpacing")
envSetVal("graphic" "xSnapSpacing" 'float 3.0)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

ySnapSpacing

```
layout ySnapSpacing float minimumDistance
graphic ySnapSpacing float minimumDistance
```

Description

Controls the minimum distance the pointer moves in the Y direction.

The layout default is `0.1` user units, the graphic default is `1.0` user units.

GUI Equivalent

Command:	*Options – Display*
Field:	*Grid Controls*: *Y Snap Spacing*

Examples

```
envGetVal("layout" "ySnapSpacing")
envSetVal("layout" "ySnapSpacing" 'float 0.3)
envGetVal("graphic" "ySnapSpacing")
envSetVal("graphic" "ySnapSpacing" 'float 3.0)
```

Related Topics

Display Options Form

© 2023 Cadence Design Systems, Inc.

yankShape

```
layout yankShape cyclic { "rectangle" | "polygon" }
```

Description

Specifies the default shape used by `leHiYank`.

The default is `rectangle`.

GUI Equivalent

Command:	*Edit – Basic – Yank*
Field:	*Yank Shape*

Examples

```
envGetVal("layout" "yankShape")
envSetVal("layout" "yankShape" 'cyclic "polygon")
```

Related Topics

Yank Form

© 2023 Cadence Design Systems, Inc.

zoomToFig

```
layout zoomToFig boolean { t | nil }
```

Description

Specifies whether the cellview zooms to objects found in a search.

The default is `nil`.

GUI Equivalent

Command:	*Tools – Find/Replace*
Field:	*Search Results: Zoom To Figure*

Examples

```
envGetVal("layout" "zoomToFig")
envSetVal("layout" "zoomToFig" 'boolean t)
```

Related Topics

Find/Replace Form

© 2023 Cadence Design Systems, Inc.

Environment Variables Used Internally

The following graphic and layout editor environment variables are used internally by the software or are reserved for future use. Changing the values of these variables may cause the layout editor or other Cadence software to behave improperly or quit. Many of these variables are read-only and cannot be changed.

coalesceLimit	markNetStopLayer
coalesceRatio	markNetViaLayer
confirmDelete	markNetHilightObj
connStatus	midPointOf2TargetsMode
criticality	minLabelSize
cycleSnap	minRefreshArea
disableLayerFuncSnapToMfgGrid	minStippleSize
eipRedraw	moreDialogs
enterFunGravityTypeStr	netName
figOrder	outlineOnly
filteredNetName	pickNCreate
grMode	probeTextOn
iterNum	propClass
jumpersOn	pruneSize
leFigGroupNamePrefix	routeMenuOn
markerShowIgnore	toggle
markNetDumpNetObj	transformMarkersOn
markNetConnectedLayer	viewNameList

Related Topics

Layout XL Basic Editing Environment Variables

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

C

Layout XL Assistants

Virtuoso® Layout Suite XL provides various assistants to enable you to work on your designs effectively and efficiently. The following table provides an overview of the assistants.

Assistant	Description
Auto Via Assistant	Lets you analyze vias in your design to check for missing vias. It also lets you create local overlaps with no vias. This assistant is available in Virtuoso Layout Suite EXL and higher tiers only.

© 2023 Cadence Design Systems, Inc.

Auto Via Assistant

Use the Auto Via Assistant to analyze and create vias. This assistant is available in Virtuoso Layout Suite EXL and higher tiers only.

The Auto Via Assistant user interface has the following components.

Analyze Tab	Lets you analyze vias in your design to check for locations where you can create vias.
Create Tab	Lets you create vias at the suggested locations.

The following table describes the options available on the Constraint Group drop-down.

Field	Description
Constraint Group	Specifies the constraint group that lists the valid vias to be used when creating vias. The available options are: *Same as Wire*, *LEFDefaultRouteSpec, foundry, skip_a_track, virtuosoDefaultExtractorSetup, virtuosoDefaultSetup,* and *wide*. The default is *Same as Wire.*

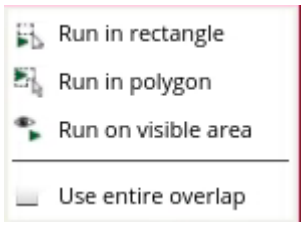
Analyze Tab

Use the *Analyze* tab to analyze vias in your design for locations where you can create vias.

The following table describes the options available on the Analyze tab of the Auto Via Assistant.

Field	Description
Toolbar Options	This section provides the toolbar options to analyze vias.
	Analyzes vias in the entire cellview.

© 2023 Cadence Design Systems, Inc.

Field	Description
	Runs analysis in the specified area. The options are: *in rectangle*, *in polygon*, or *on visible area*. Select the *Use entire overlap* option to use the entire overlap for analysis.
	
	Shows the analyzed vias.
Nets	This section provides options to analyze the possibility of via creation for the specified nets. Environment Variable: viaAutoViaOnSelectedNets
Exclude Nets	Excludes the specified nets from the analysis. Environment Variable: viaAutoViaExcludeNetsName
Exclude power/ground	Excludes the power and ground nets from the analysis. Environment Variable: viaAutoViaExcludePowerAndGroundNets
Constraint Overrides	This section provides options to specify constraint overrides. The Auto Via Assistant uses constraints from a constraint group or locally to override from environment variables. If no constraint override is selected then the assistant uses constraints from the constraint group. If any constraint override is selected then the assistant uses constraints from the environment variables.
Show All	Displays all constraint override options. Environment Variable: viaShowAllConstraintOverride

© 2023 Cadence Design Systems, Inc.

Field	Description
Min Num Cuts	Specifies the minimum number of cuts to be automatically created for overlapping areas between shapes. Environment Variable: viaOverrideMinNumCutsConstraint, viaAutoViaMinNumCuts
Alignment	Specifies the alignment of vias. The choices are *Auto*, *Top left*, *Top center*, *Top right*, *Center left*, *Center center*, *Center right*, *Bottom left*, *Bottom center*, *Bottom right*, *Edge align*, *Two edge align*, or *Two edge align center*. The default is *Auto*. Environment Variable: viaOverrideAlignmentConstraint, viaAutoViaAlignment
Cut Class Orient	Specifies the preferred orientation for the cut class. The choices are *Auto*, *Horizontal*, *Vertical,* or *Same orientation*. Environment Variable: viaOverrideCutClassOrientConstraint, viaCutClassOrientation
Cut Array Dir	Specifies the preferred direction for the cut box as *Auto*, *Horizontal*, or *Vertical*. The default is *Auto*. Environment Variable: viaOverrideCutArrayDirConstraint, viaAutoViaCutArrayDirection
Enclosure Dir	Specifies the preferred enclosure direction for the via engine in auto mode as *XY*, *Wire*, *Min area*, *Horizontal*, or *Vertical*. Environment Variable: viaOverrideEnclosureDirConstraint, viaAutoViaEnclosureDirection
Place via on centerline	Specifies whether the via is placed on the centerline. Environment Variable: viaOverrideOffCenterViaConstraint, viaPreventOffCenterVia
Add Metal	Adds a metal layer to the via. Environment Variable: viaOverrideAddMetalConstraint, viaEnableAddMetal
Enable	Enables the *Extend to orthogonal wire* option.

© 2023 Cadence Design Systems, Inc.

Field	Description
Maximize cut	Lets you get a bigger cut size at the cost of more added metal outside the overlap. Environment Variable: viaOverrideMaximizeCutsConstraint, viaMaximizeCuts
Extend to orthogonal wire	Extends vias to the width of the crossed wire.
Optimization	This section provides options to optimize via analysis.
Try with smaller cut classes	Places a via with a smaller cut class if a larger via is not DRC aware. Environment Variable: viaTryWithSmallerCutClass
Attempt to use bridge vias	Places a bridge via according to the topology. If the bridge via is invalid, a basic via is created. Environment Variable: viaAttemptToUseBridgeVias
Fix DRC on overlap shapes	Prevents the creation of a via if it introduces DRC violations on an overlap. If this option is not selected, the via is created even if it introduces DRC violations on the overlap. Environment Variable: viaFixOverlapShapeViolations
Processing	Species the processing options to be used during via analysis. The choices are Distribute on *multi CPU*, or *Single CPU*. Environment Variable: viaMultiProcessMode
Max Num CPUs	Specifies the number of CPUs to be used during via analysis. Environment Variable: viaMultiProcessNumCPU
More Options	This section provides options to specify the stack depth.
Max Stack Depth	Specifies the value of maximum stack depth to search the overlap with the same number of transitions or less than the specified number. Environment Variable: viaMaxStackDepth

© 2023 Cadence Design Systems, Inc.

Related Topics

Via Creation

Auto Via Assistant Options Form

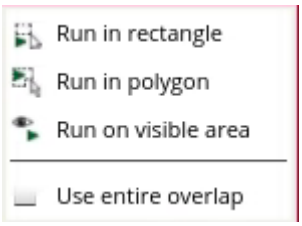
Analyzing a via using Auto Via Assistant

Creating a via using Auto Via Assistant

Create Tab

Use the *Create* tab to create the vias at the suggested locations in your design.

The following table describes the options available on the *Create* tab of the Auto Via Assistant.

Field	Description
Toolbar Options	This section explains the toolbar options to create vias.
	Creates vias in the entire cellview.
	Creates vias in a rectangular area, polygon area, or the entire visible area. Select the *Use entire overlap* to use the entire overlapped area for creating vias.
	
	Creates vias at the analyzed places.

© 2023 Cadence Design Systems, Inc.

Field	Description
Connectivity	This section provides options to create vias for a selected set of nets, exclude selected nets, and exclude power and ground nets. The choices for this field are *On same net*, *Ignored*.
Nets	Specifies the net to be used when creating vias. Environment Variable: viaAutoViaOnSelectedNets
Exclude Nets	Excludes the specified nets when creating vias. Environment Variable: viaAutoViaExcludeNetsName
Exclude power/ground	Excludes the power and ground nets when creating vias. Environment Variable: viaAutoViaExcludePowerAndGroundNets
Constraint Overrides	This section specifies constraint override options.
Show All	Displays the constraint override options. Environment Variable: viaShowAllConstraintOverride
Min Num Cuts	Specifies the minimum number of cuts to be automatically created for overlapping areas between shapes. Environment Variable: viaOverrideMinNumCutsConstraint, viaAutoViaMinNumCuts
Alignment	Specifies the alignment of vias using the alignment choices *Auto*, *Top left*, *Top center*, *Top right*, *Center left*, *Center center*, *Center right*, *Bottom left*, *Bottom center*, *Bottom right*, *Edge align*, *Two edge align*, or *Two edge align center*. The default is *Auto*. Environment Variable: viaOverrideAlignmentConstraint, viaAutoViaAlignment
Cut Class Orient	Specifies the preferred orientation for the cut class. The choices are *Auto*, *Horizontal*, *Vertical,* or *Same-orientation*. Environment Variable: viaOverrideCutClassOrientConstraint, viaCutClassOrientation
Cut Array Dir	Specifies the preferred direction for the cut box as *Auto*, *Horizontal*, or *Vertical*. The default is *Auto*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Enclosure Dir	Specifies the preferred enclosure direction for the via engine in auto mode as *XY*, *Wire*, *Min area*, *Horizontal*, or *Vertical*. Environment Variable: viaOverrideCutArrayDirConstraint, viaAutoViaCutArrayDirection
Place via on centerline	Specifies whether the via is placed on the centerline. Environment Variable: viaOverrideOffCenterViaConstraint, viaPreventOffCenterVia
Add Metal	Adds a metal layer to the via. Environment Variable: viaOverrideAddMetalConstraint, viaEnableAddMetal
Maximize cut	Lets you get a bigger cut size at the cost of more added metal outside the overlap. Environment Variable: viaOverrideMaximizeCutsConstraint, viaMaximizeCuts
Extend to orthogonal wire	Extends vias to the width of the crossed wire.
Optimization	This section provides options to optimize the via creation process.
DRC check	Prevents the creation of a via if it introduces DRC violations with it's neighboring shapes. If this option is not selected, the via is created even if it introduces DRC violations with it's neighboring shapes. Environment Variable: viaEnableDRCCheck
Try with smaller cut classes	Places a via with a smaller cut class if a larger via is not DRC aware. Environment Variable: viaTryWithSmallerCutClass
Attempt to use bridge vias	

© 2023 Cadence Design Systems, Inc.

Field	Description
	Places a bridge via according to the topology. If the bridge via is invalid, a basic via is created. Environment Variable: viaAttemptToUseBridgeVias
Fix DRC on overlap shapes	
	Prevents the creation of a via if it introduces DRC violations on the overlap. If this option is not selected, the via is created even if it introduces DRC violations on the overlap. Environment Variable: viaFixOverlapShapeViolations
Processing	Species the processing options to be used during via analysis. The choices are Distribute on *multi CPU*, or *Single CPU*. Environment Variable: viaMultiProcessMode
Max Num CPUs	Specifies the number of CPUs to be used during via analysis. Environment Variable: viaMultiProcessNumCPU
Annotations	This section provides options to create annotations displayed in the *Auto Via* tab of the Annotation Browser.
Enable	Enables annotations when creating vias. Environment Variable: viaEnableAnnotation
Vias adding metal	Creates annotations on vias that add metal to the design. Environment Variable: viaAnnotateViasWithAddedMetal
Vias not adding metal	Creates annotations on vias that do not add metal to the design. Environment Variable: viaAnnotateViasWithNoAddedMetal
Vias not retained	Creates annotations on overlaps on which vias have not been created because they add a metal or create DRC violations. Environment Variable: viaAnnotateViasNotRetained
Overlaps with no solution	
	Creates annotations on overlaps for which no via is generated. Environment Variable: viaAnnotateOverlapsWithNoValidSolution

© 2023 Cadence Design Systems, Inc.

Field	Description
More Options	This section provides options to specify the maximum stack depth and creating vias in group.
Max Stack Depth	Searches the overlap with the same number of transitions or less than the specified number. You can specify the value of the maximum stack depth. Environment Variable: viaMaxStackDepth
Create in Group	Specifies whether the via is created in a group. The choices are *None*, *New figGroup*, *Virtual_0*. Environment Variable: viaCreateInGroup

Related Topics

Via Creation

Auto Via Assistant Options Form

Analyzing a via using Auto Via Assistant

Creating a via using Auto Via Assistant

© 2023 Cadence Design Systems, Inc.

D

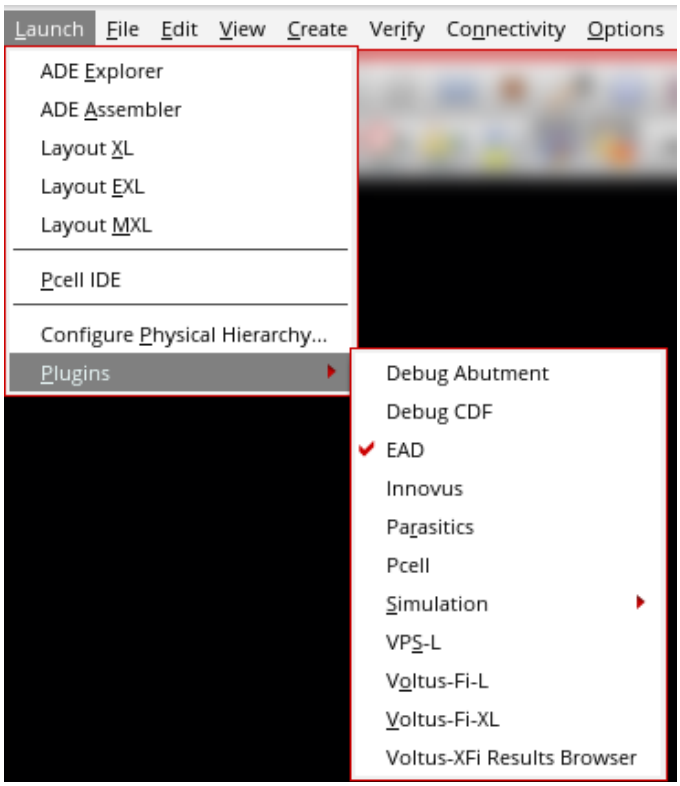
Layout Editor Plug-Ins

Virtuoso® Layout Suite cockpit supports various plug-ins. Plug-ins are applications that can be dynamically added to the layout editor user interface to perform some specialized design task, and then removed again when no longer required.

The *Launch - Plugins* menu contains all the applications that are available from the layout editor. To use any application listed in the menu, you must first load it.

When you load a plug-in application, it appears in the layout editor, typically as a separate menu on the menu bar, but sometimes as a user interface docked on the left- side of the layout window. Some plug-ins open the related applications directly on the layout canvas itself.

The following figure shows the *Plugins* submenu available on the *Launch* menu of Virtuoso Layout Suite.



© 2023 Cadence Design Systems, Inc.

The following table describes the plug-ins available in the layout editor and provides links for further information on associated features.

Plug-In	Description
Debug Abutment	Lets you debug the abutment using the Pcell IDE window. For more information, see *Debugging Pcells Abutment*.
Debug CDF	Lets you debug CDF callback procedures using simple SKILL expressions for form initialization (`forminit`), form completion (`doneProc`), and any parameter field value modification (parameter callbacks). For more information, see *Debugging CDF*.
Innovus	Adds the *Innovus* menu to the layout editor menu bar. The Innovus Implementation System is used as the design cockpit if a Verilog netlist is used for the top-level design. For more information, see Overview of Mixed Signal Interoperability.
Pcell	Adds the *Pcell* menu to the layout editor menu bar to let you create graphical Pcells. For more information, see *Creating Graphical Parameterized Cells*.
Simulation: NC-Verilog	Opens the Virtuoso Verilog Environment for NC-Verilog Integration application. For more information, see About the NC-Verilog Integration Environment.
Simulation: Other	Adds the *Simulation* menu to the layout editor menu bar. For more information, see About the NC-Verilog Integration Environment.
Simulation: Spectre	Adds the *Spectre* menu to the layout editor menu bar. This application provides a solution for digital circuit designers to netlist a Virtuoso transistor-level schematic for the Spectre circuit simulator. For more information, see Virtuoso Schematic Editor Spectre Interface.
Simulation: UltraSim	Adds the *UltraSim* menu to the layout editor menu bar. This product provides a solution for digital circuit designers to netlist a Virtuoso transistor-level schematic for the Virtuoso UltraSim circuit simulator. For more information, see Virtuoso Schematic Editor UltraSim Interface.

© 2023 Cadence Design Systems, Inc.

Plug-In	Description
VPS-L	Adds the *VPS-L* menu to the layout editor. The Virtuoso Power System L application provides a view of IR drop, electromigration (EM), and signal net EM analysis.
Voltus-Fi-L	Opens the Voltus-Fi-L application, which uses the extracted view to perform IR drop and Electromigration (EM) analysis. For more information, see Getting Started.
Voltus-Fi-XL	Opens the Voltus-Fi-XL application, which performs multi-mode simulation (MMSIM), electromigration, and voltage drop (EMIR) analysis for power and signal nets using the simulation database generated by Spectre APS/XPS. For more information, see Getting Started with Voltus-Fi-XL.
Voltus-XFi Results Browser	Adds the Voltus-XFi Results Browser assistant, which you can use to analyze the EM and IR drop violations in detail. For more information, see EM-IR Analysis Results.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

E

Bindkeys in Virtuoso Layout Suite

To view the bindkeys commands in Virtuoso Layout Suite, see Bindkeys.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.