

# **Allegro® Design Entry HDL User Guide**

**Product Version 23.1**  
**September 2023**

© 2023 Cadence Design Systems, Inc. All rights reserved.

Portions © Apache Software Foundation, Sun Microsystems, Free Software Foundation, Inc., Regents of the University of California, Massachusetts Institute of Technology, University of Florida . Used by permission. Printed in the United States of America.

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

Allegro Design Entry HDL contains technology licensed from, and copyrighted by: Apache Software Foundation, 1901 Munsey Drive Forest Hill, MD 21050, USA © 2000-2005, Apache Software Foundation. Sun Microsystems, 4150 Network Circle, Santa Clara, CA 95054 USA © 1994-2007, Sun Microsystems, Inc. Free Software Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA © 1989, 1991, Free Software Foundation, Inc. Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, © 2001, Regents of the University of California. Daniel Stenberg, © 1996 - 2006, Daniel Stenberg. UMFPACK © 2005, Timothy A. Davis, University of Florida, (davis@cise.ulv.edu). Ken Martin, Will Schroeder, Bill Lorensen © 1993-2002, Ken Martin, Will Schroeder, Bill Lorensen. Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts, USA © 2003, the Board of Trustees of Massachusetts Institute of Technology. vtkQt, © 2000-2005, Matthias Koenig. All rights reserved.

**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.

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.

**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

---

<b>Preface</b> .....	19
<u>About This User Guide</u> .....	19
<u>Finding Information in This User Guide</u> .....	19
<u>Related Documentation</u> .....	20
<u>Design Entry HDL</u> .....	20
<u>Front-to-Back Flow</u> .....	21
<u>Related Tools and Flows</u> .....	21
<u>Typographic and Syntax Conventions</u> .....	22
<b>1</b>	
<b><u>About Design Entry HDL</u></b> .....	23
<u>Design Entry HDL Features</u> .....	23
<u>Design Entry HDL in the Design Flow</u> .....	25
<b>2</b>	
<b><u>Getting Started</u></b> .....	27
<u>Starting Design Entry HDL</u> .....	27
<u>Design Entry HDL User Interface</u> .....	29
<u>Menu Bar</u> .....	30
<u>Toolbars</u> .....	32
<u>Status Bar</u> .....	37
<u>Console Command Window</u> .....	37
<u>Context-Sensitive Menus</u> .....	37
<u>Design Entry HDL Tasks</u> .....	38
<u>Creating a Schematic</u> .....	38
<u>Creating a Hierarchical Design</u> .....	39
<u>Design Entry HDL Frequently Asked Questions (FAQs)</u> .....	40
<u>Where can I enter commands?</u> .....	40
<u>Where are the setup options?</u> .....	40
<u>How do I pan drawings?</u> .....	41

# Allegro Design Entry HDL User Guide

---

<u>How do I zoom in and out of a drawing?</u>	41
<u>How do I customize Design Entry HDL?</u>	41
<u>What commands can I use to edit schematic text?</u>	41
<u>Are there menu shortcuts?</u>	42
<u>How do I browse drawings and components?</u>	42
<u>How do I add libraries?</u>	43
<u>How do I add notes?</u>	43
<u>How do I add parts?</u>	43
<u>How do I connect parts?</u>	43
<u>How do I name signals?</u>	43
<u>How do I rename signals?</u>	43
<u>How do I add properties?</u>	46
<u>How do I add ports?</u>	46
<u>How do I check my drawing for errors?</u>	46
<u>How do I save a design?</u>	46
<u>What is page locking?</u>	47
<u>How do I add additional pages to a design?</u>	47
<u>How do I go to a specific page in a design?</u>	47
<u>How do I plot a design?</u>	48
<u>What are groups?</u>	48
<u>What is different about working with groups?</u>	48
<u>How do I locate parts and wires in a design?</u>	48
<u>How do I generate a symbol view from a schematic?</u>	48
<u>How do I package my design?</u>	49
<u>How do I backannotate a design?</u>	49
<u>How do I highlight objects in a design?</u>	49
<u>How do I cross-reference a design?</u>	49
<u>Can Cross Referencer place location designators inside the schematic page border?</u>	50
<u>How do I archive a design?</u>	50
<u>How do I view bias point values in Design Entry HDL?</u>	50
<u>What can I do if copied instances and nets are not on the grid?</u>	51
<u>Can I extract the complete design information of a schematic to a .csv file?</u>	52
<u>Can I open a datasheet link for a part from a schematic?</u>	52
<u>How can I show the attachment lines of component properties while moving them on a schematic?</u>	53

<u>Can attach props.cfg be used at the CDS SITE level?</u>	55
<b>3</b>	
<b><u>Project Creation and Setup</u></b>	57
<u>Introduction to Project Manager</u>	57
<u>Project Structure</u>	58
<u>Project Files</u>	59
<u>Local Project Files</u>	60
<u>Site Project File</u>	61
<u>Installation Project File</u>	61
<u>User-Specific Project Settings File</u>	62
<u>Project Flows</u>	62
<u>Creating a Project</u>	62
<u>Files Created for Your New Project</u>	64
<u>Creating a Site Project File</u>	66
<u>Setting Up a Project</u>	68
<u>Changing the Root Design for a Project</u>	69
<u>Creating a New Root Design for a Project</u>	69
<u>Editing the cds.lib File</u>	70
<u>Selecting Libraries for a Project</u>	72
<u>Adding Physical Part Table Files to a Project</u>	73
<u>Setting Up Tools</u>	75
<u>Specifying the Application Temp Directory</u>	75
<u>Selecting a Text Editor</u>	76
<u>Selecting a Property File</u>	76
<u>Setting Up a Log File</u>	77
<u>Selecting an Expansion Type</u>	78
<u>Selecting the Configuration for Expansion</u>	79
<u>Editing a Configuration</u>	80
<u>Creating a New Configuration View</u>	80
<u>Selecting Views for the Project</u>	80
<u>Locking Project File Directives</u>	82
<u>Controlling Project Settings at Different Levels</u>	82
<u>Locking a Directive</u>	82
<u>Locking Support for Directives at Different Levels</u>	83

<u>Reading Settings from CPM Files at Different Levels</u>	84
<b>4</b>	
<b><u>Design Entry HDL Editing Environment</u></b>	87
<u>Setting Up Defaults</u>	88
<u>Setting Up Design Entry HDL Editor Options</u>	89
<u>Defining a Default Text Editor</u>	91
<u>Basic Editing Tasks</u>	91
<u>Undoing an Operation</u>	92
<u>Moving Objects</u>	92
<u>Copying Objects</u>	94
<u>Deleting Objects</u>	96
<u>Changing the Color of Objects</u>	96
<u>Drawing an Arc</u>	97
<u>Drawing a Circle</u>	97
<u>Splitting Overlaid Objects</u>	98
<u>Displaying the Console Window</u>	98
<u>Editing Text in Dialog Boxes and the Console Window</u>	98
<u>Copying Parts of a Schematic Across Pages, Designs, and Projects</u>	99
<u>Displaying Information</u>	111
<u>Displaying Schematic Information</u>	111
<u>Displaying Toolbars and Other Parts of the Design Entry HDL Window</u>	115
<u>Highlighting Objects</u>	116
<u>Turning Off Highlighting</u>	117
<u>Opening the Markers Control Window</u>	118
<u>Displaying the Markers Toolbar</u>	118
<u>Displaying the Error Status Bar</u>	118
<u>Using the QuickPick Browser</u>	119
<u>Overview</u>	119
<u>QuickPick Browser - Interface Overview</u>	119
<u>Setting up QuickPick Browser</u>	120
<u>Working with QuickPick Toolbar</u>	122
<u>Working with the QuickPick Browser Window</u>	123
<u>Filtering Information in the QuickPick Browser Window</u>	124
<u>Basic Navigation in Design Entry HDL</u>	125

# Allegro Design Entry HDL User Guide

---

<u>Panning the Drawing</u>	125
<u>Zooming In and Out of the Drawing</u>	126
<u>Navigating the Drawing Hierarchy</u>	126
<u>Moving a Window</u>	127
<u>Resizing a Window</u>	127
<u>Closing a Window</u>	128
<u>Navigating a Design</u>	128
<u>Exiting Design Entry HDL</u>	131
<u>Running Commands with Strokes</u>	131
<u>Guidelines for Strokes</u>	132
<u>Support for Fonts</u>	133
<u>Design Entry HDL Options - Font page</u>	137
<u>Setting Fonts using CPM Directives</u>	140
<u>Working with Fonts</u>	142
<u>Fonts on Different Platforms</u>	143
<u>Uprev of an Existing Design</u>	143
<u>Single Mode operation in DE-HDL</u>	144
<u>Support for Common Windows Commands and Operations</u>	145
<u>Windows Mode</u>	145
<u>Menus in the Windows Mode</u>	147
<u>Design Entry HDL Options Dialog Box</u>	148
<u>Support for Keyboard Operations</u>	148
<u>Bounding Box on Components</u>	150
<u>Global Navigate Window</u>	151
<u>Properties Window</u>	151
<u>Displaying Unconnected Pins on Components</u>	152
<u>Wire Selection</u>	154
<u>Anchor Point Stretch</u>	154
<u>Alignment and Distribution</u>	156
<u>Selection Filters</u>	160
<u>Object Visibility Layers</u>	161
<u>Page Search Toolbar</u>	164
<u>Searching Design Objects</u>	166

### 5

<u><a href="#">Creating a Schematic</a></u> .....	171
<u><a href="#">Creating a Project</a></u> .....	171
<u><a href="#">Starting Design Entry HDL</a></u> .....	172
<u><a href="#">Creating a Design Page</a></u> .....	174
<u><a href="#">Adding Page Borders</a></u> .....	174
<u><a href="#">Managing Grid Settings</a></u> .....	176
<u><a href="#">Using Part Information Manager</a></u> .....	178
<u><a href="#">Allegro EDM and Standard Mode</a></u> .....	180
<u><a href="#">Accessing the Allegro EDM Component Database from DE-HDL</a></u> .....	181
<u><a href="#">Adding a Part</a></u> .....	181
<u><a href="#">Replacing a Part</a></u> .....	182
<u><a href="#">Modifying a Part</a></u> .....	182
<u><a href="#">Adding Data Tips for Components</a></u> .....	183
<u><a href="#">Configuring Physical Property Options</a></u> .....	187
<u><a href="#">Overview</a></u> .....	187
<u><a href="#">Understanding Options Sets</a></u> .....	187
<u><a href="#">Getting Started with Property Options</a></u> .....	187
<u><a href="#">Customizing PTF Options Filters</a></u> .....	193
<u><a href="#">Connecting Parts</a></u> .....	194
<u><a href="#">Drawing a Wire Manually</a></u> .....	195
<u><a href="#">Auto-Routing a Wire</a></u> .....	195
<u><a href="#">Naming Signals</a></u> .....	195
<u><a href="#">Signal Naming Conventions</a></u> .....	196
<u><a href="#">Step Size in Signal Names</a></u> .....	199
<u><a href="#">Adding Properties</a></u> .....	201
<u><a href="#">Using PORT Symbols</a></u> .....	203
<u><a href="#">Rules for Using Port Symbols</a></u> .....	204
<u><a href="#">Port Association Restrictions</a></u> .....	205
<u><a href="#">Working with Ports and Signals</a></u> .....	205
<u><a href="#">Setting the Initial Value of a Signal</a></u> .....	206
<u><a href="#">Specifying the Size of Nets</a></u> .....	208
<u><a href="#">Specifying the Assertion Level of Pins and Signals</a></u> .....	209
<u><a href="#">Creating an Alias for a Signal</a></u> .....	209
<u><a href="#">Declaring a Base Signal</a></u> .....	211

<u>Global Signals</u>	213
<u>Unnamed Signals</u>	220
<u>Signal Concatenations</u>	221
<u>Signal Replication</u>	223
<u>Merge Symbols</u>	224
<u>Signal Slices (Bit and Part Selects)</u>	228
<u>Setting the Verilog Logic Type for Ports and Signals</u>	231
<u>Setting the VHDL Logic Type for Ports and Signals</u>	234
<u>Specifying Ranges for Ports, Signals and Aliases</u>	238
<u>Unconstrained Ranges for Ports, Signals, and Aliases</u>	239
<u>Resolved Types and Resolution Functions</u>	239
<u>Type Conversion</u>	240
<u>Abstract Data Types in VHDL</u>	241
<u>Using Iterated Instances</u>	242
<u>Saving a Design</u>	243
<u>Working With Existing Designs</u>	244
<u>Opening a Drawing</u>	244
<u>Recovering a Drawing</u>	245
<u>Reverting to the Previous Saved Version of a Drawing</u>	245
 <u>6</u>	
<u>Working with Wires</u>	247
<u>About Signals and Connectivity</u>	247
<u>About Bus Taps</u>	248
<u>About Bus Names</u>	248
<u>Drawing a Wire Manually</u>	249
<u>Auto-Routing a Wire</u>	250
<u>Stretching a Wire</u>	250
<u>Bending a Wire</u>	250
<u>Splitting a Wire</u>	250
<u>Snapping a Wire to the Nearest Pin</u>	250
<u>Naming a Signal</u>	251
<u>Wiring Bus-Through Pins</u>	251
<u>Marking Wire Connections</u>	252
<u>Naming Signals on a Bus</u>	252

<u>Specifying a Tap Symbol</u>	254
<u>Attaching Values to Bus Taps</u>	254
<u>Changing Wire Thickness and Pattern</u>	255

## 7

### Working with Libraries and Components

---

<u>About the Standard Library</u>	257
<u>Working with Libraries</u>	257
<u>Adding New Libraries</u>	258
<u>Browsing Libraries</u>	258
<u>Adding Libraries to the Search Stack</u>	259
<u>Removing Libraries from the Search Stack</u>	259
<u>Defining Library Search Order</u>	260
<u>Working with Components</u>	261
<u>Browsing the Component List</u>	262
<u>Creating Design Entry HDL Parts</u>	262
<u>Creating a Symbol in Design Entry HDL</u>	263
<u>Creating Entity Declarations from Symbols</u>	265
<u>Creating the chips.prt File</u>	268
<u>Creating a Part Table File</u>	270
<u>Adding a Component</u>	270
<u>Locking Components</u>	271
<u>Compatible Footprints Check</u>	274
<u>Using Compatible JEDEC_TYPEs</u>	274
<u>How compatible JEDEC_TYPEs Work?</u>	275
<u>Modifying Components</u>	276
<u>Replacing a Component</u>	276
<u>Breaking Up a Component</u>	278
<u>Changing Pin States on a Component</u>	279
<u>Choosing a Version of a Component</u>	279
<u>Mirroring Components or Blocks</u>	279
<u>Changing the Orientation of Components or Text</u>	280
<u>Sectioning a Component</u>	280
<u>Sectioning Multiple Components</u>	281
<u>Unsectioning Components</u>	286

## Allegro Design Entry HDL User Guide

---

<u>Swapping Pins on a Component</u>	287
<u>Ways to Determine if a Component Has Bus-Through Pins</u>	287
<u>Deleting a Library Component (Cells, Views, and Files)</u>	287
<u>Creating a Page Border Symbol</u>	288
<u>Customizing a Page Border in the Standard Library</u>	288
<u>Creating a Page Border of Your Own</u>	290
<u>Using Component Revision Manager</u>	293
<u>Overview</u>	293
<u>How Component Revision Manager Works</u>	293
<u>Getting Started with Component Revision Manager</u>	294
<u>Using Component Revision Manager</u>	296
<u>Setting Default View</u>	301

## 8

<u>Working with Properties and Text</u>	303
<u>About Properties</u>	303
<u>Locking Properties</u>	304
<u>Copying Properties</u>	305
<u>Moving Properties</u>	306
<u>Adding Properties</u>	306
<u>Making an Attribute File</u>	311
<u>Adding a URL</u>	311
<u>Adding Text</u>	313
<u>Modifying Text</u>	314
<u>Changing the Text Editor</u>	316
<u>Adding Port Names from the Corresponding Symbol</u>	316
<u>Swapping Notes or Properties</u>	316
<u>Reattaching a Property from One Object to Another</u>	316
<u>Specifying Property Display for a Specific Object</u>	317
<u>Specifying Property Display for Objects Globally</u>	317
<u>Specifying Visibility of Pin Properties</u>	318
<u>Specifying Visibility of Pin Numbers</u>	320
<u>Master and In-context Properties</u>	321
<u>Adding Properties to a Flat Design or Root (Top) Level Design</u>	323
<u>Adding Properties in Non-Replicated Designs</u>	329

## Allegro Design Entry HDL User Guide

---

<u>Adding Properties in Replicated Designs</u>	337
<u>Editing Existing Properties</u>	347
<u>Deleting Properties</u>	349
<u>Case-Sensitivity in Property Names and Values</u>	349
<u>Case Insensitive Property Values</u>	350
<u>Case Insensitive Property Names</u>	351
<u>Case Sensitivity and PPTs</u>	353
<u>Constructing PPTs with Case-Sensitive Values</u>	354
<u>Assigning Power Pins</u>	355
<u>Viewing Properties on Power Pins</u>	355
<u>Using the Assign Power Pins Dialog Box</u>	362
<u>Controlling the Overwriting of POWER_PINS Property</u>	369
<u>Subtype Names in POWER_GROUP Property</u>	372
<u>Working with Text Macros</u>	372
<u>About Text Macros</u>	373
<u>How to use Text Macros</u>	373
<u>Defining Text Macros on a Drawing Using the DEFINE Symbol</u>	374
<u>Defining Text Macros Using \PARAMETER or \PARAM</u>	375
<u>Defining a Text Macro in a File</u>	376
<u>Working with Custom Text</u>	377
<u>Using Custom Text</u>	382

9

<u>Working with Block Designs</u>	385
<u>About Blocks</u>	385
<u>About View Generation in Hierarchical Designs</u>	385
<u>Generating Views for Top-Down Design</u>	386
<u>Generating Views for Bottom-Up Design</u>	386
<u>Creating Hierarchical Designs</u>	386
<u>Top Down Method</u>	387
<u>Bottom Up Method</u>	389
<u>Adding a Block</u>	389
<u>Using Read-only Blocks in Your Design</u>	392
<u>Navigating the Drawing Hierarchy</u>	392

### 10

<b>Working with Hierarchical Split Symbols</b> .....	399
<u>Overview – Hierarchical Split Symbols</u> .....	399
<u>Creating Split Symbols for Hierarchical Blocks</u> .....	400
<u>Assigning Ports to Split Symbols</u> .....	406
<u>Using Hierarchical Split Symbols in a Design</u> .....	410
<u>Editing Split Symbols</u> .....	416

### 11

<b>Working with Designs</b> .....	423
<u>Finding Nets and Cells in Your Design</u> .....	423
<u>Navigating Nets in Your Design</u> .....	425
<u>Global Modification</u> .....	426
<u>Modifying Component, Pin, and Net Properties</u> .....	426
<u>Deleting Component, Pin, and Net Properties</u> .....	430
<u>Selecting the Component to Replace from the Physical Part Filter</u> .....	432
<u>Selecting the New Component from the Physical Part Filter</u> .....	434
<u>Selecting the Component to Replace from the Design (Schematic Pick)</u> .....	436
<u>Selecting the New Component from the Design (Schematic Pick)</u> .....	437
<u>Erroneous Conditions</u> .....	441
<u>Design Entry HDL Issues Affecting the Global Modify Solution</u> .....	445
<u>The Batch Mode Operation</u> .....	445
<u>Part Manager</u> .....	447
<u>How Part Manager Works</u> .....	447
<u>The Part Manager User Interface</u> .....	448
<u>Working with Part Manager</u> .....	453
<u>Updating Part Instances</u> .....	454
<u>Updating All Part Instances with Mismatched Injected Properties</u> .....	456
<u>Applying Changes to Part Instances</u> .....	456
<u>Resetting Changes Made to Part Instances</u> .....	457
<u>Updating Multiple Parts with a Single PTF Row</u> .....	457
<u>Refreshing Contents of Part Manager</u> .....	458
<u>Highlighting Part Instances on the Schematic</u> .....	459
<u>Displaying the Canonical Path</u> .....	459

# Allegro Design Entry HDL User Guide

---

<u>Running Scripts</u>	460
<u>Running a Script</u>	462
<u>Stopping a Script</u>	463
<u>Sample Scripts</u>	463
<u>Highlighting (Cross-Probing) Objects</u>	464
<u>Distributing Design Changes between Physical and Logical Designs</u>	465
<u>Applying Connectivity Changes from the Physical Design to Your Schematic</u>	465
<u>Back Annotating Your Design</u>	466
<u>Module Ordering</u>	466
<u>Module Ordering in Design Entry HDL</u>	466
<u>Using the Hierarchy Viewer</u>	467
<u>Excluded Modules File</u>	470
<u>Retaining Module Ordering While Importing Blocks</u>	471
<u>Sheet Names in Hierarchy Viewer</u>	472
<u>Setting Up Page Numbers</u>	474
<u>Sheet Name Input in New Designs</u>	475
<u>Performing Page Management Operations</u>	476
<u>Inserting a Page</u>	476
<u>Deleting a Page</u>	480
<u>Compressing Pages</u>	482
<u>Moving a Page</u>	482
<u>Definitions</u>	485
<u>Points to Remember</u>	485
<u>Displaying and Working with Schematic Page Numbers</u>	487
<u>Displaying Page Numbers in a Schematic</u>	487
<u>Modifying Custom Text for Page Numbers</u>	491
<u>Updating Custom Text Variables for Page Numbers</u>	492
<u>Managing Schematic Pages</u>	492
<u>Importing Designs</u>	497
<u>Overview</u>	497
<u>Importing a Design Sheet</u>	497
<u>Re-importing a Block</u>	503
<u>Re-importing Read-only Blocks</u>	504
<u>Baselining a Design</u>	506
<u>How Baselining Works</u>	506
<u>Setting Preferences for Metadata Creation</u>	507

<u>Baselining a Design</u>	508
<u>Creating the Table of Contents for a Design</u>	511
<u>Adding the Table of Contents to a Design</u>	512
<u>Creating a Symbol for TOC</u>	518
 <b>12</b>	
<b><u>Netlisting Your Design</u></b>	523
<u>Netlisting for Simulation</u>	524
<u>Netlisting for Digital Simulation</u>	524
<u>Netlisting for Analog and Mixed Signal Simulation</u>	529
<u>Generating Netlist in ISCF format</u>	530
 <b>13</b>	
<b><u>Managing Variants in Design Entry HDL</u></b>	533
<u>New Variant Menu Options in Schematic Canvas</u>	534
<u>Creating Variants</u>	536
<u>Editing Variants</u>	537
<u>Removing Variants</u>	541
<u>Launching Variant Editor</u>	543
<u>Opening Variant Schematic View</u>	545
<u>Working with Hierarchical Variants</u>	546
<u>Marking Components for Variants</u>	552
<u>Removing Components from Variants</u>	555
<u>Marking Components as 'Do Not Install'</u>	556
<u>Marking Components as Preferred Components</u>	557
<u>Modifying Components</u>	558
<u>Adding Alternates for a Component</u>	560
<u>Modifying Properties</u>	561
<u>Replacing Components in Functions or Variants</u>	563

**14**

**Design Management in Design Entry HDL** ..... 565

**15**

**Plotting Your Design** ..... 567

Windows Plotting on Windows ..... 567

Setting Up Windows Plotting Options ..... 568

Previewing the Design ..... 571

Plotting the Design ..... 574

Plotting in Batch Mode ..... 576

Setting up HPF Plotting Options ..... 583

Plotting the Design ..... 589

Plotting the Design from the Console Window ..... 591

Hierarchical Plotting ..... 599

Changing the Order in Which Designs Are Plotted ..... 599

Plotting Hierarchical Designs ..... 601

Frequently Asked Questions in Plotting ..... 605

**16**

**Design Techniques** ..... 613

Introduction ..... 613

Flat Designs ..... 613

Creating a Flat Design ..... 614

Concurrent Engineering of Flat Designs ..... 615

Considerations of Flat Designs ..... 615

Structured Designs ..... 616

Creating a Structured Design ..... 616

Benefits of Structured Designs ..... 621

Considerations of Structured Designs ..... 621

Hierarchical Designs ..... 621

Creating a Hierarchical Design ..... 622

Creating Symbols ..... 624

Editing Symbols ..... 624

Benefits of Hierarchical Designs ..... 628

<u>Comparing Design Techniques</u>	628
------------------------------------	-----

## 17

<u>Simulating using PSpice Simulator</u>	631
--	-----

<u>What is PSpice?</u>	631
<u>Notes for Using Design Entry HDL with PSpice</u>	632
<u>Conversion of Old PSpice Projects to New Project Format</u>	632
<u>New Directory Structure for PSpice Projects</u>	634
<u>Design Entry HDL Bias Display with PSpice</u>	641
<u>What is a bias point value?</u>	641
<u>Bias Display in Design Entry HDL</u>	641
<u>Loading Bias Point Values</u>	643
<u>Displaying bias point values</u>	643
<u>Bias Point Display Settings</u>	646
<u>Split Part</u>	649
<u>Displaying PSpice Names</u>	650

## A

<u>Glossary</u>	651
-----------------	-----

## B

<u>Working with Groups</u>	661
<u>Creating a Group by Rectangle</u>	661
<u>Creating a Group by Polygon</u>	662
<u>Creating a Group by Specifying an Expression</u>	662
<u>Creating a Group by Including Objects</u>	663
<u>Grouping the Entire Schematic</u>	663
<u>Including Additional Objects in a Group</u>	663
<u>Excluding Objects from a Group</u>	663
<u>Setting the Current Group</u>	664
<u>Viewing Group Contents</u>	664
<u>Moving a Group</u>	664
<u>Rotating a Group</u>	665
<u>Rotating a Group of Properties</u>	665

## Allegro Design Entry HDL User Guide

---

<u>Spinning a Group</u>	665
<u>Mirroring a Group</u>	665
<u>Copying a Group</u>	668
<u>Deleting a Group</u>	669
<u>Specifying Color for a Group</u>	669
<u>Highlighting a Group on the Schematic</u>	669
<u>Replacing Components in a Group</u>	669
<u>Replacing Component Symbols in a Group (Versioning)</u>	670
<u>Breaking Up Components in a Group</u>	670
<u>Adding Properties to a Group</u>	670
<u>Specifying Property Display for a Group</u>	671
<u>Specifying Property Justification for a Group</u>	671
<u>Changing the Text Size in a Group</u>	671
<u>Modifying Components with the Same Part Name in a Group</u>	672
<u><b>Index</b></u>	673

# Preface

---

## About This User Guide

The *Allegro® Design Entry HDL User Guide* explains how to use the Allegro® Design Entry HDL schematic editor.

This user guide assumes that you are familiar with the development and design of electronic circuits at the system or board level.

## Finding Information in This User Guide

This user guide covers the following topics:

<b>See...</b>	<b>For Information About...</b>
<a href="#"><u>Chapter 1, “About Design Entry HDL”</u></a>	Lists the various features of Design Entry HDL
<a href="#"><u>Chapter 2, “Getting Started”</u></a>	Describes the Design Entry HDL user interface and basic steps for creating a schematic
<a href="#"><u>Chapter 3, “Project Creation and Setup”</u></a>	Introduction to creating and setting up design projects
<a href="#"><u>Chapter 4, “Design Entry HDL Editing Environment”</u></a>	Setting up and using the Design Entry HDL editing environment
<a href="#"><u>Chapter 5, “Creating a Schematic”</u></a>	Describes the tasks to be performed for creating a schematic
<a href="#"><u>Chapter 7, “Working with Libraries and Components”</u></a>	Describes how to use the default Design Entry HDL libraries and perform operations on components
<a href="#"><u>Chapter 6, “Working with Wires”</u></a>	Describes various operations that can be performed on wires in drawings

## Allegro Design Entry HDL User Guide

### Preface

---

---

#### See...

#### For Information About...

<a href="#"><u>Chapter 8, “Working with Properties and Text”</u></a>	Lists procedures for different types of properties and text in Design Entry HDL
<a href="#"><u>Chapter 9, “Working with Block Designs”</u></a>	Describes various operations that can be performed on blocks in a design
<a href="#"><u>Chapter B, “Working with Groups”</u></a>	Describes various operations that can be performed on groups of components
<a href="#"><u>Chapter 11, “Working with Designs”</u></a>	Describes various advanced features of Design Entry HDL
<a href="#"><u>Chapter 12, “Netlisting Your Design”</u></a>	Describes the different types of netlists that you can create for a Design Entry HDL design
<a href="#"><u>Chapter 15, “Plotting Your Design”</u></a>	Describes the different methods of plotting a Design Entry HDL design
<a href="#"><u>Chapter 16, “Design Techniques”</u></a>	Discusses the three different design techniques - flat, structured and hierarchical
<a href="#"><u>Chapter 17, “Simulating using PSpice Simulator”</u></a>	Describes how to use Design Entry HDL with the PSpice A/D simulator
<a href="#"><u>Appendix A, “Glossary.”</u></a>	The Design Entry HDL glossary

---

## Related Documentation

You can also refer to the following documentation to know more about related tools and methodologies:

### Design Entry HDL

- For information about new features in 17.2, see *Allegro Design Entry HDL: What’s New in Release 17.2-2016*.
- To learn Design Entry HDL, see *Allegro Design Entry HDL Tutorial*.
- To learn how to use Constraint Manager with Design Entry HDL, see *Allegro Constraint Manager with Design Entry HDL Tutorial*.
- For information about the SKILL interface to Design Entry HDL, see *Allegro Design Entry HDL SKILL Reference*.

## Front-to-Back Flow

- For information on the front-to-back flow for PCB design, see *Allegro Front-to-Back User Guide*.
- For information on the Design Synchronization solution, see *Design Synchronization and Packaging User Guide* and *Design Synchronization Tutorial*.
- For information about packaging your design, see *Packager-XL Reference*.
- For information on the Design Variance solution, see the *Design Variance User Guide* and *Design Variance Tutorial*.

## Related Tools and Flows

- For information on various PCB design working environments such as a team of designers working on a Design Entry HDL project, implementing FPGAs in designs, working with high-speed constraints, importing IFF files for radio-frequency designs, and reusing existing modules, see *PSD Design Flows*.
- To learn how to create new Design Entry HDL projects and define various settings, see *Project Manager User Guide*.
- To learn how to use Design Entry HDL utilities such as Cross Referencer, Archiver and BOM, see *Design Entry HDL Utilities User Guide*.
- For information on maintaining and modifying the Design Entry HDL digital libraries, see *PCB Librarian Expert User Guide*, *PCB Librarian User Guide*, and *Allegro Design Entry HDL Libraries Reference*.
- For information on the digital simulation interface provided by Design Entry HDL, see *Allegro Design Entry HDL Digital Simulation User Guide* and *Allegro Design Entry HDL Digital Simulation Tutorial*.
- For information on capturing electrical constraints in Constraint Manager, see *Allegro Constraint Manager User Guide*.
- For information on Design Entry HDL data management, see *Design Manager User Guide*.
- For information on Design Entry HDL Rules Checker, see *Allegro Design Entry HDL Rules Checker User Guide*.
- For information on creating custom interfaces to translate the HDL database into a format that can be used by an external system and to update the HDL database with changes from a physical design system, see *CAE Views Programming Guide*.

## Typographic and Syntax Conventions

This list describes the syntax conventions used for this user guide:

- literal      Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names.
- argument      Words in italics indicate user-defined arguments for which you must substitute a name or a value.
- |      Vertical bars (OR-bars) separate possible choices for a single argument. They take precedence over any other character.
- [ ]      Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices. You can choose one argument from the list.
- { }      Braces are used with OR-bars and enclose a list of choices. You must choose one argument from the list.

---

## About Design Entry HDL

---

Design Entry HDL is a design environment that supports behavioral and structural design descriptions captured in text and graphics. It incorporates block editing functions for quick architectural design.

Design Entry HDL organizes schematic information into pages. It captures and displays only one page of schematic information at a time. Design Entry HDL is a by-reference editor because it references all parts in a schematic from various libraries that reside at the reference or local area. A standalone, self-contained database of the libraries and the design can be created using the Archiver utility.

## Design Entry HDL Features

- A top-down (hierarchical) design that lets you quickly draw blocks and connect wires between blocks. A cross-view generator (Genview) to create blocks from HDL descriptions or automatically generate HDL text from high-level diagrams.
- A customizable user interface that lets you customize menus and toolbars, map keys to functions, and create new commands.
- A hierarchy editor that lets you view the structure of your design.
- An attribute editor that lets you annotate properties on a design to drive the physical layout.
- Integration with the Design Synchronization toolset. This toolset lets you view differences between your schematic and the board layout and then synchronize them.
- Cross-probing between Design Entry HDL and other Cadence tools.
- Support for design reuse. You can associate logical components with a layout section to create reusable components. This component can be reused in other areas in your design and also in the designs you create later.
- Integration with Design Entry HDL Rules Checker, an advanced rule checking and rule development system.

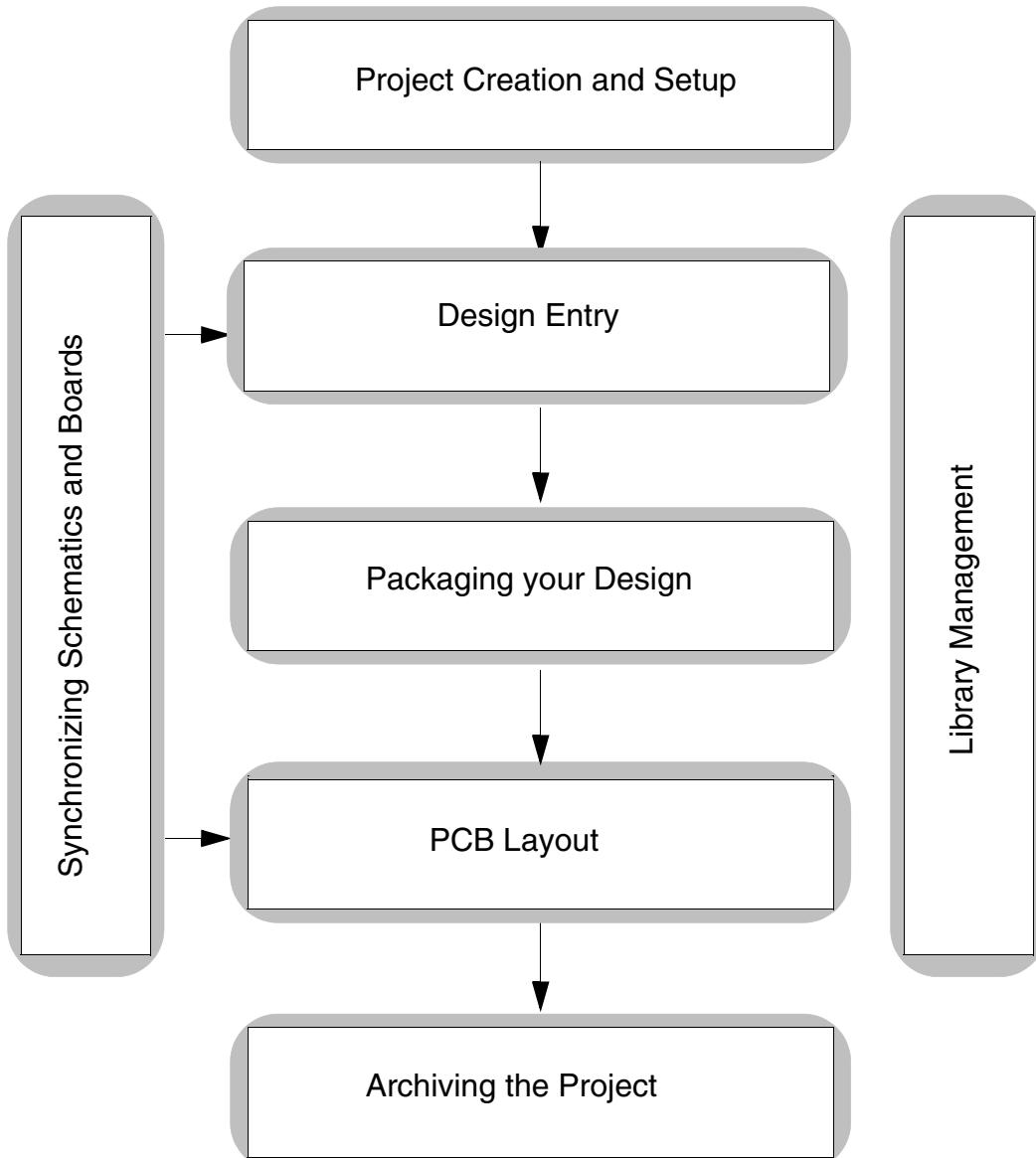
## **Allegro Design Entry HDL User Guide**

### About Design Entry HDL

---

- Integration with the Constraint Manager tool that allows you to capture and manage electrical constraints as you implement logic.
- Support for importing Intermediate File Format (IFF) files that can be created for radio-frequency (RF) designs. You can create radio-frequency (RF) designs using tools such as ADS or MDS by Agilent Technologies, Inc. The ADS tool supports the creation of Intermediate File Format (IFF) files. Once the RF design is ready, you can create IFF files for the schematic and layout of the design. You can import a schematic IFF file into Design Entry HDL to transfer the graphics and connectivity data of the RF design into Design Entry HDL and then use the RF design as a block in a larger Design Entry HDL design.
- Design Entry SKILL, the SKILL programming interface to Design Entry HDL.

## Design Entry HDL in the Design Flow



## **Allegro Design Entry HDL User Guide**

### About Design Entry HDL

---

---

## **Getting Started**

---

This chapter contains the following information:

- [Starting Design Entry HDL](#) on page 27
- [Design Entry HDL User Interface](#) on page 29
- [Design Entry HDL Tasks](#) on page 38
- [Design Entry HDL Frequently Asked Questions \(FAQs\)](#) on page 40

### **Starting Design Entry HDL**

After you open a design project in Project Manager, the Cadence Board Design flow is displayed in Project Manager. In the Board Design flow, click the *Design Entry* icon.

**Note:** You must be on the Common Desktop Environment (CDE) on a Sun workstation to run the Design Entry HDL set of tools.

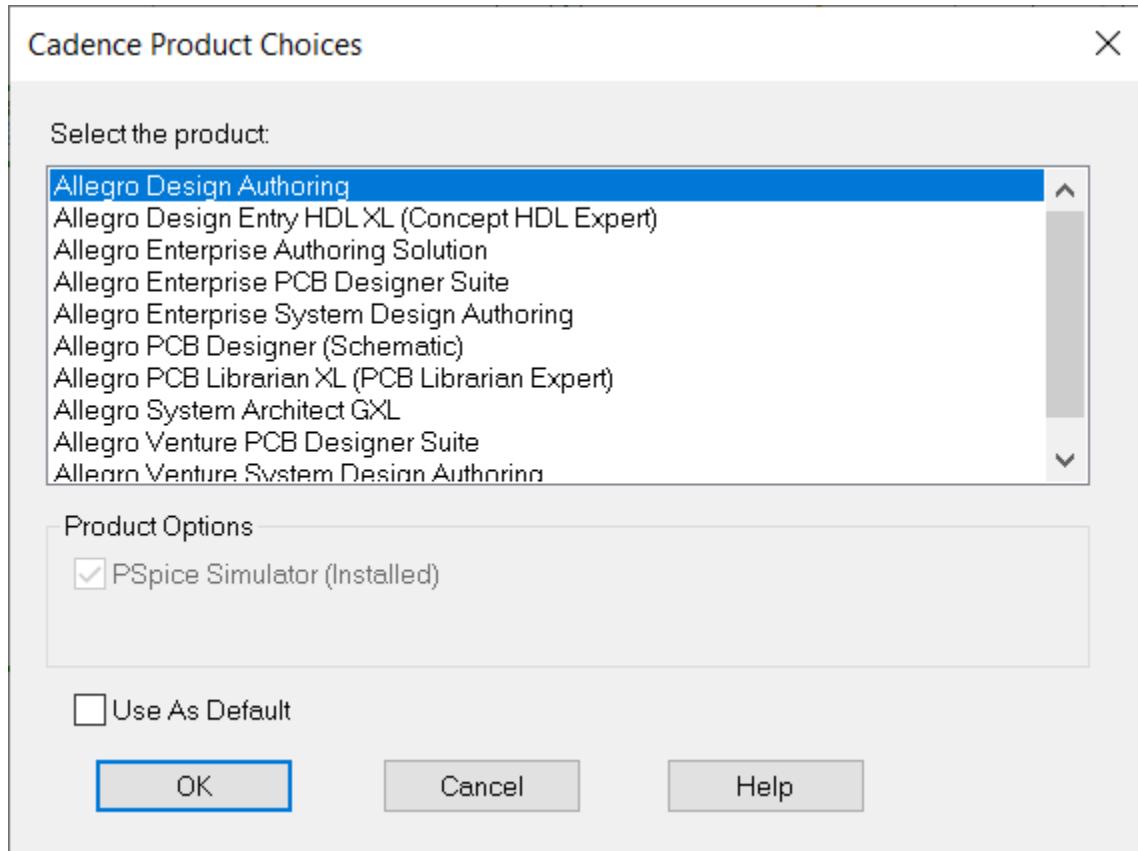
To launch Design Entry HDL, do one of the following:

- In Project Manager, open the `.cpm` file of a project and then click the *Design Entry* icon.
- Choose *Start – Cadence PCB 2022 – Design Entry HDL 2022*.
- At the command prompt, type `concepthdl` and press ENTER.

## Allegro Design Entry HDL User Guide

### Getting Started

The Cadence Product Choices dialog appears.



1. Select the product and product option from the Cadence Product Choices dialog
2. Click *OK*.

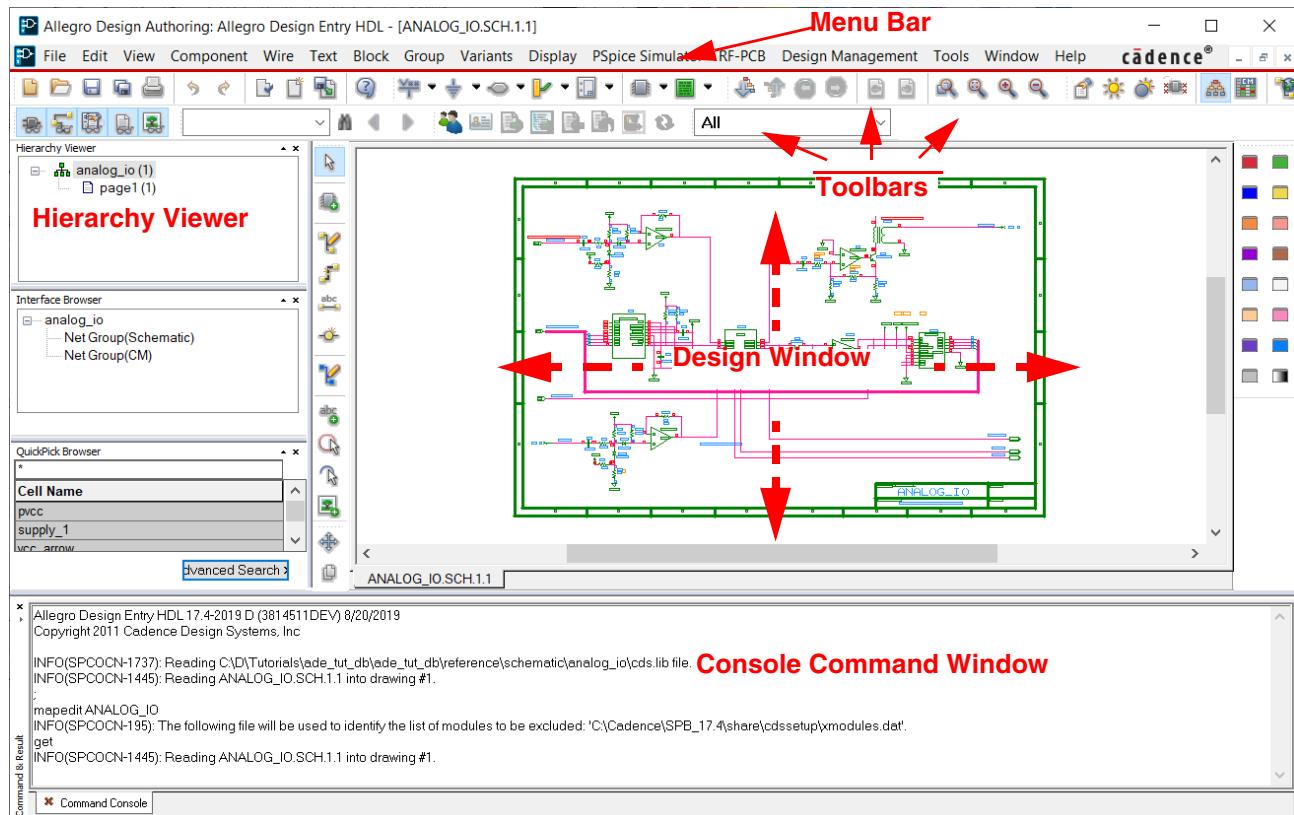
**Note:** If you open Design Entry HDL from the command line, you can use the `-product` parameter to prevent the Cadence Product Choices dialog box from appearing every time you run the command. For more information, refer to [Specifying Product Choice from Command Line](#) in Allegro Design Entry HDL Reference Guide.

## Allegro Design Entry HDL User Guide

### Getting Started

# Design Entry HDL User Interface

When you launch Design Entry HDL, the Design Entry HDL user interface appears as illustrated:



The Design Entry HDL interface consists of the following elements:

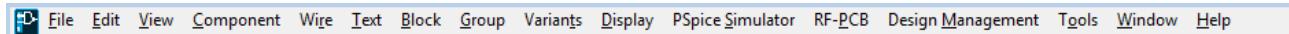
- Design window
- Menu bar
- Toolbars
- Status bar
- Global Navigation window
- Console command window
- Context-sensitive menus

# Allegro Design Entry HDL User Guide

## Getting Started

---

### Menu Bar



The Design Entry HDL menu bar includes the following menus:

- **File**  
For operations such as opening, saving, and plotting a drawing.
- **Edit**  
For operations such as Undo, Copy, Paste, Delete, Spin, and Color.
- **View**  
For operations such as Zoom, Pan, and Grid.
- **Component**  
For operations that can be done on a part such as adding, replacing, and modifying a part.
- **Wire**  
For operations such as connecting parts and naming signals.
- **Text**  
For operations such as adding properties and notes.
- **Block**  
For operations such as adding blocks.
- **Group**  
For operations such as creating groups and performing editing functions on groups.
- **Variants**  
For variant selection and viewing.
- **Display**  
For operations such as highlighting and de-highlighting components.
- **PSpice Simulator**  
For operations related to analog, digital and mixed-signal simulation using PSpice Simulator. This menu is visible only if you have installed PSpice Simulator.

## Allegro Design Entry HDL User Guide

### Getting Started

---

#### ■ RF-PCB

Opens the RF-PCB IFF Import dialog which enables you to import radio frequency design into your schematic.

#### ■ Design Management

Allows you enable team and data management for a design.

A team-enabled design allows multiple engineers to collaborate on a single design project facilitating collaboration between various groups. You can manage a multiblock hierarchical, and flat designs, with multiple users concurrently modifying portions of a logical and physical design.

Data management allows you to manage version control and version history for design changes by checking designs into SharePoint/Windchill.

This menu is visible only if you have the Allegro(R) Design Authoring Team Design Option and/or the Allegro Data Manager license.

#### ■ Tools

For operations such as setting up defaults, customizing, updating a schematic with layout changes, updating the layout with schematic changes, finding nets and instances in your design, global navigation, checking your design, and running scripts.

#### ■ Window

For operations such as opening a new window, cascading and tiling it.

#### ■ Help

Opens the Design Entry HDL help page and web resources such as Cadence Online Support and Education Services.



In the Windows Mode, menu bar and menu options are different from the normal mode. See Menus in the Windows Mode for more information.

### Toolbars

Design Entry HDL has the following toolbars:

- [Standard Toolbar](#)
- [Navigate Toolbar](#)
- [Tools Toolbar](#)
- [Block Toolbar](#)
- [Add Toolbar](#)
- [Edit Toolbar](#)
- [Color Toolbar](#)
- [Markers Toolbar](#)
- [Group Toolbar](#)
- [QuickPick Toolbar](#)
- [Object Visibility Layers Toolbar](#)
- [Page Search Toolbar](#)
- [Variant Toolbar](#)

If you have installed PSpice Simulator A/D, the following six additional toolbars are available. For more information on these toolbars, refer to *PSpice® User Guide*.

- Analog
- Passive
- Source
- Linear
- Discrete
- Misc

If you have the Allegro(R) Design Authoring Team Design Option or the Allegro Data Manager licenses, an additional toolbar, Design Management, is available.



## Allegro Design Entry HDL User Guide

### Getting Started

---

For more information on the tasks you can perform using this toolbar, refer to *Allegro Design Management User Guide*.

#### Standard Toolbar



You can use the *Standard* toolbar for the standard functions on a drawing (open, save, save all, print, undo, redo, check, expand, add new page, and import sheets).

#### Navigate Toolbar



You can use the *Navigate* toolbar to navigate a drawing (descend, ascend, previous drawing, next drawing, previous page, next page, zoom points, zoom fit, zoom in, and zoom out).

#### Tools Toolbar



You can use the *Tools* toolbar to perform actions such as displaying the Attributes form, highlighting, dehighlighting, showing/hiding unconnected pins, hiding/displaying Hierarchy Viewer, and launching Constraint Manager and Part Manager.

#### Block Toolbar



# Allegro Design Entry HDL User Guide

## Getting Started

---

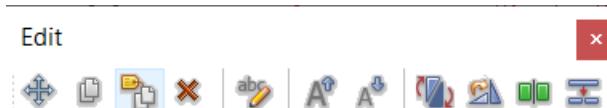
You can use the *Block* toolbar to add blocks, add pins on blocks and draw wires to connect blocks.

### Add Toolbar



You can use the *Add* toolbar to add objects (components, wires, and text) and graphics such as dots and circles.

### Edit Toolbar



You can use the *Edit* toolbar to perform edit operations such as copy, paste, delete, and spin.

### Color Toolbar



The *Color* palette lists the colors supported in Design Entry HDL and allows you to quickly change the colors of various objects.

### Markers Toolbar



The *Markers* toolbar helps you traverse through schematic errors.

# Allegro Design Entry HDL User Guide

## Getting Started

---

### Group Toolbar



The *Group* toolbar has all the commands for creating and modifying a group. A group is a collection of objects such as notes, components, wires, and properties.

### QuickPick Toolbar



The QuickPick toolbar helps you quickly add commonly-used cells, parts, and local blocks to a design.

### Object Visibility Layers Toolbar



You can use the object visibility toolbar to control the visibility of each object layer. The visibility of each of the object layers can be controlled by toggling the toolbar buttons.

### Page Search Toolbar



You can use the search toolbar to search for text on the current page. The text could be symbol text, a net name, property or part of a note.

This toolbar is by default set to search the current page. If you want to modify the scope of the search, use the Find dialog. See [Searching Design Objects](#) for details.

### Variant Toolbar



You can select the variant by clicking the *Variants* icon available in the Variant toolbar in Design Entry HDL. When you click this icon, a list of available variants is displayed. You can choose to view a specific variant from the list.

To view variant-specific information on the schematic, you need to open the design in Variant Editor and save the data once. On saving the design, Variant Editor generates files specific to each variant. These files contain variant-specific information and are used for dynamic display of data in DE-HDL schematics. Variant details can be defined at the time of creating a new variant or editing existing variant details. Once saved, use of the Variants icon and the *View – Variant* menu in Design Entry HDL list the variants in the design.

You need to ensure that both Variant Editor and Design Entry HDL are launched from the same instance of Project Manager. Otherwise, the variant data in these tools will not be in sync and there might discrepancies in the variant data being displayed.

## Status Bar

The status bar displays a single line about the action you are performing or when Design Entry HDL expects you to perform an action.

## Console Command Window

You can type commands in this window. The window can also be used to manually test any scripts that you have written for Design Entry HDL. To enable or disable the console command window, choose *View – Console Window*.

If you want to save the output in the command console to a file, set the following command: `console_dump on` in the console. This writes the console window text to a file named `consoledump.txt` in the temp area.

## Context-Sensitive Menus

Every object in Design Entry HDL has a context-sensitive menu attached to it. The menu appears when you right-click on the object. The menu contains options to perform certain operations that are relevant to the current object and its context. Examples of operations on a symbol are copy, delete, edit, and rotate.

## Design Entry HDL Tasks

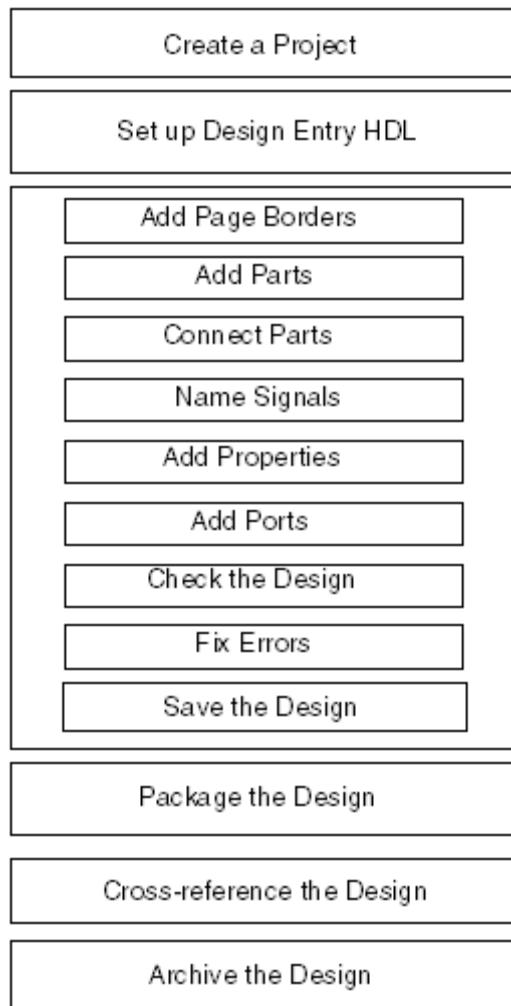
The Design Entry HDL tasks covered in this section are as follows:

- [Creating a Schematic](#) on page 38
- [Creating a Hierarchical Design](#) on page 39

### Creating a Schematic

The following figure illustrates the sequence of tasks you perform in Design Entry HDL to create a schematic.

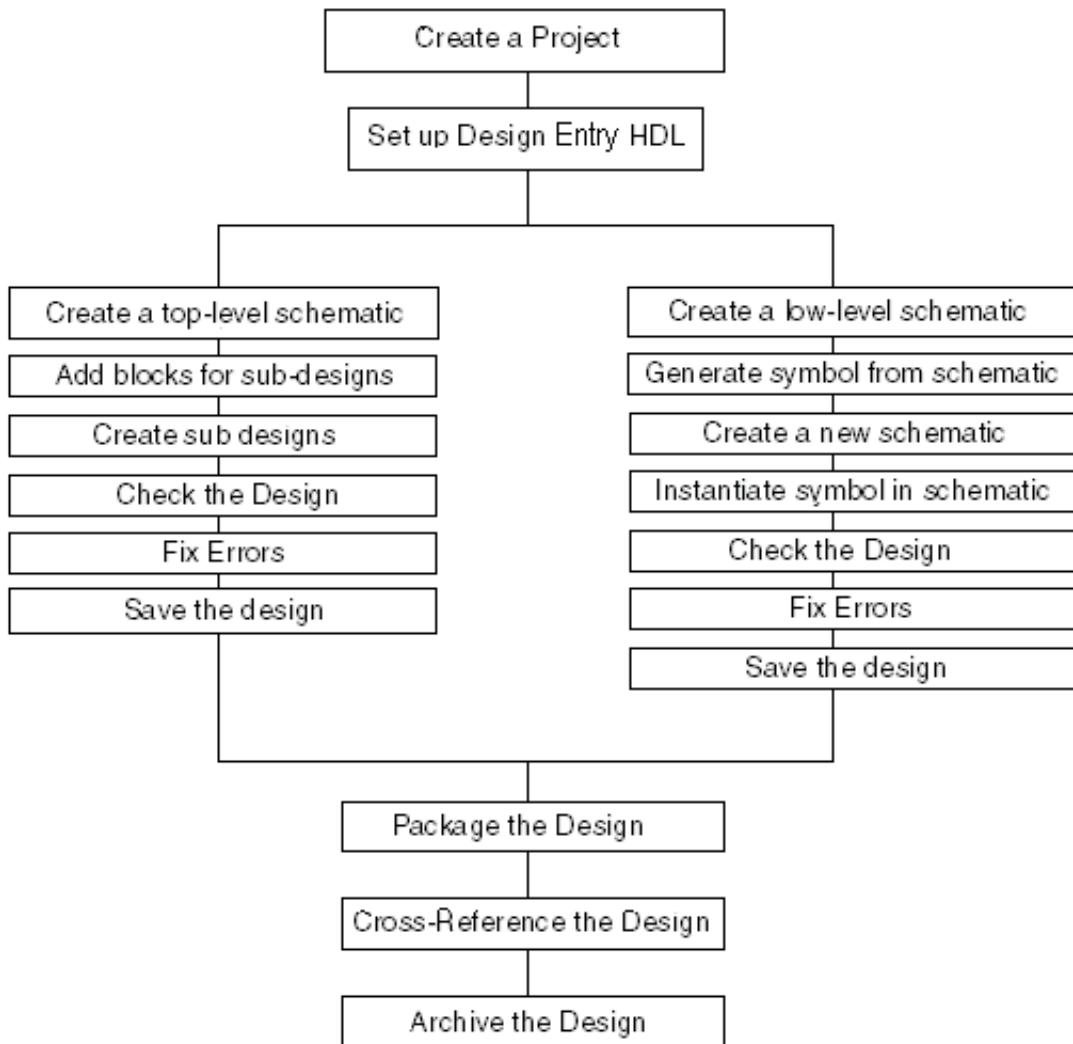
### Schematic Creation Tasks



### Creating a Hierarchical Design

The following figure illustrates the sequence of tasks you perform to create a hierarchical design.

#### Hierarchical Design Creation Tasks



## Design Entry HDL Frequently Asked Questions (FAQs)

This section answers common questions that are useful when you start working in Design Entry HDL.

### Where can I enter commands?

You can type commands in the console window that appears below the drawing area when you choose *View – Console Window*. If you exit Design Entry HDL with the console window option enabled, the console window will appear automatically the next time you start Design Entry HDL.

### Command Conventions and Entering Commands

Each menu item has an associated Design Entry HDL command. To run a command, do one of the following:

- Choose a command from a menu.
- Type a command in the console window, which appears below the drawing area when you choose *View – Console Window*.
- Click a toolbar icon.
- Press the control keys, which are noted next to the frequently used menu commands.
- Draw a stroke pattern.
- Write commands in a script file and run the script.

You can abbreviate Design Entry HDL commands. Design Entry HDL recognizes the smallest unique portion of the command name and arguments. Design Entry HDL commands are not case-sensitive.

### Where are the setup options?

Global setup options are located in the Project Manager. You can access Design Entry HDL setup options through the Project Manager and through the *Tools* menu in Design Entry HDL (*Tools – Options*).

## How do I pan drawings?

You can pan a drawing using the mouse, scroll bars, the keyboard, or the *View* menu.

## How do I zoom in and out of a drawing?

To zoom into a drawing, do one of the following:

- Choose *View – Zoom In*.
- Choose *View – Zoom Scale* and enter a scale factor such as 2.
- Choose *View – Zoom by Points* and stretch a rectangle around the area you want to zoom into then do the following:
  - a. Click slightly above and to the left or right of the objects you want to group.
  - b. Drag the cursor down diagonally from where you first clicked.
  - c. Click again.

To zoom out of a drawing, choose *View – Zoom Out* or *View – Zoom Scale* and enter a scale factor such as 0.5.

To fit a drawing in the screen, choose *View – Zoom Fit*.

## How do I customize Design Entry HDL?

You can customize toolbars, commands, menus, and keys in Design Entry HDL using *Tools – Customize*.

## What commands can I use to edit schematic text?

You can use the following keyboard commands when running the change command (*Text – Change*):

---

To	Press
Move the cursor backwards	Left Arrow
Move the cursor forward	Right Arrow

## Allegro Design Entry HDL User Guide

### Getting Started

---

To	Press
Move the cursor to the beginning of the line	Home - or - Right-click and select <i>Position at BOL</i> .
Move the cursor to the end of the line	End - or - Right-click and select <i>Position at EOL</i> .
Delete the previous character	BackSpace
Delete the next character	Del
Start a text editor	Ctrl + E - or - Right-click and select <i>Editor</i> .

---

### Are there menu shortcuts?

- Toolbars provide shortcuts to several functions. You can turn on any or all of the toolbars with the *View – Toolbars* menu command.
- Control keys also provide shortcuts to several menu commands. Control-key shortcuts are noted next to the frequently-used menu commands.
- Press predefined function keys (*F1-F12*).
- Standard Windows *Alt* key functions are also available.

### How do I browse drawings and components?

You can add and edit components using Part Information Manager by doing one of the following:

- Choose *File – Open* to display a file browser from which you select the drawing you want to edit.
- Choose *Component – Add* to display Part Information Manager from which you can select components to add to your drawing.

Part Information Manager appears. For details about Part Information Manager, see [Using Part Information Manager](#) on page 178 and refer to *Part Information Manager User Guide*.

## How do I add libraries?

You add libraries using *Tools – Setup* in Project Manager. Within Design Entry HDL, you can control the available library list and the search order for libraries using *File – View Search Stack*.

## How do I add notes?

You can add notes and attach them to the schematic using *Text – Note*.

## How do I add parts?

You can add parts using *Component – Add*.

## How do I connect parts?

You can connect parts with wires using *Wire – Draw* or *Wire – Route*. *Wire – Draw* lets you manually route around objects while *Wire – Route* automatically routes the wire around objects. Alternatively, right-click the component where you want to add the wire, and choose *Add Wire* from the pop-up menu.

## How do I name signals?

You can name signals using *Wire – Signal Name*. You can also create buses by naming signals in the appropriate manner. If you name a wire as `DATA<15..0>`, Design Entry HDL converts the wire to a 16-bit bus.

## How do I rename signals?

You can rename signals using the popup menu. Select the signal that you want to rename, right-click and use *Rename Signal*. When you rename a net, all its associated constraints and properties are retained.

You can also use the `_netrename` console command to rename a net using the following syntax: `_netrename <old_net_name> <new_net_name>`

When renaming a net, the net must be present in the design block in which it is being renamed. For example, when a local signal `CLK` is renamed in the `full_adder` block, the

## Allegro Design Entry HDL User Guide

### Getting Started

signal will only be renamed in `full_adder`. If the signal is in multiple pages, it will be renamed across all the pages.



Multiple-bit vector signals can be renamed only if the new vector signal has the same width.

After renaming nets, it is recommended that you perform an explicit Electrical Constraint Set (ECSet) audit in Constraint Manager.

Design Entry HDL does not support the following:

- Nets cannot be renamed when working with read-only pages or when the design is in use by another user in a team design environment.
- The signal scope cannot be modified when renaming a net. For example, a global signal cannot be renamed as a local signal, or vice versa. If, however, you rename a signal that involves a scope change, you will be prompted that the net will be renamed but the scope of the old signal will be retained. You can then proceed with or cancel the net renaming operation.
- Only entire buses with the same width can be renamed. Individual bus bits cannot be renamed. For example, you can rename `Z10<3..0>` to `A12<3..0>`, or `Z10<3..0>` to `A12<0..3>` but `Z10<0..3>` cannot be renamed to `A12<4..8>`.

---

**Before net rename**

**After net rename**

---

`Z10<3..0>`

`A12<3..0>`

## Allegro Design Entry HDL User Guide

### Getting Started

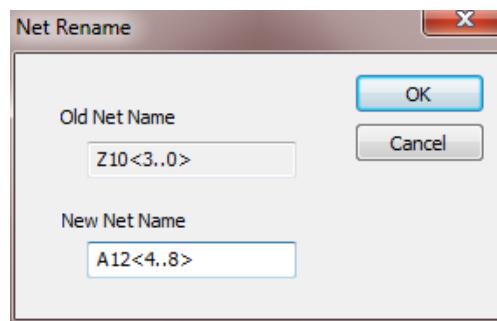
Before net rename

Z10<3..0>

After net rename

A12<0..3>

Z10<3..0>



When you try and change the width of a bus, Design Entry HDL displays the following message:



- You cannot rename scalar signals to single-bit vector signals and vice versa.



Net renaming operations are auto-saved. Any unsaved changes prior to the net rename auto-save are saved at the same time. You cannot undo a net renaming operation.

**Note:** You can change a net name for one instance or change it across a design. You can change a net name for one instance by selecting the net text and choosing *Text — Change*, or by right-clicking and choosing *Change*, or by using the Attributes form. When you change a net name for one instance, it is equivalent to deleting the net name and re-adding it, which deletes the constraints on the net. The constraints on that net are reset to the default.

When you rename a net across a design, the changed net name is propagated across the design and the constraint information is preserved. Choose the *Rename Signal* option to change a net name across a design.

If you often use methods other than the *Rename Signal* option to rename a net, even when you wanted to rename a net across a design, use the ASK\_RENAME\_SIGNAL\_OPTION directive. When set to ON, and a user tries to change a net name, DE-HDL prompts users to confirm that they want to change the name only for the instance or rename the net across the design.

## How do I add properties?

You can add properties on parts, pins, and signals using *Text – Property*. You can view, add, and modify the visibility of properties using *Text – Attributes*.

## How do I add ports?

You can use the ports available in the Standard Library using *Component – Add*.

## How do I check my drawing for errors?

- You control settings for error checks in the Design Entry HDL Setup options accessed through the *Tools* menu in Design Entry HDL.
- The *Tools – Check* menu choice or the Check icon in the Standard toolbar lets you run a check.
- You can view error messages and locate them in your design using the Markers control window (*Tools – Markers*). This window also lets you view long, detailed error messages that correspond with the short error messages that are typically displayed.
- Error Status Bar

Using the status bar in combination with the Markers toolbar, you can view short error messages without the Markers control window. *Tools – Error* controls to navigate the markers file.

## How do I save a design?

You can save a design using *File – Save*.

## What is page locking?

When a user who has write permissions is editing a page in a design, Design Entry HDL locks the page. If a second user opens the same page for editing, Design Entry HDL displays a message that the page is locked by the first user and that the second user cannot save any changes made in the page.

Conditions related to page locking are as follows:

- If the root design schematic page is a read-only page, or is locked by another user, then the *Save* option is completely disabled. You cannot make any in-context or schematic changes to the page.
- If the root schematic context property data file (dcf) is Read Only, then the context save is skipped. You will not be able to change the in-context attributes in DE-HDL. However, if a lower-level schematic page is read only or is locked by another user, then you can only make context changes. You will not be able to edit the schematic page. On saving the page, the dcf file is saved with the context changes.

Design Entry HDL creates a lock file called `pagen_csb.lock` in the schematic view when you open a schematic page.

## How do I add additional pages to a design?

Design Entry HDL supports multiple page schematics. Choose *File – Edit Page/Symbol – Add New Page* to add a new page to the schematic.

## How do I go to a specific page in a design?

1. Choose *File – Edit Page/Symbol – Go To*.

The *Go To Page/Symbol* dialog box appears.

2. Enter the page number and click *OK*.

To go to a specific page in a hierarchical design, select the *Calculate page number in hierarchy* check box, enter the page number and click *OK*.

**Note:** If you do not select the *Calculate page number in hierarchy* check box, you can only go to a page within the cell in which the currently open schematic page exists. For example, if the currently open schematic page is `LAPTOP.SCH.1.1`, you can only go to pages within the `LAPTOP` cell.

**Note:** You can also use the `gotosheet` console command to go to a specific page in a hierarchical design. When the *Calculate sheet number in hierarchy* option is selected, you are navigated to the sheet number. When this option is not selected, the specified page number is edited in the current cell. In this case, the sheet number used by the `gotosheet` command is the sequential numbering of pages in the entire design hierarchy, while the page number used by the `edit` command is the physical page number in the current block.

For more information on page numbering in Design Entry HDL, see [Displaying and Working with Schematic Page Numbers](#) on page 487.

## How do I plot a design?

You can plot a design using *File – Plot*.

## What are groups?

When you wish to perform a common edit operation like Copy, Move, or Delete on a collection of objects on the schematic, you can define the collection as a group and carry out the operation using the options available in the Group menu.

## What is different about working with groups?

- Functions for creating and working with groups are contained in one group menu.
- A separate toolbar contains the frequently-used group operations.
- Design Entry HDL makes it easy to set the current group. When you work with grouped items, it clearly indicates the group name in brackets next to the grouped menu items.

Design Entry HDL provides a *Group Contents* dialog box using which you can see the contents of the groups defined in the schematic.

## How do I locate parts and wires in a design?

You can locate parts and wires in a design using *Tools – Global Find*. You can also use wildcards on names and narrow down the search using properties and values.

## How do I generate a symbol view from a schematic?

You can generate symbol views from schematics using *Tools – Generate View*.

## How do I package my design?

You can open Packager-XL using the Design Synchronization tool of Project Manager. You can also use *File – Export Physical* in Design Entry HDL. For more information on packaging, see [Design Synchronization and Packaging User Guide](#).

## How do I backannotate a design?

Backannotation updates a schematic with layout changes. It annotates your schematic with physical information such as pin numbers and location designators produced by the Design Synchronization process. Choose *Tools – Back Annotate* to specify the file (typically `pstback.dat`) containing the physical information with which you update the schematic.



### Caution

***Do not run backannotation if any other user who has write permission is working on the design. Running backannotation when another user is working on the design results in incomplete backannotation.***

## How do I highlight objects in a design?

To highlight an object in a drawing, choose *Display – Highlight* and click on the object to be highlighted.

You might want to highlight objects in your design for the following reasons:

- To trace a signal on multiple pages of an expanded drawing
- To trace a signal in the drawing hierarchy between expanded drawings
- To correlate the circuit logic to changes you made in the schematic or to navigate the nets between a physical layout and the corresponding schematic between Design Entry HDL and other system tools.

Choose *Display – Dehighlight* to remove highlighting.

## How do I cross-reference a design?

When you view the plot of a schematic, it is often difficult to trace a signal or instances of a part. Cross Referencer traces the signals and parts in a schematic and annotates the location of each one.

On a cross-referenced design, Cross Referencer writes the page number and the location of the part or signal in relation to the page border. These annotations can be found beside each signal and part that has been cross-referenced.

Choose *Tools – CRefer* in Project Manager to cross-reference your design.



To generate flat cross references for nets in hierarchical blocks that are either instantiated multiple times, or are instantiated using split hierarchical symbols, you need to add offpage connectors to the nets.

## Can Cross Referencer place location designators inside the schematic page border?

In Cross Referencer, the outer boundary of a page border is the maximum limit to draw cross references on a schematic. To work around this, you can do one the following:

- Add location designators to offpage symbols and place them so as to stack over the offpage symbols. This will ensure that the cross references generated are placed according to the positions defined for the location designators in the offpage symbols.
- Place the nets slightly off the page border leaving enough space for cross references to be drawn inside the page border.

## How do I archive a design?

You can use the Archiver tool to archive your design. This tool copies over all the libraries that are referenced by your design to the archived area. Archiving lets you work on the design at a location where connectivity to the library server is not available.

To archive your design, choose *Tools – New Archive* in Project Manager.

## How do I view bias point values in Design Entry HDL?

You can enable the bias display feature of Design Entry HDL to view bias point information, such as bias point voltage, bias point current, and bias power on the schematic. To view bias point values on the schematic, do the following:

1. Load bias point values.

From the PSpice Simulator menu, choose *Bias Point – Preferences*. In the Bias Point Preferences dialog box, select the *Update Bias Point Information Automatically* check box and click *OK*.

#### 2. Choose *PSpice Simulator – Bias Points – Enable*.

Menu options for displaying bias point voltage, bias point current, and bias power are enabled.

#### 3. Specify the bias point information to be displayed on the schematic.

- To display bias point voltages on a schematic, choose *PSpice Simulator – Bias Points – Enable Bias Voltage Display*.
- To display bias currents on the schematic, choose *PSpice Simulator – Bias Points – Enable Bias Current Display*.
- To display bias power values, choose *PSpice Simulator – Bias Points – Enable Bias Power Display*.

**Note:** If you do not want the bias point values to be loaded automatically, skip 1. Instead, select *PSpice Simulator – Bias Points – Annotate Bias Values* whenever you want to load the latest bias point information on to the schematic.

To know more about the bias display feature in Design Entry HDL, see [Chapter 17, “Simulating using PSpice Simulator”](#).

## What can I do if copied instances and nets are not on the grid?

When you reuse schematic designs created by other users, nets or instances in the design might be off grid. This makes it difficult to wire components.

For DE-HDL to be able to place a symbol on the grid, the schematic grid setting should be compatible with the symbol grid setting. If the grid settings are incompatible, the pin-pitch (distance between the pins of a component) would not be compatible with the schematic grid. This can result in the symbol being placed off grid.

Therefore, ensure that the schematic grid is compatible with the grid settings of the symbol when the symbol was created.

To move off-grid components to the grid, you can also use the `_movetogrid` command. This command works on the currently opened schematic page.

DE-HDL iterates through each component in the page and moves it to the grid if the component is off the grid. All properties attached to a component are also moved to the grid.

DE-HDL reads the grid values as defined in the Design Entry HDL Options dialog (*Tools — Options*).

**Note:** If you want to revert the schematic page to its previous state, you will have to undo as many times as there are components on the page.

The command does not impact notes or dangling wires in the schematic. If a wire is connected to a component which is on grid but whose other end is unconnected and off the grid, the wire is not considered off grid and is not impacted by this command.

## Can I extract the complete design information of a schematic to a .csv file?

You can extract the complete design information from the schematic, — components, part numbers, connected signals, connectors pinout, and so on — into a .csv file using the `dsreportgen -ui` command.

In the *Generate Report* dialog, select the project file and create a new block-based template. You can create various kinds of report templates and generate reports.

For details, see the [Using the dsreportgen Command](#) and [Create Report Template](#) sections in *System Connectivity Manager User Guide*.

## Can I open a datasheet link for a part from a schematic?

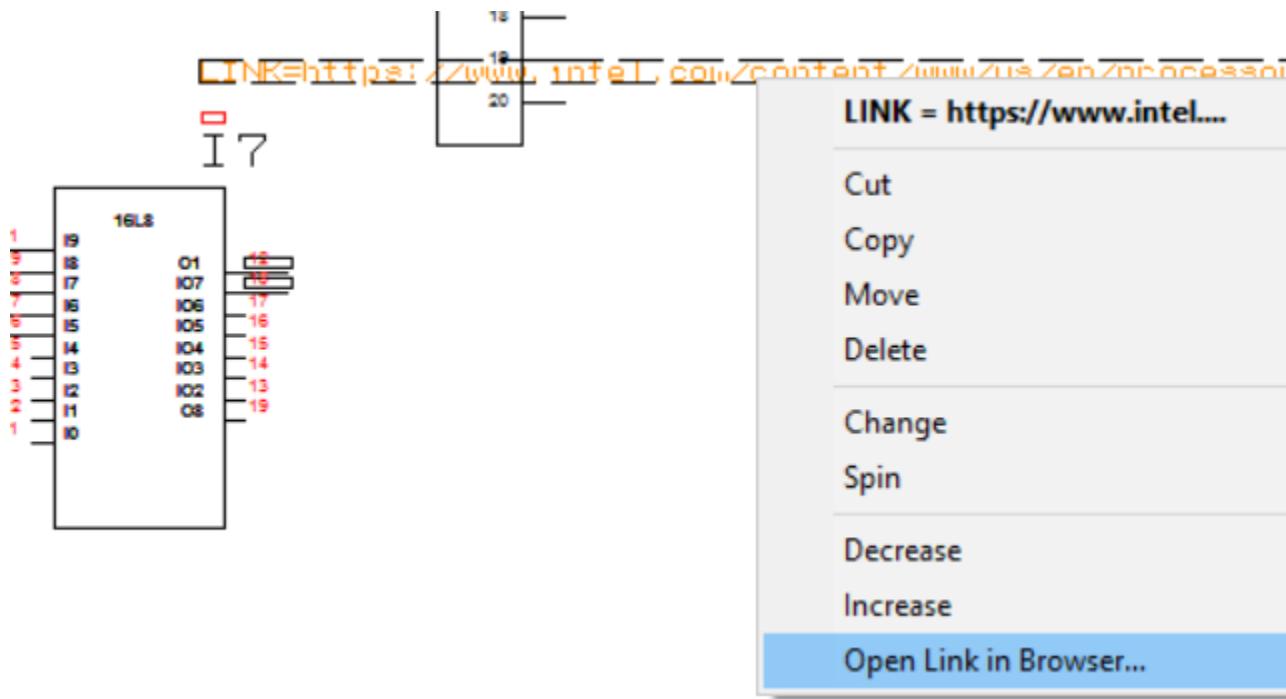
If you have an injected property with a value that is a link to a datasheet, you can access the datasheet link in two ways:

- Annotate the injected property on the schematic and access the link using the Attribute dialog.

## Allegro Design Entry HDL User Guide

### Getting Started

- Annotate the injected property as a visible value on the schematic and open the datasheet by right-clicking and choosing *Open Link in Browser*.



You can also open a datasheet link from Part Manager or Part Information Manager. For details on opening a datasheet from Part Information Manager, see *Part Information Manager User Guide*.

### How can I show the attachment lines of component properties while moving them on a schematic?

You can use *Display — Attachments* to view the visible properties that are associated with components or entities. This option displays attachments temporarily. The attachment lines are hidden when you pan or edit the schematic.

To ensure that the display of attachments persists for specific properties of objects, such as wires, pins, or components, you can specify the property names in the `attach_props.cfg` file.

Modify the `<installation_directory>/share/cdssetup/attach_props.cfg` file and include attachment flags to define when to show or not show the attachments.

Copy this file to `CDS_SITE/$HOME/<project>` to override the default `attach_props.cfg` file in the installation directory.

The attachment flags are as follows:

- steady: displays the property attachment only when property is at its location and is not being moved
- always: displays the property attachment when the component is being moved and when it is steady in its place.
- move: displays the property attachment only while the property is being moved

Here is a sample of the default Cadence `attach_props.cfg` file:

```
; ; Following is the list of properties with which permanent attach  
; ; always ==> Property attachment is shown permanently as well as ;  
; ; steady ==> Property attachment only while the property is steady  
; ; move ==> property attachment is shown only while it is dragged  
  
(  
    ("BIASVOLTAGE" "steady")  
    ("BIASCURRENT" "steady")  
    ("BIASPOWER" "steady" )  
)
```

The default Cadence `attach_props.cfg` file only has three PSpice-related properties. You can delete the default properties, and add the properties you want then specify the action. For example, in the following lines, DE-HDL has been instructed to do the following:

```
; ; Following is the list of properties with which permanent attach  
; ; always ==> Property attachment is shown permanently as well as ;  
; ; steady ==> Property attachment only while the property is steady  
; ; move ==> property attachment is shown only while it is dragged  
  
(  
    ("BIASVOLTAGE" "steady")  
    ("BIASCURRENT" "steady")  
    ("BIASPOWER" "steady" )  
    ("LOCATION" "always")  
    ("XR" "move")  
    ("VALUE" "move")  
)
```

- Always show the attachment lines for the LOCATION attribute with respect to the component or entity on the schematic canvas.

## Allegro Design Entry HDL User Guide

### Getting Started

---

- Show the attachment lines while moving the XR and VALUE properties, and/or while moving the component or entity specific to these properties.

You can use the same syntax for any object properties on the schematic.

For example, if you want to see the attachment lines when SIG\_NAME is moved, you can add the following statement in the attach\_props.cfg file:

```
("SIG_NAME" "always")
```

**Note:** You do not need to provide different entries for the LOCATION and \$LOCATION properties. The attach\_props.cfg reads the attachment flag for both LOCATION and \$LOCATION.

**Note:** If you have set the mode to always for certain attachments, and want these attachments in a schematic PDF, use a third-party PDF converter. PDF Publisher does not currently support publishing attachments to PDF.

### Can attach\_props.cfg be used at the CDS\_SITE level?

The default Cadence attach\_props.cfg file that is part of the installation directory can be overridden at the site level. The file should be in <site>/cdssetup for it to be used by all users accessing the site.



The complete attach\_props.cfg file must be in CDS\_SITE. The contents of the default Cadence attach\_props.cfg file will not be merged with the contents of the file in CDS\_SITE.

# **Allegro Design Entry HDL User Guide**

## Getting Started

---

---

## **Project Creation and Setup**

---

This section introduces you to the Project Manager tool that you use to create and set up projects and describes the procedures for creating and setting up projects. This section contains the following topics:

- [Introduction to Project Manager](#) on page 57
- [Project Structure](#) on page 58
- [Project Files](#) on page 59
- [Project Flows](#) on page 62
- [Creating a Project](#) on page 62
- [Setting Up a Project](#) on page 68
- [Locking Project File Directives](#) on page 82

### **Introduction to Project Manager**

Project Manager is the interface to the Cadence board design solution and library management. The Project Manager tool can be used for the following tasks:

- Create design projects or library projects  
Design projects are created by designers, while library projects are created by librarians.
- Set up projects  
You can choose libraries for your project, as well as options such as Physical Part Table files, property files, expansion types, configurations and view names. The configurations of a design are also stored as views in the design directory.
- Import, export, and archive projects
- Launch tools such as Design Entry HDL, PCB Editor, Allegro SI, Library Explorer, Part Developer, and PadStack Editor.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

- View project settings and libraries in a convenient tree form, and keep track of all the tools running in a session.

The Project Manager flow can be customized to your requirements. You can add and remove tools from the flow and use custom icons. The flow can also be converted to a toolbar to conserve space on your desktop. For details on customizing the Project Manager flow and icons, refer to the *Project Manager User Guide*.

## Project Structure

The logical directory structure for designs is Lib -> Cell -> View -> Files. Each Lib, Cell, and View is a physical directory.

---

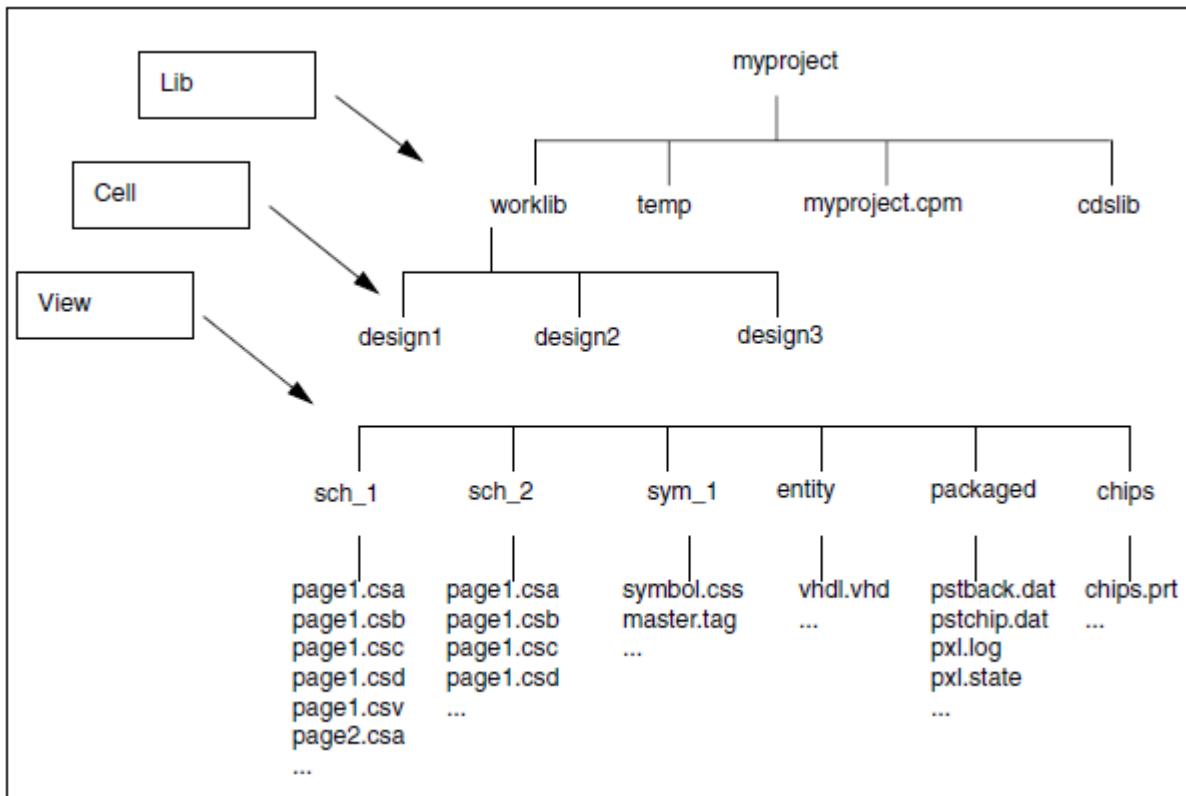
Lib	Is a directory that contains designs (cells).
Cell	Is the design directory. It contains all the data organized into views for that design. For example, the design <i>cpu</i> can have schematic, symbol, chips, packaged, and physical views. The configurations of a design are also stored as views in the design directory.
View	Is a directory that contains all the data for a particular unit of a design. For example, all the files for a schematic are in the <code>sch_n</code> view (where <i>n</i> is the version number). All the information for a symbol representation of the design is in the <code>sym_n</code> view (where <i>n</i> is the version number). The packaged design data is in the packaged view. Configurations, including the four default configurations— <code>cfg_package</code> , <code>cfg_verilog</code> , <code>cfg_pic</code> , and <code>cfg_vhdl</code> —are also views under a cell.

---

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

#### Example



## Project Files

Project Manager manages all the information about a project, such as its libraries, physical part table files, log files, property files, and setup defaults for tools, through project files.

There are four types of project files:

- Local Project Files
- Site Project File
- Installation Project File
- User-Specific Project Settings File

## Local Project Files

When you create a new project, Project Manager creates a project file called `<projectname>.cpm` in the project directory. Each project has one project file. The `<projectname>.cpm` file contains all the setup information that you specified for your project. It has the following:

- The name of the top-level design and the library in which it is located
- The list of project libraries
- The physical part tables selection
- Changes to default view names
- The name and location of the text editor for editing text files from Cadence tools
- The name and location of the property file
- The name and location of the log file
- The name and location of the application temp directory, which is the directory in which applications such as Design Entry HDL store temporary files
- Setup directives for individual tools such as Design Entry HDL, Packager-XL, and Project Manager
- Directives for customizing Project Manager (a customized Tools menu or customized flows)
- Directives for controlling Bias Display settings

To know more about these directives, see [Chapter 17, “Simulating using PSpice Simulator.”](#)

The default setup information is maintained in an installation project file (`cds.cpm`) shipped by Cadence. The defaults in the `cds.cpm` file apply to all your projects. If you want to change these defaults, create a site project file (`site.cpm`) for your site.

When you open a project, Project Manager gets the setup directives you specified for that project from the `<projectname>.cpm` file and the defaults for the others from the `site.cpm` and `cds.cpm` files. Your setup directives always have precedence over the `site.cpm` directives, which in turn have precedence over the `cds.cpm` directives.

You can view the project settings for a project with the *View – Project Settings* command.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

#### Project File (.cpm) Example

```
( Machine generated file created by SPI )
( Last modified was 11:38:31 Thursday, April 05, 2016 )
( NOTE: Do not modify the contents of this file. If this is regenerated by )
( SPI, your modifications will be overwritten. )

START_GLOBAL
use library_ppt 'ON'
design_name 'poa'
design_library 'poa'
library 'poa' 'standard' 'pic' 'poa_lib' 'element'
temp_dir 'temp'
cpm_version '@'
session_name 'ProjectMgr12919'
cdsprop_file ''
ppt './ptf/poa.ppt'
EXCLUDE_PPT
INCLUDE_PPT
END_GLOBAL

START_PKGRXL
state_wins_over_design 'ALL'
END_PKGRXL
```

#### Site Project File

Create the site project file, called `site.cpm`, in the `<your_inst_dir>/share/local/cdssetup/projmgr` directory when you want to specify default setup options for all the projects at your site. The directives in this file have precedence over the installation project file (`cds.cpm`) and the local project file (`<projectname>.cpm`) has precedence over the `site.cpm` file. You can override these directives for individual projects by specifying the changes in the local project file (`projectname.cpm`). For more information on creating a site project file, see [Creating a Site Project File](#) on page 66.

#### Installation Project File

The installation project file called `cds.cpm` is shipped by Cadence and is in the `<your_install_dir>/share/cdssetup/projmgr` directory. The `cds.cpm` file contains the default setup directives for all projects and tools. Project Manager obtains defaults from this file for setup options that are not defined in the `projectname.cpm` or `site.cpm` files. Do not modify this file. If you want to change the defaults for your projects, create a site project file (`site.cpm`).

The setup directives you specify, that is, the directives in the `projectname.cpm` file, always have precedence over the `site.cpm` directives, which in turn have precedence over the `cds.cpm` directives. When you open a project, Project Manager gets the setup directives you

specified for that project from the `projectname.cpm` file and the default values for the other directives from the `site.cpm` file. If the directives are not defined in the `site.cpm` file either, Project Manager obtains the default values from the `cds.cpm` file.

## User-Specific Project Settings File

You define user-specific project settings in the `user.cpm` file. The project settings in this file apply to all the projects that a user opens. The `user.cpm` file is located at `$HOME/cdssetup/projmgr/`, provided the environment variable `$HOME` is set.

## Project Flows

Project Manager is also an HTML browser, and the project flow is defined in a simple HTML document. You can customize the project flow by replacing it with HTML pages that you create for individual projects or for all the projects at your site.

### Note:

The standard Cadence Board Design Flow consists of two HTML files, `main.htm` and `home.htm`, which are in the Cadence installation hierarchy at `<your_install_dir>/share/cdssetup/projmgr/flows`. The `home.htm` file is loaded when no project file is currently open. It has links for opening an existing project or creating a new project. The `main.htm` file is loaded when a project file is opened.

The HTML files contain HREFs. When Project Manager evaluates an HREF, it first looks at its own list of tools to see if the URL matches a tool name. If it does, the tool is launched as a separate process. If no match is found, Project Manager attempts to load the referenced URL. This allows Project Manager to be used as a flow manager, tool launcher, and an internet/intranet browser.

## Creating a Project

To create a new project in Project Manager, do the following:

1. Choose *File – New*.

The New Project Wizard appears.

2. In the *Name* box, type your project name.

3. In the *Location* box, do one of the following:

- Type the complete path of the folder in which you want to create the new project.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

- Click *Browse*, select a folder in the *Choose Directory* dialog box, and then click *OK*. The path to the selected folder is displayed in the *Location* box.

**Note:** If you want to create the project in a new folder, append a name for the new folder to the path (for example: `\cpu`). Project Manager will create the folder.

#### 4. Click *Next*.

The *Project Libraries* dialog box appears with the list of available libraries and project libraries. If you created the project in a new folder or in a folder that does not contain a `cds.lib` file, a `cds.lib` file is automatically created; this file contains a `projectname_lib` entry. You will see the `projectname_lib` entry in the *Project Libraries* list.

#### 5. Select the libraries for your project by placing them in the *Project Libraries* list.

- To add one library to the *Project Libraries* list, select the library in the *Available Libraries* list and then click *Add*.
- To add more than one library to the *Project Libraries* list, press *Ctrl* and select the libraries. Then, click *Add*.
- To add all the libraries in the *Available Libraries* list, click *Add All*.
- To remove one library from the *Project Libraries* list, select the library and then click *Remove*.
- To remove more than one library from the *Project Libraries* list, press *Ctrl* and select the libraries. Then, click *Remove*.
- To remove all the libraries from the *Project Libraries* list, click *Remove All*.

#### 6. Choose the search order for your project libraries. The order in which libraries are listed in the *Project Libraries* list determines their search order.

- To move a library one level up, select the library and then click *Up*.
- To move a library one level down, select the library and then click *Down*.

**Note:** You cannot rearrange the order of the *Available Libraries* list.

#### 7. Click *Next*.

#### 8. In the *Design Name* dialog box, specify the top-level drawing for your design. You can choose an existing design from the project libraries or create a new one in any of the project libraries. The design name must conform to [Design Naming Conventions](#).

To create a new design, do the following:

- a. In the *Library List*, select the library in which you want to create the new design.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

- b.** In the *Design Name* box, type a name for the new design.

To select an existing design, do the following:

- a.** In the *Library List*, select the library that contains the design.
- b.** Click *Browse*, select a design from the *Existing Cell Names* list, and then click *OK*.

#### 9. Click *Next*.

The *Finish* dialog box displays your project specifications.

#### 10. Do one of the following:

- To create the project, click *Finish*.
- To change the project name, project location, design library, design name, or project libraries, click *Back* and edit the information you entered in each dialog box. When you finish, click *Next* until the Finish page appears. Click *Finish* to create the project.

Project Manager displays the default project flow with icons for Design Entry HDL and PCB Editor.

## Design Naming Conventions

The following characters can be used in design names without any restrictions:

alphanumeric characters

\_ (underscore)

- (hyphen)

No other characters can be used in design names.

For more information on naming conventions, see [Allegro Design Entry HDL Reference Guide](#).

## Files Created for Your New Project

When you create a new project, Project Manager creates the following:

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

- A project file (*<projectname>.cpm*)

The project file contains all the setup information you have specified for the project: the name and location of the top-level design, libraries, view names, physical part tables, and tool setup directives. For more information, see [Project Files](#).

- Four configuration views in the design directory (cell) for each new project

The four configurations are: `cfg_package`, `cfg_verilog`, `cfg_pic`, and `cfg_vhdl`. Each configuration is a directory and contains an `expand.cfg` file.

- An application temp directory (temp)

Temporary files created by applications such as Design Entry HDL are placed in the `temp` directory. You can delete the contents of this directory. You can also specify your own application temp directory from Project Setup.

In addition, if you created the project in a new directory or in a directory that does not contain a `cds.lib` file, Project Manager creates the following:

- A `cds.lib` file

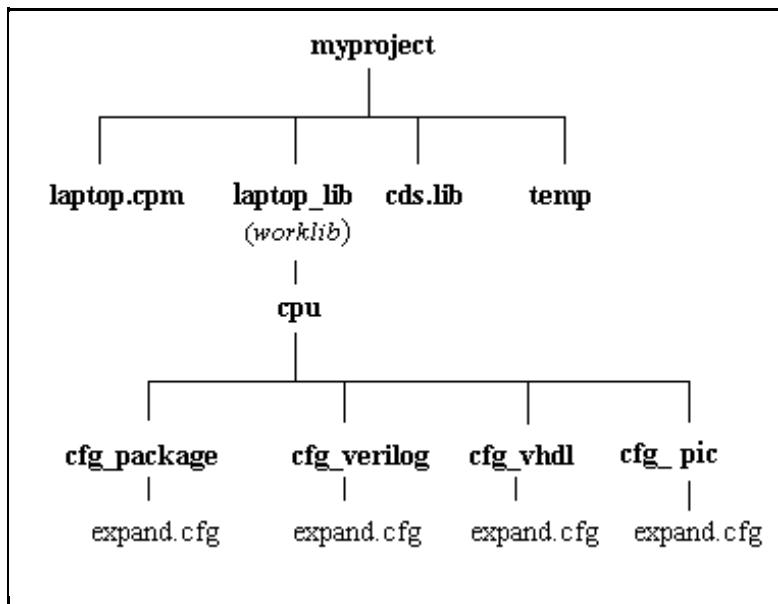
The `cds.lib` file determines the list of available libraries from which you can choose the project libraries for your project. It contains the logical names of libraries and their physical locations. By default, it includes the path to the installed Cadence libraries.

- A `worklib` directory

The physical name of this directory is `worklib`. You can place your design directories (cells) in this directory. When you use Design Entry HDL and its related utilities and create designs, non-view log files are added to the project directory, and design data and view-related log files are added to the `worklib` directory.

## Example

When you create a new project, `laptop`, in a new directory called `myproject` and a design, `cpu`, in `laptop.lib`, you will have the following file structure:



## Creating a Site Project File

You can customize the default settings for all your projects by creating the `site.cpm` file. To create a `site.cpm` file, either use a copy of an existing project file, or create a dummy project and use its project file to define your site settings.

To create a site project file for all the projects at your site, do the following:

1. Choose *Tools – Setup*.
2. In each tab of the *Project Setup* dialog box, specify the default setup information you want for all projects at the site level. For information about the setup options, click the *Help* button in the dialog box.
3. Click *Apply* to save your changes.
4. Close *Project Setup* by clicking *OK*.
5. Choose *File – Export*.
6. In the *Export Project* dialog box, do the following:

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

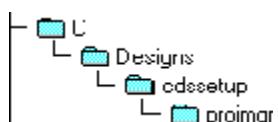
- a. Type site.cpm in the *File Name* box.
  - b. In the *Folders* list, select <your\_inst\_dir>/share/local/cdssetup/projmgr, where <your\_inst\_dir> is the directory in which you have installed Cadence tools.
  - c. Ensure that the *Save File as Type* box displays Project Files (\*.cpm).
  - d. Ensure that the *Full Settings* option is not selected.
7. Click *OK* in the *Export Project Setup* dialog box.

### Creating a Custom Site Environment

If you do not place the site.cpm file in the <your\_inst\_dir>/share/local/cdssetup/projmgr directory, you must set a CDS\_SITE = location environment variable that specifies the location of the site project file. The site location must have the following directory structure:

cdssetup/projmgr/site.cpm

For example, if you want to set CDS\_SITE as C:\Designs, you must create the following directory structure and place the site.cpm file in the projmgr directory:



If you have set the CDS\_SITE environment variable to another location, such as /hm/common/, you need to ensure that the concepthdl.scr file is installed at /hm/common/cdssetup/concept/ so that backannotation from Variant Editor works as required. The site.cpm should be at /hm/common/cdssetup/projmgr/. You also need to copy any file under /share/cdssetup/ that has been customized to /hm/common/cdssetup/. This will ensure that the customized information is available even when you install a newer version of Cadence PSD software.

If you have any custom Project Manager flows, maintain them at \$CDS\_SITE/cdssetup/projmgr/flows using the same directory structure as at <your\_inst\_dir>/share/cdssetup/projmgr/flows/.

If you have customized any of the following files and want the changed version to be available for all projects at your site, copy them into the location as recommended in the *Site Project File* section of [Allegro Front-End CPM Directive Reference Guide](#).

## **Allegro Design Entry HDL User Guide**

### Project Creation and Setup

---

**Note:** The `cdsprop.txt` file need not be copied as you should not modify this file.

#### ***Customized Menu Options***

Customized changes to Design Entry HDL menus are saved in the `concepthdl_menu.txt` file, stored by default in the `HOME\cdssetup\concept` directory. These customized changes are site specific. If you install a newer version of Design Entry HDL, do the following to ensure that you retain your customized changes and see menu changes in the newer version of Design Entry HDL:

1. Take a backup of your customized `concepthdl_menu.txt` file before installing a newer version of Design Entry HDL.
2. Copy the customized menu changes from the backed up `concepthdl_menu.txt` file to the new `concepthdl_menu.txt` file.

## **Setting Up a Project**

The Project Setup window displays the current global, physical part table, tools, expansion, and views settings for your project.

You can perform the following tasks from Project Setup:

- [Changing the Root Design for a Project](#)
- [Creating a New Root Design for a Project](#)
- [Editing the `cds.lib` File](#)
- [Selecting Libraries for a Project](#)
- [Adding Physical Part Table Files to a Project](#)
- [Setting Up Tools](#)
- [Specifying the Application Temp Directory](#)
- [Selecting a Text Editor](#)
- [Selecting a Property File](#)
- [Setting Up a Log File](#)
- [Selecting an Expansion Type](#)
- [Selecting the Configuration for Expansion](#)

- [Editing a Configuration](#)
- [Creating a New Configuration View](#)
- [Selecting Views for the Project](#)

## Changing the Root Design for a Project

### ***To change the root design for a project from Project Manager***

1. Open the project for which you want to change the root design.
2. Choose *Tools – Setup*.  
The *Project Setup* window appears.
3. Choose the *Global* tab.
4. In the *Library Name* list, select the library containing the design.
5. In the *Design Name* field, type the name of the design or click *Browse* and select the design from the *Select Cell* list.
6. Click *Apply* to save the changes, or click *OK* to save the changes and exit *Project Setup*.

**Note:** You can also create a new root design from Project Manager.

## Creating a New Root Design for a Project

### ***To create a new root design for a project from Project Manager***

1. Open the project in which you want to create a new design.
2. Choose *Tools – Setup*.  
The *Project Setup* window appears.
3. Choose the *Global* tab.
4. In the *Library Name* list, choose the library in which you want to create the new design.  
**Note:** The *Library Name* list is the list of project libraries.
5. In the *Design Name* field, delete the text and type the new design name. Click *Browse* to see a list of existing cell names for the library you have selected.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

6. Click *Apply* to save your changes, or click *OK* to save the changes and exit *Project Setup*.

## Editing the `cds.lib` File

Design Entry HDL is a by-reference schematic editor. This means that Design Entry HDL references all parts in the schematic from various libraries that reside at the reference or local area.

Each project has a `cds.lib` file. Project Manager creates the `cds.lib` file when you create a project in a new folder or in a folder that does not contain a `cds.lib` file. The new `cds.lib` contains the following:

- A directive to include the installed Cadence libraries. (For example: `INCLUDE <your_install_dir>/share/cdssetup/cds.lib`)
- A define statement that maps the logical project library (`projectname_lib`) to its physical name (`worklib`). (For example: `DEFINE myproject_lib worklib`.)

You can edit the `cds.lib` file and add directives to include any other libraries such as company libraries. You can add libraries to `cds.lib` directly by specifying their logical names and physical locations. Alternatively, you can add a file that contains a list of libraries and their physical locations.

The `cds.lib` file determines the list of available libraries from which you can choose the project libraries for a project.

To edit the `cds.lib` file, do the following:

1. Open the project.
2. Choose *Tools – Setup*.  
The *Project Setup* window appears.
3. Choose the *Global* tab.
4. Click the *Edit* button next to the `cds.lib` field.  
The `cds.lib` file is opened in the default text editor.

5. Edit the `cds.lib` file.

You can add libraries to the `cds.lib` file directly by specifying their logical names and their physical locations. (For example, `DEFINE MYLIB C:/Libraries/IEEE`). You can also add files that contain a list of libraries and their locations, for example, `INCLUDE C:/`

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

Libraries/company.lib, where company.lib contains a list of libraries and their locations.) See [Adding Libraries to the cds.lib File](#).

6. Save the file and exit the text editor.
7. In the confirmation window, click *Yes* to update the library list.
8. Click *Apply* to save your changes, or *OK* to save your changes and exit *Project Setup*.

### Adding Libraries to the cds.lib File

To add a library, add the following statement to the cds.lib file for the project:

```
DEFINE libraryname libpath
```

where libraryname is the logical name for the directory specified in libpath.

The libraryname is the name that appears in the list of *Available Libraries* in Project Setup.

Example:

```
DEFINE MYLIB C:/Libraries/IEEE
DEFINE lsttl C:/Libraries/lsttl
```

### Syntax limitations for the cds.lib file

- Embedded spaces are not allowed in the libpath. For example, the following statement is invalid:  

```
DEFINE mylib "D:\user\name\My Library"
```
- Embedded forward (\)and backward (/) slash characters are allowed as path delimiters.
- Alias can either be in small case or upper case. Although mixed case is also supported, it is generally not recommended. For example, all of the following statements are valid:

```
DEFINE MYLIB C:/Libraries/IEEE
DEFINE mylib C:/Libraries/IEEE
DEFINE MYlib C:/Libraries/IEEE
```

### Adding a File Containing a List of Libraries

To add a file containing a list of libraries, add one of the following statements to the cds.lib file for the project:

```
INCLUDE filename
SOFTINCLUDE filename
```

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

where *filename* is the name of a file containing a list of libraries and their locations. (*filename* can also be another `cds.lib` file). Using the `INCLUDE` statement generates an error message when Cadence tools cannot find this file. Using the `SOFTINCLUDE` statement generates no error message when Cadence tools cannot find this file.

All the libraries in the `cds.lib` file will appear in the list of *Available Libraries* in Project Setup.

Example:

```
INCLUDE C:/Libraries/mycompany.lib
```

### ***Removing a Library***

To remove a library, add the following statement to the `cds.lib` file for the project:

```
UNDEFINE libraryname
```

where *libraryname* is the name of the library you want to remove.

Use this statement when you want to remove some of the libraries defined in a file you included with `INCLUDE` or `SOFTINCLUDE` statements.

### **Selecting Libraries for a Project**

1. Open the project.

2. Choose *Tools – Setup*.

The *Project Setup* window appears.

3. Choose the *Global* tab.

4. If you want to view the contents of a library, choose the library and click *View*. A window displaying the contents of the library appears. You cannot make any changes in this window.

5. Modify the *Project Libraries* list under *Library*.

6. Do one of the following:

- To add one library, select the library in the *Available Libraries* list and click *Add*.
- To add all the libraries in the *Available Libraries* list, click *Add All*.
- To remove one library, select the library in the *Project Libraries* list and click *Remove*.

- To remove all the libraries in the *Project Libraries* list, click *Remove All*.
7. Choose the search order for the project libraries. The order in which the libraries are listed in the *Project Libraries* list determines their search order.
8. To move a library one level up, select the library and then click *Up*.
9. To move a library one level down, select the library and then click *Down*.
10. Click *Apply* to save your changes, or *OK* to save the changes and exit *Project Setup*.

### **Available Libraries and Project Libraries**

#### **Available Libraries**

These are the libraries available to you for any project. They are determined by the directives in the `cds.lib` file. Cadence-installed libraries are included in the `cds.lib` file as default libraries. You can edit the `cds.lib` file to add other libraries to the list of available libraries.

#### **Project Libraries**

These are the libraries you select for your project from the list of available libraries. You can select project libraries when you create a project or at any other time using the Setup tool. If you create a project in a new folder or in a folder that does not have a `cds.lib` file, a `projectname.lib` file is also created and placed in the *Project Libraries* list.

You can modify the *Project Libraries* list from Project Manager.

### **Adding Physical Part Table Files to a Project**

The Physical Part Table (`.ptf`) file contains the physical properties of a symbol. You can add this data or modify it. PTF files can be located at the cell level under the Part Table view, or in any other directory. Cell-level `.ptf` files contain information about the primitives for that cell.

To access the information contained in the Physical Part Table files, you must include them in your project. When you include cell-level Physical Part Tables, all the `.ptf` files in the Part Table view of that cell are included. You can also include other `.ptf` files by specifying their location. You can include cell-level `.ptf` files and other `.ptf` files in the same project.

If you have a cell-level `.ptf` file and the `INCLUDE_PPT` directive is set, Packager-XL does not read the `.ptf` file. To include the cell-level `.ptf` file, you will have to add it in the `INCLUDE_PPT` directive.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

To add physical part tables to your project, you can add either .ptf files directly or directories that contain .ptf files. For example, if the lsttl directory contains the lsttl.ptf file, you can add either the complete path to the lsttl.ptf file or just the path to the lsttl directory. When you add a directory, all the .ptf files in that directory are added to the project. You can then exclude some of the .ptf files if you do not want them in the project.

Do the following to add Physical Part Table files to a project:

1. Open the project.

2. Choose *Tools – Setup*.

The *Project Setup* window appears.

3. Select the *Part Table* tab.

4. To add cell-level .ptf files to your project, select the *Use Cell-Level Physical Part Table Files* check box. All the .ptf files contained in the Part Table view of the cells will be read by Packager-XL.

5. To add other Physical Part Table files,

a. Under *Physical Part Table Files*, click *Add*.

The *Add Physical Part Table* dialog box appears.

b. Do one of the following:

- Type the name and the path of the .ptf file or the directory containing the .ptf files. To add more than one path, separate each path with a space.
- To add a file, click *File* and select the .ptf file in the *Choose Physical Part Table Files* dialog box. (To select more than one file, select the first one, then press *CTRL* and select the others.) To add a directory, click *Directory* and select a directory in the *Choose Directory* dialog box.

c. Click *OK*.

6. To exclude any unwanted .ptf files contained in the directories you have added, do one of the following:

- ❑ Under *Exclude Physical Part Table Files*, click *Add* and enter the name and path of the .ptf file you want to exclude from your project. Repeat this step for all the files you want to exclude.
- ❑ Under *Include PTFs*, click *Add* and enter the name and path of the .ptf file you want to include in your project. Repeat this step for all the files you want to include.
- ❑ To remove a Physical Part Table file or directory, select the file and click *Remove*.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

- Select the *Merge Physical Part Table Files* check box to merge the information in all the included Physical Part Table files.
  - Select the *Perform Case Sensitive Row Match* check box to match case-sensitive key properties for a part in the physical part table files.
7. Click *Apply* to save your changes, or *OK* to save your changes and exit *Project Setup*.
8. You can include cell-level .ptf files and other .ptf files in the same project; Packager-XL reads the contents of each.

## Setting Up Tools

The Tools tab in the Project Setup window allows you to select the setup options for PCB Editor, Design Entry HDL, Allegro Project Manager, Packager-XL, Programmable IC, Simulation, and Mixed Signal simulation. You can specify setup directives for these tools from Project Manager or directly from the tools.

The Tools tab also displays the default settings for the property file, the text editor, the project log file, and the temp directory.

In this tab, you can do the following:

- Specify the setup directives for PCB Editor, Design Entry HDL, Project Manager, Packager-XL, Programmable IC, Simulation, and Mixed Signal Simulation. Simulation Interface provides a simulation environment to simulate your schematics from Design Entry HDL with Verilog or Leapfrog. You can specify setup directives for Design Entry HDL and Packager-XL directly from the tools or from Project Manager.
- Select a default text editor.
- Specify an application temp directory.
- Select a property file.
- Set up a log file.

**Note:** You can specify the setup directives for Design Entry HDL and Packager-XL directly from the tools or from Project Manager.

## Specifying the Application Temp Directory

The Application Temp directory is the directory in which applications such as Design Entry HDL store temporary files. You can delete the contents of this directory.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

An Application Temp directory, `temp`, is created automatically when you create a new project. However, you can specify any directory as the Application Temp directory for a project.

To specify an Application Temp directory, do the following:

1. Choose *Tools – Setup*.

The *Project Setup* window appears.

2. Select the *Tools* tab.

3. In the *Temp Directory* field, type the full path to the `temp` folder, or click *Browse* and use the file browser to select the location of the `temp` folder.

4. Click *Apply* to save your changes, or click *OK* to save your changes and exit Project Setup.

## Selecting a Text Editor

For each project, you can select a text editor as the default text editor for Cadence tools. When you view or edit any text file from a Cadence tool, it will be displayed in the text editor you have specified. The default editor is WordPad.

To select a text editor, do the following:

1. Choose *Tools – Setup*.

The *Project Setup* window appears.

2. Select the *Tools* tab.

3. In the *Default Text Editor Path* field, type the full path to the text editor you want to use, or click *Browse* and use the file browser to select the text editor.

4. Click *Apply* to save your changes, or *OK* to save your changes and exit Project Setup.

## Selecting a Property File

The property file for a project contains directives that control how properties are handled during expansion. It specifies whether a property is inherited by other objects, whether it is a parameter, what objects it can be attached to, and whether it is passed to the destination tool.

Cadence provides a default property file called `cdsprop.paf`, which is located in the `<your_install_dir>/share/cdssetup` directory. Do not modify this file. You can use your own property file by specifying its path in *Project Setup*.

To select a property file, do the following:

1. Choose *Tools – Setup*.

The *Project Setup* window appears.

2. Select the *Tools* tab.

3. In the *Property File* field, type the full path of the property file you want to use, or click *Browse* and use the file browser to select the file.

4. Click *Apply* to save your changes, or *OK* to save your changes and exit Project Setup.

**Example** cdsprop.paf **file**

```
FILE_TYPE=ATTRIBUTES;
{ Default attributes for properties.
  Attributes for user designed properties should be added to the user's
  property attribute file. This file should not be modified. }
ALLOW_CONNECT: inherit(signal), permit(pin, body, signal);
AUTO_GEN: inherit(), permit(body);
BIDIRECTIONAL: inherit(), permit(pin);
BODY_NAME: inherit(), permit(body);
CHIP_DELAY: inherit(pin), permit(pin, signal);
CLOCK_DELAY: inherit(pin), permit(pin, signal);
COMMENT_BODY: filter;
COUPLED: inherit(), permit(body);
DELAY: inherit(), parameter, permit(body);
DIR: inherit(), permit(body), case_sensitive;
EVAL: inherit(pin), permit(pin, signal);
EXPR: filter;
FALL: inherit(), parameter, permit(body);
GROUP: inherit(body), permit(body);
HAS_FIXED_SIZE: inherit(), permit(body);
HIGH: inherit(), permit(body);
INPUT_LOAD: inherit(), permit(pin);
IO_NET: inherit(signal), permit(signal);
LAST_MODIFIED: inherit(), filter;
LOCATION: inherit(body), permit(body);
LOCATION_CLASS: inherit(body), permit(body);
LOW: inherit(), permit(body);
MODEL: inherit(), permit(body);
```

## Setting Up a Log File

A log file for a project tracks information such as the date and time of any activity, the tools launched from the project, the user's name, MPS sessions and hosts.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

If you want to maintain a log file for a project, you must select the option in Project Setup. A log file will not be generated by default.

To set up a log for a project, do the following:

1. Choose *Tools – Setup*. The *Project Setup* window appears.
2. Select the *Tools* tab.
3. In the *Project Log File* field, type a name for the log file. The file will be created in the project directory. To specify an existing file, click *Browse* and use the file browser to select it.
4. Click *Apply* to save your changes, or *OK* to save your changes and exit Project Setup.

## Selecting an Expansion Type

The expansion type and configuration you select in Project Setup determines the current configuration for Design Entry HDL and Hierarchy Editor. If you change the expansion type in Project Setup, the current configuration for Design Entry HDL and Hierarchy Editor changes. The default expansion type is Physical Layout.

To select the expansion type for your design, do the following:

1. Choose *Tools – Setup*.  
The Project Setup window appears.
2. Select the *Expansion* tab.
3. Do one of the following:
  - Select the *Physical Layout* option to expand your design for PCB Editor and other back-end tools.
  - Select the *Verilog Simulation* option to expand your design for simulation with Verilog-XL and other Verilog-based simulators.
  - Select the *VHDL Simulation* option to expand your design for simulation with Leapfrog and other VHDL-based simulators.
  - Select the *PIC Configuration* option to expand your design for simulation with Programmable IC such as Verilog-XL.
  - Select the *Mixed Signal* option to expand your design for mixed-signal simulation.
4. In the *View* field next to the expansion type you selected, click *Browse*.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

The *Select View* dialog box appears. Select the configuration you want to expand and click *OK*.

5. If you want to view or edit the configuration, click *Edit*.

The configuration is opened in the Cadence Hierarchy Editor. Edit the configuration, save it with the *File – Save* command, and exit the Cadence Hierarchy Editor.

6. Click *Apply* to save your changes, or click *OK* to save your changes and exit Project Setup.

**Note:** When you package a design, Packager-XL always uses the Physical Layout expansion, irrespective of the expansion type you select in Project Setup.

### Selecting the Configuration for Expansion

When you create a new project, the default configurations for each expansion type are created automatically. These are as follows:

- `cfg_package` for Physical Layout
- `cfg_verilog` for Verilog Simulation
- `cfg_vhdl` for VHDL Simulation
- `cfg_pic` for PIC Simulation
- `cfg_mixed` for Mixed Signal Simulation

You can select a different configuration for each expansion type.

To select the configuration for each expansion type, do the following:

1. Choose *Tools – Setup*.

The *Project Setup* window appears.

2. Select the *Expansion* tab.

3. Click the *Browse* button next to expansion type.

The *Select View* dialog box appears. Select the configuration you want to use and click *OK*.

4. Click *Apply* to save your changes, or *OK* to save your changes and exit Project Setup.

**Note:** You can also create a new configuration view from Project Manager.

## Editing a Configuration

You edit a configuration with the Hierarchy Editor, a graphical tool for creating and editing configurations.

To edit a configuration, do the following:

1. Choose *Tools – Setup*.

The Project Setup window appears.

2. Select the *Expansion* tab.

3. In the *View* field next to an expansion type (Physical Layout, Verilog Simulation, VHDL Simulation, PIC Configuration and Mixed Signal), click *Browse* and select the configuration you want to edit.

4. Click *Edit*.

The configuration you specified in the *View* field is opened in the Cadence Hierarchy Editor.

5. Make the required changes in the Hierarchy Editor, save the changes, and exit the Hierarchy Editor.

6. Click *Apply* to save your changes, or *OK* to save your changes and exit Project Setup.

## Creating a New Configuration View

1. Choose *Tools – Setup*.

The Project Setup window appears.

2. Select the *Expansion* tab.

3. In the *View* field next to the *Expansion Type* option you have chosen, delete the existing view name, and type the name for the new view.

4. Click *Apply* to save your changes, or *OK* to save your changes and exit Project Setup.

Project Manager creates the new view as well as an `expand.cfg` file in the view.

## Selecting Views for the Project

Views are created when you work with your designs. When you package a design, a packaged view is created and all Packager-XL output files and log files are stored in it. The

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

chips.prt file is placed in the chips view, cell-level physical part tables in the part\_table view, and PCB data in the physical view.

Similarly, the root design views for Board Design, PIC Design, Verilog Simulation and VHDL Simulation will be used for expanding your design for physical layout, PIC simulation, Verilog simulation, and VHDL simulation, respectively.

Project Manager assigns default view names. These are:

Type of View	View Name
Packaged	packaged
Chips	chips
Part Table	part_table
Physical	physical
Constraints	constraints
Board Design	sch_1
Programmable IC Design	sch_1
Verilog Simulation	sim_sch_1
VHDL Simulation	sim_sch_1

You can change the view names for each project.

### Changing View Names

You can create a new view name or select an existing view name. Open the project for which you want to change view names.

1. Choose *Tools – Setup*.

The *Project Setup* window appears.

2. Select the *Views* tab.

- To change the view name, click on the drop-down list to select from an existing view name.
- To create a new view name, type the new view name.

3. Click *Apply* to save your changes, or *OK* to save your changes and exit Project Setup.

## Locking Project File Directives

Design Entry HDL lets you lock project (.cpm) file directives providing you increased control over configuring and controlling project settings. It provides a mechanism by which you can control user access and modification permissions on project settings. You can also configure settings that will be reflected in all the projects you open, irrespective of the settings in the project's .cpm file.

### Controlling Project Settings at Different Levels

Project settings of a Design Entry HDL project are configured in the Cadence default .cpm file (`cds.cpm`), the CDS\_SITE area (`site.cpm`), and in the local project (`<project>.cpm`).

However, you might want to define user-specific settings, which you can customize according to your needs and retain the same settings for any project you open irrespective of the local project settings. Some examples of user-specific settings include: default printer, text editor, and panning.

In Design Entry HDL, the directive-locking feature of a CPM file provides control over the list of directives which you can configure at the user level and which will reflect in all the projects irrespective of the project settings. This is achieved by `user.cpm`, where user-specific settings are defined. If the environment variable `$HOME` is set, the `user.cpm` file is located at `$HOME/cdssetup/projmgr/`. This feature allows you to lock a directive at any of the four levels, including the `user.cpm` level, defining the level at which the value of the directive is honored.

### Locking a Directive

A locked directive is defined with the keyword `LOCK` in the .cpm file. Locking implies that the directive is locked for all levels down from the level at which it is locked.

For example, locking a directive at `project.cpm` implies that the directive will be honored at the `project.cpm` level if the directive is in `project.cpm`. If it is not in `project.cpm`, the directive will be honored from `site.cpm` or `cds.cpm` as the case may be. However, the directive, if in `user.cpm`, will not be honored.

### Example

To lock the `PINNUMBER_SIZE` directive, the following two sections are required in the .cpm file:

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

The PINNUMBER\_SIZE directive will need to be in the following section:

```
START_CONCEPTHDL
PINNUMBER_SIZE      0.090
...
END_CONCEPTHDL
```

And it will need to be in this section too in the cpm file:

```
START_CONCEPTHDL_CONTROL_SETTINGS
PINNUMBER_SIZE  LOCK
END_CONCEPTHDL_CONTROL_SETTINGS
```

When a project is loaded, the directive in the user.cpm file will be honored only if the install (cds.cpm) or the site-level cpm file (site.cpm) allow the directive to be read and set at the user level.

You need to specifically allow user-level settings for a given directive in the install or site level cpm files with the ALLOW\_USER\_CPM keyword as illustrated:

```
START_CONCEPTHDL_CONTROL_SETTINGS
PINNUMBER_SIZE      ALLOW_USER_CPM
END_CONCEPTHDL_CONTROL_SETTINGS
```

## Locking Support for Directives at Different Levels

The locking mechanism for directives allows better control over configuring settings, such as part table settings, PXL property settings, CHECK command rules, grid settings, and so on. You can lock any directive in the cpm files at different levels. The directive locking feature uses the following precedence to check the locking status of directives:

- \$CDSROOT: The CDSROOT project file (cds.cpm) is the first cpm file to be read.
- \$CDS\_SITE: The next cpm file in the load process is from CDS\_SITE (site.cpm). You can lock the directives in this file also, but this setting overrides the directive settings in the user's HOME account and the local <project>.cpm file.
- Project: The <project>.cpm file is the next file to be read.
- \$HOME: After the CDSROOT and CDS\_SITE project files, the HOME cpm file (user.cpm) is loaded. Directives in user.cpm will be honored only when the ALLOW\_USER\_CPM setting is provided at the install or CDS\_SITE level cpm for those directives.

The \$HOME settings will be honored only when the directives in user.cpm are not locked at the install, site, or project levels. Locks found in the HOME account can be used to keep local project settings from masking your preferences. This is how preferences are honored in each project that you open.

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

**Note:** You cannot edit the directives locked in the CDSROOT, CDS\_SITE, or HOME areas in a local project environment.

## Reading Settings from CPM Files at Different Levels

The following describes how settings at different levels impact your project:

CPM File	Description
user.cpm	<p>Contains user-level settings</p> <p>Directives defined in the <code>user.cpm</code> file are honored if:</p> <ul style="list-style-type: none"><li>■ They are not locked at the <code>&lt;project&gt;.cpm</code>, <code>site.cpm</code>, or <code>cds.cpm</code> levels</li><li>■ The <code>ALLOW_USER_CPM</code> keyword is in <code>site.cpm</code> or <code>cds.cpm</code> for the directive</li></ul> <p>Modified directives honored in <code>user.cpm</code> are written to <code>user.cpm</code> and not to <code>&lt;project&gt;.cpm</code>.</p>
<code>&lt;project&gt;.cpm</code>	<p>Contains settings that are local to a project</p> <p>Directives defined in the <code>&lt;project&gt;.cpm</code> file are honored if:</p> <ul style="list-style-type: none"><li>■ They are not honored in <code>user.cpm</code></li><li>■ They are not locked at the <code>site.cpm</code> or <code>cds.cpm</code> level</li></ul> <p>Changes to a directive locked at the <code>&lt;project&gt;.cpm</code> level are allowed and the new value is written in the <code>&lt;project&gt;.cpm</code>.</p> <p>The modified directive in the <code>&lt;project&gt;.cpm</code> file does not have the <code>LOCK</code> keyword associated with it.</p>

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

CPM File	Description
site.cpm	<p>Contains site-level settings</p> <p>Directives defined in the <code>site.cpm</code> file are honored if:</p> <ul style="list-style-type: none"><li>■ They are not honored in <code>user.cpm</code> or <code>&lt;project&gt;.cpm</code></li><li>■ They are not locked at the <code>cds.cpm</code> level</li></ul> <p>Changes to the directives are not honored if they are locked in <code>site.cpm</code> itself. Otherwise, all modifications to the directives are written to the <code>&lt;project&gt;.cpm</code> file.</p> <p>A modified directive in the <code>&lt;project&gt;.cpm</code> file does not have the <code>LOCK</code> keyword associated with it.</p>
cds.cpm	<p>Contains install-level settings</p> <p>Directives defined in the <code>cds.cpm</code> file are honored if they are not honored in <code>user.cpm</code>, <code>&lt;project&gt;.cpm</code>, or <code>site.cpm</code>.</p> <p>Modifications to the directives are not honored if they are locked in the <code>cds.cpm</code> itself. Otherwise, all the modifications to the directives are written to the <code>&lt;project&gt;.cpm</code> file. The modified directive in the <code>&lt;project&gt;.cpm</code> file does not have the <code>LOCK</code> keyword associated with it.</p>

### Example

Consider the following example of the `PINNUMBER_SIZE` directive defined at the four levels of `cpm` files:

CPM File	Directive Definition
user.cpm	<pre>START_CONCEPTHDL ... PINNUMBER_SIZE 0.075 ... END_CONCEPTHDL ...</pre>

## Allegro Design Entry HDL User Guide

### Project Creation and Setup

---

<b>CPM File</b>	<b>Directive Definition</b>
<project.cpm>	START_CONCEPTHDL ... PINNUMBER_SIZE 0.072 ... END_CONCEPTHDL
site.cpm	START_CONCEPTHDL ... PINNUMBER_SIZE <b>0.090</b> ... END_CONCEPTHDL ... START_CONCEPTHDL_CONTROL_SETTINGS PINNUMBER_SIZE LOCK END_CONCEPTHDL_CONTROL_SETTINGS
cds.cpm	START_CONCEPTHDL ... PINNUMBER_SIZE 0.082 ... END_CONCEPTHDL

---

Here, the directive PINNUMBER\_SIZE with a value of *0.090* (in site.cpm) will be honored. Modification to this value is not allowed.

---

## **Design Entry HDL Editing Environment**

---

Design Entry HDL enables you to make changes in the editing environment according to your preferences. The following topics are described below:

- [Setting Up Defaults](#) on page 88
- [Basic Editing Tasks](#) on page 91
- [Displaying Information](#) on page 111
- [Using the QuickPick Browser](#) on page 119
- [Basic Navigation in Design Entry HDL](#) on page 125
- [Running Commands with Strokes](#) on page 131
- [Support for Fonts](#) on page 133
- [Uprev of an Existing Design](#) on page 143
- [Single Mode operation in DE-HDL](#) on page 144
- [Support for Common Windows Commands and Operations](#) on page 145
  - [Windows Mode](#) on page 145
  - [Menus in the Windows Mode](#) on page 147
  - [Design Entry HDL Options Dialog Box](#) on page 148
  - [Support for Keyboard Operations](#) on page 148
  - [Bounding Box on Components](#) on page 150
  - [Global Navigate Window](#) on page 151
  - [Properties Window](#) on page 151
  - [Displaying Unconnected Pins on Components](#) on page 152
  - [Wire Selection](#) on page 154

- [Anchor Point Stretch](#) on page 154
- [Alignment and Distribution](#) on page 156
- [Selection Filters](#) on page 160
- [Object Visibility Layers](#) on page 161
- [Page Search Toolbar](#) on page 164
- [Searching Design Objects](#) on page 166

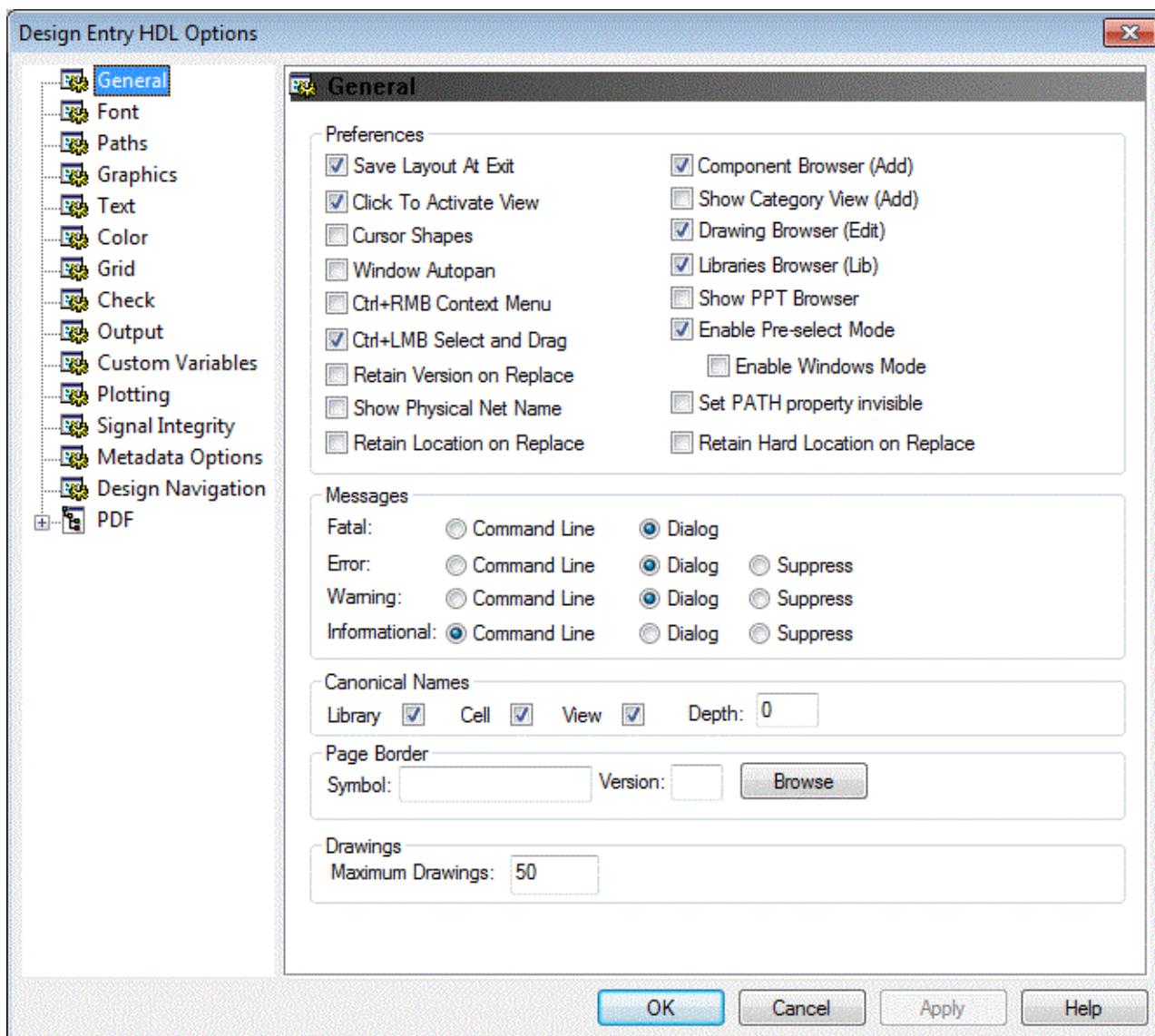
## Setting Up Defaults

This section covers the following information:

- [Setting Up Design Entry HDL Editor Options](#)
- [Defining a Default Text Editor](#)

## Setting Up Design Entry HDL Editor Options

1. Choose *Tools – Options*.



2. Click an option on the left pane to display the setup options you want to view. For example, click *Grid* to display the grid options.
3. Click *OK*.

### Enabling the Pre-Select Menu

The post-select menu is the default. You can set the preferred menu to the pre-select menu.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

1. Choose *Tools – Options – General*.
2. Select the *Enable Pre-select Mode* check box.

**Note:** Clicking the Enable Windows Mode option at this point enables the features available under this mode of Design Entry HDL.



If you customize Design Entry HDL menus in the pre-select mode, the changes are not reflected in the post-select mode. Similarly, changes made in the post-select mode are not honored in the pre-select mode.

3. Click *Apply*.
4. Click *OK*.

For operations on schematics, select the object first, then the menu option for the operation you want to perform. For example, to rotate a component using the pre-select model, first click on the component and then choose the *Edit – Rotate* menu option.

**Note:** In the Windows Mode of Design Entry HDL, these operations are much easier to handle. See Support for Common Windows Commands and Operations on page 145 for more details.

1. Select the component.
2. Do one of the following:
  - Choose *Edit – Rotate*.
  - Choose *Edit – Component – Rotate* in the Windows Mode.

**Note:** Design Entry HDL supports the pre-select mode only through menus; the pre-select mode is not supported through Design Entry HDL commands. For example, in this mode, if you first select an object, and then type *delete* in the console window, the object is not deleted.

## Setting Automatic Page Borders

To automatically set a page border for a new schematic, do the following:

1. Choose *Tools – Options – General*.
2. In the *Page Border* section,
  - a. Specify the name of the page border you want to add in the *Symbol* field.

- b. Specify the version of the page border in the *Version* field. By default, the version is 1.
3. Click *Apply*.
4. Click *OK*.

## Defining a Default Text Editor

If you open a text file in Design Entry HDL, the default text editor is used to display and edit the file.

***To set the default text editor for a project, do the following:***

1. Open Project Manager.
2. Click on the *Setup* icon.  
The Project Setup dialog box appears.
3. Select the *Tools* tab.
4. Specify the Default Text Editor Path.

The default text editor is `write` on Windows. These default settings are read by Project Manager from `<your_install_dir>/share/cdssetup/projmgr/cds.cpm`.

5. Click *Apply*.

## Basic Editing Tasks

This section covers the steps to be performed in Design Entry HDL to complete basic editing tasks such as undo, copy, paste, delete, and so on.

**Note:** Before you perform these basic editing tasks, ensure that the *CTRL+LMB Select and Drag* check box is selected in the *General* page of the Design Entry HDL Options dialog box.

- [Undoing an Operation](#) on page 92
- [Moving Objects](#) on page 92
- [Copying Objects](#) on page 94
- [Deleting Objects](#) on page 96
- [Changing the Color of Objects](#) on page 96

- [Drawing an Arc on page 97](#)
- [Drawing a Circle on page 97](#)
- [Splitting Overlaid Objects on page 98](#)
- [Displaying the Console Window on page 98](#)
- [Editing Text in Dialog Boxes and the Console Window on page 98](#)
- [Copying Parts of a Schematic Across Pages, Designs, and Projects on page 99](#)

## Undoing an Operation

To undo an operation, do the following:

1. Choose *Edit – Undo*.
2. Continue choosing *Edit – Undo* to back out of operations progressively.
3. You can reverse an undo operation by choosing *Edit – Redo*.
4. If you have moved objects between drawings and want to undo the operation, you must choose *Edit – Undo* once in each of the drawings.

**Note:** You can also run the *undo* command using the following stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#).

## Reversing an Undo Operation

To redo an operation, do the following:

1. Choose *Edit – Redo*.
2. Continue choosing *Edit – Redo* to reverse undo operations progressively.
3. You can reverse the redo operation by choosing *Edit – Undo*.

## Moving Objects

To move text, wires, or an unwired component, do the following:

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

1. Select the object.
2. Do one of the following:
  - Move the object to a new location and click again.
  - Choose *Edit – Move*.
3. Click on the object.
4. Move the object to a new location and click again.

To move a wired component, do the following:

1. Ensure that *Auto Route on Move* is checked in *Tools – Options* for Graphics.
2. Choose *Edit – Move*.
3. Click on the wire nearest to the component you want to move.

The following three cases can occur depending on where you click the mouse button and which button you click:

- Click near the open edge of the wire.**  
Only the wire moves; the component attached to it does not move.
- Click on the wire segment near the component pin.**  
Both, the component, and the wire attached to it, move.
- Right-click and select the Change Attachment context menu item.**

The following three states can occur:

- The first time the *Change Attachment* menu item is selected, the component is moved along with all the wires connected to it. The attached wires are directly routed.
- Selecting the *Change Attachment* menu item a second time disconnects the wire segment from the pin of the component and moves the wire segment.
- Selecting the *Change Attachment* menu item a second time moves just the wire segment while keeping it connected to the component pin.

**Note:** You can also run the move command using the following stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#).

To move multiple objects, do the following:

1. Hold down the left mouse button and drag the mouse to select multiple objects, or use *Ctrl+click* or *SHIFT+click* to select multiple objects.

To exclude components, properties or wires from the selected objects, right-click and choose *Exclude* to exclude components, properties or wires from the selected objects.

2. Click on one of the selected objects.
3. Move the objects to a new location and click again.

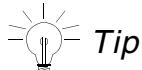
## Copying Objects

To copy an object, do the following:

1. Choose *Edit – Copy*.

To copy an object with its properties, right-click and choose *All* from the pop-up menu.

2. Click an object.



You can also use the keyboard command *Ctrl+c* to copy and *Ctrl+v* paste in the [Windows Mode](#).

The object is attached to the cursor. You can place the object on the drawing.

3. To place several copies of the object without selecting the object again, right-click and choose *Retain Selection* from the pop-up menu.
4. Click in the same drawing or in another window to place the copies.
5. If you chose *Retain Selection* from the pop-up menu and you finish placing copies but want to remain in the *Edit – Copy* mode, choose *Terminate Selection* from the pop-up menu. To exit *Edit – Copy* entirely, choose *Done* from the pop-up menu.

**Note:** You can also run the copy command using the following stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#).

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

To copy an object and its properties, do the following:

1. Choose *Edit – Copy All*.

2. Click an object.

The object is attached to the cursor. You can place the object on the drawing.

3. To place several copies of the object without selecting the object again, right-click and choose *Retain Selection* from the pop-up menu.
4. Click in the same drawing or in another window to place the copies.
5. If you chose *Retain Selection* from the pop-up menu and you have placed copies but want to remain in the *Edit – Copy All* mode, choose *Terminate Selection* from the pop-up menu. To exit *Edit – Copy All* entirely, choose *Done* from the pop-up menu.

**Note:** You can also copy properties when you choose *Edit – Copy* or *Edit – Array*, right-click, and choose *All* from the pop-up menu.

To make multiple copies of an object, do the following:

1. Choose *Edit – Array*.

2. Type the number of copies you want to make in the *Array Size* box, and click *OK*.

3. To copy an object with its properties, right-click and choose *All* from the pop-up menu.

4. Click in the same drawing or in another window to place the copies.

Objects copied in the array are offset from each other by the same distance as the first object in the array from the original.

5. To place another copy of the array without specifying the array again, right-click and choose *Retain Selection* from the pop-up menu.

6. Click in the same drawing or in another window to place the array.

A copy of the selected object remains attached to the cursor. You can place the copy in several unrelated places on the drawing.

7. If you selected *Retain Selection* from the pop-up menu and you have placed the array but want to remain in *Edit – Array* mode, choose *Terminate Selection* from the pop-up menu. To exit *Edit – Array* entirely, choose *Done*.

To copy multiple objects, do the following:

1. Hold down the left mouse button and drag the mouse to select multiple objects, or use *Ctrl+click* or *SHIFT+click* to select multiple objects.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

To exclude components, properties or wires from the selected objects, right-click and choose *Exclude* to exclude components, properties or wires from the selected objects.

2. Click the right mouse button and choose *Copy*, *Copy All* or *Array*.

The objects are attached to the cursor. You can place the objects on the drawing.

**Note:** For more information about copying and pasting parts across schematic pages and designs, see [Copying Parts of a Schematic Across Pages, Designs, and Projects](#) on page 99.

## Deleting Objects

To delete an object, do the following:

1. Choose *Edit – Delete*.



You can also press the *Delete* key to delete an object in the [Windows Mode](#).

2. Click on the objects you want to delete.

To reverse a deletion, choose *Edit – Undo*.

**Note:** You can also run the delete command using the following stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#).

To delete multiple objects, do the following:

1. Hold down the left mouse button and drag the mouse to select multiple objects, or use *Ctrl+click* or *SHIFT+click* to select multiple objects.

To exclude components, properties or wires from the selected objects, right-click and choose *Exclude* to exclude components, properties or wires from the selected objects.

2. Click the right mouse button and choose *Delete* to delete the objects.

## Changing the Color of Objects

To change the color of an object, do the following:

1. Choose *Edit – Color*.

2. Click on the objects whose color you want to change.

## Drawing an Arc

To draw an arc, do the following:

1. Choose one of the following:
  - Edit – Arc.*
  - Place – Arc in the Windows Mode.*
2. First, click in the schematic then move the cursor to approximately the diameter of the arc and click again.

These first two points define the endpoints of the arc.
3. Click a third time between the two points.

The curvature of the arc is defined by how close to the first two points you click. You can also draw a circle from an arc.

## Drawing a Circle

To draw a circle, do the following:

1. Choose *Edit – Circle*.  
 *Tip*

Choose *Place – Circle in the Windows Mode.*
2. Click in the schematic.

This defines the center of the circle.
3. Size the circle by dragging the cursor away from the center.

To draw a circle from an arc, do the following:

1. Choose one of the following:
  - Edit – Arc.*
  - Edit – Arc in the Windows Mode.*

2. Click in the schematic, and move the cursor to mark the desired diameter of the circle and click again.
3. Right-click and choose *Done* from the pop-up menu.

## Splitting Overlaid Objects

1. Choose *Edit – Split*.
2. Click the overlaid objects.
3. Click a clear location nearby to place the overlaid objects.

## Displaying the Console Window

Choose *View – Console Window*.

The console window appears. The next time you display the *View* menu, you will see a check next to *Console Window*.

## Editing Text in Dialog Boxes and the Console Window

---

To	Press
Move the cursor left, right, up, or down	Arrow keys
Move the cursor to the beginning of line	Home
Move the cursor to the end of line	End
Select text (or extend selection) to the left, right, up, or down	Shift + Arrow keys
Select text (or extend selection) to the end of line	Shift + End
Select text (or extend selection) to the beginning of line	Shift + Home
Delete the previous character	Backspace
Delete the next character	Delete
Delete selected text	Delete

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

To	Press
Copy selected text	Ctrl + Insert - or - Ctrl + C
Cut selected text	Ctrl + X
Paste	Shift + Insert - or - Ctrl + V
Undo the last cut or paste	Alt + Backspace

**Note:** On Windows, you can also use commands from the pop-up menu to edit text in dialog boxes.

## Copying Parts of a Schematic Across Pages, Designs, and Projects

Design Entry HDL provides support for the industry-standard copy and paste feature, which enables you to copy and paste parts of a schematic from one design to another. In addition, you can copy text and bitmaps from text editors and graphics editors respectively to the schematic canvas. You can also paste previously-entered commands in the console window for repeated execution.

You can open multiple projects in Design Entry HDL and select a few components, wires, notes, or page border in one project. When you select the *Copy*, *Copy All*, or *Cut* commands from the *Edit* menu, the selected objects are placed in the clipboard. You can then paste the contents of the clipboard onto another schematic page, design or project.

To copy part of a schematic, such as a component, wire, border, or a block, do the following:

1. Select the part on the schematic.
2. Choose *Edit – Copy*, or right-click the portion to be copied and choose *Copy* from the pop-up menu.
3. Go to the desired page in the design or in another design.
4. Do one of the following:
  - Choose *Edit – Paste* and click at the target location on the schematic canvas.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

- Right-click at the target location and select *Paste* from the pop-up menu.
5. Click multiple times if you want to place several instances of the copied part on the schematic.
  6. Right-click and choose *Done* from the pop-up menu.

**Note:** You need to deactivate the *Paste* command by pressing *Done* from the pop-up menu in the target schematic page to remove the copied part from the clipboard. This is recommended if you want to copy another schematic part across schematic pages or designs or run any other command.

### Copying Components

If you copy a component, all the visible properties of the component are copied and pasted. Packaging properties are copied only if you choose the *Copy All* command.

The RETAIN\_HARDLOCATION\_ON\_COPY directive when set to OFF in the .cpm file ensures that the value of the LOCATION property is reset to ? when you copy an instance with a hard location. This directive is set to ON by default because of which the value of the LOCATION property is retained when a component is copied.

### Copying Wires

When you copy a wire, all the visible properties of the wire are copied. While pasting wires, you can choose between the *Paste* or *Paste Special* options. If you choose the *Paste* option, the wire is placed at the target location on the schematic. If you choose the *Paste Special* option, you can change the signal name before pasting the wire at the desired location.

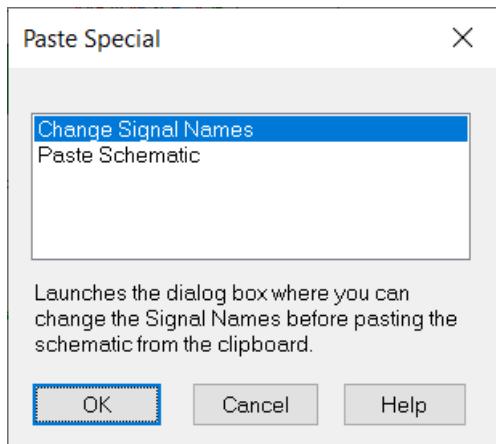
### Using Paste Special

1. Right-click and choose *Paste Special* from the pop-up menu.

## Allegro Design Entry HDL User Guide

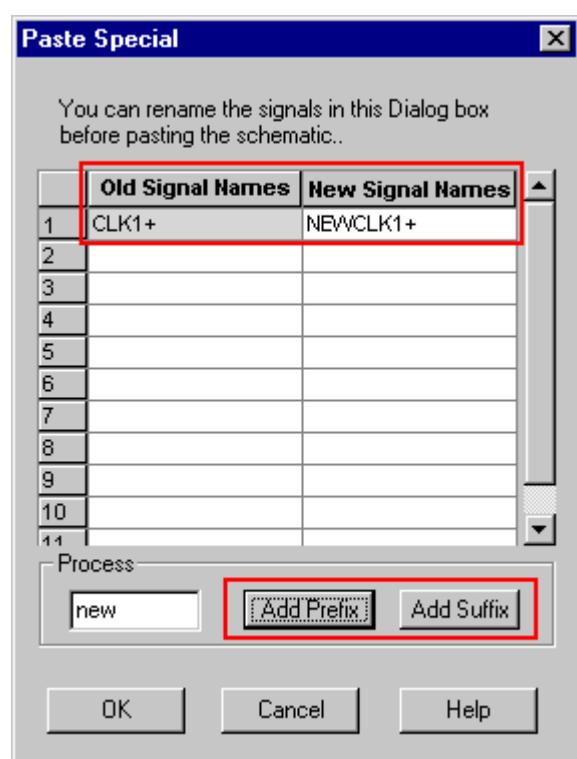
### Design Entry HDL Editing Environment

The following dialog box appears. You can paste copied parts on to the schematic by selecting the *Paste Schematic* option or change the signal names of selected parts and then copy them on to the schematic by selecting the *Change Signal Names* option.



2. To change signal names, select the *Change Signal Names* option and click *OK*.

A two-column dialog box with the current signal names and Design Entry HDL-recommended signal names appears.



## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

You can choose to edit these signal names or leave a signal unnamed. You can also add a prefix or a suffix to the existing signal name.

3. To edit signal names, enter a new value in the *New Signal Name* field and click *OK*.

The signal names are changed and the copied part is highlighted.

**Note:** You can only select a contiguous block of signal names in the Paste Special dialog. If you need to work on separate groups, repeat steps 1-3. After you make your changes, click *OK* to update the data in the graphics which are attached to your cursor.

4. Click at the target location on the schematic canvas.
5. Click again to open the Paste Special dialog box and repeat steps 2 to 4 for another paste special operation.

### Copying Blocks

If the block you are copying is in the local libraries, performing the paste operation would be the same as instantiating the block in a schematic. If the required libraries are not included in the *cds.lib* file, a warning is displayed.



### Copying Page Border

When you copy a page border, the placeholders for custom variables added to the symbol or on the canvas are also copied. If a user-defined custom variable that is not defined in the new project is pasted, it becomes a dangling property.

**Note:** An error message is flagged in the following cases:

- A non-standard page border is used in a manner where the border is not included in the project.
- A primitive is not defined in any library.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

- A user-defined block is not included in the local libraries of the project.

#### Copying Text from a Text Editor to a Schematic

When you copy text from a text editor such as WordPad, Notepad, or from HTML or Acrobat Reader files, it is added to the schematic as a note.

To copy a piece of text from a text editor, do the following:

1. Select the text in the text editor.
  2. Do one of the following:
    - Choose *Edit – Copy*.
    - Right-click on the text and select *Copy* from the pop-up menu.
  3. Click the location on the schematic canvas where you want to place the text.
- The selected text is placed on the schematic as a note. The casing is all upper case characters. To retain the original text casing, make changes in the Design Entry HDL Options – Text page.
4. To display text setup options, choose *Tools – Options* and select *Text* in the left pane.
  5. To retain the casing of the original text, uncheck the *Upper-case Input* box in the Design Entry HDL Options dialog box.
  6. Click again to paste another copy of the text.

The note is copied in original casing. However, the original text size is not retained. The size of the input text depends on the size specified in the *Size* spin box in the Design Entry HDL Options – Text page. The default font, *Concept Font* is used when placing a piece of text on the schematic.

7. Right-click and select *Done* to complete the operation.

Based on the nature of the copied text, Design Entry HDL processes it as shown in the table below:

When you copy a ...	Design Entry HDL ...
---------------------	----------------------

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

Table	Copies text in plain format. The formatting of the table is not retained and text formatting information, such as indentation, lists, bullet lists, and paragraphs is ignored while copying text. Every paragraph of text is added as a separate note.
Hyperlink	Copies it as an active link in the form of a note. If you double-click on this note, the appropriate link opens in a web browser.
Special character	Copies it as a note.

### Copying a Group

You can also create a group of components on the schematic and then copy the contents of the group to another page or design.

To copy a group, do the following:

1. Create the group.
2. Choose *Group – Copy [group\_name]* or *Group – Copy All [group\_name]*.
3. Right-click the target location on the schematic and choose *Paste* or *Paste Special* from the pop-up menu.

**Note:** A warning may appear if all the required libraries are not included in the `cds.lib` file of the target project

4. Right-click and choose *Done* from the pop-up menu.

### Copying Previously Executed Commands in the Console Window

In addition to copying and pasting to and from the schematic canvas, you can copy previously-executed commands in the console window. This saves you the trouble of re-entering frequently used commands in the console window. You select a piece of text in the console window and copy and paste it to execute it again. If Design Entry HDL correctly matches the pasted text to an existing command, it is executed. Otherwise, an error message is displayed.

To copy a previously executed command in the console window, do the following:

1. Select the command or text in the console window that you want to re-execute.
2. Press **<CTRL> + <C>** or **<CTRL> + <INSERT>**
3. Go to the next line in the console window and press **<CTRL> + <V>** or **<SHIFT> + <INSERT>**.

The command is executed again.

 *Important*

If you have customized the  $<CTRL> + <C>$  and  $<CTRL> + <V>$  key combinations for the *Edit – Copy* and *Edit – Paste* menu commands respectively, you cannot use the key combinations in the console window for copying text to re-execute a previously-entered command.

**Note:** You can also copy commands from other text editors. Valid commands terminating with *Enter* are executed.

### Copying Bitmaps to the Schematic Canvas

In addition to copying text and previously-executed commands, Design Entry HDL also supports copying of *.bmp* and *.jpeg* format bitmaps from an external graphics editor to the schematic canvas. You could use this functionality to paste a company logo to a page border.

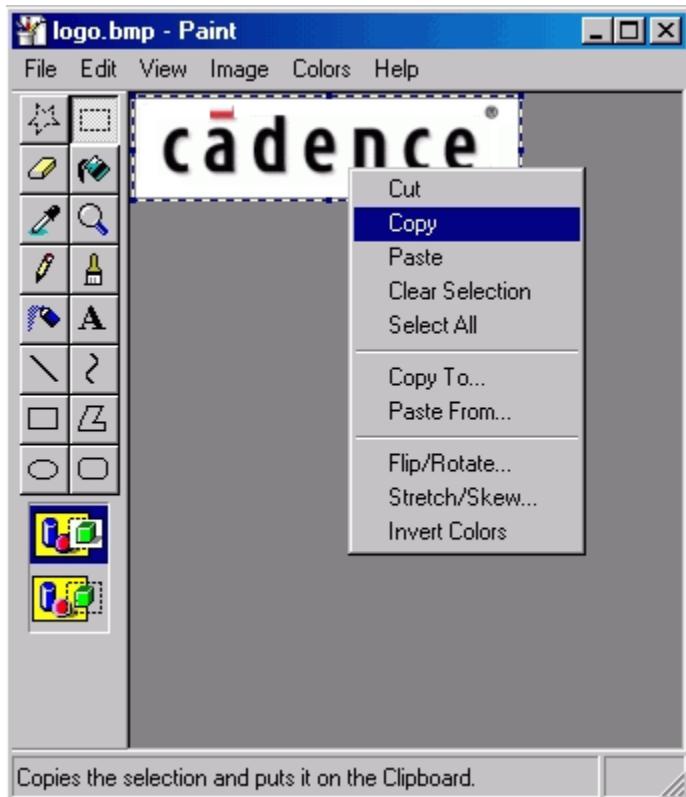
To copy a bitmap from a graphics editor to the schematic canvas, do the following:

1. Open the bitmap to be pasted in a graphics editor and select it.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

2. Select *Edit – Copy* from the main menu of the graphics editor or select *Copy* from the pop-up menu.



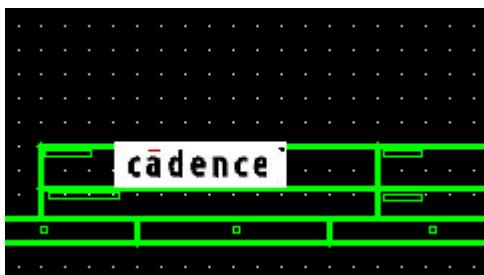
3. Go to the desired page on the schematic canvas.

4. Right-click the target location on the schematic canvas and select *Paste* from the pop-up menu.

The bitmap is attached to the cursor.

5. Click at the target location.

The bitmap is pasted on the schematic canvas.

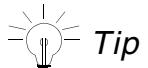


6. Right-click and choose *Done* from the pop-up menu.

# Allegro Design Entry HDL User Guide

## Design Entry HDL Editing Environment

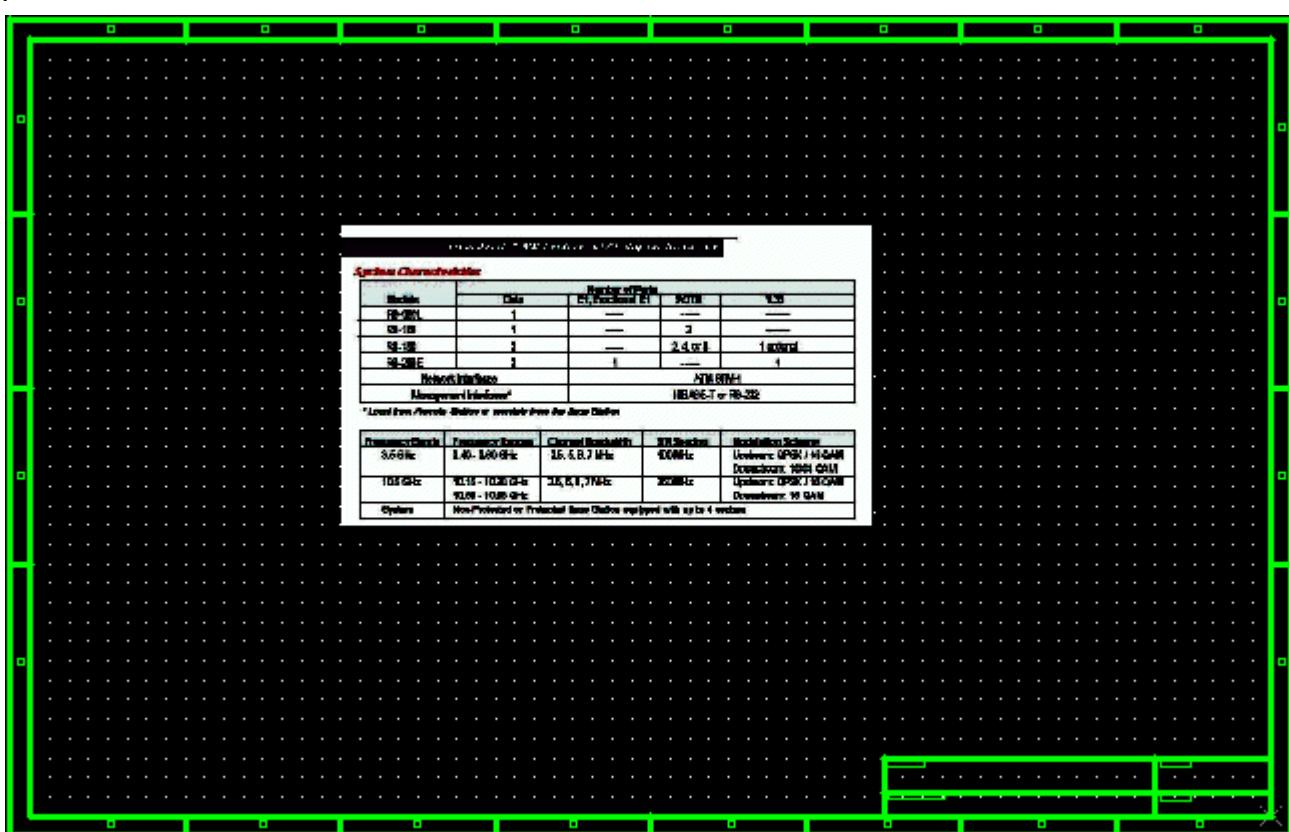
**Note:** When you paste an image, the resolution of the image is set to 72 dots per inch (dpi). The dpi value defines the resolution or the pixel density of the image pasted on the schematic canvas. If you increase this value, the size of the graphic reduces resulting in sharper images. To change the value, you need to add the IMAGE\_DEFAULT\_DPI directive in the project.cpm file. For example, to change the value to 150, add IMAGE\_DEFAULT\_DPI '150'.



You can paste logo images to the page border symbols. These symbols can be saved in a library for future component retrieval, inclusion, and browsing.

### Example

In addition to adding logos, you can add datasheets into schematics. While it is possible to copy text within a datasheet on to a schematic, table-based formatting would be lost in the text paste operation. You can take a snapshot of the datasheet and then paste it on a schematic page. The following is an example of a datasheet copied as an image:



### Important

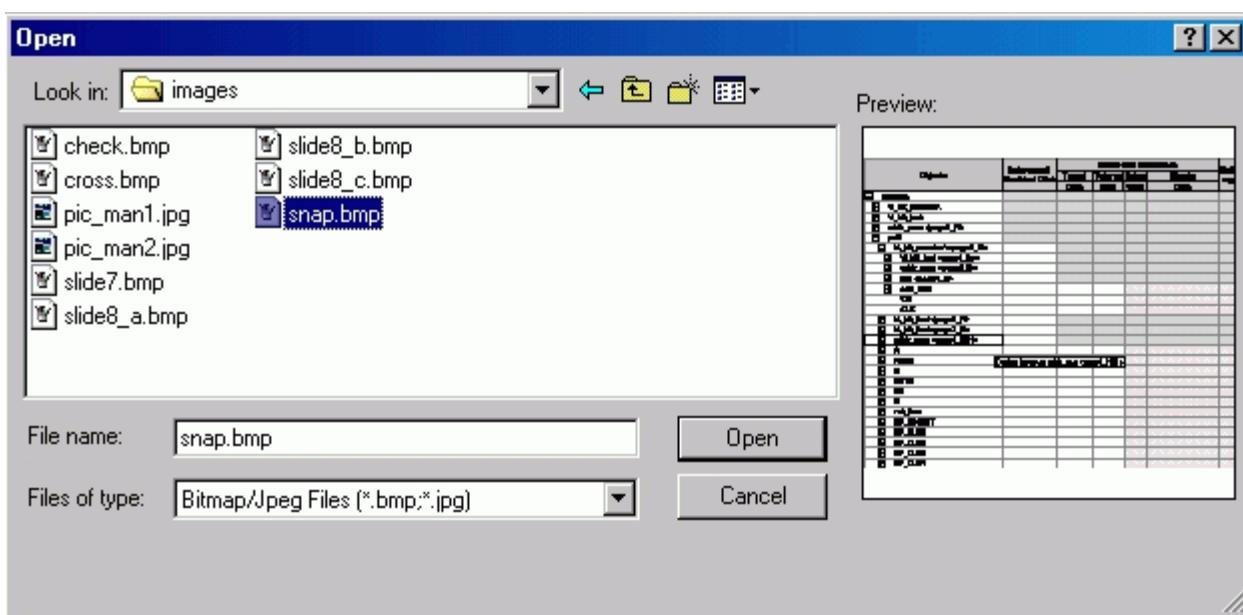
You cannot paste a combination of image and text from a text editor to the schematic canvas. If you copy a piece of text and graphic together from a text editor such as Microsoft Word, the image is lost and only the text is available for pasting on the schematic.

### Alternative Ways of Inserting an Image

You can also import an image by clicking the *Insert Image* icon on the Add toolbar.

To insert an image, do the following:

1. Click the *Insert Image* icon on the Add toolbar  
The Open dialog box displays.
2. Select the image that you want to place on the schematic canvas.  
You can also preview the image in this dialog box.



3. Click *Open*.

The image is attached to the cursor.

4. Click at the desired location.
5. Right-click and choose *Done*.

### The Image – Insert menu

Another way of inserting an image in a schematic is using the *Edit – Image – Insert* menu command. Alternatively, you can choose *Image – Insert* from the pop-up menu that displays when you right-click a schematic or a symbol. This action opens the Open dialog box from where you can select the image that you want to insert.

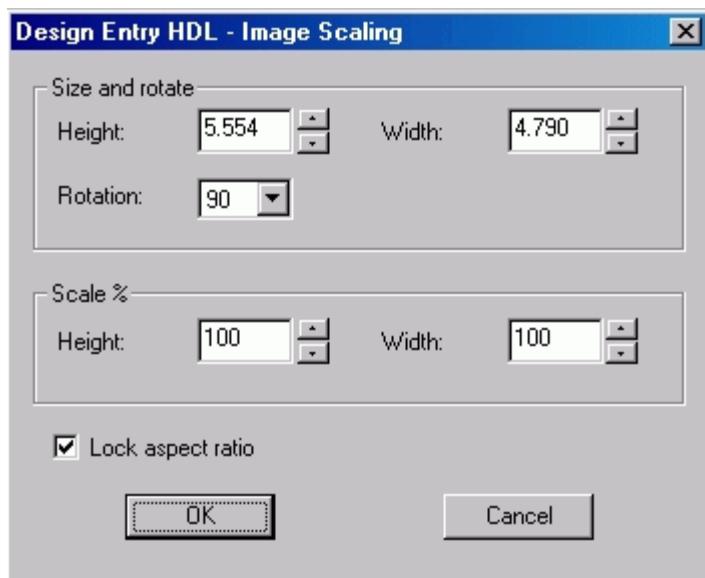
### Scaling an Image

After pasting an image on the schematic canvas, you can scale its height and width according to your specifications. The *Image Scaling* dialog box provides you the option of scaling the height and width of an image.

To scale an image, do the following:

1. Right-click on the image.
2. Choose *Properties* from the pop-up menu.

The Design Entry HDL - Image Scaling dialog box appears.



#### Field Description

**Size and rotate: Height** Sets the height of the image (in inches) according to the value you specify in this field.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

<b>Size and rotate: Width</b>	Sets the width of the image according to the value you specify in this field.
Rotation	Lets you rotate the image to a specified angle. You can choose from 0, 90, 180, and 270 degrees.
Scale %: Height	Lets you scale the height of the image by a specified percentage. The height of the image (in inches) is adjusted to reflect the percentage you specify in this field.
Scale %: Width	Lets you scale the width of the image by a specified percentage. The width of the image (in inches) is adjusted to reflect the percentage you specify in this field.
Lock aspect ration	Select this check box to ensure that scaling does not change the width:height ratio. With this check box selected, when you change one of the options, (height, width, or the scale value for height or width), DE-HDL automatically adjusts the values of the other three options based on your input for the selected option. Uncheck this box to scale the width and height independently.

3. Specify the desired values in the Height and Width fields.

4. Click *OK*.

The image is scaled according to the values that you specify.

**Note:** You can also stretch an image by choosing the *Edit – Image – Stretch* menu item.

### Capturing an Image on Schematic Canvas

You can also capture an image of part of a schematic. Use the *Image – Capture* menu item from the pop-up menu to capture screen shots of a selected part on a schematic. When you capture an image, it is copied to the clipboard from where it can be pasted into any graphics editor or a graphics-aware text editor such as Microsoft Word.

To capture a screen shot of any part of a schematic or symbol, do the following:

1. Right-click the schematic canvas or symbol.
2. Choose *Image – Capture* from the pop-up menu.

A rectangular cursor appears

3. Drag a rectangle to select an area of the schematic you want to capture.

The selected area is copied to the clipboard.

4. In a graphics editor (or a graphics-enabled text editor), choose *Edit – Paste* or *Paste* from the pop-up menu.

The captured screen shot is pasted in the editor.

## Displaying Information

This section covers the following information:

- [Displaying Schematic Information](#) on page 111
- [Displaying Toolbars and Other Parts of the Design Entry HDL Window](#) on page 115
- [Highlighting Objects](#) on page 116
- [Turning Off Highlighting](#) on page 117
- [Opening the Markers Control Window](#) on page 118
- [Displaying the Markers Toolbar](#) on page 118
- [Displaying the Error Status Bar](#) on page 118

### Displaying Schematic Information

Design Entry HDL lets you highlight selected objects in drawings, between drawings, and between Design Entry HDL and other applications.

Design Entry HDL also displays information about these items:

Attachments	Modified (drawings)
Color	Nets
Component	Origins (objects)
Connections (wires)	Pins (locations)
Coordinates	Pin Names
Directory (current)	Properties
Distance (point to point)	Return
History (drawings)	Text Size
Keys (assignments)	

## Displaying Attachments Between Properties and Objects

Choose *Display – Attachments*. Attachments display in red.

**Note:** Choose *Window – Refresh* to clear your selections or any Design Entry HDL display information.

## Displaying Component Information

1. Display the console window.
2. Choose *Display – Component*.

Component information is displayed in the console window.

## Displaying the Color of Objects

1. Display the console window.
2. Choose *Display – Color*.

Component information is displayed in the console window.

## Displaying Wire Connections

Choose *Display – Connections*. An asterisk appears at each wire connection.

**Note:** Choose *Window – Refresh* to clear your selections or any Design Entry HDL display information.

## Displaying Coordinates

1. Display the **console** window.
2. Choose *Display – Coordinates*.
3. Click in the schematic.

The x.y location of a point is displayed in the console window.

**Note:** The point you specify will clear with your next menu selection.

## Displaying the Current Directory

1. Display the console window.

#### 2. Choose *Display – Directory*.

The current directory is displayed in the console window.

### Displaying the Distance Between Points

#### 1. Display the console window.

#### 2. Choose *Display – Distance*.

#### 3. Click at one point in the schematic and then at a second point.

The distance between them is displayed in the console window.

**Note:** The points you specify will clear with your next menu selection.

### Displaying the Drawings Read Into Design Entry HDL in the Current Session

#### 1. Display the console window.

#### 2. Choose *Display – History*.

Drawing names are listed in the console window.

### Displaying Function Key Assignments

Choose *Display – Keys*.

Function key assignments are listed in a message box.

### Displaying Modified Drawings

#### 1. Display the console window.

#### 2. Choose *Display – Modified*.

Drawings that were modified but not saved are listed in the console window.

### Displaying Nets

#### 1. Choose *Display – Net*.

#### 2. Click a wire (net).

The selected net is displayed in red.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

**Note:** Choose *Window – Refresh* to clear your selections or any Design Entry HDL display information.

#### Displaying Origins of Objects

Choose *Display – Origins*.

An asterisk appears at the origin of all objects in the schematic.

**Note:** Choose *Window – Refresh* to clear your selections or any Design Entry HDL display information.

#### Displaying Pins

Choose *Display – Pins*.

An asterisk appears at all pin locations in the schematic.

**Note:** Choose *Window – Refresh* to clear your selections or any Design Entry HDL display information.

#### Displaying Pin Names

1. Zoom in on the component whose pin names you want to display.
2. Choose *Display – Pin Names*.
3. Click the component.

Pin names appear next to each of the pins.

**Note:** Choose *Window – Refresh* to clear your selections or any Design Entry HDL display information.

#### Displaying Property Information

Choose *Display – Properties*.

Property names and their values appear for each property in the schematic.

**Note:** Choose *Window – Refresh* to clear your selections or any Design Entry HDL display information.

#### Displaying the Name of the Previous Drawing

1. Display the console window.
2. Choose *Display – Return*.

The names of drawings that were edited in the current window are listed in the console window.

#### Displaying Text Size

1. Display the console window.
2. Choose *Display – Text Size*.
3. Select some text.

The size of the selected text is displayed in the console window.

**Note:** Choose *Window – Refresh* to clear your selections or any Design Entry HDL display information.

#### Displaying Pages in a Multipage Drawing

- To view the next page of a drawing, choose *File – Edit Page – Next*.
- To view the previous page of a drawing, choose *File – Edit Page – Previous*.
- To view a specific page of a drawing, choose *File – Edit Page – Go To*.
- To display the previous window, choose *View – Previous View*.

Design Entry HDL displays the drawing as it appeared before you changed the view.

#### Displaying Toolbars and Other Parts of the Design Entry HDL Window

**To display toolbars, do the following:**

1. Choose *View – Toolbars*.
2. Click the check boxes next to the toolbar you want to display:
  - Standard
  - Add

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

- Block
- Markers
- Edit
- Color
- Analog
- Passive
- Source
- Linear
- Discrete
- Misc

Design Entry HDL places a check in the box next to the toolbar you want to display. If a toolbar is already checked (selected) and you click the check box next to the item, Design Entry HDL removes the check, turning off that toolbar.

3. Click *OK*.

#### ***To display a grid, status bars, or the console window, do the following:***

Choose *View* and then choose one of the items you want to display:

- Grid
- Status Bar
- Error Status Bar
- Console Window

When you choose one of these items, Design Entry HDL places a check next to it, turning it on. If an item is already checked (selected), Design Entry HDL turns it off and removes the check.

## Highlighting Objects

To highlight an object, do the following:

1. Choose *Display – Highlight*.

#### 2. Click an object.

To highlight multiple objects, do the following:

1. Hold down the left mouse button and drag the mouse to select multiple objects, or use *Ctrl+click* or *SHIFT+click* to select multiple objects.

To exclude components, properties, or wires from the selected objects, right-click on the objects and choose *Exclude*.

2. To highlight objects, right-click and choose *Highlight*.

### Retaining Highlighting of Multiple Objects

By default, a highlighted object is dehighlighted when a new object is selected in Allegro PCB Editor or the Global Navigate window in Design Entry HDL.

In the case of groups, when you select multiple objects in PCB Editor by dragging the mouse, DE-HDL highlights *any one* of the selected objects in the group on the same schematic page. For example, if you group C1, C2, C3, and C4 on a board file in PCB Editor, DE-HDL will highlight any one of these objects in the schematic page. However, if you group objects by choosing *Edit – Groups*, DE-HDL will only highlight the last selected object in the group.

If you want all the selected objects on the same page to be highlighted at the same time, set the `RETAIN_PREVIOUS_HILITE` directive to `ON` in the project (`.cpm`) file. This ensures that all the selected objects are highlighted.

### Turning Off Highlighting

To de-highlight an object, do the following:

1. Choose *Display – Dehighlight*.
2. Click the pin, wire (net), or component that you want to de-highlight.

To de-highlight multiple objects, do the following:

1. Hold down the left mouse button and drag the mouse to select multiple objects, or use *Ctrl+click* or *SHIFT+click* to select multiple objects.

To exclude components, properties, or wires from the selected objects, right-click and choose *Exclude* to exclude components, properties or wires from the selected objects.

2. Click the right mouse button and choose *Dehighlight* to de-highlight the objects.

## Opening the Markers Control Window

To view markers, do one of the following:

- Choose *Tools – Markers*.
- Click  in the *Markers* toolbar.

## Displaying the Markers Toolbar

1. Choose *View – Toolbars*.

The *Toolbars* dialog box is displayed.

2. Check *Markers*.

3. Click *OK*.

## Displaying the Error Status Bar

Do one of the following:

- Choose *View – Error Status Bar*.
- Click  in the *Markers* toolbar.

To enlarge a drawing, do the following:

1. Choose *View – Zoom by Points*, or click the *Zoom Points* icon () in the standard toolbar.
2. Draw a rectangle around the component to be sectioned.

Alternatively, you can choose *View – Zoom In*, or click the *Zoom In* icon () to zoom in incrementally.

# Using the QuickPick Browser

## Overview

You might have designs that require the use of common parts and frequently-used library components. In such cases, searching and opening the same part repeatedly using the Part Information Manager window can be a time-consuming task that impacts productivity and efficiency. To save time, you can use the QuickPick Browser to do the following:

- Add frequently-used parts quickly in a design.
- Add commonly-used standard library components directly without using the Part Information Manager window.
- Add previously-used local blocks in a design.

**Note:** The QuickPick Browser is available only with Part Information Manager.

## QuickPick Browser - Interface Overview

You can access the QuickPick Browser feature using a toolbar that is located next to the standard Allegro Design Entry HDL toolbar. The QuickPick toolbar is visible as soon as you launch Allegro Design Entry HDL provided you have set the QuickPick Browser display (*View – Toolbars*). You can use the QuickPick toolbar to quickly add commonly-used cells, parts, and local blocks to a design.



The following table lists the various icons of the QuickPick toolbar.

Icon..	Lets you..
--------	------------



Add a cell listed under the Power category icon.

Power



Add a cell listed under the Ground category icon.

Ground

# Allegro Design Entry HDL User Guide

## Design Entry HDL Editing Environment

---

### Icon.. Lets you..

---



Add a cell listed under the Ports category icon.

Ports



Add a cell listed under the Taps category icon.

Taps



Add a cell listed under the Page Borders category icon.

Page  
Borders



Add a previously-used part in the design.

Parts



Add a previously-used block in the design.

Blocks

---

**Note:** The Parts list option ( of the QuickPick browser provides a list of previously-added physical parts used in a design. Logical parts added to the design will not be available in the Parts list.

## Setting up QuickPick Browser

Before using the QuickPick toolbar in Allegro Design Entry HDL, ensure that you specify the display of standard library cells under appropriate component categories such as Power, Ground, Ports, Taps, and Page Borders. This information, by default, is stored in the `qpsetup.qps` file at the CSF search location. For example, a location can be: `cdssetup/concept` in the site area.

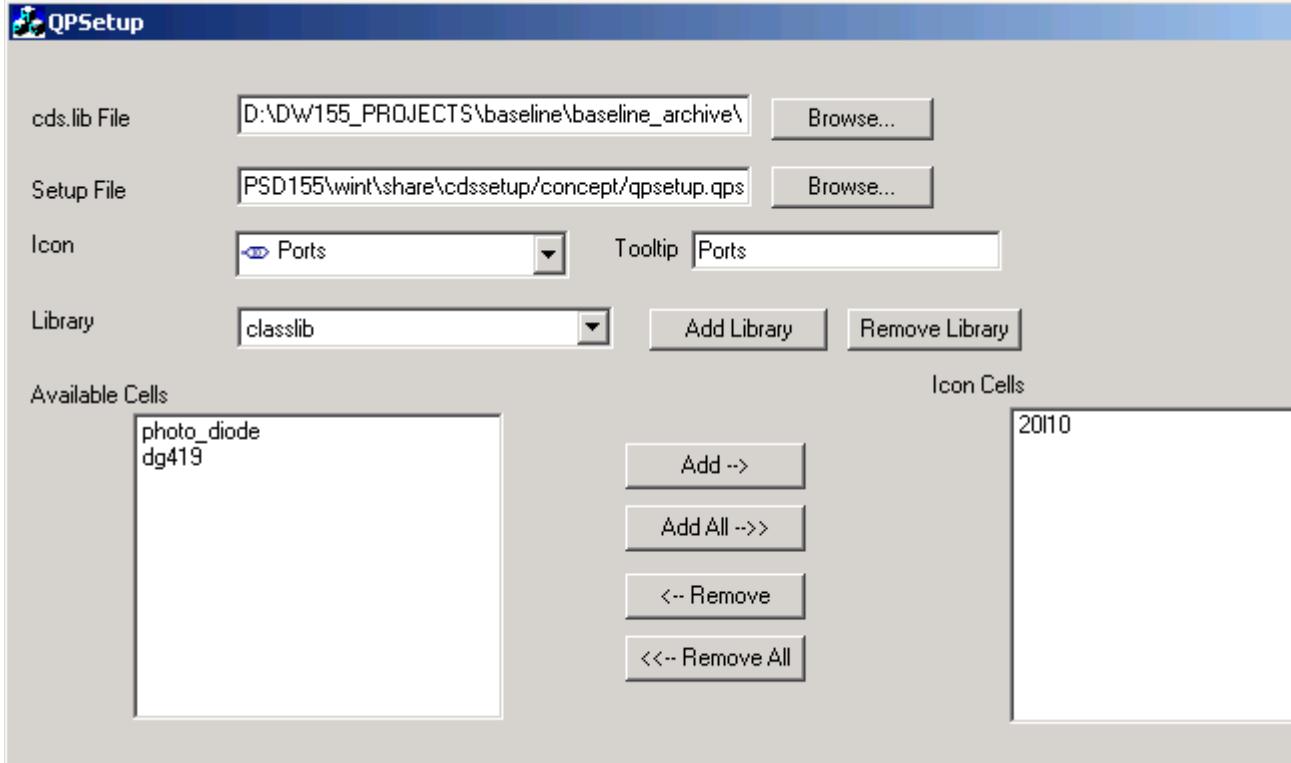
To set up the QuickPick Browser, do the following:

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

1. Open an instance of the command window.
2. Type `qpseteditor` and press ENTER.

The QPSetup dialog box appears.



3. In the *cds.lib File* field, specify the location of the `cds.lib` file of the design project. Alternatively, click *Browse* to navigate to the project file.
4. The *Setup File* field enables you to specify the location of the `qpsetup.qps` file that has existing QuickPick setup information for the design project. If the `qpsetup.qps` file is present at the project level (where the cpm file is present) or is available at the CSF search location, then, by default, the *Setup File* field will be populated with the location. Alternatively, click *Browse* to navigate to the file available at any other location.

#### *Important*

For a default display in this field, make sure that the name of the setup file is `qpsetup.qps`. QPSetup dialog box does not recognize any other name.

5. Select a component category from the *Icon* drop-down list. Your options are: Power, Ground, Ports, Taps, and Page Borders.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

6. Enter the text to appear as a tool tip when you position the cursor over a component category.

7. Select a standard library from the *Library* drop-down list.

By default, all the libraries specified in the `cds.lib` file are listed as drop-down list values. As soon as you select a library, all cells that belong to the library appear in the *Available Cells* list box.

8. Select a cell in the *Available Cells* list box and click *Add -->* to include the selected cell in the component category.

The selected cells move to the *Icon Cells* list box.

9. To include all cells of the selected standard library under a component category, click *Add All-->>*. Alternatively, click *Add Library*.

All cells in a library move to the *Icon Cells* list box.

10. To remove a cell, select it in the *Icon Cells* list box and click *<-- Remove*.

The cell moves to the *Available Cells* list box.

11. To remove all the cells listed in the *Icon Cells* list box, click *<<--Remove All*. Alternatively, click *Remove Library*.

**Note:** Hold down the left mouse button and drag the mouse to select multiple cells, or use Ctrl+click or Shift+click to select multiple cells.

12. To save the configuration for component categories in the default file (`qpsetup.qps`), click *Save*. This information, by default, is saved in the `qpsetup.qps` file located in your project directory. Click *Save As* to save the current setup information in a new file in any location.

13. Click *OK* to close the QPSetup dialog box.

## Working with QuickPick Toolbar

You can use the QuickPick toolbar to add standard library cells, frequently-used parts and local blocks to a design.

To add a library cell, do the following:

1. Click the down arrow beside a category icon.

A drop-down list containing all the cells relevant to the category specified in the `qpsetup.qps` file appears.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

2. Choose a cell from the drop-down list.
3. Place the cell on the schematic.

To add a part or block, do the following:

1. Click the down arrow beside the *Parts* and *Blocks* icons.

A drop-down list containing all frequently-used parts or blocks local to the design appears.

2. Choose a part or local block from the drop-down list.
3. Place the part or block on the schematic.

**Note:** A cell with a schematic view (`sch_n`) that contains the `pc.db` file is considered as a block. All such cells appear as blocks in the drop-down list (beside the *Blocks* icon).

**Note:** If you are using a design (made in previous versions of Allegro Design Entry HDL) and want to use the QuickPick toolbar functionality with it, first run the following command-line tool with the options:

```
partmgr -proj <path to cpm file> -ptfMode shoppingCart -product  
Concept_HDL_expert
```

To run this script successfully, you must have the Allegro Design Entry HDL Expert license. However, the QuickPick feature is available regardless of the license you have.

When you remove a part from the design, it is not removed from the Parts list. This might result in the toolbar containing parts that are no longer in the design. In such situations, Cadence recommends that you use the aforementioned command-line tool to update the QuickPick toolbar design. This will ensure that the toolbar contains all the parts currently in the design, at any moment.

## Working with the QuickPick Browser Window

You can use the *QuickPick Browser* window to view descriptions of frequently-used local parts and blocks, and apply filters to narrow the list of cells, parts or blocks.

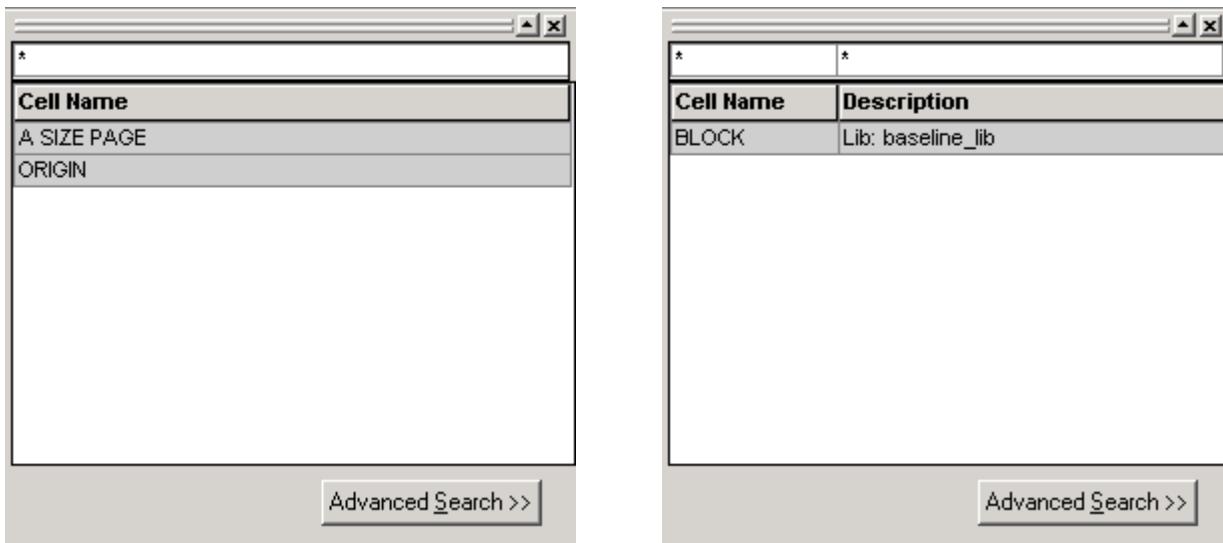
To add a cell, part or block using the *QuickPick Browser* window, do the following:

1. Double-click a category icon in the toolbar.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

All the available cells under the category appear in the *QuickPick Browser* window. The name and description of the cells appear in either a two-column or a single-column grid depending on whether you are adding a cell, part or block.



2. Select a row containing cell, part or local block.
3. Place the cell, part or block on the schematic.

**Note:** The QuickPick Browser window does not refresh when a new part is added from Part Information Manager. However, the new parts you add are updated in the corresponding drop-down lists. To refresh the QuickPick Browser window, click on the respective category icon in the QuickPick toolbar.

### Filtering Information in the QuickPick Browser Window

Both columns have filters that support wildcard characters. These filters have a drop-down combo box that shows all the valid values which you can use to filter out the required values.

To use a filter, do one of the following:

- Enter a string in the filters to filter cell, part or block names, and press Enter.
- Choose a value from the filters.

The required values appear in the column.

## Basic Navigation in Design Entry HDL

This section covers the following information:

- [Panning the Drawing](#) on page 125
- [Zooming In and Out of the Drawing](#) on page 126
- [Navigating the Drawing Hierarchy](#) on page 126
- [Moving a Window](#) on page 127
- [Resizing a Window](#) on page 127
- [Closing a Window](#) on page 128
- [Navigating a Design](#) on page 128
- [Finding Nets and Cells in Your Design](#) on page 128
- [Navigating Nets](#) on page 130
- [Exiting Design Entry HDL](#) on page 131

### Panning the Drawing

***To pan using the mouse, do the following:***

1. Press and hold the right mouse button or press *SHIFT* and hold the right mouse button.
2. Move the mouse to view portions of the drawing.

**Note:** Enabling the *Window Autopan* option causes the window to move over the drawing. Turning off *Window Autopan* causes the drawing to move inside the window. (See the *General* tab under *Tools – Options*).

***To pan using scroll bars, click the slider in either the vertical or horizontal scroll bar and drag it.***

***To pan using the keyboard, do the following:***

1. Press and hold *Ctrl*.
2. Press any arrow key (up, down, left, right).

**To pan using the View menu, choose one of these:**

*View – Pan Up*

*View – Pan Down*

*View – Pan Left*

*View – Pan Right*

## **Zooming In and Out of the Drawing**

**To zoom into the drawing, do the following:**

1. Select one of the following:
  - Choose *View – Zoom In*.
  - Choose *View – Zoom Scale* and enter a scale factor, such as 2.
  - Choose *View – Zoom by Points* and stretch a rectangle around the area you want to zoom in.
2. Click slightly above and to the left or right of the objects.
3. Move the cursor down diagonally from where you first clicked.
4. Click again.

**To zoom out of the drawing, choose *View – Zoom Out* or *View – Zoom Scale* and enter a scale factor, such as .5.**

**To fit the drawing in the screen, choose *View – Zoom Fit*.**

## **Navigating the Drawing Hierarchy**

**To view a block diagram from the top-level schematic, do the following:**

1. Choose *File – Edit Hierarchy – Descend*.
2. Click a block in the schematic.

3. Click *OK* in the message box that appears.  
The block diagram is displayed.
4. Continue descending the drawing hierarchy by repeating steps 1 and 2.

***To ascend the drawing hierarchy from a lower level block diagram, do the following:***

1. Choose *File – Edit Hierarchy – Ascend*.
2. Click *OK* in the message box that appears.
3. Continue ascending the drawing hierarchy by repeating steps 1 and 2.

***To return to the previous drawing, do the following:***

1. Choose *File – Return*.

**Note:** You can view the list of the drawings that Design Entry HDL will return to and the order in which the drawings will be accessed by choosing *Display – Return*.

***Descending into the Drawing:***

1. Choose *File – Descend*.
2. Click the block or component whose logic you want to view.
3. Choose *File – Return* or *File – Ascend* to descend to the previous drawing.

## Moving a Window

1. Place the cursor in the title bar of the window.
2. Press and hold the left arrow key.
3. Slide the window to a new location.

## Resizing a Window

1. Place the cursor in a corner of the window.  
The cursor changes to a diagonal two-headed arrow.
2. Press and hold the left arrow key.

3. Stretch or reduce the window to a new size.

## Closing a Window

Choose *File – Close*.

Closing a window does not save the design. Design Entry HDL saves your design only if you choose *File – Save* or when you exit Design Entry HDL.

## Navigating a Design

- Choosing *File – Edit Hierarchy – Descend* or *File – Edit Hierarchy – Ascend* lets you navigate the drawing hierarchy.
- Choosing *Tools – Global Navigation* lets you navigate the entire design and helps in crossprobing between Design Entry and PCB Editor.
- Using *Tools – Expand Design* builds your design based on the views specified in the current expansion configuration. When you expand the drawing, views are derived from the configuration and not from the setup default. After expanding a drawing, you can navigate the drawing hierarchy using the *File* menu, by double-clicking objects, or by using the Hierarchy Editor.

The *Next Page* and *Previous Page* icons in the Standard toolbar let you move between the pages of a multipage drawing.

## Finding Nets and Cells in Your Design

To find nets and cells in your design, do the following:

1. Choose one of the following:
  - Choose *Tools – Expand Design*.
  - Choose *View – Expand Design* in the Windows Mode.
2. Select one of the following:
  - Tools – Global Find*.
  - Edit – Global Find* in the Windows Mode.
3. In the *Name* box, do one of the following:
  - The *Global Find* dialog box appears.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

- Type the name of the net or cell to be located. For example, typing `1s04` will find all instances of the `1s04` part.

**Note:** To find a vectored signal, `DATA[3..0]` or `DATA(3..0)`, type `DATA` or `DATA<3..0>` in the *Name* box.

- Select a previously-entered name from the list.

Select the *Using Wild Card* check box if you want to search using wildcards in the name.

4. Select the object type to be located, either *Net* or *Cell*.
5. If you want to restrict the search by property, do the following in the *With Property* section of the dialog box:

- Type or select a property name in the *Name* box.
- Type a property value in the *Value* box or type an asterisk (\*) to locate all objects that have the specified property name and any property value.

When you enter the property name and value, they are added to their respective list boxes so that they can be reused during the same design session.

6. Select an option to specify that you want the search results to be listed by full *Hierarchical Names* or by *Library Location*.
7. Click *Find* to begin the search.

The *Find* button changes to *Stop Find*, which you can use to cancel a lengthy search in progress. You can also click *Close*.

A message at the bottom of the dialog box tells you how many instances were found. The (unlabeled) status area box displays the instances of the object found, either by *Hierarchical Names* or by *Library Locations*.

8. Choose how you want a selected search result to be viewed: *Zoom to Object*, *Navigate* or both.
9. Click on a search result to view it in your design.

When you select a result to view, the page containing the object appears with the object highlighted. If you chose *Navigate*, the *Global Navigation* window appears so that you can move across the design to view all net instances listed in the search results box.

10. Perform one of the following steps:

- Click on another search result to view in the design.
- Search for a different net or cell by entering a new net or cell name in the *Name* box.

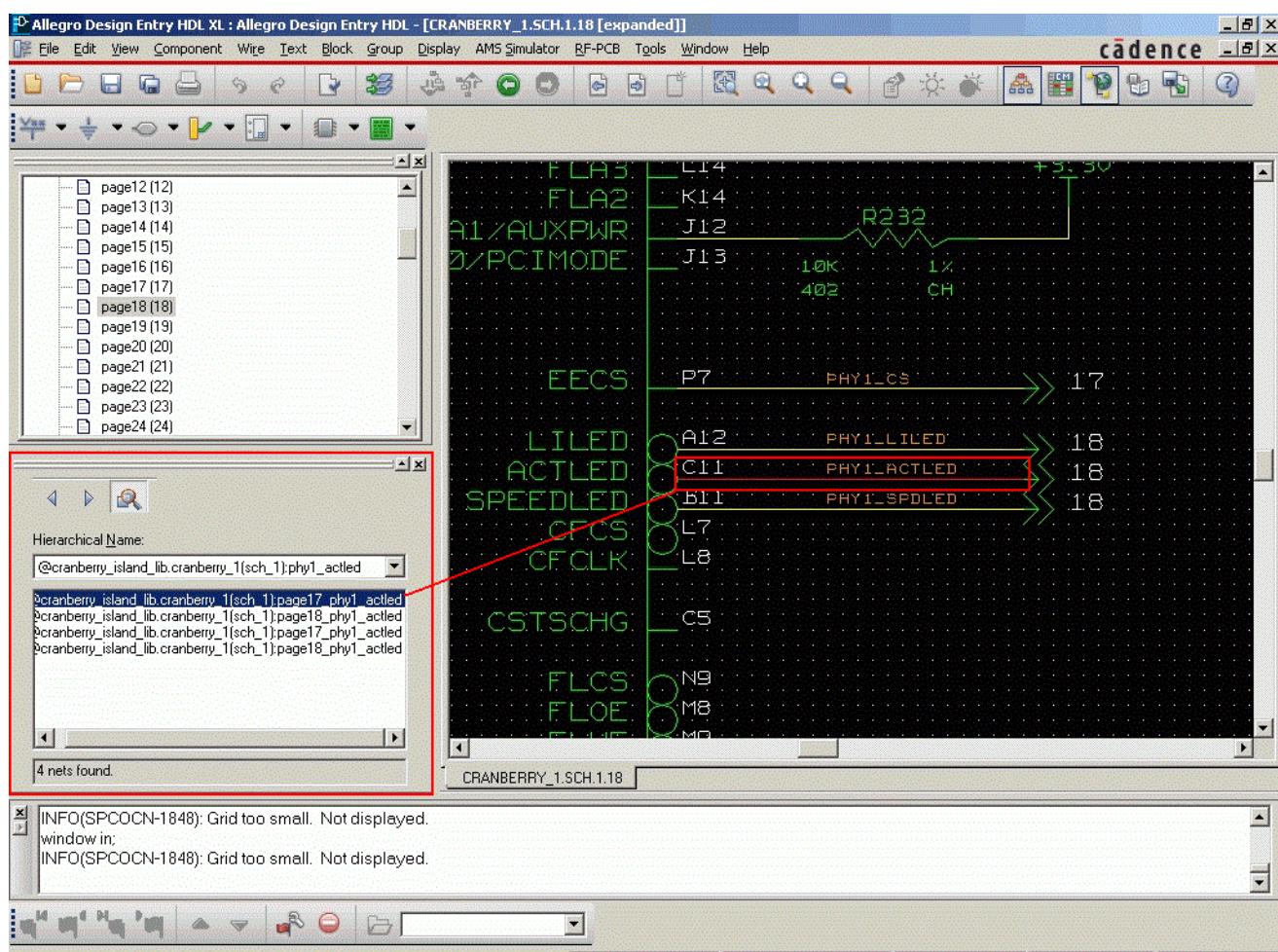
# Allegro Design Entry HDL User Guide

## Design Entry HDL Editing Environment

- Click *Close*.

### Navigating Nets

You can navigate nets in your design using the Global Navigation window. When you highlight a net or a part instance in PCB Editor, the Global Navigation window docks and all the aliases for the net or the part instance are listed in the Results window. You can navigate to the net or the part instance by selecting the aliases in the list. Additionally, you can select a net or an instance from the Hierarchical Names drop-down list box in the Global Navigation window. The aliases displayed in the Results area will correspond to the net or the part instance. The dockable Global Navigation window also makes it easier to manage the window along with the Design Entry HDL user interface.



## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

The Global Navigation window includes:

- The window includes Back and Next icons. These icons help you navigate between different nets and part instances that you select through the Global Navigation window during a single session of Design Entry HDL.
- The window is displayed in the Design Entry HDL user interface as a dockable window, by default.
- The window is not docked by default when you open Design Entry for the first time. When you dock the window and open it again or access Design Entry HDL the next time, the previous settings prevail and the window appears docked in the same location.
- The window allows you to clear the search results and search history. To do this, right-click the *Global Navigation* window and choose *Clear Results* or *Clear History* as required.
- During cross-probing:
  - On highlighting an instance in PCB Editor, the Global Navigation window docks, if it is undocked. The canonical name for the instance is seeded in the Global Navigation window. You can navigate to the instances by clicking on the canonical name listed in the Results area.
  - In case you highlight a net from PCB Editor, Design Entry will highlight the net.
  - All the aliases for the net are seeded in the Global navigation window. you can navigate to the net by clicking on any of the canonical names.

For more information about the *Global Navigation* window, refer to the [\*Global Navigation\*](#) section of Allegro Design Entry HDL Reference Guide.

To navigate nets in your design, refer to the [Navigating Nets in Your Design](#) section.

## Exiting Design Entry HDL

- Choose *File – Exit*.

Design Entry HDL prompts you whether or not to save any unsaved changes.

## Running Commands with Strokes

In the drawing area, place the cursor over an object, or if you are zooming or panning, over the region you want to view.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

**Note:** You can draw stroke patterns, such as undo (U) and zoom fit (W), anywhere in the drawing area because they do not create a bounding box or act on a specific object.

1. Press the Shift or Ctrl keys and hold down the left mouse button and make a stroke.

**Note:** If the *Ctrl+LMB Select and Drag* check box in the *General* tab of the *Design Entry Options* dialog box is selected, you need not press *Ctrl* or *SHIFT*. You only need to hold down the left mouse button and make a stroke.

In the *Windows* mode, the *Ctrl+LMB Select and Drag* check box is *disabled*, which means that you have to press either the *Ctrl* or the *SHIFT* key along with the left mouse button action to make a stroke.

2. As you move the mouse, you see the stroke pattern being drawn.
3. Release the left mouse button when the stroke is complete.
4. If Design Entry HDL recognizes the stroke, the command runs as though you typed it into the console window.

Design Entry HDL provides these default strokes (red indicates the starting point for the stroke pattern):

	attribute		note		zoom
	change		property		zoom fit (world view)
	copy		route		pan down
	delete		select		pan up
	exit		undo		pan left
	move		version		pan right

## Guidelines for Strokes

You apply the following guidelines when using strokes:

- Strokes must be entered in the same direction that they were created.
- For strokes that act on a single object, the object under the first point of the stroke is selected.
- For strokes that act on a group of objects, such as zoom (Z) and select (O or S), objects within the first and last points of the stroke are selected.

- Strokes that do not create a bounding box or act on a specific object can be drawn anywhere in the drawing area - for example, zoom fit (W) or undo (U).

**Note:** If you do not want to use strokes, you can turn off the feature by entering the following command in the console window:

```
set lmb off
```

To turn the feature on, enter the following command:

```
set lmb on
```

If you want to enable strokes for the right mouse button, you can turn it on by entering the following command:

```
set stroke_rmb on
```

To turn strokes off for the right mouse button, enter the following command:

```
set stroke_rmb off
```

## Support for Fonts

The Design Entry HDL environment supports different fonts. This allows you to set fonts for different categories of text objects such as the following:

- Symbol Text
- Symbol / Component / Instance Properties
- RefDes and Section
- Net Name
- Net Properties
- Cross References
- Pin Name
- Pin Text
- Pin Properties
- Custom Text
- Text / Notes

**Note:** When you cut/copy and paste a note on the canvas, the size of the copied text note is lost and the default text size is used to display the note.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

Fonts and font attributes, such as font name, style, and effects, can be set for a category of text objects. You cannot specify different font attributes on individual instances of text objects on the schematic. For example, if you specify ARIAL as the font to display net names, all the net names will be displayed in ARIAL. These font attribute settings you specify for each category of text object are written in the .cpm file but are not written to the database.

The font color and size attributes however are stored in the database for a specific instance of a text object. As a result, Font Color and Font Size can be defined differently for each individual text object **provided the text object already exists on a design**. New text objects that you add to a design inherit the font color and size defined in the font settings. You can modify the color and size of the text object after you add the object to the design.

If you open a pre-16.3 design, text objects are rendered in the selected fonts and attributes. However, the font color and size for existing objects do not change.

#### *Important*

When you add a part to a schematic, the font size and color defined for the symbol are used. These font attributes might not be the same as the font settings defined in *Tools — Options* for symbol and pin text. If you change the font size or color for symbol or pin text, they will be applied while adding text to the symbol drawing.

### Retaining Font Settings for Symbols

By default, DE-HDL applies the font settings you define in *Tools — Options* to symbol properties such as symbol text, or \$LOCATION. If you want the symbol font settings that were defined when creating the symbol to be preserved when instantiating a component on a schematic, set the RETAIN\_FONT\_SETTINGS\_ON\_SYMBOL\_ADD directive to ON in the START\_CONCEPTHDL...END\_CONCEPTHDL section of the .cpm file.

For example, assume that the text color is defined as red for a symbol in Part Developer. When you instantiate this symbol in DE-HDL, the symbol color will still be red regardless of the symbol text font color in DE-HDL.

If you have property placeholders, such as \$LOCATION and \$PN, with a defined size on library instances, the sizes are retained when you instantiate the symbol on a design. However, when you package and backannotate the design, DE-HDL applies the font settings defined in *Tools — Options*.

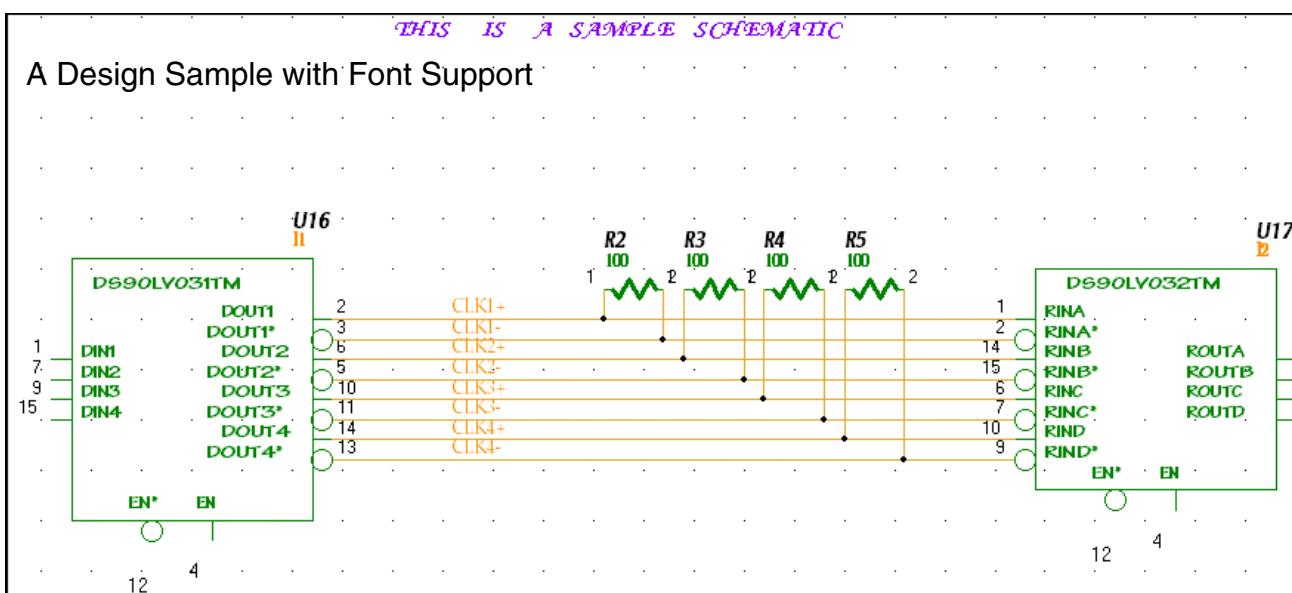
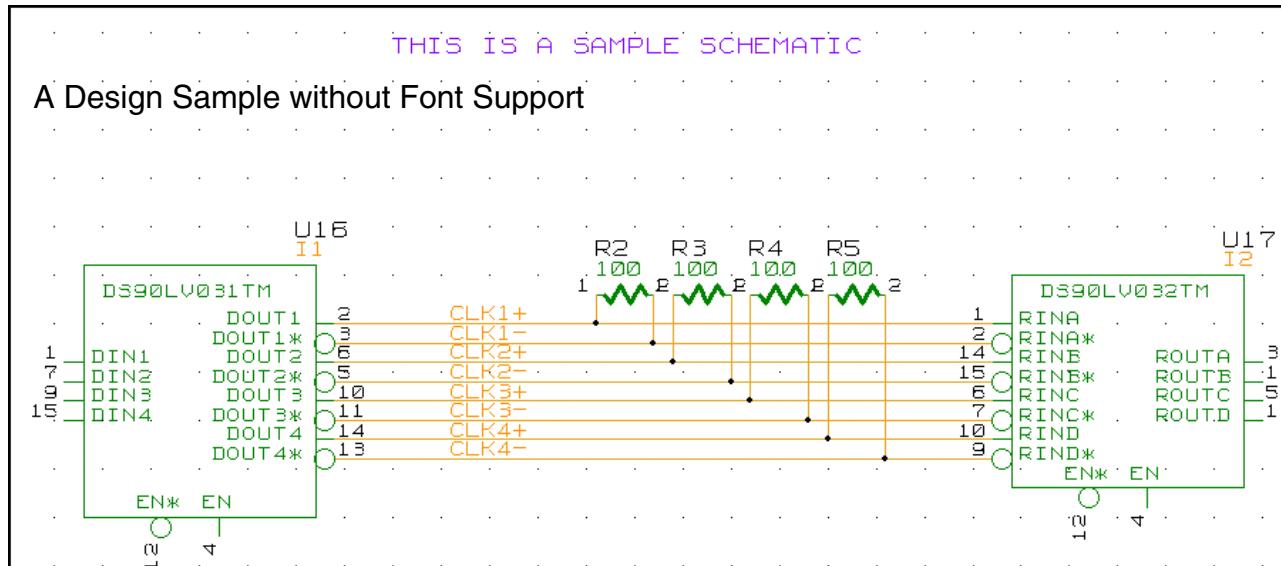
If you want the symbol font settings to be retained during backannotation, set the RETAIN\_FONT\_SETTINGS\_ON\_SYMBOL\_BA directive to ON in the START\_CONCEPTHDL...END\_CONCEPTHDL section of the .cpm file.

# **Allegro Design Entry HDL User Guide**

## Design Entry HDL Editing Environment



There are no UI options for these two directives in Design Entry HDL Tools — *Options*. The directives must be set in the `.cpm` file.

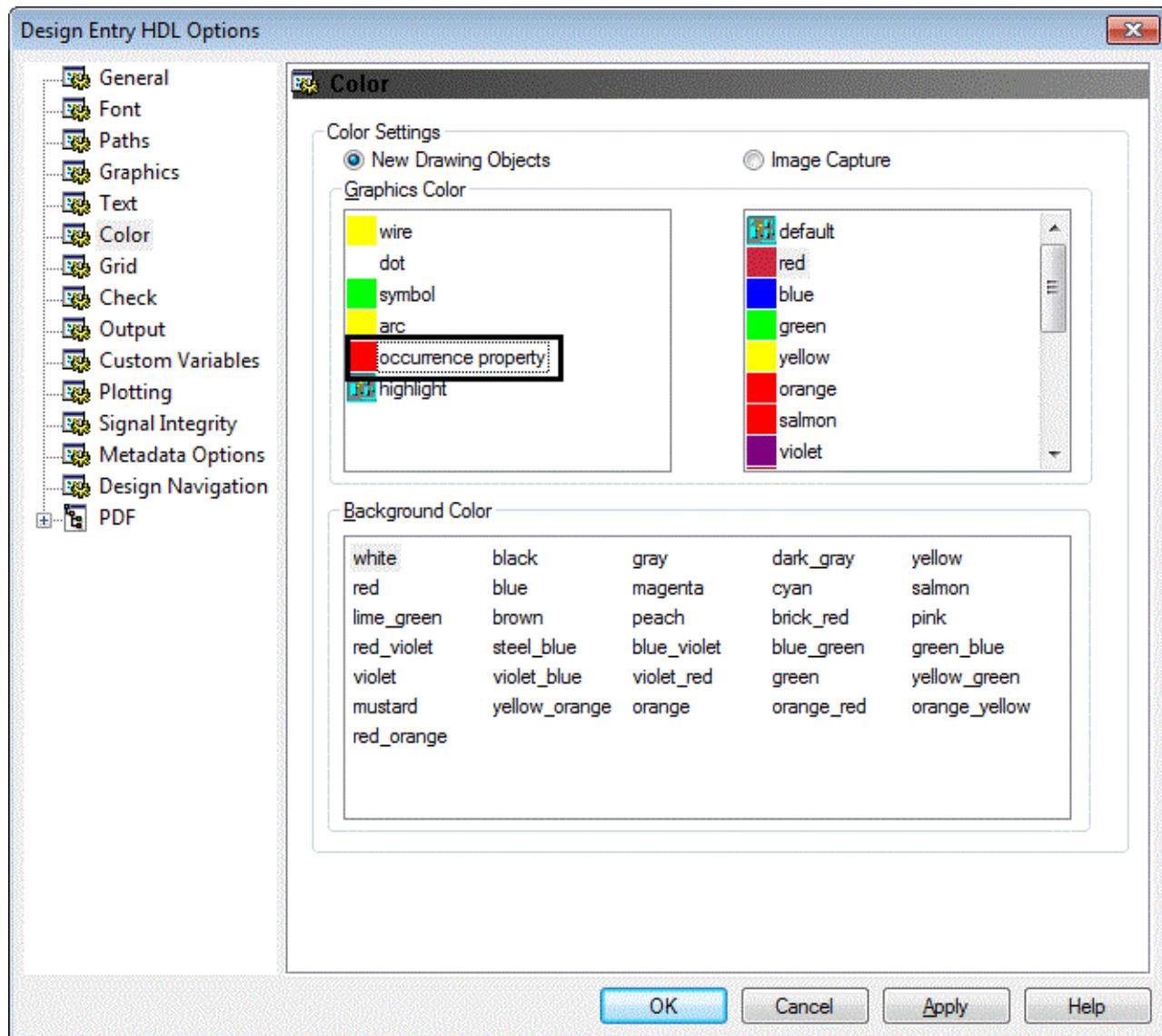


**Note:** If you open a pre-16.4 design, you do not need to change modes from the hierarchy to the expanded mode to the occurrence edit mode. The uprev of designs to the latest release leads to a single mode of operation in which the winning values of properties are always displayed on the schematic canvas. Therefore, the instance property settings are honoured only for master properties and not for occurrence properties. To change the settings for

## Allegro Design Entry HDL User Guide

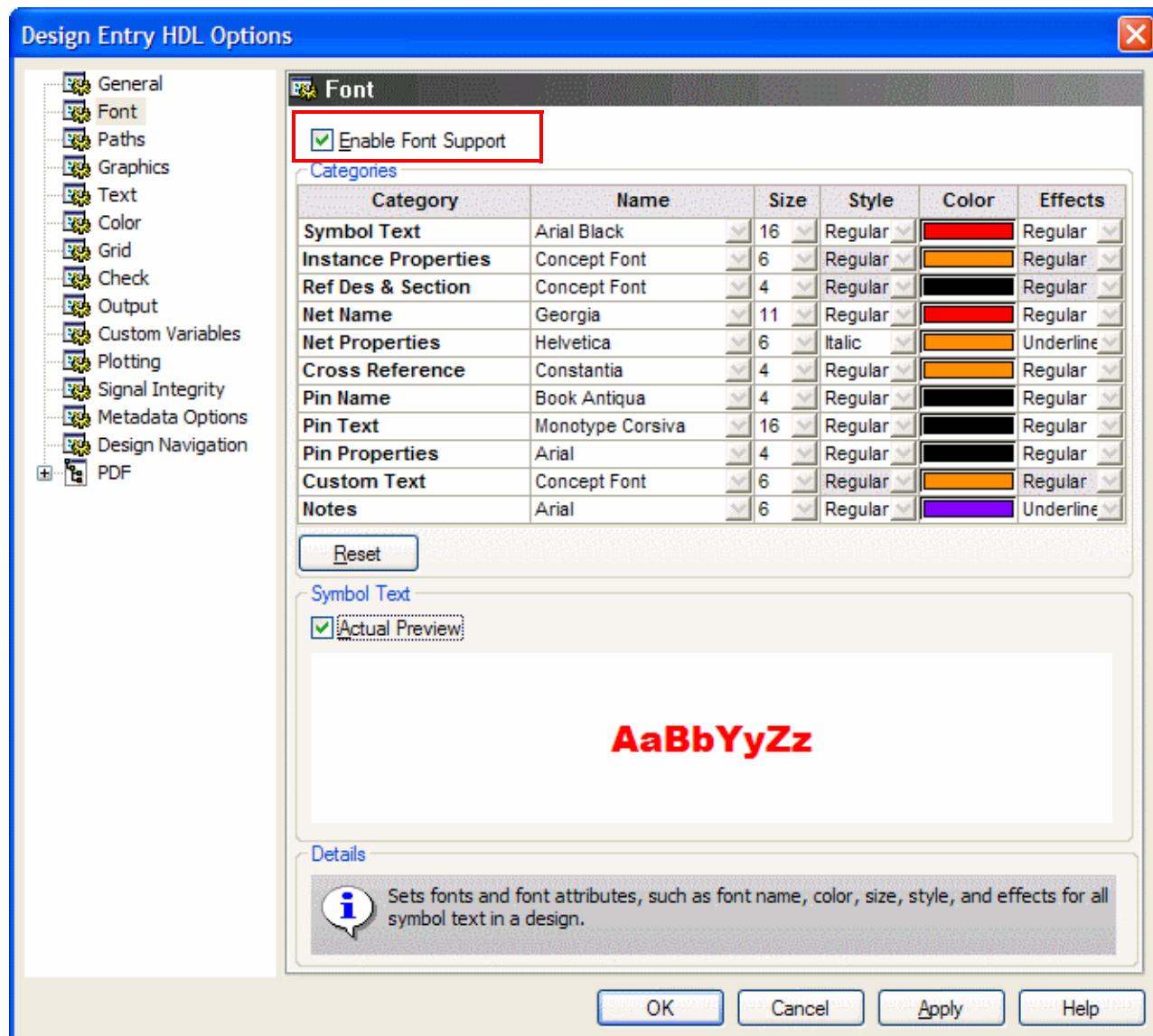
### Design Entry HDL Editing Environment

occurrence properties, use the following option: *Tools – Options – Color Occurrence Properties*.



## Design Entry HDL Options - Font page

In the Fonts page of the Design Entry HDL Options dialog box, you can specify font attributes for different types of schematic text objects. To enable support for displaying different fonts, ensure that the *Enable Font Support* option is selected.



## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

When you create a new design or turn on the font support feature in an existing design, font attributes are set to default values. You can change these values using this dialog box.

UI Option	Description
Enable Font Support	Select this check box to enable support for fonts in DE-HDL. This option is selected by default when you open any design in 16.3 or later releases.
Category	Select the schematic text object for which you want to set the font and font attributes. You can set font attributes for different categories of text objects simultaneously.  <b>Note:</b> Names of NetGroups use the same font size and color set for <i>Net Properties</i> .
Name	Select a font name to display a specific category of text objects. For example, you can select the Courier font to display all the net names in the design. The Name drop-down list includes all the fonts installed on the local system.
Size	Specify a font size with which all the newly-added text objects for the category are to be displayed. This size is also known as point size, where one point size equals 1/72 of an inch.  Currently, the font size stored in the DE-HDL database is a ratio of size in inches to 0.082. Therefore, a 1 inch font size is written in the database as $1/0.082 = 12.095$ . With font support in DE-HDL, the sizes being displayed are point sizes where 72 points = 1inch. Therefore, all the text sizes which are currently available in the database are converted to the point size and displayed accordingly.  <b>Note:</b> The text size is currently stored in the database. Therefore, you can specify the font size for individual objects. All the objects which are already on the canvas have a font size specified for them and the same font size is honored.
Style	Select a font style from the four font styles: Regular, Bold, Bold Italic, and Italic. All fonts do not support all the styles. Therefore, you can specify only those styles which are supported for a specific font. For example, you can specify all the four styles for the Arial font, while only Regular style is supported for Arial Black.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

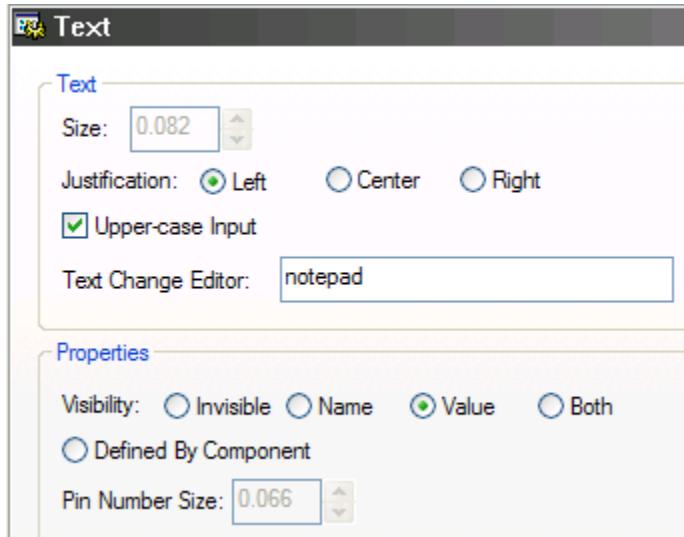
---

UI Option	Description
Color	Select a color from the following list of colors with which all newly added text for the category is to be displayed: Red, Blue, Green, Yellow, Orange, Salmon, Violet, Brown, Sky blue, White, Peach, Pink, Purple, Aqua, Gray, Mono.  <b>Note:</b> The text color, like the text size, is currently stored in the database. Therefore, you can specify the font color for a specific set of text objects. All the objects which are already on the canvas have a font color specified for them and the same font color is honored.
Effects	Select the Underline effect to display underlined text. By default, all text objects display regular text.
Reset	Resets the font settings back to the Cadence default.
Actual Preview	Shows the preview of sample text with the selected fonts and font attributes in the actual size it will appear on the schematic canvas. If this check box is not selected, the sample text appears in the default size.
Details	Displays information about the selected font attribute.

**Note:** The font style and font effects attributes are not applicable to the default vector font, the Concept Font. The only value which you can set for style and effect for the Concept font is 'Regular'.



**If font support is enabled for a design, you cannot modify the text or pin number size from the Text page of the Design Entry HDL Options dialog box.**



## Setting Fonts using CPM Directives

The specified font attributes for a text object are stored in the project (.cpm) file. You can bypass the Font dialog box and directly set the font attributes for each category of text objects in the `START_FONTS ... END_FONTS` section of the cpm file of the project.

To enable support for fonts using CPM directives, ensure that the `CDS_ENABLE_FONTS` directive is set to `ON` in the cpm file as shown here:

```
START_FONTS
CDS_ENABLE_FONTS 'ON'
:
:
END_FONTS
```

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

To set the values for the five font attributes of any category, set the following cpm directives:

#### Use this directive...

CDS\_<CATEGORY\_NAME>\_FONT  
CDS\_<CATEGORY\_NAME>\_FONTSIZE  
CDS\_<CATEGORY\_NAME>\_FONTSTYLE  
CDS\_<CATEGORY\_NAME>\_FONTCOLOR  
CDS\_<CATEGORY\_NAME>\_FONTEFFECTS

#### To set the value of...

Name  
Size  
Style  
Color  
Effects

Where,

#### Replace <CATEGORY\_NAME> with...

SYMBOLTEXT  
SYMBOL  
REFDES  
NETNAME  
NETPROP  
CROSSREF  
PINNAME  
PINTEXT  
PINPROP  
CUSTOMTEXT  
TEXTNOTES

#### For this category

SYMBOL TEXT  
INSTANCE PROPERTIES  
REF DES & SECTION  
NET NAME  
NET PROPERTIES  
CROSS REFERENCE  
PIN NAME  
PIN TEXT  
PIN PROPERTIES  
CUSTOM TEXT  
NOTES

Settings for the fonts can be stored in the project (.cpm) file. Therefore, these settings can also be stored in the site level project (.cpm) file and the common settings can be used by an entire company. To mandate a universal font setup for a company, the administrator can lock these settings in the site area and ensure that all users follow the same standards. PCB Librarian tools also provide support for fonts and use the same directives to display symbols. Therefore, the symbol editor canvas of PCB Librarian and the DE-HDL canvas have the same look and feel.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

For information on locking CPM file directives, refer to [Locking Project File Directives](#) on page 82.

When you set the fonts, the schematics are rendered in the specified fonts. When you plot the schematic or publish a PDF for the schematic, the generated output is WYSIWYG.

## Working with Fonts

When you set a value for a specific text object, the values of three font attributes—name, style, and effects—are applied across the design to the existing as well as newly created objects. The size and color attributes apply only to newly added text objects. These changes are not reflected on the existing text objects.

If you need to change font attributes for existing objects, you can write a script. For example, the following sample script changes the size of the `SIG_NAME` property:

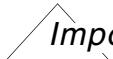
```
set nextgroup A
find sig_name
include properties
exclude notes
exclude bodies
textsize 0.104 A
write
```

**Note:** The script runs only on the currently open page. If you want to run the script on a complete design, you need to run the following command from the console window:

```
runscript <script name>
```

## Fonts and Allegro Design Publisher

When you plot a design with the font support feature on, the design is printed with the text objects appearing in the same fonts as they appear on the schematic canvas. Similarly, when you publish a PDF of the schematic design using Allegro Design Publisher (the Publish PDF utility), the published document displays the fonts as they appear on the schematic.



During PDF generation, Allegro Design Publisher ignores fonts that are not available and the default font is used to create the PDF document.



If a font support-enabled design is opened on a system where the fonts used in the design are not installed, an error is displayed and the text object is displayed in a vector font.

## Fonts on Different Platforms

All fonts available, or installed on the Windows platform, are available to DE-HDL. Cadence provides the Cadence Vector Font with the standard installation of Design Entry HDL, which is the current DE-HDL default font.

## Uprev of an Existing Design

When you open designs created in releases prior to the current release, the uprev utility of Design Entry HDL auto-detects the need to uprev the design.

The uprev utility is also available as a batch command, which can you execute on designs without opening DE-HDL. You can also write some simple scripts and run an operation to uprev designs. The process works like the Save Hierarchy operation; each hierachal block of the design is visited and then upreved to the current release.

**Note:** To open pre-16.5 release designs that have constraints on the schematic, you must first synchronize the constraints. The design is upreved only after the constraints are synchronized.

To synchronize constraints in pre-16.5 releases, you can do one of the following:

- Run the following command in batch mode:

```
concept2cm -proj < .cpm > -export -forward -uprevor
```

- Use the following option:

*Tools – Constraints – Synchronize*

The uprev process does not modify any of the schematic sheets. However, you will notice that some changes are made to the file structure.

Properties and constraints reside in a file, the `*.dcf` file, which is stored in the `sch_1` view.

Design connectivity information is stored in a design connectivity file that is XML based. This file has the extension `*.xcon`. The file helps load designs with the complete design connectivity information and also enables incremental netlisting.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

The table below summarizes the changes that occur to the schematic files.

File names in sch_1 view	Changes
master.tag	Changed to add xcon and dcf file names
verilog.v	Not created
viewprps.prp	Not created
hdldirect.dat	Not created
vlog004u.sir	Not created
design.xcon	New file created
design.dcf	New file created

**Note:** While upreving, if there is a block created with no schematic for the block, a warning message is displayed prompting you that the property block is not valid for the object.

## Single Mode operation in DE-HDL

If you work with pre-16.5 designs, you need to change the mode from hierarchy to occurrence. Post 16.5, you do not need to change the modes from the hierarchy to the expanded mode, to the occurrence edit mode. The uprev of designs to the 16.5 release leads to a single mode of operation in which the winning values of properties are always displayed on the schematic canvas.

Pre-16.5, different modes of operations were used to capture and modify connectivity or capture and modify properties. In the Hierarchy mode, all the connectivity and property changes made were written directly to the block. In the Occurrence Edit mode, only property changes are allowed and these changes were written in the property file (opf) in the root (top) level design. Post 16.5, there are no modes, but the connectivity and property changes can still be made in a similar manner with a lot more of flexibility.

## Support for Common Windows Commands and Operations

A number of Design Entry HDL features provide support for common Windows commands and operations so that DE-HDL conforms to the general usability standards of Windows-based applications such as Adobe Reader and Microsoft Office applications.

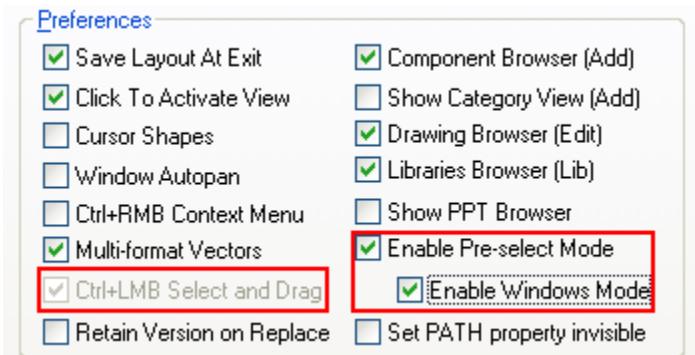
For example, you can use the selection filter and object visibility layers features to quickly select objects to create groups, then work on the objects as a group. You can quickly search an occurrence of text on the same page or across a complete design. The various features are as follows:

- [Windows Mode](#) on page 145
- [Menus in the Windows Mode](#) on page 147
- [Design Entry HDL Options Dialog Box](#) on page 148
- [Support for Keyboard Operations](#) on page 148
- [Bounding Box on Components](#) on page 150
- [Global Navigate Window](#) on page 151
- [Properties Window](#) on page 151
- [Displaying Unconnected Pins on Components](#) on page 152
- [Wire Selection](#) on page 154
- [Anchor Point Stretch](#) on page 154
- [Alignment and Distribution](#) on page 156
- [Selection Filters](#) on page 160
- [Object Visibility Layers](#) on page 161
- [Page Search Toolbar](#) on page 164

### Windows Mode

The Windows mode provides support for common Windows operations in Design Entry HDL, such as cut, copy, paste, delete on schematic objects, and reorganized menus that conform to Windows standards. By default, the Windows mode is disabled. To enable the Windows mode, do the following:

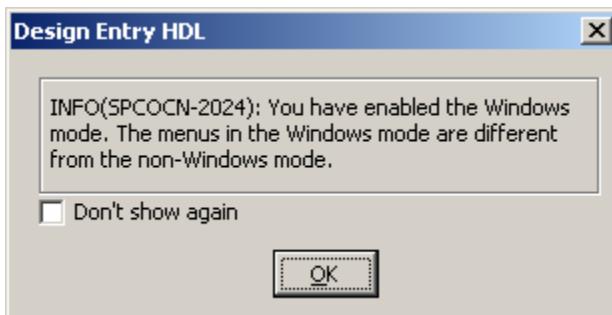
1. Choose *Tools – Options*.
2. On the General page of the *Design Entry HDL Options* dialog, select the *Enable Windows Mode* check box in the *Preferences* section.



**Note:** Note that the *Ctrl+LMB Select* and the *Drag* check box is disabled in the Windows mode implying that you need to hold down the Ctrl or Shift keys along with clicking the left mouse button to make a stroke on the drawing.

3. Click *OK*.

As you switch to the Windows mode, a message pops up about differences in the menus in the Windows mode.



**Note:** This message does not appear if you have customized Design Entry HDL menus, and if the `concepthdl_menu.txt` file is in the `HOME\cdssetup\concept` directory. There is no change in the menus even in the Windows mode and your custom settings are honored. If however, you customize menus now, the customization settings are stored in the `concepthdl_menu_win.txt` file.

Similarly, when you switch to the non-windows mode and the `concepthdl_menu_win.txt` file is in the `HOME` directory, you will continue to see the menus in the Windows mode.

To see the menus in the Windows mode as they were in the original settings (factory

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

settings), click the *Reset* button in the Menus page of the Customize dialog box (*Tools – Customize*). In the Windows mode, this will reset the menus to their original settings stored in the default `concepthdl_menu_win.txt` file available at `<your_install_dir>/share/cdssetup/concept`.

## Menus in the Windows Mode

In the Windows mode, the most prominent change you will note is the difference in how the menus are organized.



In the non-Windows mode on the Windows platform, DE-HDL has 14 menus. In the Windows mode, there are 12 menus. On other platforms, there are 11 menus in the Windows mode as compared to 13 menus in the non-Windows mode.

Some of the main menu names have changed. Some menu commands are accessible from the same locations with little or no change in the sequence, while some others have either moved to a new menu or are now nested inside a new or existing submenu for increased granularity.

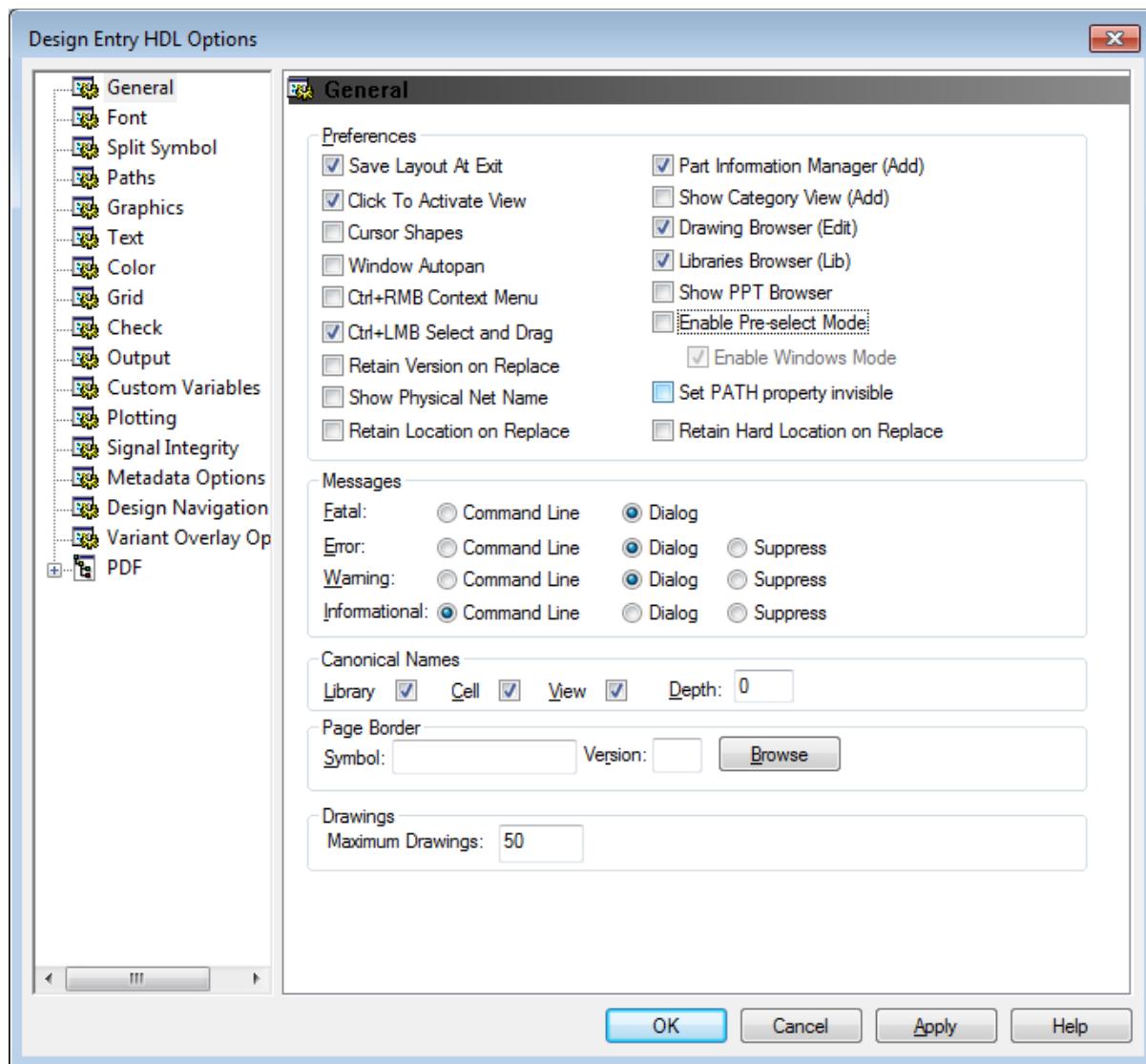


*Important*

If you customize menus in the pre-select mode, changes are not reflected in the post-select mode. Similarly, changes made in the post-select mode are only available in that mode.

## Design Entry HDL Options Dialog Box

The Design Entry HDL Options dialog box has a look and feel similar to the Preferences dialog box of Adobe Reader. Instead of tabbed pages, page names appear in the left panel.



## Support for Keyboard Operations

Another important feature which makes Design Entry HDL similar to a standard Windows application is support for keyboard functions, such as CTRL+C for copy and CTRL+V for

## **Allegro Design Entry HDL User Guide**

### Design Entry HDL Editing Environment

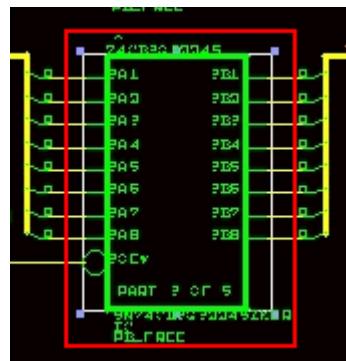
---

paste. You can copy objects from the Windows Clipboard to a schematic through keyboard shortcuts. Other keyboard operations supported are listed in the following table:

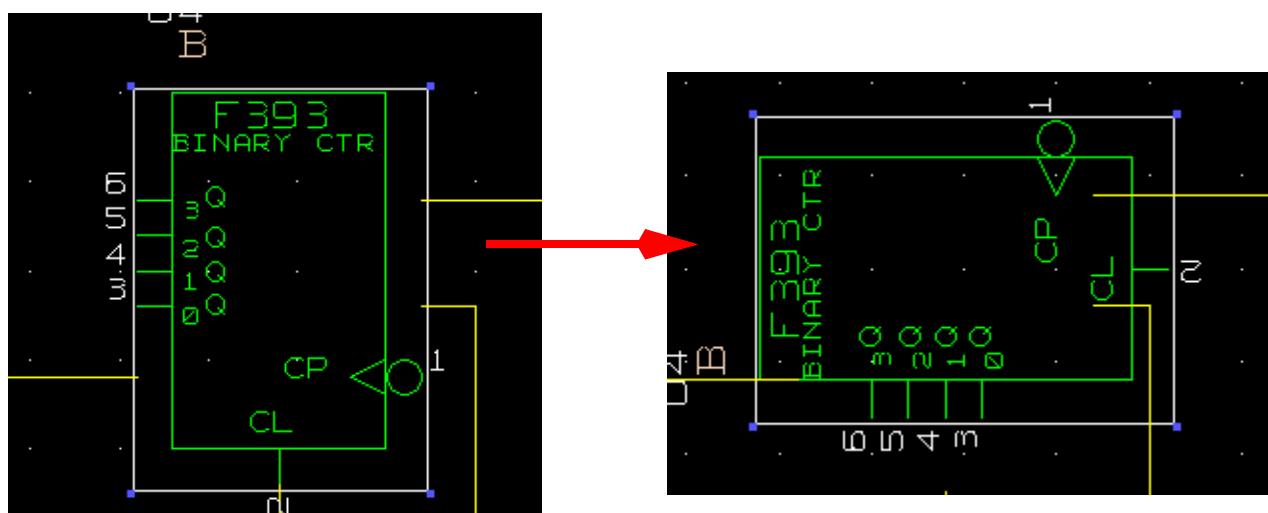
<b>Keyboard Combination</b>	<b>Operation</b>
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + A	Select All
Ctrl + X	Cut
Delete	Delete
Home	Go to the first sheet within a module
End	Go to the last sheet within a module
Shift + Up Arrow	Move component up to the next grid point
Shift + Down Arrow	Move component down to the next grid point
Shift + Left Arrow	Move component left to the next grid point
Shift + Right Arrow	Move component right to the next grid point
Arrow Keys	Move objects in small increments
Page Up	Move one page up
Page Down	Move one page down

## Bounding Box on Components

A bounding box with anchor points or handles around a component appears when you click a component in a schematic in the Windows mode. The bounding box acts as a selection indicator.



You can rotate a component by pressing the *Ctrl+R* key combination. Each time you press this key combination, the component is rotated anti-clockwise.

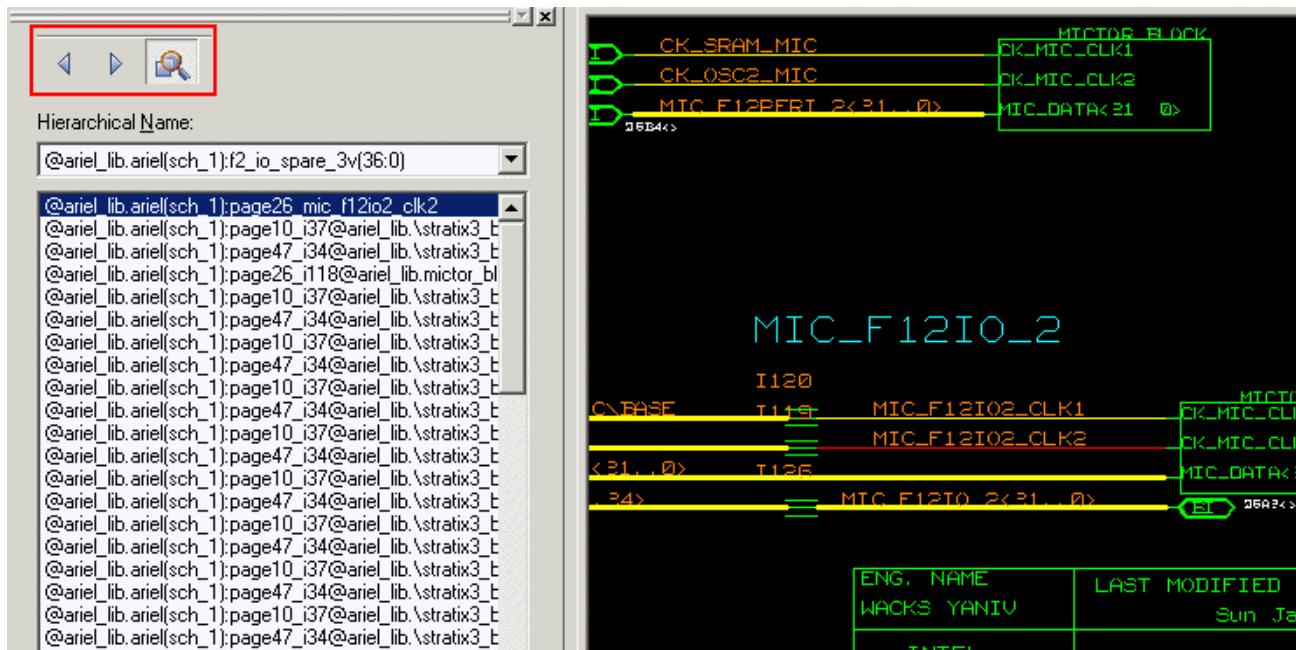


## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

## Global Navigate Window

The Global Navigate window displays three icons: *Next*, *Previous*, and *Zoom to object*. If the *Zoom to object* icon is pressed and you select nets from the list using the *Next* and *Previous* icons, the corresponding region on the schematic is zoomed in.



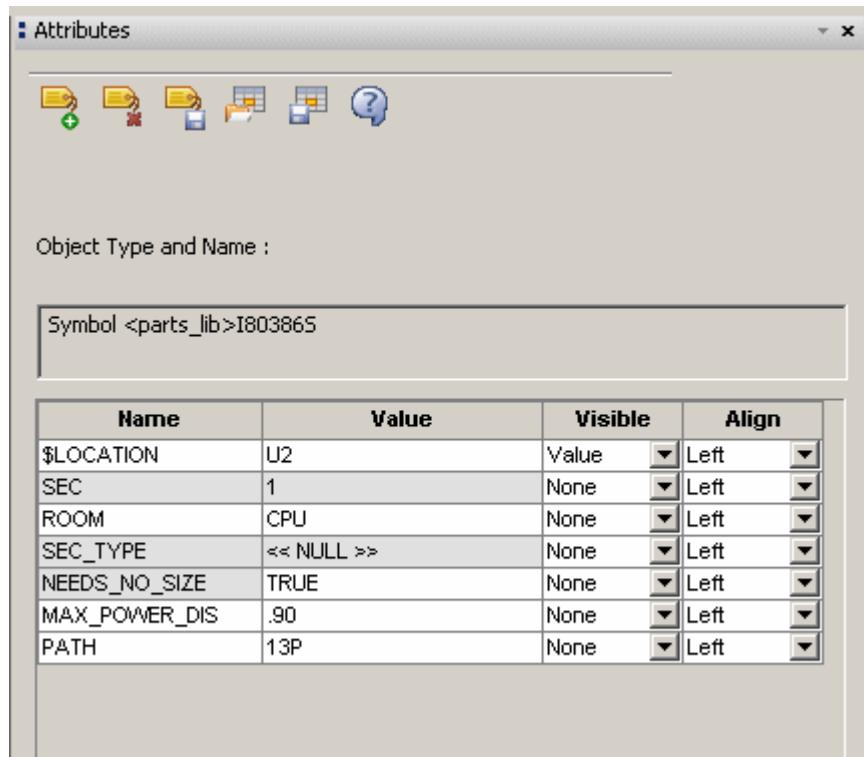
## Properties Window

In the Windows mode, the Attributes form appears as a dockable window. When you select an object on the schematic, the Properties window is populated with the properties

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

information of the component or net. The window has icons to add, delete, and save properties and to load and save a property (.att) file.



## Displaying Unconnected Pins on Components

This option in the *Edit – Component – Unconnected Pins* menu (*Component – Unconnected Pins* menu in the non-Windows mode), acts as a toggle for showing or hiding unconnected pins on components. You can also use the following console command to show unconnected pins:

```
Set SHOW_UNCONNECTED_PIN ON
```

## Allegro Design Entry HDL User Guide

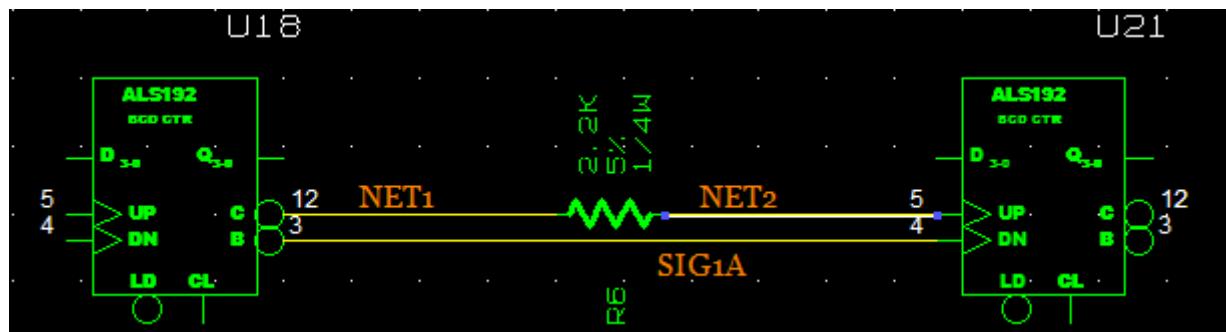
### Design Entry HDL Editing Environment

When this option is on, all unconnected pins on the components of a drawing are marked with pink dots.



### Wire Selection

On selecting a wire, only the wire segment is selected. A bounding box no longer appears around the wire segment. Instead, the wire is highlighted with an anchor point at each end.



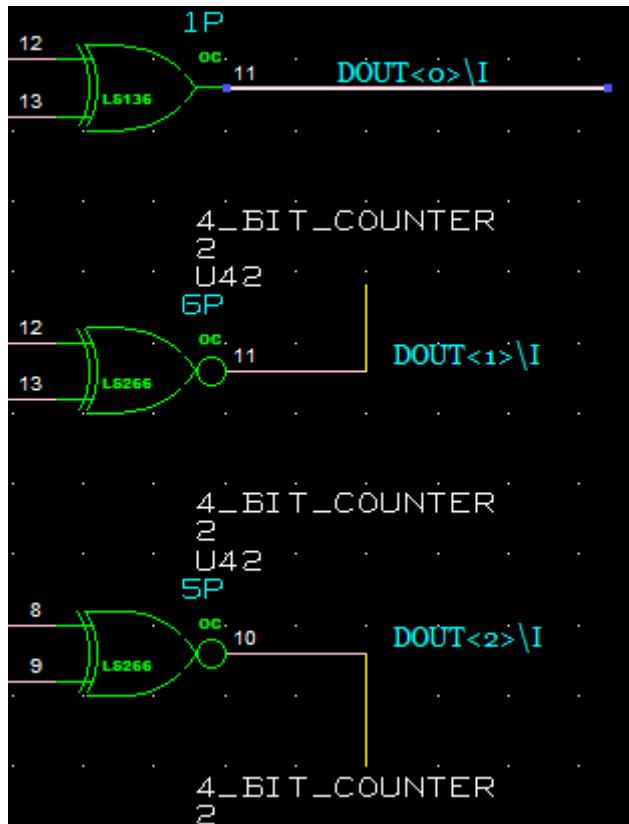
### Anchor Point Stretch

You can stretch a wire in any direction, but only from the free end of the anchor point. When you select a wire and click and drag it from its anchor point, a new wire is drawn from that anchor point. Effectively, the wire command is called when you click the anchor point of a wire

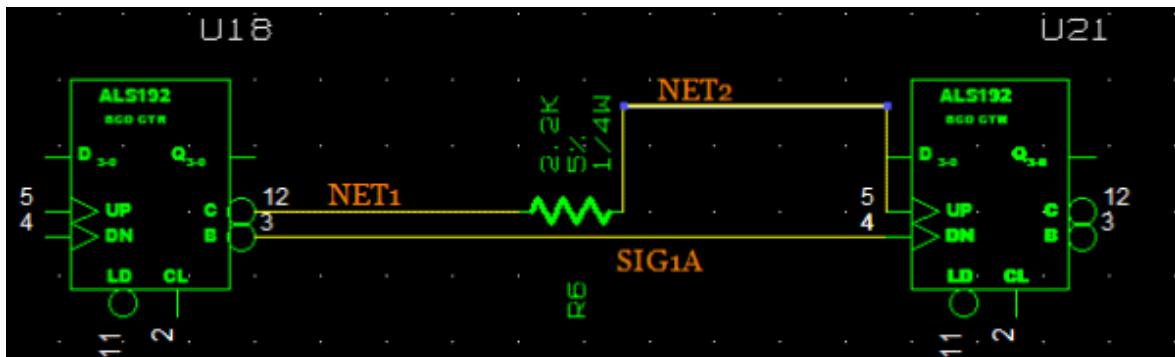
## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

stub and drag it. For example, in the following image, three wires are stretched from the free-end towards the right, top, and bottom directions respectively.



When you select a wire stub to move it, only that stub moves. The other wire stubs or components connected to it do not move. In the following image, the selected net, NET2, is moved towards the top. Note that only the wire segment moves and not the attached components. New wire segments are added to preserve the connectivity.



**Note:** Wire Selection and Anchor Point Stretch are available only in the Windows mode.

## Alignment and Distribution

Schematic creation requires that components, wires and text be placed neatly and effectively. Aligning and distributing objects on a schematic can be a time-consuming task if it is done manually by inserting or moving objects.

You can use the align or distribute functions to speed up the process of designing. Only a selected set of objects can be aligned or distributed. These include the following:

- Components
- Wire segments
  - Vertical
  - Horizontal
- Text
  - Properties
  - Notes

You can align and distribute objects either vertically or horizontally. The Align function aligns a selected set of objects with respect to a common axis. The Distribute function equally spaces a group of objects according to the type of distribution—horizontal or vertical.



Align and distribute operations work only on a single type of object at a time. For example, you can align or distribute either text or wire segment at a time.

**Note:** During the align and distribute operations, connectivity is always maintained. If the components are already connected, the wires are stretched to maintain connectivity. There is no loss in connectivity, shorting, or overlapping of objects after the align or distribute operations. Design Entry HDL checks whether the origins overlap only after the distribute operation is completed.

### Aligning and Distributing Schematic Objects

To align or distribute a set of objects, you need to select similar objects and then run the *Align* or *Distribute* commands. Use one of the following ways to run the commands:

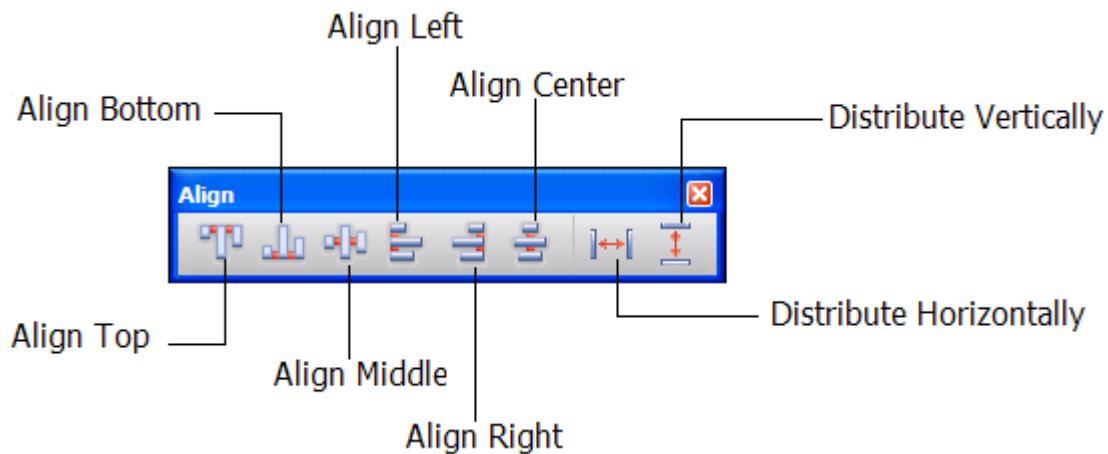
- Main menu: Choose the *Edit – Align or Distribute* menu and then select any of the following menu commands:
  - Align Left

# Allegro Design Entry HDL User Guide

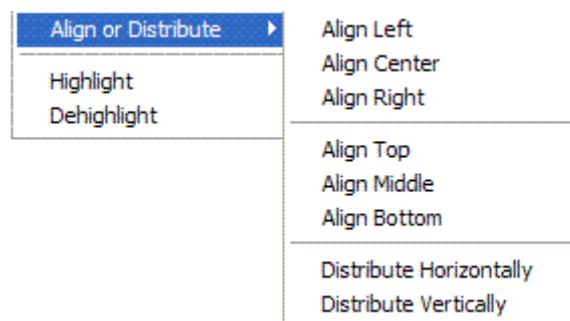
## Design Entry HDL Editing Environment

- Align Center
- Align Right
- Align Top
- Align Middle
- Align Bottom
- Distribute Horizontally
- Distribute Vertically

### ■ Toolbar:



### ■ Pop-up menu (right-click menu)



### Aligning Objects

When you use the Alignment functionality, objects are aligned with respect to a common axis. First select the objects to be aligned to activate the icons on the toolbar or the Align menu options. Depending on the alignment option you choose, a common axis is calculated. You can align schematic objects in the following manner:

- Left
- Center
- Right
- Top
- Middle
- Bottom



#### Caution

***When you align objects, they may stack on top of each other. Therefore, before aligning objects, you must ensure that the objects are positioned relative to one another in the desired manner.***

Alignment Option	Description
Left	The objects move with reference to the left edge of the left-most object, such that the left edge of each object is aligned, that is, has the same x coordinate value.
Center	Aligns objects vertically through the centers of the objects. The objects are moved in such a way that they lie on the same x coordinate as that of the component placed at the center.
Right	The objects move with reference to the right edge of the right-most object, such that the right edge of each object is aligned, that is, has the same x coordinate value.
Top	The objects move with reference to the top edge of the top-most object, such that the top edge of each object is aligned, that is, has the same y coordinate value.
Middle	Aligns objects horizontally through the middle of the objects.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

Bottom

The objects move with reference to the bottom edge of the bottom-most object, such that the bottom edge of each object is aligned, that is, has the same y coordinate value.

### ***Distributing Objects***

The Distribution functionality distributes objects with an equal amount of space between them. First select the objects to be distributed to activate the icons on the toolbar or the Distribute menu options. The distribution command moves the objects so that the space between all the objects is the same. You can distribute schematic objects vertically or horizontally.

#### **Distribution Option**

Distribute Horizontally

#### **Description**

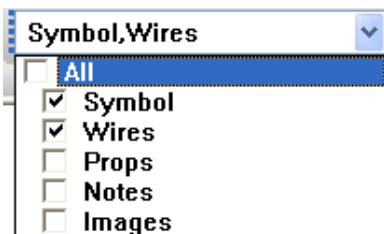
The left boundary of the left-most object and the right boundary of the right-most object are used to calculate the mean distance. This mean is used to displace each object such that the horizontal distance between each object is equal.

Distribute Vertically

The top boundary of the top-most object and the bottom boundary of the bottom-most object are used to calculate the mean distance. This mean is used to displace each object such that the vertical distance between each object is equal.

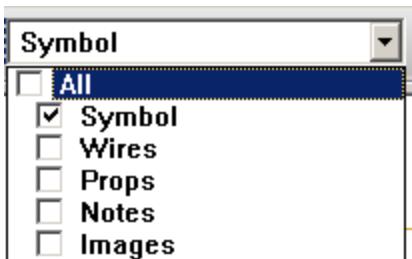
## Selection Filters

The selection filters, which are available as a toolbar, can be docked to any side of the DE-HDL window and can be used to select one or more objects on the schematic. After selecting the objects, you can perform various operations on these objects such as aligning, distributing, or moving them to a specific area on the page. The selected objects can also be grouped. You can treat the objects as a group. The following image shows the selection filter with symbols and wires selected.

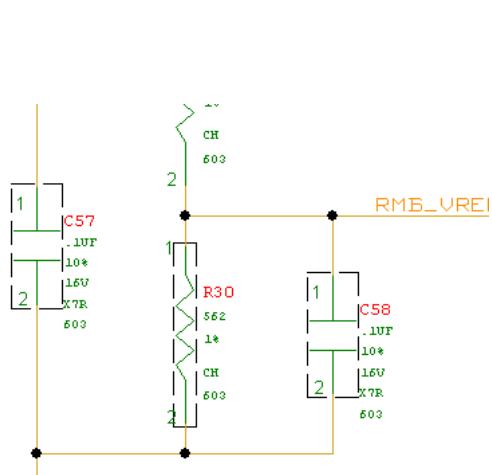
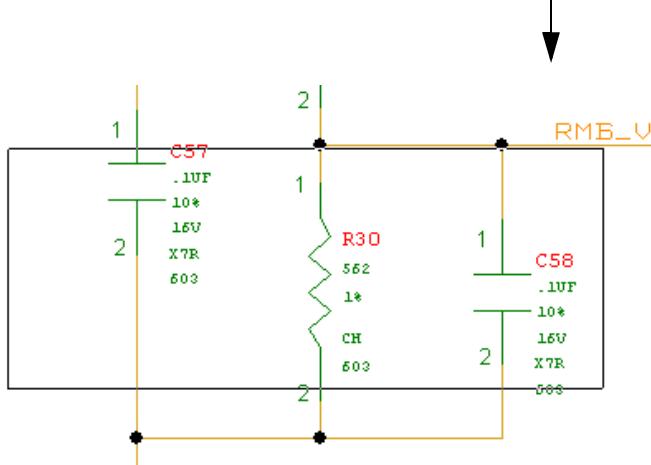


# Allegro Design Entry HDL User Guide

## Design Entry HDL Editing Environment



After you select an object type in the selection filter such as Symbols, drawing a selection box around the schematic objects selects only the objects of type which have been set in the selection filter; in this case, it is Symbols.



## Object Visibility Layers

The different objects in DE-HDL are available on different layers. Using a toolbar, you can control the visibility of each of the object layers.

# Allegro Design Entry HDL User Guide

## Design Entry HDL Editing Environment

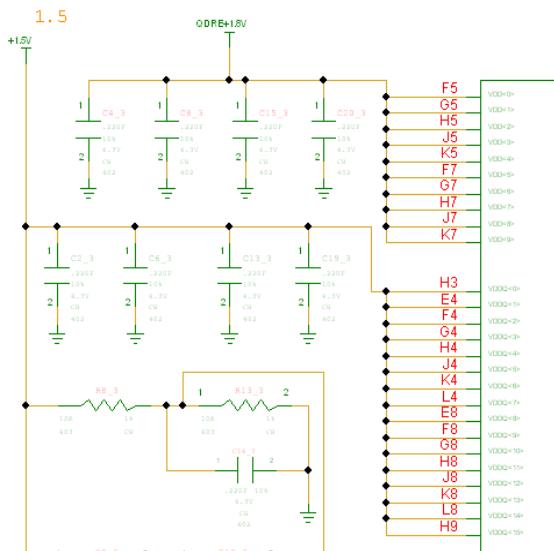
The different object types in DE-HDL are as follows:

- Components / Symbols
- Nets
- Properties
- Notes
- Images



The visibility of each of the object layers can be controlled by pressing or releasing the toolbar buttons. Pressing an object layer button makes the objects in that layer visible; releasing the button hides the objects of that layer.

On a schematic sheet, all the objects are visible by default.

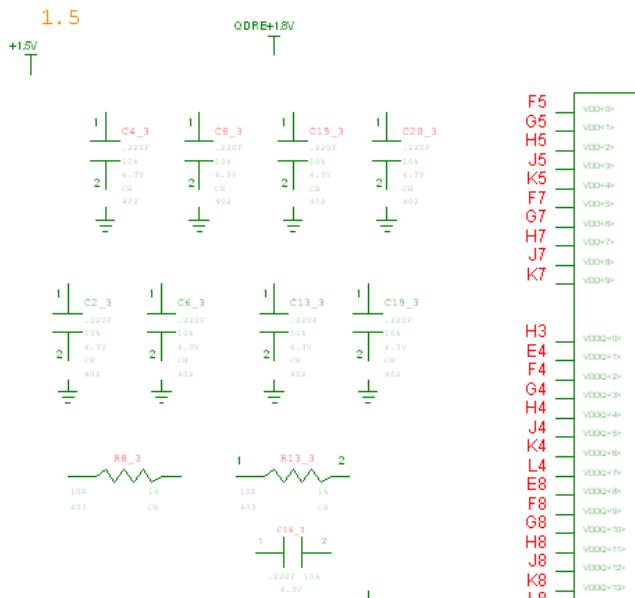


# Allegro Design Entry HDL User Guide

## Design Entry HDL Editing Environment



On clicking any of the icons of the Layers toolbar, for example, nets, all the net objects become invisible on the schematic canvas as illustrated.



You can use the *Layers* toolbar to select objects of the same type to perform common operations. In case of large schematics, this feature can also be used to hide properties to make the connectivity of the design more clearly visible.

To select objects, do the following:

1. Draw a selection box.  
All the objects are selected.
2. Click all the Visibility layer icons one by one and see the objects of the particular layer disappear from the schematic canvas.
3. Click the *Symbols* icon and make all the symbols visible.
4. Choose *Group – Create – By Rectangle*.
5. Draw a selection box.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

The symbols in the selection box become part of the group.

6. Choose *Group – Add Property*.

7. Specify the property value pair as FOO = BAR.

The property is added to all the symbols that are members of the group.

## Page Search Toolbar

Using the search toolbar, you can search for text on the current page. The text can be symbol text, a net name, property or part of a note. After you type the search string, press Enter for the results.

**Note:** Along with the results, the Search Results window also displays the number of results, and the time taken to search.



The search can also be limited to one or more types of objects by selecting the object category in the Filter options in the Find dialog. Based on the selection, Design Entry HDL only searches for those objects. The search results are displayed in a docking window which can be docked on any side of the DE-HDL window.

The Search Results window displays the following:

- Design Name
- Physical Page Number
- Result Object Type
- Object Details

Design Name	Page	Type	Object Name	Prop Name	Prop Value
QDR_CHANNEL	page1 (12)	Component	RES (8)		
QDR_CHANNEL	page1 (12)	Component	RES (8)		

Showing: 8 of 8      Search took: 0.094000 secs

Command & Result    Command Console    Search Result

The different rows display the different objects that contain the text that was searched for. If you double-click on any of the rows, Design Entry HDL navigates to the object and zooms in to display the object.

## **Allegro Design Entry HDL User Guide**

### Design Entry HDL Editing Environment

---

You can select one or more of the result rows (using Ctrl or Shift+select) and create a group of the selected objects. Once the group has been created, you can run group commands on the group. You can also choose to save the search results as a text file.

## Searching Design Objects

In Design Entry HDL, you can search a design for components, nets, properties, or a combination of any of these. Alternatively, you can search for all the objects in the design.

To access the FInd dialog, use *Edit — Search — Option*.

To search for components, nets, properties, or a combination of these in a design, you must first select the objects. If you only select the Nets option when searching, Design Entry HDL displays results for all bus instances in the design, including tapped bus bits and sections.

The option to search for all the objects in a design is automatically enabled in the Find dialog when, in addition to components, nets, or properties, other filters such as notes, pins, images, and plumbing bodies are selected.

The following table provides a quick look at the differences between both kinds of search:

Search for	Components, nets, properties, or a combination	All objects in a design
Buses	Works with a search string as "busname". For example: my_bus) and with search strings such as "busname< .. >"	Works only with the search string as "busname< .. >". For example, my_bus<3..0>)
Global Signals	Works with search strings such as "net name" and "net name \g"	Works only with search strings such as "net name\g"
Interface Nets	Works with a search string such as "net name" and "net name \l" format	Works only with the "net name\l" format.

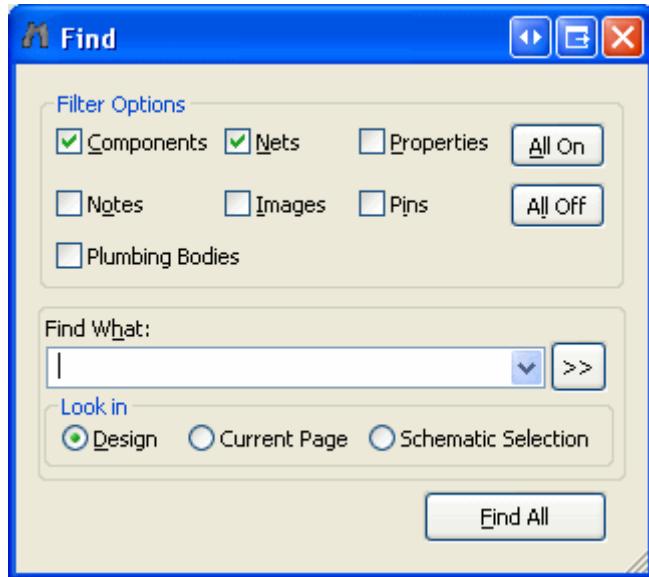
By default, the Find dialog is set to search the entire design for nets and components. In the dialog, you can limit the scope of the search. On a schematic page, searches are performed

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

based on your selected objects, or, based on selected filter options such as components, nets, properties, notes, images, pins, and plumbing bodies.

You can also specify where to search: the current page, design, or the schematic. You can search by using wildcards, property names, and values.



The following table lists the elements in the Find dialog with descriptions:

Elements	Description
Filter Options	<p>The check boxes in this panel allow you select the objects that you want to search for based on the specified search query string.</p> <p>Use the <i>Look in</i> radio buttons to specify where Design Entry HDL should search:</p> <ul style="list-style-type: none"><li>■ Current page</li><li>■ Design</li><li>■ Schematic Selection</li></ul> <p><b>Note:</b> If you select <i>Schematic Selection</i>, the <i>Plumbing Bodies</i> filter option is disabled.</p>

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

---

#### Find What

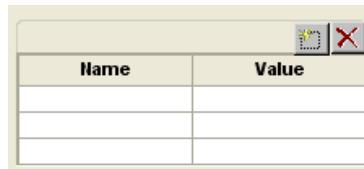
Use this to enter the search query string. The Find What combo box supports

- Directly typing into the search query string
- Last 20 search query strings in the drop-down list in the current session
- Copy and paste

#### [>>] Button

You can click this button to open the Property Grid window where you can type the property names and values. This button is enabled if the *Components*, *Nets*, or *Pins* filter options are selected.

The name and value that you type in the window are automatically updated in the Find What combo box.



For example, property Name: Foo and property value: Bar will be displayed as `%prop('Foo', 'Bar')` in the combo box.

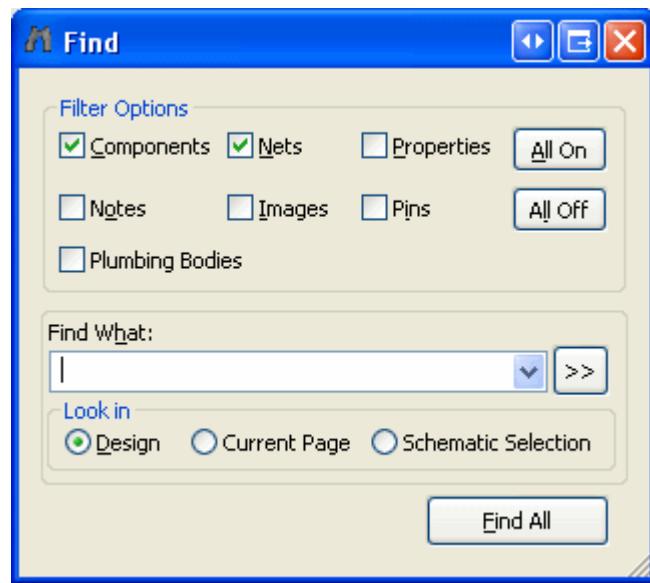
**Note:** Wildcard search in the property grid is only supported for property values and not for property names.

## Allegro Design Entry HDL User Guide

### Design Entry HDL Editing Environment

Look in

You can specify the scope of the search in this box. you can choose to search the entire design, current page or schematic.



The search results are displayed in a dockable Search Results window. The dockable window has a grid control to display the search results.

A screenshot of the Allegro Search Results window. The window has a title bar 'Search Result' and contains a grid table with columns: 'Design Name', 'Page', 'Type', 'Object Name', and 'Pro'. The grid shows three rows of results for 'DESEXAMPLE':

Design Name	Page	Type	Object Name	Pro
DESEXAMPLE	2	Component	(i9)	
DESEXAMPLE	2	Component	(i10)	
DESEXAMPLE	3	Component	(i3)	

At the bottom of the window, there is a status message 'Showing: 3 of 3' and 'Search took: 0.000000 secs'. A toolbar at the bottom includes 'Command Console' and 'Search Result' buttons.

## **Allegro Design Entry HDL User Guide**

### Design Entry HDL Editing Environment

---

The Search Results window displays the following in columns:

- Design Name
- Physical Page Number
- Result Object Type
- Object Details (includes Object Name, Property Name, and Value)

**Note:** Only sch\_1 properties are returned if you search the design. If you search the current page, only Schvar\_1 properties are searched for.

---

## **Creating a Schematic**

---

To create a schematic in Design Entry HDL, do the following:

- Create a project.
- Start Design Entry HDL.
- Create a design page.
- Add a page border.
- Add parts using Part Information Manager.
- Connect parts.
- Name signals.
- Add properties.
- Add ports.
- Save the design.
- Work with designs.

## **Creating a Project**

You use Project Manager to create and set up a project. Project Manager creates a project file (*<project name>.cpm*) that stores paths to local libraries (also known as design libraries), the top-level design name (also known as the root design), part tables (files that map logical components to corresponding physical components), tool settings (defaults), global settings, view directory names, and other related settings for a design project.

A design project consists of the following:

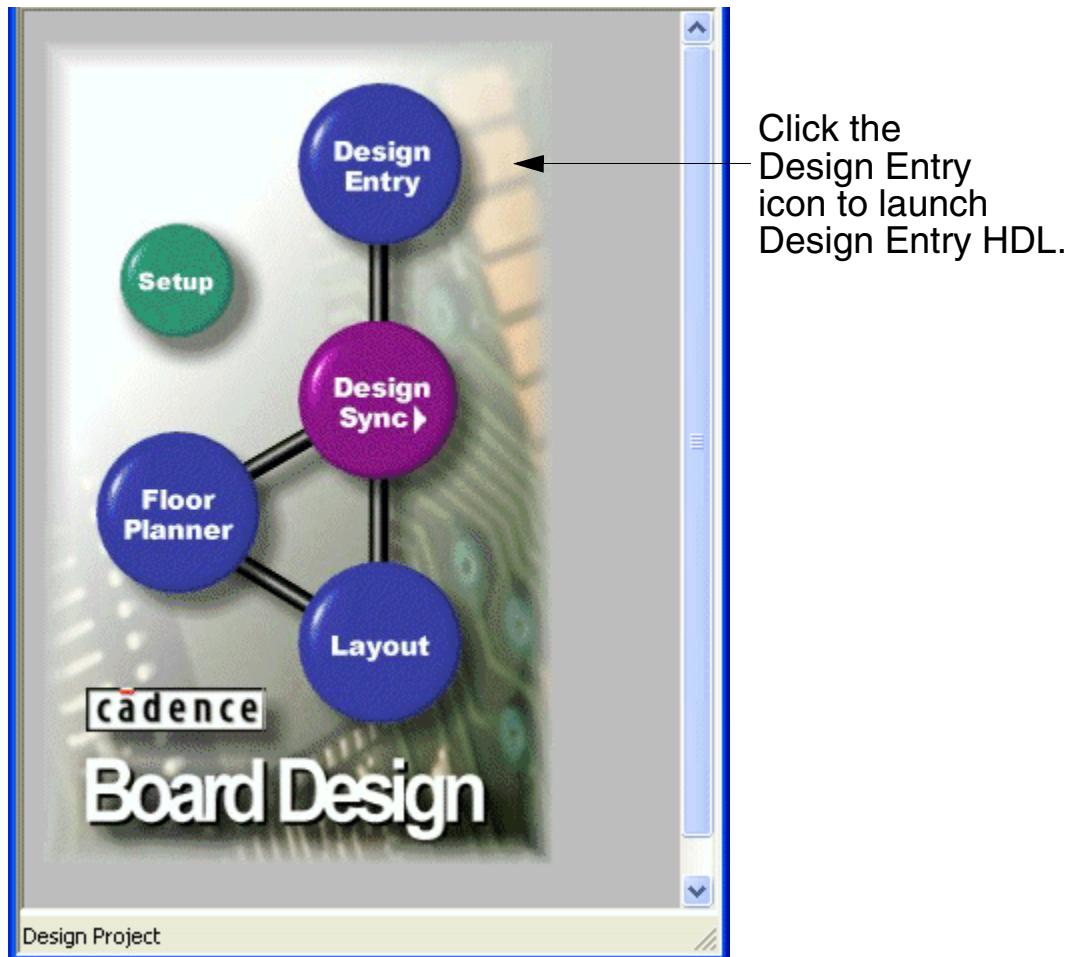
- Reference libraries
- Local libraries

- cds.lib file
- Project file (.cpm file)

For more information on creating a project using Project Manager, see [“Creating a Project”](#) on page 62.

## Starting Design Entry HDL

After you create a design project in Project Manager, the flow area of Project Manager displays the Cadence Board Design flow. In the board design flow, click the *Design Entry* icon.



## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

#### About Drawing Names

The drawing name identifies your design. Drawing names can have several fields separated by periods:

```
[<library>] cell [.view] [.version] [.page]
```

The cell name is the only required part of the drawing name.

Library Name	Identifies the library containing the component. Angle brackets surround a library name.
Cell Name	The cell name describes the intended function of the drawing.
View Name	Describes the representation of a schematic drawing. For a symbol, the view name must be SYM.  Standard views in Design Entry HDL are:  <b>Schematic drawings</b> -These can be any name (SCH is the default) and contain symbols that represent library components or hierarchical cells.  <b>Symbol drawings</b> -These must be defined as SYM. This is a graphical representation of a library component. It defines the shape, pins, and general properties of the component. When making a hierarchical design, you make a symbol (SYM) drawing to represent an entire schematic (SCH) drawing.
Version	Numbers the schematic or symbol view representation. If you don't specify a version number, Design Entry HDL assumes version 1.  <b>Note:</b> Do not use the version field of the drawing name to store revisions of a drawing.
Page	Used for schematics only. The page field identifies the pages of a multi-page drawing.

#### ***General Rules to Ensure Compatibility Between Schematics and Other Design Tools***

- You should not change the drawing view type when you save the drawing. For example, if you are editing shifter.sch, you cannot save it as shifter.sym. You must use the *File – Save As* menu command to change the name and type of a drawing and then save it.

- If you add a component into a symbol drawing, either for comparison purposes or as part of a new symbol, Design Entry HDL will not let you save the drawing. You must first smash the copied component or group into individual wires, arcs, and notes.

## Creating a Design Page

***To create a single page,***

1. Choose *File – New*.
2. Add a border around the drawing.
3. Choose *File – Save As* to save and name the new drawing.

Use the other system design tools to compile, simulate, and package the design.

To create a multiple-page

1. Choose *File – New*.
2. Place a border in the main design area.
3. Choose *File – Save As* to save and name the new drawing, and specify page 2 in the *Page* field of the *View Save As* dialog box.

**Note:** Alternatively, while viewing page 1, you can choose *File – Edit Page – Next*, or select the *Next Page* icon in the Standard toolbar.

Use the other system design tools to compile, simulate, and package the design.

## Adding Page Borders

Page borders provide a convenient way to document information such as the date of creation, the name of the engineer, the page number, the name of the design, and company logo on the schematic. Page borders also serve as a border to demarcate the borders for a schematic.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

The Standard library supplied by Cadence contains the following symbols that can be used as page borders

Page Name	Specifications
A Size Page	8.5 x 11 inch border
B Size Page	11 x 17 inch border
Cadence A Size Page	8.5 x 11 inch border with the Cadence Design Systems logo
Cadence B Size Page	11 x 17 inch border with the Cadence Design Systems logo
C Size Page	17 x 22 inch border
D Size Page	22 x 34 inch border
E Size Page	34 x 44 inch border
F Size Page	44 x 68 inch border

---

### Placing a Border in the Design Area

Design Entry HDL helps you to automatically add a page border when you create a new page. For more information, see [Setting Automatic Page Borders](#).

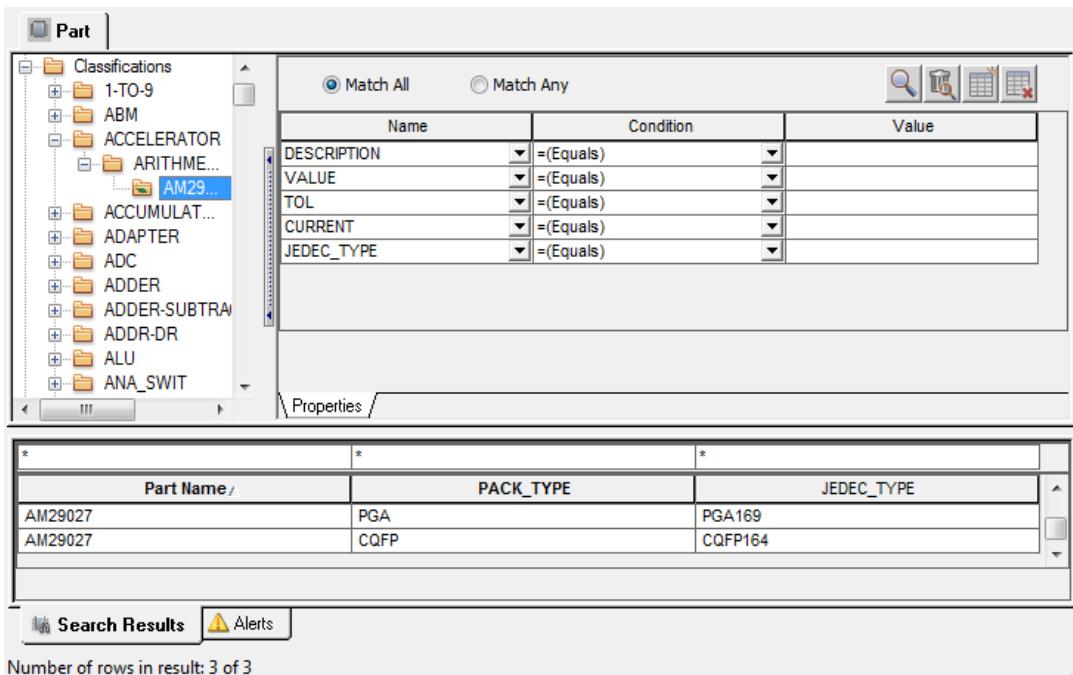
#### ***To add a page border***

1. Choose *Component – Add*.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

Part Information Manager appears.



2. Click the *Browse Libraries* folder.
3. In the *Library* box, select the *standard* library.
4. In the scroll area, select a border for your drawing.
5. Click *OK* in the message box that appears.  
The border is attached to the cursor.
6. Place the border in the main design area.
7. Add note text as appropriate in boxes in the lower-right corner of the border.

## Managing Grid Settings

A grid helps you place objects and ensure wire alignment and pin connections. Note that grids are saved with individual designs, so grid settings can differ between designs. You can hide or display grids using *View – Grids*.

Using the Design Entry HDL Options dialog (*Tools — Options*), you can change the display options for the grids. You can view the grid as either dashed lines or dots, specify the distance at which you would like to view the grid lines, and define the grid type.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

The grid type can be:

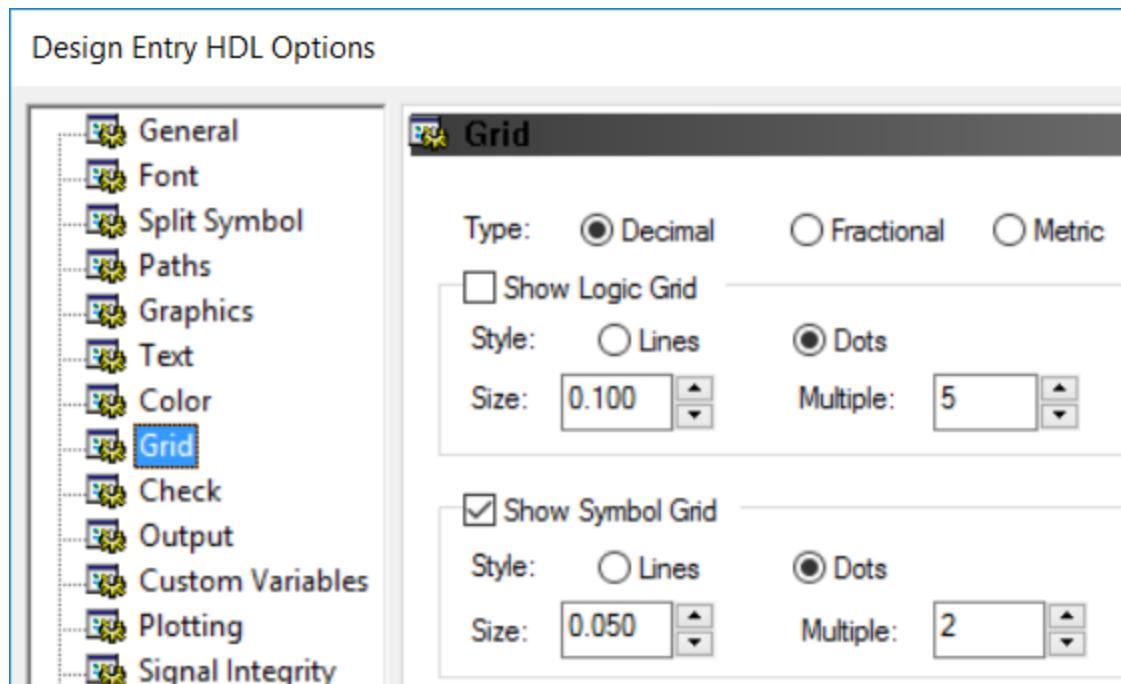
- Decimal: Bases drawings on the decimal system (500 units per physical inch).
- Fractional: Bases drawings on 400 units per inch. Components will appear 25 percent larger.
- Metric: Bases drawings on the metric system (20 units per millimeter; 508 units per inch).

**Note:** For your schematics, you should use the same grid type as used when creating symbols for components instantiated in the schematics. If you use different grid types, the symbols can be off-grid and cause connectivity problems.

You can also specify a multiple for the grid. This displays every nth grid line to define where objects can be placed so that pins do not fall off-grid. This ensures the correct connectivity of wires and symbols.

DE-HDL provides support for two kinds of grids:

- Logic grid - used for placement of electrical objects on the canvas. Electrical objects include components, buses, ports, connecting wires — the objects that define the electrical connectivity of a design. Setting up a grid ensures that all the objects are on the grid.
- Symbol grid - appears on the canvas when you open the symbol view of a component instance.

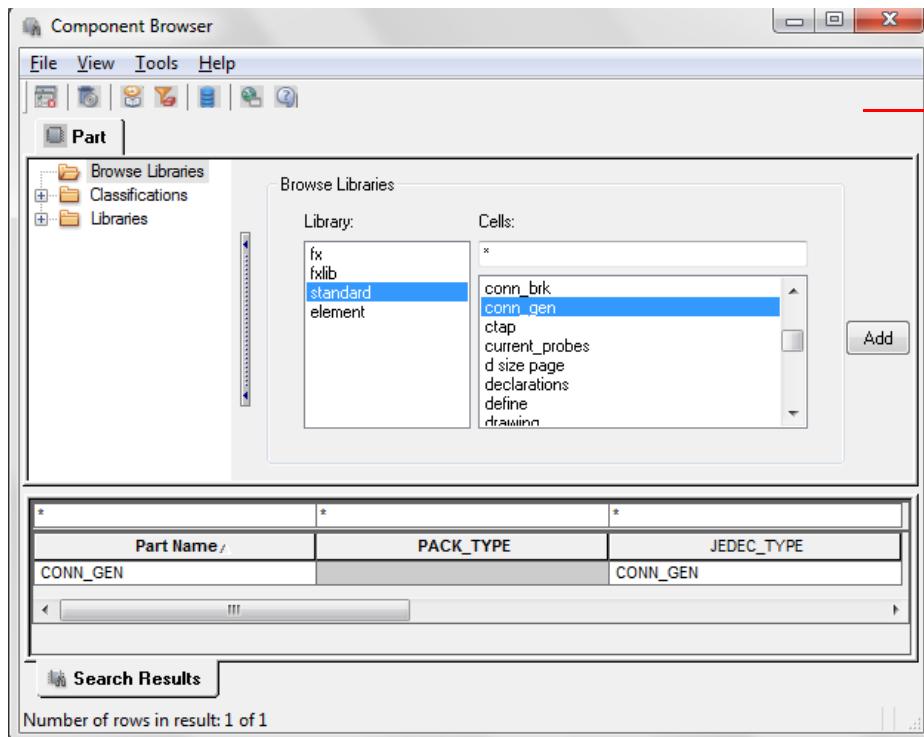


## Using Part Information Manager

You can access Part Information Manager in two modes — the standard mode and the Allegro EDM mode. In DE-HDL documentation, the standard mode is also referred to as the offline mode, and the Allegro EDM mode as the online mode.

# Allegro Design Entry HDL User Guide

## Creating a Schematic



Part Information Manager - Standard (offline) mode

This screenshot shows the Allegro Component Browser window in Allegro EDM (Online) Mode. The interface is similar to the offline version but includes a search bar at the top right labeled '<Enter Text To Search>' with a magnifying glass icon. The left sidebar shows a tree view of categories: 'Browse Libraries', 'Classifications', 'Block', 'Electrical' (which is selected), 'Mechanical', and 'Mechanical'. Under 'Electrical', there are three sub-folders: '\_UNCLASSIFIED\_1', '\_UNCLASSIFIED\_2', and '\_UNCLASSIFIED\_3'. The main area features a query builder with four tabs: 'Relations', 'Attributes', 'Properties', and 'Relations' (which is currently selected). Below the query builder is a table with columns: PPL, Model Name (Footprint), Part Number, TOLERANCE, MATERIAL, VOLTAGE, VALUE, and SIGNAL\_MODE. Four rows of data are listed: 628955-001, 628955-004, 20%, OSCON, 6.3V, 10MF, capacitor330uF; CPRP20\_40\_CPRP20\_50, 628955-001, 20%, OSCON, 6.3V, 100000UF, capacitor330uF; SMC2626, 201828-020, 20%, ALUM, 6.3V, 1000000NF, capacitor100uF; and SMC\_10MM, 201828-050, 20%, EMPTY, 6.3V, 1500UF, DEFAULT\_CAPACITOR\_1.5e+. At the bottom of the table are 'Search Results' and 'Shopping Cart' buttons. The status bar at the bottom displays 'Number of rows in result: 4 of 4' and 'Connected to Database'.

## Allegro EDM and Standard Mode

- In the Allegro EDM mode, Part Information Manager is connected to the Allegro EDM component database and library data is accessed from the Allegro EDM component database. This allows you access to a larger, and accessible-from-anywhere database, of components. In the standard mode, data is accessed from libraries specified in the `cds.lib` file.
- In the standard (offline) mode, you can search for a component using the ECAD properties defined in the PTF. However, in the Allegro EDM mode, you can do a further search for components using non-ECAD properties that your librarian may have defined for parts in the Allegro EDM part database. You can also search for parts by specifying a life cycle value that your librarian may have associated with each part or by specifying the PPL to which the part belongs.

You can also search for a part using associated part names, associated footprints, and associated mechanical parts. Unlike in the standard mode, in the Allegro EDM mode, you can additionally search for properties or attributes of a part simply by typing in a free text search window.

- In the Allegro EDM mode, the following is also possible:
  - A library administrator can associate a list of parts to a preferred part list (PPL). As a designer, you can then define the PPLs to be used at the project level.
  - You can also enable a strict mode in Allegro EDM Part Information Manager that allows users to add parts only from selected PPLs.
  - The librarian can associate a life cycle status with a part, such as *nearing EOL, RoHS non-compliant, in short supply*. This status is visible in the Allegro EDM mode and helps you decide whether or not to add a part to a design. In addition, when you try and add such parts to your design, a warning may be displayed.
  - One electrical part can be associated with a number of Manufacture Part Numbers (MPNs). You can view all the related MPNs in the part details window.
  - You can use the Shopping Cart tab to find parts that are selected or added to a design.
  - The shopping list can be used to import parts from other designs or from a shopping cart created by another designer.
  - In the Allegro EDM mode, you can switch between the PTF and metadata view to view search results. The PTF view displays the PTFs of a component. In the metadata view, the attributes, properties, and relations of a component are displayed.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

- Deleted parts used in designs are indicated with a red, cross icon in the Shopping Cart.

## Accessing the Allegro EDM Component Database from DE-HDL

To access the Allegro EDM component database through Part Information Manager launched from DE-HDL, in the START\_COMPBROWSER section of your project .cpm file, specify the following two directives:

- Online\_Mode 'TRUE'
- server\_url <server\_URL>

where server\_url points to the Allegro EDM database. For example, `http://srv-testsrv01:1111`

To access symbols, the `cds.lib` file in your design project should point to the Allegro EDM server libraries, and the PPT directive in the Global section of your .cpm file should point to the server PTF to access parts.

Local cells/blocks other than those in the design library are not available in the Allegro EDM mode. To access cells and blocks from your local libraries, you can switch to the offline mode by selecting File—Switch to Offline or by clicking the Switch to Offline button (  ) in Allegro EDM Part Information Manager.

For more information about Part Information Manager, refer to *Part Information Manager User Guide*.

## Adding a Part

To add a part to a schematic, do the following:

1. In Allegro Design Entry HDL window, choose *Component – Add*.

Part Information Manager appears.

2. Search for a part.

For information on how to search for parts, refer to *Part Information Manager User Guide*.

3. Click a part to choose in the *Search Results* pane.

The <Part Name> tab appears with the part information.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

4. To add a part, click *Add* in the *Search/Details* pane. Alternatively, right-click the part table row for the part, and choose *Add to Design* from the pop-up menu. You can also double-click the part table row to add the selected part to the design.
5. Click on the schematic where you want to place the part.

**Note:** All the parts you add, for the first time, are available to you in the QuickPick toolbar, subsequently.

## Replacing a Part

To replace a part:

1. In Allegro Design Entry HDL window, choose *Component – Replace* to display Part Information Manager.
2. Search for a part. For information on how to search for parts, see *Part Information Manager User Guide*.
3. Click a part in the *Search Results* pane, and click *Replace* in the *Search/Details* pane. Alternatively, right-click the part table row for the part, and choose *Replace* from the pop-up menu.
4. Click on the component on the schematic to replace it.

If you are in the pre-select mode in Allegro Design Entry HDL, you can replace multiple components by doing the following:

1. Use Ctrl+click or SHIFT+click to select multiple components.
2. Choose *Component – Replace* to display Part Information Manager. Alternatively, choose *Replace* from the RMB pop-up menu.
3. Select the component that should replace all the components.

## Modifying a Part

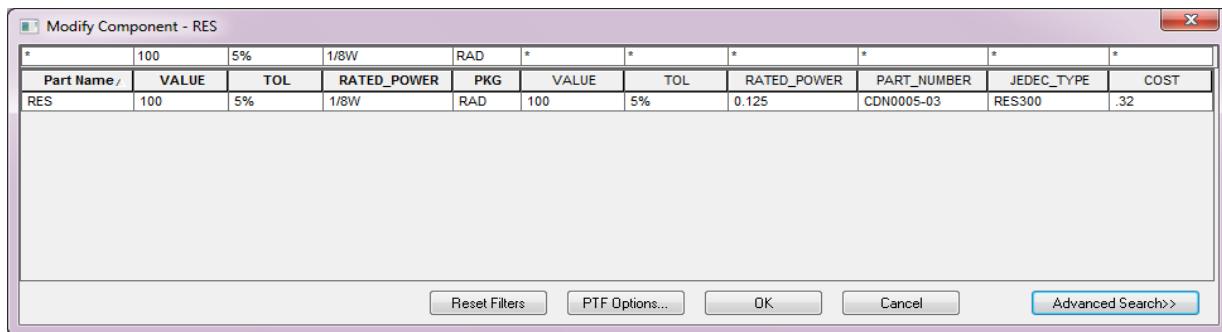
To modify a part:

1. In Allegro Design Entry HDL window, choose *Component – Modify*.
2. On the schematic, click on the component that you want to modify.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

The Modify Component dialog appears. The part attributes already annotated in the component are, by default, filtered, and displayed in the header fields. To modify the attribute values, click in the relevant header field and select a new value.



Click *Filters* if you want to display all the part names associated with the selected component.

3. Click *OK* to apply the changes in the selected component on the schematic.

## Adding Data Tips for Components

Companies often want to provide part-specific recommendations to designers for guidance purposes. You can now add these recommendations and guidelines as data tips. Designers will see these data tips in the schematic after adding components to a design.

Data tips need to be defined in a text file with the name `datatips.txt`. The data tips file can contain text, hyperlinks as well as Japanese characters.

A property is used to associate the part on the schematic canvas with the entry in the data tips file. The value of the property on the part is matched with the entry in the data tips file and the corresponding data tips are displayed. The name of the property to be used to associate the part needs to be specified as a `.cpm` directive, set in the `START_CONCEPTHDL` section.

```
DATATIPS_PROP_NAME '<Part_Property_Name>'
```

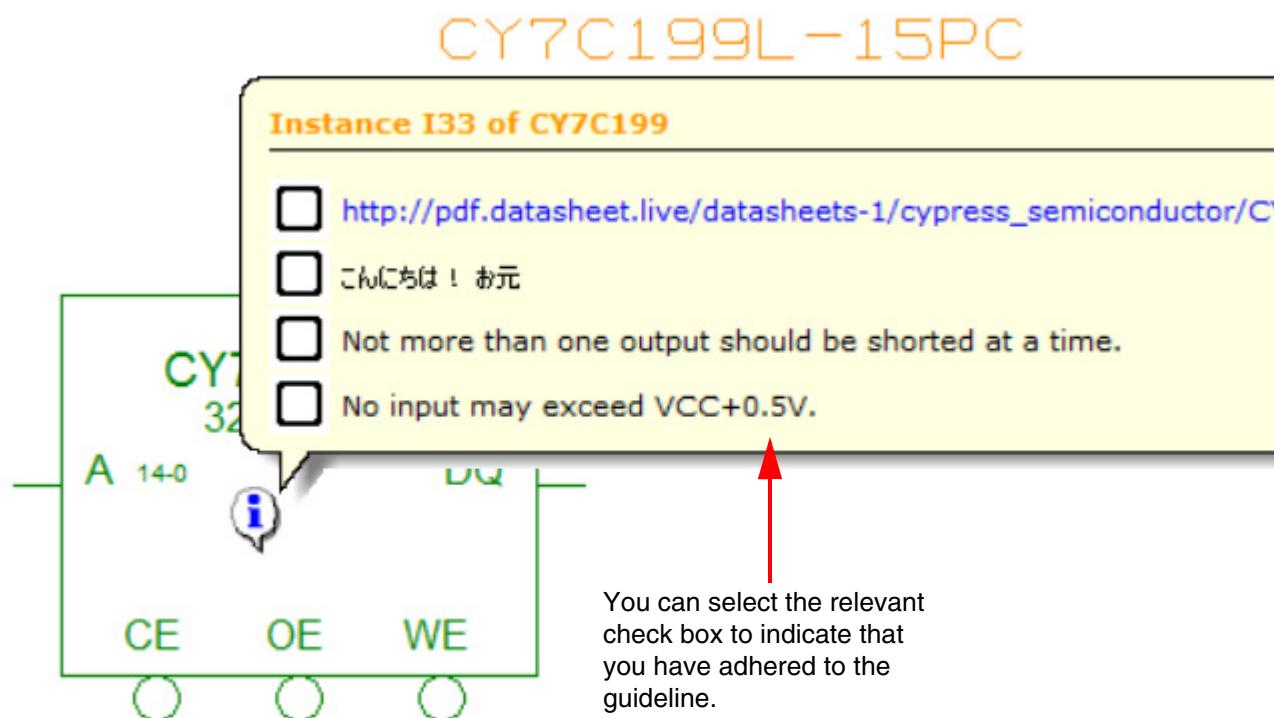
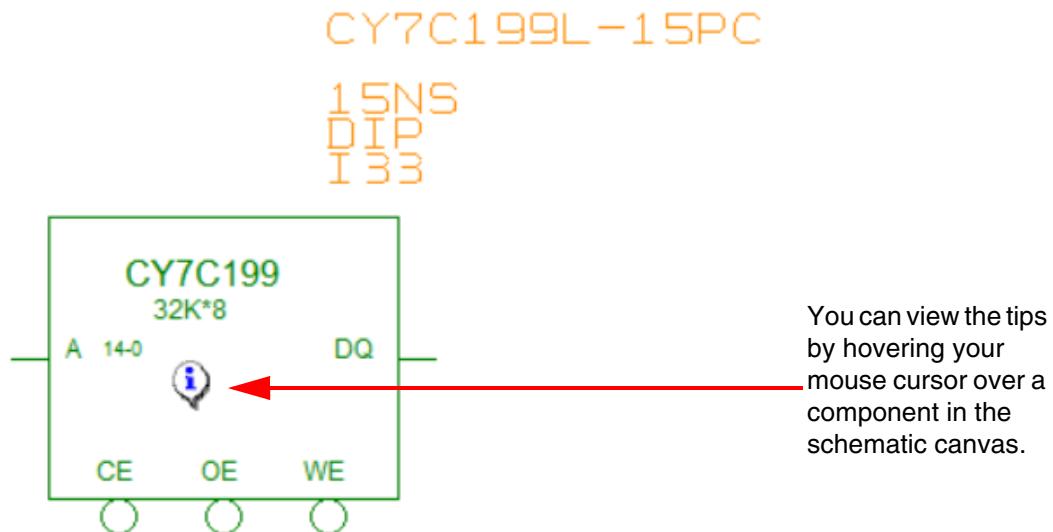
The directive can be configured at the project or site level.

For example, if some parts in a design have a common property such as `PART_NUMBER`, this property can be used to define data tips. The data tips file can contain entries corresponding

## Allegro Design Entry HDL User Guide

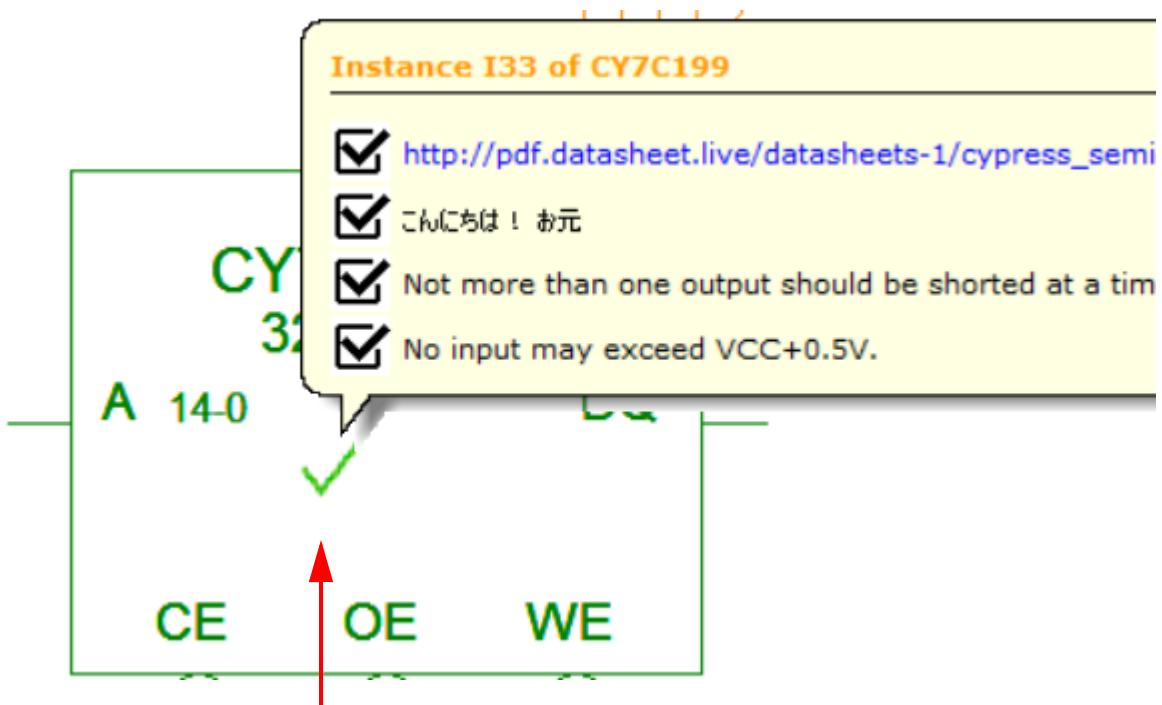
### Creating a Schematic

to different values of the PART\_NUMBER property. An information icon is displayed on components that have data tips in a schematic.



## Allegro Design Entry HDL User Guide

### Creating a Schematic



Information icon changes to a check mark if all the guidelines have been followed

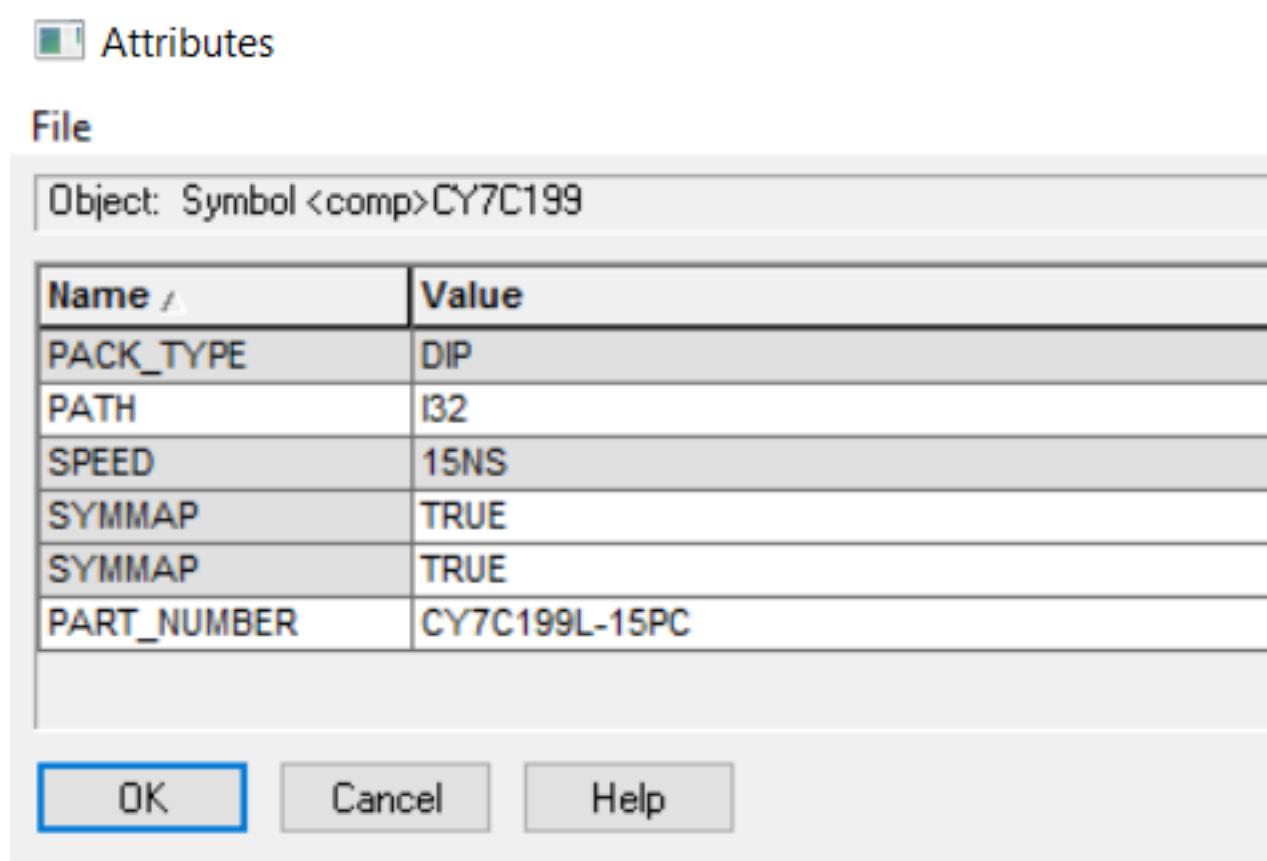
To add data tips for a part, do the following:

1. Specify the DATATIPS\_PROP\_NAME '<Part\_Property\_Name>' directive in the START\_CONCEPTHDL...END\_CONCEPTHDL section in the .cpm file.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

You can add and view the part property names and values in the Attributes box on the schematic canvas as illustrated:



For example, specify PART\_NUMBER as the part property name in the .cpm file.  
DATATIPS\_PROP\_NAME 'PART\_NUMBER'

2. Create a .txt file for the data tips that you want to define and name it datatips.txt. Store the file in the cdssetup folder at the project, \$HOME, or site level.
3. Define the part property value in this .txt file. For example: PART "CY7C199L-15PC". PART is a keyword to define the part for which data tips will be defined.
4. Define the data tips you want for this part in the following format:

<NUM>:<Data Tip Details>

For example:

```
PART "CY7C199L-15PC"
1: "http://pdf.datasheet.live/datasheets-1/cypress\_semiconductor\_data\_sheets/CY7C199L-15PC.pdf"
2: "こんにちは！ お元"
3: "Not more than one output should be shorted at a time."
4: "No input may exceed VCC+0.5V."
```

The data tips are displayed in the same order as the defined number in the data tip file. The same number is used to keep track of which data tip has been marked checked. If you check all the guidelines, the information icon changes to a check mark.

5. Save the file.

## Configuring Physical Property Options

### Overview

Allegro Design Entry HDL stores and retrieves physical part numbers and their attribute information for your project in a Part Table file (PTF). Before utilizing this information on a schematic, you need to configure physical property options (of a part) to define the format and visibility of the properties. For example, you can define the physical property options in such a way that properties such as PART\_NUMBER and TOL do not appear on the schematic.

Using the property options feature, you can:

- Configure various physical property options for parts.
- Control the appearance of PTF columns in the Part Information Manager window.

### Understanding Options Sets

The default physical property settings and definitions are stored in a text file named ppt\_optionset.dat for a given project. The settings for a specific part stored in a ppt\_optionset.dat file is called an option set. The ppt\_optionset.dat file is located in the project directory.

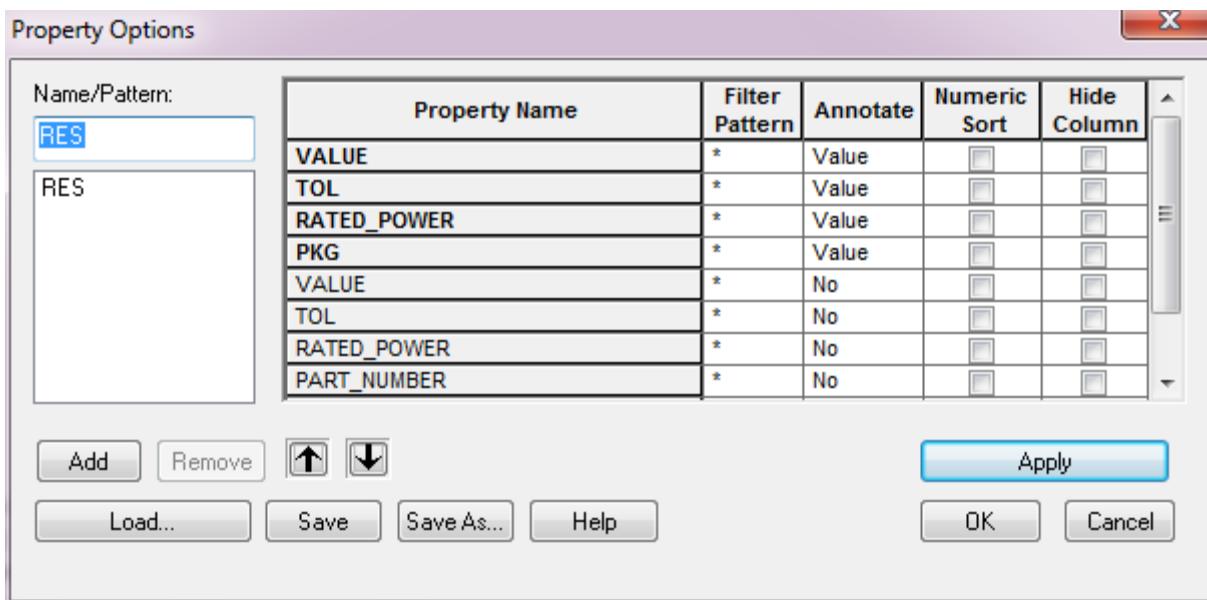
### Getting Started with Property Options

The Property Options dialog box is opened when you do the following:

## Allegro Design Entry HDL User Guide

### Creating a Schematic

- Click *PTF Options* in the Modify Component dialog box.
- Choose *Property Options* from the pop-up menu that appears when you right-click on a part row in the Part Information Manager window.



### Defining Physical Property Options

1. Select a physical property listed under the *Property Name* column, and click the Up (  ) or the Down (  ) to modify the order in which the property appears in the Property Options dialog box and on the schematic.

For example, if you move the PACK\_TYPE property to the lowest level in the list, the PTF row in the Part Information Manager window displays it as the last column. When you click on a row to place a part with physical information, Allegro Design Entry HDL displays PACK\_TYPE at the bottom of the list of properties.

2. Use the *Filter Pattern* column to filter physical property values based on the string you enter. For example, if you want the search results pane in the Part Information Manager window to display only that row of the PTF that has the value of VOLTAGE as 63V, select the VOLTAGE property under the *Property Name* column, enter 63 in the filter, and click *Apply*.
3. Select a value from the *Annotate* drop-down list to specify a visibility level of the physical property on the schematic.
  - a. If you select *No*, physical properties do not appear on the schematic.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

- b. If you select *Name*, only the names of the physical properties appear on the schematic.
  - c. If you select *Value*, only the values of physical properties appear on the schematic.
  - d. If you select *Both*, the names as well as the values of physical properties appear on the schematic.
  - e. If you select *Invisible*, the physical properties are added on the schematic, but are not displayed on the schematic.
- 4. Select the *Numeric Sort* check box to sort the property columns. Numeric sort treats property values as numbers and sorts them accordingly. In addition to alphanumeric and numeric sort, the property columns can be sorted on the basis of MKS units such as micro, milli, kilo, and so on.



You can perform numeric sort on property columns in the search results pane of Part Information Manager provided all the part table rows have the same part name.



You cannot use numeric sort on a property value (in the part tables) that contains strings.

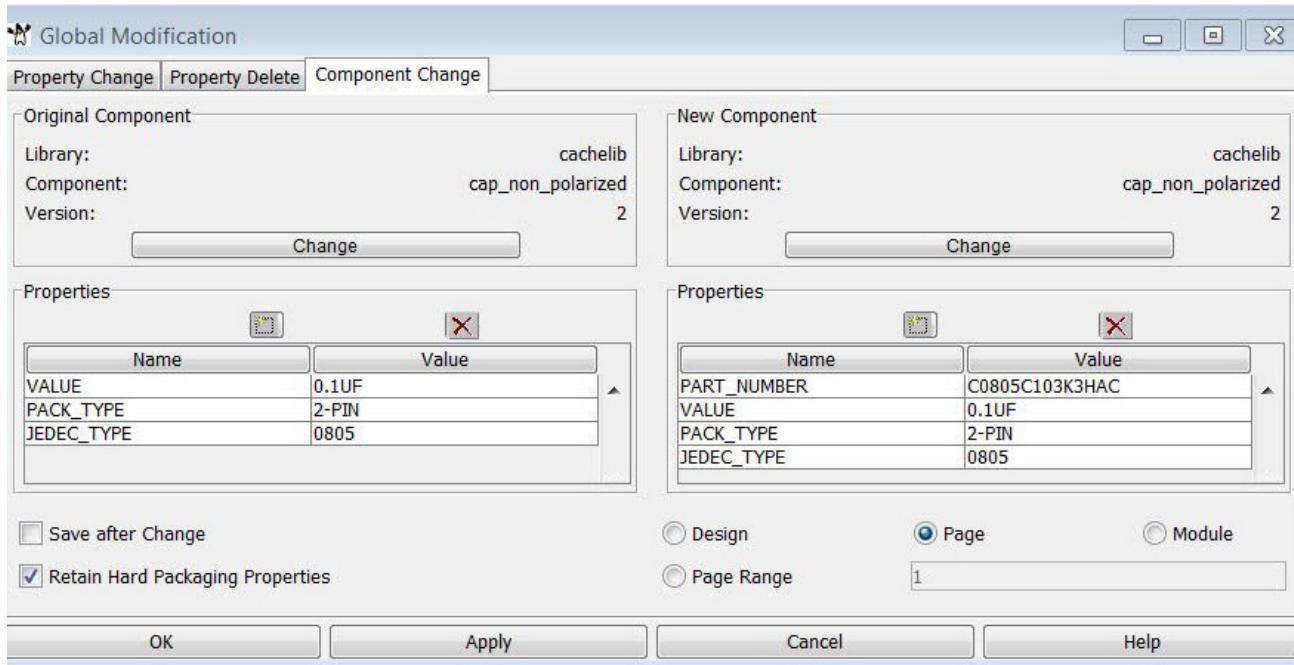
- 5. Select the *Hide Column* check box to hide the selected property column in the Part Information Manager window.
- 6. Click *OK* or *Apply* to apply the options you define.

**Note:** Properties set using property options are not updated for parts that are already in a design. If you want to apply properties to existing parts, you will have to modify or replace the existing part. You can use the global modify option to modify all the parts in the design. If you use this option, delete the rows of the properties that you want to

## Allegro Design Entry HDL User Guide

### Creating a Schematic

modify from the Original Component property list.



You will need to run the global modify option for every unique part in the design. You can use a global batch script for this.

7. Click **Save** to save all options to the `pptoptionset.dat` file. You can load these options in later sessions from the `pptoptionset.dat` file using the *Load* button. To save all the options in a new file, click *Save As*.



In a `ppt_optionset.dat` file, if more than one entry matches a given name, the last entry for option settings wins.

### Sample `ppt_optionset.dat` File

The syntax of the `ppt_optionset.dat` file is as follows:

```
( "VERSION 3.0"  
(  
( OPTION_SET_ATTRIBUTES )  
( PROPERTY_ATTRIBUTES )  
)
```

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

)

OPTION\_SET\_ATTRIBUTES has the following four fields:

Field	Description
optionsetname	A character string to identify the option set
totalprops	Total number of properties in the option set
Keyprops	Number of key properties in the set
Injprops	Number of injected properties in the set

PROPERTY\_ATTRIBUTES has the following fields

:

Field	Lets you
Proptype	Define the type of property. Use 1 for key property and 0 for the injected property.
Propname	Specify the name of the property.
Filterpattern	Define a filter pattern to be used for the property.
SortType	Use 1 to specify numeric sort; use 0 to ensure that numeric sort is not applied
Annotate	Define 0 to not use the annotation, 1 for name, 2 for value, 3 for both, and 4 for invisible.
Visible	Use 1 to make a property visible in the Part Information Manager window; use 0 to hide a property in the Part Information Manager window.
Propcol	Specifies the column number of the property. Use 0 for the property in the first column.

**Note:** SortOrder and Annotate Type are obsolete properties that are no longer supported.

Given below is a sample ppt\_optionset.dat file with the following option sets:

```
20L10
DG419
( "VERSION 3.0"
```

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

```
(  
  ("20L10" 4 1 3)  
  (1 "PACK_TYPE" "LST" 0 1 0 1 1 0)  
  (0 "PART_NUMBER" "*" 1 0 0 0 1 3)  
  (0 "DESCRIPTION" "*" 1 0 0 0 1 1)  
  (0 "JEDEC_TYPE" "*" 0 0 0 0 1 2)  
)  
(  
  ("DG419" 4 1 3)  
  (1 "PACK_TYPE" "*" 1 0 0 2 1 0)  
  (0 "PART_NUMBER" "*" 0 0 0 0 1 1)  
  (0 "DESCRIPTION" "*" 0 0 0 0 1 2)  
  (0 "JEDEC_TYPE" "*" 0 0 0 0 1 3)  
)  
)
```

Where, ("20L10" 4 1 3) and ("DG419" 4 1 3) represent the OPTION\_SET\_ATTRIBUTES for the two option sets: 20L10 and DG419. The following table lists the OPTION\_SET\_ATTRIBUTES details for the two option sets.

<b>Field</b>	<b>20L10</b>	<b>DG419</b>
optionsetname	20L10	DG419
totalprops	4	4
Keyprops	1	1
Injprops	3	3

The following table explains the first row of PROPERTY\_SET\_ATTRIBUTES for the two option sets.

<b>Field</b>	<b>20L10</b>	<b>DG419</b>
Proptype	1	1
Propname	PACK_TYPE	PACK_TYPE

Field	20L10	DG419
Filterpattern	LST	*
SortType	0	1
Annotate	1	2
Visible	1	1
Propcol	0	0

### Copying an Option Set

To copy an option set:

1. Select a part name in the Name/Pattern list.
2. Change the name in the text field of the *Name/Pattern* list.
3. Click *Add*.

The new part name is added to the Name/Pattern list.

### Removing an Option Set

To remove an option set:

1. Select a name in the Name/Pattern list.
2. Click *Remove*.

**Note:** You cannot remove the current option set that you are working with.

### Customizing PTF Options Filters

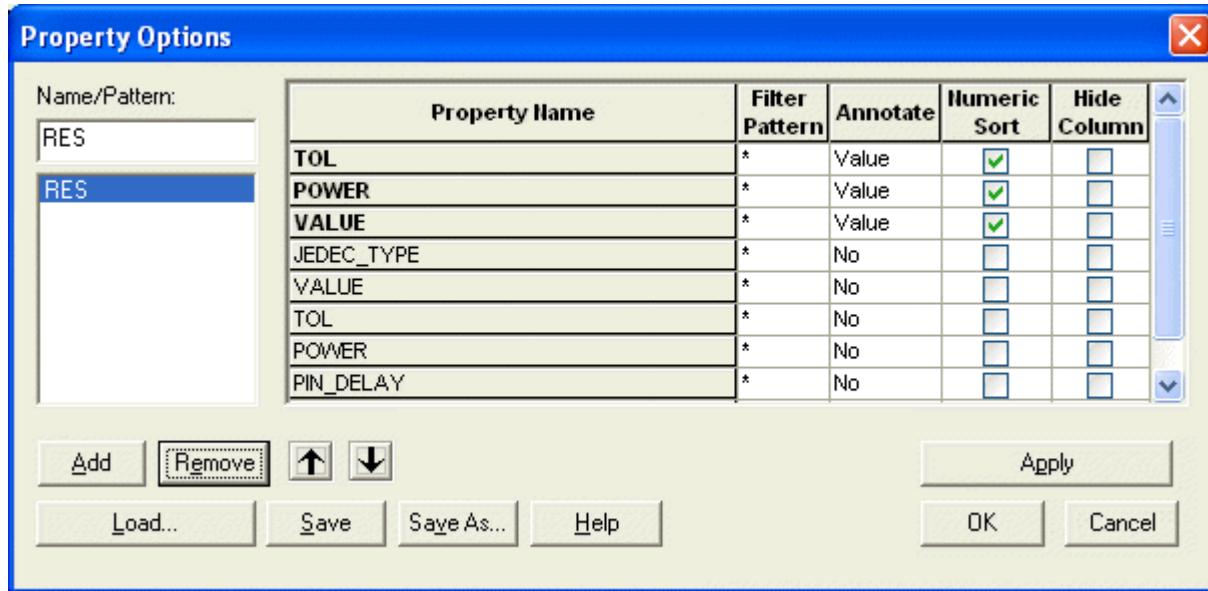
You can also seed initial part table filter values in the Property Options dialog box in addition to the default \* filter. To create a custom filter, perform the following steps

1. In Part Information Manager, select a part.
2. Right-click a row in the search results pane and choose *Property Options* from the pop-up menu.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

The Property Options dialog box is displayed.



3. Type \* in the Name/Pattern field and click Add.

An entry will be created automatically by copying the properties from the existing part from which the Property Options dialog box was launched.

4. Right-click a row on the grid and choose *Insert Row* or *Remove Row* from the pop-up menu to add and remove properties.

**Note:** This pop-up menu is available for the part name \*.

5. Specify the filter values.
6. Click Save to save the ppt\_optionset.dat file.
7. Click OK to close the Property Options dialog box.

When a search is performed on a part for which PTF options exist, the options are honored. Otherwise, the default settings saved as \* are honored.

## Connecting Parts

You can connect parts in Design Entry HDL using wires. Parts can be connected manually using *Wire – Draw* or automatically connected using *Wire – Route*.

## Drawing a Wire Manually

### **To draw a wire without naming it**

1. Choose *Wire – Draw*.
2. Click a pin on a component.
3. Click again wherever you want the wire to bend, or click a pin on another component.

### **To name a wire when you draw it**

1. Choose *Wire – Draw*.
2. Right-click and choose *Signal Name...* from the pop-up menu.
3. Type a signal name in the *Signal Name* box.
4. Click on the wire that extends from the component.
5. Save the design.
6. Launch Constraint Manager from De HDL. SIG B is now displayed as the renamed net.

## Auto-Routing a Wire

1. Choose *Wire – Route*.
2. Click the edge of a component, then click the edge of another component.

**Note:** You can also run the `route` command using this stroke pattern



For more information on working with wires, see [Chapter 6, “Working with Wires.”](#)

## Naming Signals

To name an existing wire:

1. Choose *Wire – Signal Name*.
- The *Signal Name* dialog box appears.

2. Type one or more signal names on separate lines.
3. Select the wires you are naming in the same order you entered them in the *Signal Name* dialog box.

To name a wire when you draw it:

1. Choose *Wire – Draw*.
2. Right-click and choose *Signal Name* from the pop-up menu.
3. Type a signal name in the *Signal Name* box.
4. Click wherever you want the wire to bend, or click a pin on another component.

## Signal Naming Conventions

Signal names must adhere to the following conventions:

- Names can start with a letter, numeral or supported special characters.
- Names cannot be VHDL and Verilog keywords.
- Names cannot be NC as this name is reserved for unconnected nets.
- Design Entry HDL is not case-sensitive. Design Entry HDL treats two names that differ only in uppercase or lowercase as the same name.

The following characters have special significance in signal names. Follow these conventions while naming signals.

**Note:** These conventions also apply to component path names.

For more information, see the [Naming Rules and Conventions](#) chapter of *Allegro Design Entry HDL Reference Guide*.

- : Used for concatenating signals, and for specifying the range and step size of a bus. While concatenating signals, a colon needs an operand on both sides. For example, A:B is legal, while AB: is illegal. To understand the usage of colon in a bus name, refer [Step Size in Signal Names](#).

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

- & When the MULTI\_FORMAT directive is ‘ON’, an ampersand represents concatenation and needs an operand on both sides. If you set the MULTI\_FORMAT directive to ‘OFF’, the ampersand character has no special meaning and can be used anywhere in a signal name.
- ,
- Used for concatenating signals. To use a comma for concatenating signal names, the MULTI\_FORMAT directive must be ‘ON’. The operator needs an operand on both sides. For example, A,B is legal, while AB, is illegal.
- \ Must be followed by one of the following characters:
- BASE - For declaring the name of the signal as the “base” signal name for all its aliases or synonyms. For more information, see [Declaring a Base Signal on page 211](#).
- G - For *Global*. Implies the signal is a global signal. For more information, see [Global Signals on page 213](#).
- I - For *Interface*. Implies the signal is a port. Cadence recommends that you use the port symbols IMPORT, EXPORT, or INOUT instead of the \I suffix. The \I suffix declares ports as INOUT ports. If you use the port symbols, you can explicitly declare a port as an IN, OUT or INOUT port. For more information on using port symbols, see [Adding Ports on page 203](#).
- L - For *Local*.
- / When used at the beginning of a name, this indicates a global signal.
- ! When used at the beginning of a name, this indicates a global signal.
- { } Are not special characters and can be used without any restriction.
- < > Indicates that the signal is a bus. The angle brackets must be matched correctly and must contain either a parameter or an integer. Cannot be used anywhere else in a signal name.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

- ( ) When the MULTI\_FORMAT directive is ON', the parentheses indicate that the signal is a bus. Must be matched correctly and contain either a parameter or an integer. If you set the MULTI\_FORMAT directive to 'OFF', the parentheses have no special meaning and can be used anywhere in the signal name.  
**Note:** If you set the MULTI\_FORMAT directive to 'OFF', do not use 'space' and parenthesis.
- [ ] When the MULTI\_FORMAT directive is ON', the square brackets indicate that the signal is a bus. Must be matched correctly and contain either a parameter or an integer.  
If you set the MULTI\_FORMAT directive to 'OFF', the square brackets have no special meaning and can be used anywhere in the signal name.
- \*
- 0 0 is converted to ZERO in the netlist. The name ZERO indicates a low signal.
- 1 1 is converted to ONE in the netlist. The name ONE indicates a high signal.
- \_N (underscore suffixed with N) When used at the end of a signal name or a pin name, indicates that the signal or pin is low-asserted. Cadence recommends that you use the \_N suffix to indicate a low-asserted signal or pin. For more information, see [Specifying the Assertion Level of Pins and Signals](#) on page 209.

The following characters can be used in signal names without any restrictions:

---

- (dash/minus/  
hyphen)

#

\$

%

+

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

=

|

?

@

^

,

The following characters cannot be used in signal names:

;

~

'

"

!

\* can be used only at the end of the signal name. ab\*c is illegal.

## Step Size in Signal Names

Bit subscripts specify the number of bits that a signal represents, and identify the bits.

### Syntax

```
<bit1..bit2:step>  
<bit1:bit2:step>
```

The syntax specifies a sub-range of bits beginning with bit1 or bit2, whichever is the LSB, and including every bit that is step bits apart up to bit1. The step value is usually a positive integer. Use a negative integer to reverse the bit order. A step value of 1 is equivalent to no step value.

### Examples

Subscript	Result
<31..0:2>	30 28 26 ... 6 4 2 0
<11..0:4>	8 4 0
<9:1:3>	7 4 1
<0..31:-1>	31 30 29 ... 3 2 1 0
<15..0:20>	0
<0..6:-2>	0 2 4 6
<0..7:-2>	0 2 4 6

Consider the following examples:



In this example,  $B<60..0:4>$  results in  $B[60], B[56], B[52], B[48]...B[16], B[12], B[8], B[4], B[0]$ .



In this example,  $Z<31..0:2>$  is synonym to  $A<30..45>$ .

$Z<31..0:2>$  results in  $Z[30], Z[28], Z[26], Z[24], ... Z[6], Z[4], Z[2], Z[0]$ . Therefore, Design Entry HDL does the following assignment:

```
assign a[30:45] =  
{z[30], z[28], z[26], z[24], z[22], z[20], z[18], z[16], z[14], z[12], z[10],  
z[8], z[6], z[4], z[2], z[0]};
```

#### ***Limitations of Signal Naming***

- The Design Entry SCALD syntax <bit:width> is not supported. The syntax <bit:width> is supported as <bit..width>. For example, <31:8> is treated as <31..8>.
- The syntax <bit1..bit2:step> is supported only for signals. This syntax is not supported for pin names. If you create a symbol from a schematic, interface signals become pins and therefore an interface signal cannot use the above syntax. For example, if you have used AA<7..0:2> as the interface signal and created a symbol from this, Design Entry HDL will not generate AA<7..0:2> as the pin. It will generate AA<0..7> as the pin.
- The syntax <bit1..bit2:step> is not supported for PATH properties.
- You cannot perform Global Find or Global Navigate operations on buses that have step size in their names. For example, a signal with the name A<31..0:2> is not found by Global Find.
- Bits tapped from a bus that has step size in signal names are not supported. For example, consider a bus named ADDR<11..0:4>. Tapping bits ADDR<8>, ADDR<4>, and ADDR<0> from the ADDR bus ADDR<4..0> is not supported.
- The scalar signal name NC needs to be used even for vector signals connected to unconnected vector pins of a component. You cannot connect a signal of the form NC<0..n> to a vectored pin. Design Entry HDL automatically treats all bits of the unconnected vector pin as NC.

**Note:** Signals that are named with incorrect syntax are not included in the connectivity database. Saving a design in Design Entry HDL with a syntax error for a bus, will delete it from Constraint Manager as well.

## **Adding Properties**

#### ***To add one property at a time***

1. Choose *Text – Property*.
2. In the *Property* dialog box, enter a name in the *Property Name* box and a value in the *Property Value* box.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

**Note:** Spaces are not supported in the LOCATION and \$LOCATION property values. If you use spaces while specifying LOCATION and \$LOCATION properties, the Property dialog box exits with an error message. Similarly, colon (:) is not supported in reference designator values. Its use will result in an error.

**Note:** You can define the visibility of pin text on the schematic using the PIN\_TEXT\_VISIBLE property. Set it to TRUE to honor visibility on the schematic. FALSE means that PIN\_TEXT will always be invisible on the schematic.

**Note:**

3. Click *OK*.
4. Click the object to which you are attaching the property.
5. Click near the object to indicate where to display the property information.

As the default, Design Entry HDL displays only the property value. Choose *Text – Property Display* to modify how properties are displayed.

**Note:** You can also run the *property* command using this stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#) on page 131.

You can also add a property using the *property* command.

**To add many properties at the same time using the Attributes dialog box**

1. Choose *Text – Attributes* or type *attribute* in the console window.
2. Select an object.

The object you select appears highlighted, and the *Attributes* dialog box displays attributes for the object.

3. Click *Add* in the *Attributes* dialog box.

An empty row appears.

4. Type a property name and a value in the appropriate columns.
5. Adjust property visibility and alignment as needed.
6. Click *OK* to add, or click *Cancel*.

#### ***Adding Ports***

When you are creating a Design Entry HDL schematic, you must place port symbols on the page to indicate the ports on the entity. Although a signal name with a \I suffix is acceptable, it is preferable to use a port symbol instead. The Standard library has the following port symbols:

- INPORT (input)
- IOPORT (bi-directional: Input/Output)
- BUFPORt (used only for VHDL)
- OUTPORT (output)
- LNKPORT (used only for VHDL)
- AOUTPORT (for behavioral assignments)

#### **Using PORT Symbols**

To use a port symbol:

1. In Design Entry HDL, choose *Component – Add*.  
Part Information Manager appears.
2. Select standard from the *Library* list in the search pane.
3. Select a port symbol from the *Cells* list.
4. Click in the Design Entry HDL drawing area to place the symbol.
5. Close Part Information Manager.
6. Attach a wire to the pin on the port and connect it to an instance.

**Note:** A VHDL and Verilog restriction prohibits you from wiring different ports of an entity together. Design Entry HDL gives a warning if different ports of an entity are wired together in your schematic.

7. Choose *Wire – Signal Name* and name the wire.

This signal name is the port declaration in the VHDL and Verilog text.

8. Define the VHDL logic type and Verilog logic type of the port.

The default VHDL logic type for ports in Design Entry HDL schematics is STD\_LOGIC for scalar ports and STD\_LOGIC\_VECTOR for vectored ports. The default Verilog logic type

for all ports in Design Entry HDL schematics is `WIRE`. You can change these defaults for a drawing or for the entire project. You can also override these default logic types by choosing a different type for individual ports. See [Setting the Verilog Logic Type for Ports and Signals](#) on page 231 and [Setting the VHDL Logic Type for Ports and Signals](#) on page 234 for details.

If you want to use custom port symbols instead of those supplied in the Standard library, copy all the visible and invisible properties from the port symbols to the new symbol.



#### *Important*

Note the following when you use port symbols:

- ❑ If you leave an output port of an instance unconnected or if you attach the signal `NC` to a port, the port is represented as open in VHDL and as unconnected in Verilog.
- ❑ Although the VHDL language allows you to create ports with an unconstrained range, you cannot create Design Entry HDL schematics with unconstrained ports. If you want an unconstrained port, use a parameterized range for the port. For more information, see [Unconstrained Ranges for Ports, Signals, and Aliases](#) on page 239.
- ❑ To connect a signal of one type to a port of another type in VHDL, use a type conversion function. Verilog does not use type conversion functions; in Verilog you can connect a signal of type `WIRE` to ports of other types. While generating Verilog text, Design Entry HDL ignores any type conversion properties that you place on the schematic for VHDL. For more information about type conversion, see [Type Conversion](#) on page 240.
- ❑ Verilog does not support abstract data types. In VHDL, however, ports can be of abstract data types such as integers and floating point numbers. For more information, see [Abstract Data Types in VHDL](#) on page 241.

## Rules for Using Port Symbols

- Name the signal to which a port is attached.
- Do not leave an input port of an instance unconnected; it will generate both a VHDL error and a Verilog error.
- Do not connect ALIAS symbols to ports.
- Do not wire different ports of an entity together.
- Follow port association rules. For more information, see [Port Association Restrictions](#).

## Port Association Restrictions

The VHDL language has strict rules regarding the port associations allowed between ports of component instances within an architecture and the ports of the entity declaration.

Port association rules are not as strict for Verilog as they are for VHDL. If you use the `VHDL_USER=NO` property on the `VERILOG_DECS` symbol, you do not need to follow the rules described here.

The following table shows the port associations allowed in VHDL.

Formal Port	Actual Port
IN	IN, INOUT, BUFFER
OUT	OUT, INOUT
INOUT	INOUT
BUFFER	BUFFER

A formal port is the port on an instance; an actual port is the port in the entity description.

For example, if a formal port is an INOUT port and it is connected to ports higher up in the design hierarchy, the other ports must also be declared as INOUT ports. Similarly, BUFFER ports must remain BUFFER ports as they ascend the design hierarchy.

## Working with Ports and Signals

This section describes the following:

- [Setting the Initial Value of a Signal](#) on page 206
- [Specifying the Assertion Level of Pins and Signals](#) on page 209
- [Creating an Alias for a Signal](#) on page 209
- [Global Signals](#) on page 213
- [Unnamed Signals](#) on page 220
- [Signal Concatenations](#) on page 221
- [Signal Replication](#) on page 223

- [Merge Symbols](#) on page 224
- [Signal Slices \(Bit and Part Selects\)](#) on page 228
- [Setting the Verilog Logic Type for Ports and Signals](#) on page 231
- [Setting the VHDL Logic Type for Ports and Signals](#) on page 234
- [Specifying Ranges for Ports, Signals and Aliases](#) on page 238
- [Unconstrained Ranges for Ports, Signals, and Aliases](#) on page 239
- [Resolved Types and Resolution Functions](#) on page 239
- [Type Conversion](#) on page 240
- [Abstract Data Types in VHDL](#) on page 241

## Setting the Initial Value of a Signal

You can use the `VHDL_INIT` property to assign an initial value to a scalar signal or bits of a vector signal. The signal can be a local signal or a global signal.

Add the `VHDL_INIT` property as follows:

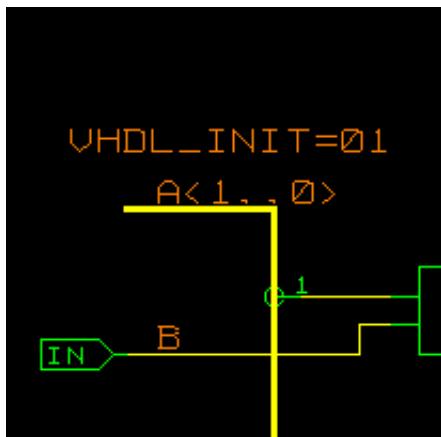
1. In Design Entry HDL, choose *Text – Attributes*.
2. Click on the signal to display the *Attributes* dialog box.
3. Click *Add*.
4. Type `VHDL_INIT` in the *Name* text box.
5. In the *Value* text box, type the initial value (0, 1, L, or H) of the signal. If the signal is a vector signal, type the values for all bits. Specifying values for only some bits produces an error.

**Note:** Do not use quotes around the value. Design Entry HDL automatically adds quotes to the value.

6. Click *OK* to save the changes and close the *Attributes* dialog box.

### Example

To set the initial value of a vector signal A<1..0> to 0 and 1, set the following.



The vhdl.vhd file then contains the initial value for signal A:

```
signal A: std_logic_vector (1 downto 0) := "01";
```

**Note:** The VHDL\_INIT property can also be attached to the pins of a symbol. When a VHDL\_INIT property is attached to a power symbol or to its pin, its power signal is initialized with that value.

### Example

Assume you have added the VHDL\_INIT property to the pin of VCC\_ARROW as follows:



The vhdl.vhd file then contains the initial value for signal VCC\_ARROW:

```
global VCC_ARROW: std_logic ;
BEGIN
  VCC_ARROW <= '1' ;
```

## Allegro Design Entry HDL User Guide

### Creating a Schematic

**Note:** The `SIG_NAME` property must exist on the signal before the `VHDL_INIT` property is assigned. For example, if you attach the following `VHDL_INIT` property to an unnamed signal:



You can see that the `vhdl.vhd` file does not contain the initial value for the unnamed signal  
`signal UNNAMED_1_54ALS00_I1_Y: std_logic ;`

## Specifying the Size of Nets

Use the `SLASH` symbol in the Standard library to specify the size of nets. The `SLASH` symbol is useful for:

- Documenting the width of signals. Normally, the width of a signal is apparent from the bit subscript on the signal name. However, if the signal name is not visible on a certain part of the schematic, you can use the `SLASH` symbol to document the width. The width you specify on the `SLASH` symbol must be the same as the actual width of the signal.
- Specifying the size of nets that do not have a size. For example, if you want to make an unnamed net a bus, you can put a `SLASH` symbol on the net and specify its size. Similarly, you can use a `SLASH` symbol to specify the width of unnamed outputs of `CONCAT` and `MERGE` symbols.

## Using a SLASH Symbol

To use a `SLASH` symbol:

1. In Design Entry HDL, choose *Component – Add*.  
Part Information Manager appears.
2. Select `standard` from the *Library* list in the search pane.
3. Select the `SLASH` symbol from the *Cells* list.
4. Click in the Design Entry HDL drawing area to place the symbol.
5. Close Part Information Manager.
6. Attach the `SLASH` symbol to the net you want to size.
7. From the *Text* menu, choose *Attributes*.

8. Click on the SLASH symbol to display the *Attributes* dialog box.
9. Specify the size of the net by changing the value of the SIZE property. The default value of the SIZE property is 1.

**Note:** If you are using a SLASH symbol only for documentation and the width of the signal is already set, the value of the SIZE property on the SLASH symbol must match the actual width of the signal.
10. Click *OK* to save the changes and close the *Attributes* dialog box.

## Specifying the Assertion Level of Pins and Signals

Cadence recommends a notation for representing the assertion level of pins and signals. By convention, a signal is active high for positive logic and active low for negative logic. Two signals with the same name but with different assertion levels are considered to be different signals.

The assertion level of a signal is determined by the `_N` or `*` suffix or a `-` prefix. Cadence recommends that you use a `_N` suffix to indicate a low-asserted signal or pin.

### **To specify a low-asserted signal or pin**

- Suffix `_N` or `*` to the signal name or pin name.

### **Example**

`ENABLE_N` is an active low scalar signal.

`DATA<15..0>_N` is an active low vector signal.

`ENABLE*` is an active low scalar signal.

`DATA<15..0>*` is an active low vector signal.

Any signal or pin that does not have the `_N` or `*` suffix is assumed to be an active high signal.

**Note:** `DATA_N<15..0>` is also supported for low vector signal names.

## Creating an Alias for a Signal

Use the ALIAS symbol in the Standard library to specify another name for a signal.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

For example, the following ALIAS



creates the following alias declaration in VHDL

```
alias opcode: std_logic_vector (0 to 2) is instr (7 downto 5);
```

and the following alias declaration in Verilog

```
alias_vector alias_inst1 (opcode[0:2], instr[7:5]);  
defparam alias_inst1.size = 3;
```

The ALIAS symbol is similar to the SYNONYM symbol in the Standard library. If you use SYNONYM symbols, Design Entry HDL will not detect all the VHDL-related errors and your VHDL output will be inaccurate. Therefore, if you want to generate VHDL text from the schematic, you must use ALIAS symbols instead of SYNONYM symbols.

If you currently use SYNONYM symbols in your designs, replace them with ALIAS symbols. If you are not generating VHDL text from your schematic, you can use either the ALIAS symbol or the SYNONYM symbol.

### Creating an ALIAS

To create an ALIAS

1. In Design Entry HDL, choose *Component – Add*.  
Part Information Manager appears.
2. Select standard from the *Library* list in the search pane.
3. Select ALIAS from the *Cells* list.
4. Click in the Design Entry HDL drawing area to place the symbol.



5. Close Part Information Manager.
6. Attach the original signal to the left pin of the ALIAS symbol.
7. Attach the ALIAS signal name to the right pin of the ALIAS symbol.

**Note:** The right pin is the pin of the ALIAS symbol which has the VHDL\_ALIAS property. The left pin is the other one.

#### Rules for Using ALIAS Symbols

##### ALIAS symbols

- can be connected to a SLICE.
- should not be connected to ports.
- should not be connected to global signals.
- should not be connected to the output of taps.
- should not be connected to the output of CONCAT signals.
- should not be connected to unnamed signals.



If a scalar signal is connected to a vector pin, HDL-Direct throws an error. However, if a scalar GLOBAL signal is connected to a vector pin, all bits of the pin are aliased to the signal, and no errors are generated.

#### Declaring a Base Signal

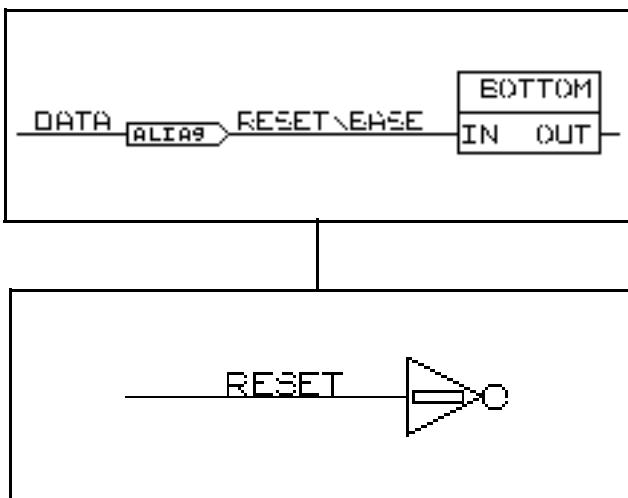
When two signals are aliased or synynomed, Design Entry HDL selects one of the signal names as the base signal. The name of the base signal becomes the name of corresponding physical net in PCB Editor. A signal is its own base signal if it is not aliased or synynomed to any other signal or if it is selected as the base signal.

You may want the name of a particular aliased or synynomed signal to be passed to PCB Editor as the physical net name. You can force Design Entry HDL to use a particular aliased or synynomed signal by declaring it as a base signal, as below:

## Allegro Design Entry HDL User Guide

### Creating a Schematic

- Suffix \BASE to the signal name or add the MAKE\_BASE=TRUE property on the signal at the highest schematic level at which the signal exists.



Schematic for block MID

Schematic for block BOTTOM

For example, in the above hierarchical design, signal DATA is aliased to signal RESET in the schematic for block MID. The RESET signal is also present in the schematic for the lower level block BOTTOM. To declare signal RESET as the base signal, you must suffix \BASE or add the MAKE\_BASE=TRUE property on the signal RESET in the schematic for block MID and not on the signal RESET in the schematic for the lower level block BOTTOM.

Because aliased or synched signals at higher schematic levels always supersede aliased or synched signals at lower schematic levels, it is only meaningful to use the \BASE suffix or add the MAKE\_BASE=TRUE property to the aliased or synched signal at the highest schematic level at which the signal exists.

**Note:** If a global signal is aliased or synched to a signal that has the \BASE suffix or the MAKE\_BASE=TRUE property, the global signal will always be treated as the base signal. For more information, see [Rules for Choosing the Base Signal](#) on page 212.

**Note:** If a power symbol `vcc` is aliased or synched to another power symbol or global signal `5V` and you want to make the `5V` signal as the base signal, name the signal that is connected to the power symbol `5V` as `5V\BASE\G`, where `\G` represents a global signal. For more information on global signals, see [Global Signals](#) on page 213.

### Rules for Choosing the Base Signal

The rules for selecting a base signal are listed below, in the order in which Design Entry HDL applies them. If Design Entry HDL cannot select the base signal name from the first rule, the next rule is applied, and so on.

#### 1. Select a global signal over a non-global signal.

For example, if a signal `CLOCK` is aliased to a global signal `SWITCH\G`, the global signal will be the base signal.

**Note:** The global signal will be the base signal even if it is aliased or synched to a non-global signal that has the `\BASE` suffix or the `MAKE_BASE=TRUE` property.

#### 2. Select the signal that has the `\BASE` suffix in its name or has the `MAKE_BASE=TRUE` property.

#### 3. Select a constant signal over a non-constant signal.

For example, if a constant signal `123` is aliased to a non-constant signal `CLOCK`, the constant signal will be the base signal.

#### 4. Select the lower bit number of two signals with the same name (for example, `X<0>` is selected over `X<3>`).

#### 5. Select a user-assigned signal name over an unnamed signal.

For example, if a signal `CLOCK` is aliased to an unnamed signal, the signal `CLOCK` will be treated as the base signal.

#### 6. Select a scalar signal over a vector signal.

#### 7. Select the signal that is lexicographically smaller (for example, `CLK` is selected over `CLOCK`).

## Global Signals

If you want to use a global signal in your schematic, suffix `\G` to the signal name.

Verilog global signals modules and VHDL global packages are created automatically for you from schematics that contain global signals; you do not have to create these manually. Global modules and packages are created when you either expand a design or package it.

When you expand or package, a cell called `gbl1` is created in the design library. This cell has `RootDesign_Configuration` views, which contain a `verilog.v` file with the Verilog global signal module and a `vhdl.vhd` file with the VHDL globals package.

## Shorting of Global Signals

If a global signal in your design is shorted with another global signal, errors are displayed when you save or package the design. For example, if a `+5V` global is shorted with a `GND` global signal:

## Allegro Design Entry HDL User Guide

### Creating a Schematic

- The following error message is displayed in the *Markers* dialog box when you save the design:

ERROR:275: Two global signals are shorted.

- The following error message can be seen in the Export Physical Progress window or in the `pxl.log` file when you package the design:

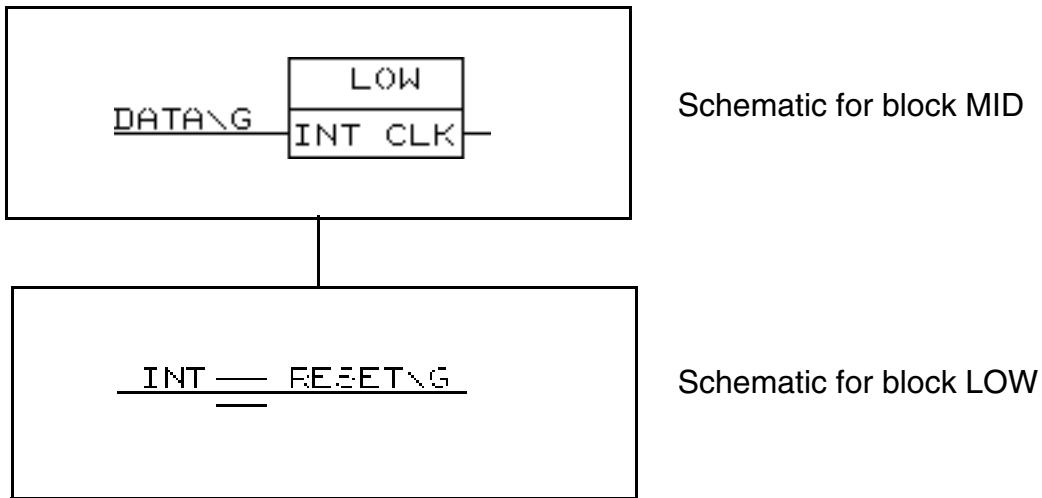
Two global signals are shorted. If you want to short these global signals, add them in Allowed Global Shorts list in Design Entry HDL setup options.

Net name: '@<root\_design\_name>.glbl(<root\_design\_name>\_cfg\_package):+5v'

Net name: '@<root\_design\_name>.glbl(<root\_design\_name>\_cfg\_package):GND'

You might have intentionally shorted some global signals in your design. If you want to allow such global signals to be shorted, add them in the *Allowed Global Shorts* list of the *Design Entry HDL Options* dialog box. Design Entry HDL will not display this error message if the global signals listed in the *Allowed Global Shorts* list are shorted.

When you save a design in Design Entry HDL, error messages are displayed only for the global signals that are shorted within the block you are currently editing. When you package the design, error messages can be seen in the Export Physical Progress window or in the `pxl.log` file for the global signals that are shorted across different blocks in the design.



In the above figure, signal `DATA\G` is connected to the pin `INT` of block `LOW` in the schematic for block `MID`. In the schematic for block `LOW`, signal `INT` is synynomied with signal `RESET\G`. This results in the shorting of the global signals `DATA\G` and `RESET\G` across the blocks in the design. When you save the schematic for block `MID` or the schematic for block `LOW` in Design Entry HDL, no error message for global signal short is displayed. However, when you

package the design, error messages for the shorting of global signals DATA\G and RESET\G can be seen in the Export Physical Progress window or in the px1.log file.

#### ***Using the Allowed Global Shorts List***

For example, suppose that there are 5 global signals—GND\G, +5V\G, DATA\G, CLOCK\G and SWITCH\G—in a design, where:

- Signal +5V\G gets aliased to signal GND\G by mistake
- Signal +5V\G is synonymed to signal DATA\G to intentionally short the signals
- Signal CLOCK\G gets synonymed to signals SWITCH\G and DATA\G, where signals CLOCK\G and SWITCH\G are intentionally shorted, but signal CLOCK\G got synonymed to signal DATA\G by mistake

To allow signal +5V\G to be shorted with signal DATA\G and signal CLOCK\G to be shorted with signal SWITCH\G, do the following:

1. Choose *Tools – Options* in Design Entry HDL.

The *Design Entry HDL Options* dialog box appears.

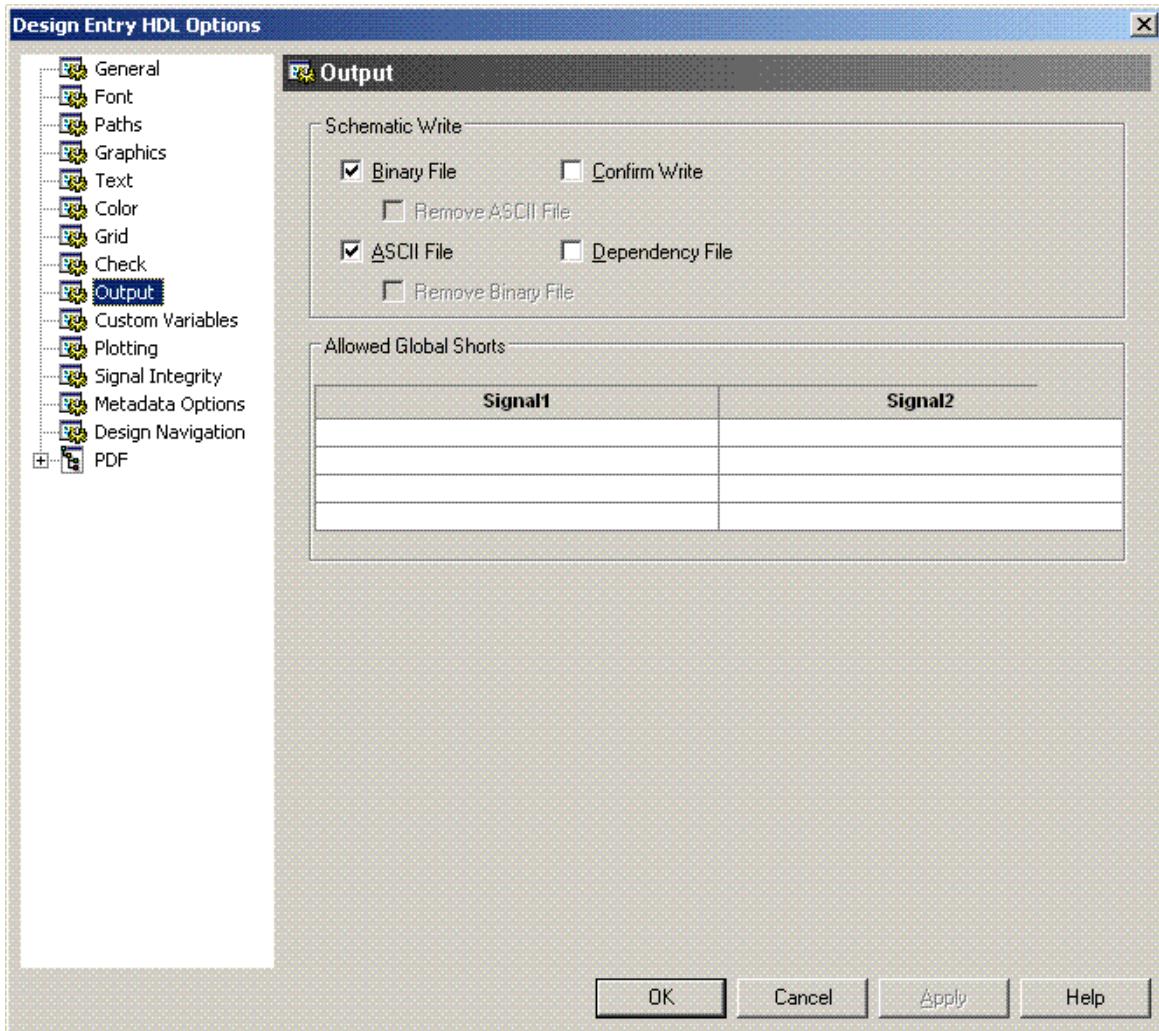
2. Select the *Output* tab and do the following in the *Allowed Global Shorts* list:

- a. Type +5V in the *Signal1* field and DATA in the *Signal2* field next to the +5V signal.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

- b. Type CLOCK in the *Signal1* field and SWITCH in the *Signal2* field next to the CLOCK signal.



The *Allowed Global Shorts* list allows you to add pairs of global signals that you want to remain shorted in the design.

Click *OK*.

When you save or package the design, error messages are displayed only for the shorted global signals that are not specified in the *Allowed Global Shorts* list.

If two shorted global signals are specified in the *Allowed Global Shorts* list and if:

- you have specified one of the signals in the *Supply 0* list in the *Verilog* netlisting options dialog box or have added the `VLOG_NET_TYPE=SUPPLY0` property on it, and

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

- you have specified the other signal in the *Supply 1* list in the *Verilog* netlisting options dialog box or have added the `VLOG_NET_TYPE=SUPPLY1` property on it,

Design Entry HDL displays the following error message when you netlist the design for digital simulation using *Tools – Simulate*:

ERROR:275: Two global signals are shorted.

This error is generated because in a design, *Supply 0* and *Supply 1* signals should not be shorted.

For example, if you have specified two shorted global signals `CLOCK\G` and `SWITCH\G` in the *Allowed Global Shorts* list and have also added `CLOCK\G` in the *Supply 0* list and `SWITCH\G` in the *Supply 1* list in the *Verilog* netlisting options dialog box, the above error message is displayed by Design Entry HDL when you netlist the design for digital simulation.

**Note:** This error message will not be displayed when you save or package the design because the shorted global signals are specified in the *Allowed Global Shorts* list.

For more information on setting up Verilog netlisting options and netlisting the design for digital simulation, see [Netlisting for Digital Simulation](#) on page 524.

### ***Supported Syntax in the Allowed Global Shorts List***

The syntax that is supported in the *Allowed Global Shorts* list is explained below using examples:

---

Example	Enter the following in Signal1 field of the Allowed Global Shorts list	Enter the following in Signal2 field of the Allowed Global Shorts list
■ To allow shorting of a scalar global signal <code>CLOCK\G</code> with another scalar global signal <code>DATA\G</code>	<code>CLOCK</code>	<code>DATA</code>
■ To allow shorting of any bit of a vectored global signal <code>CLOCK&lt;3..0&gt;\G</code> with any bit of another vectored global signal <code>DATA&lt;4..0&gt;\G</code>	<code>CLOCK</code>	<code>DATA</code>

## Allegro Design Entry HDL User Guide

### Creating a Schematic

Example	Enter the following in Signal1 field of the Allowed Global Shorts list	Enter the following in Signal2 field of the Allowed Global Shorts list
■ To allow shorting of two vectored global signals <code>INT&lt;3..0&gt;\G</code> and <code>DATA&lt;4..7&gt;\G</code> that have the same width	<code>INT&lt;3..0&gt;</code>	<code>DATA&lt;4..7&gt;</code>
	This means that Design Entry HDL allows shorting of bit <code>INT&lt;3&gt;</code> with bit <code>DATA&lt;4&gt;</code> , <code>INT&lt;2&gt;</code> with <code>DATA&lt;5&gt;</code> , <code>INT&lt;1&gt;</code> with <code>DATA&lt;6&gt;</code> and <code>INT&lt;0&gt;</code> with <code>DATA&lt;7&gt;</code> .	
	<b>Note:</b> Design Entry HDL displays an error message if the width of the shorted signals specified in the <i>Allowed Global Shorts</i> list is not the same.	
■ To allow shorting of the third bit of a vectored global signal <code>CLOCK&lt;3..0&gt;\G</code> with the fifth bit of another vectored global signal <code>DATA&lt;7..0&gt;\G</code>	<code>CLOCK&lt;3&gt;</code>	<code>DATA&lt;5&gt;</code>
	The shorting of any other bit of <code>CLOCK</code> with any other bit of <code>DATA</code> , except <code>CLOCK&lt;3&gt;</code> and <code>DATA&lt;5&gt;</code> is reported as a global signal short error.	
■ To allow shorting of any bit of a vectored global signal <code>DATA&lt;3..0&gt;\G</code> with a scalar global signal <code>VCC\G</code>	<code>DATA&lt;3..0&gt;</code>	<code>VCC</code>
	You can also use the following syntax to allow shorting of any bit of a vectored global signal <code>CLOCK&lt;3..0&gt;\G</code> with a scalar global signal <code>DATA\G</code>	
■ To allow shorting of a bit of global signal <code>CLOCK(3)\G</code> with another bit of a global signal <code>DATA[5]\G</code> .	<code>CLOCK&lt;3&gt;</code>	<code>DATA&lt;5&gt;</code>

## Allegro Design Entry HDL User Guide

### Creating a Schematic

Example	Enter the following in Signal1 field of the Allowed Global Shorts list	Enter the following in Signal2 field of the Allowed Global Shorts list
<p><b>Note:</b> If you set the MULTI_FORMAT directive to 'OFF', the signals with the above syntax are treated as scalar signals. To short the scalar global signal CLOCK(3)\G with the scalar global signal DATA[5]\G.</p>	CLOCK(3)	DATA[5]



Note the following when adding signals in the *Allowed Global Shorts* list:

- ❑ Shorting is allowed only between a pair of signals specified in the *Allowed Global Shorts* list and not between all the signals in the *Allowed Global Shorts* list.

For example, suppose that signal DATA\G is aliased to signal RESET\G and signal RESET\G is aliased to signal CONTROL\G. If you add signals DATA and RESET, and signals RESET and CONTROL in the *Allowed Global Shorts* list, it does not inherently mean that the shorting of signals DATA and CONTROL is allowed. Design Entry HDL will not display any error message for global signal short when you save the design. However, when you package the design, error messages can be seen in the Export Physical Progress window or in the px1.log file that the global signals DATA\G and CONTROL\G are shorted. To allow shorting of signals DATA\G and CONTROL\G, type DATA in the *Signal1* field and CONTROL in the *Signal2* field next to the DATA signal.
- ❑ Suppose that two vectored global signals INT<2..0>\G and CLOCK<2..0>\G are shorted. To allow shorting of the signals, you must specify INT<2..0> and CLOCK<2..0> in the *Allowed Global Shorts* list. If you specify the shorting of bits:
  - INT<2> with CLOCK<2>
  - INT<1> with CLOCK<1>
  - INT<0> with CLOCK<0>in the *Allowed Global Shorts* list, the global signal short error will not be suppressed. Also, if you specify the shorting as below in the *Allowed Global Shorts* list, the global signal short error will not be suppressed:
  - INT<2> with CLOCK<2>

- INT<1..0> with CLOCK<1..0>
- INT with CLOCK<1..0>
- If you set the MULTI\_FORMAT directive to 'OFF', and you have a scalar global signal CLOCK(3)\G aliased or synonmed to another scalar global signal INT[4]\G in your schematic, and you want to allow the signals to be shorted, add the signals in the *Allowed Global Shorts* list using the same signal syntax—CLOCK(3) and INT[4].

If you reset the MULTI\_FORMAT directive to 'ON', Design Entry HDL treats the signals CLOCK(3)\G and INT[4]\G as vectored signals. When you save or package the design, Design Entry HDL displays an error message that the two global signals CLOCK<3> and INT<4> are shorted. To correct this problem, modify the syntax of the signal names to CLOCK<3> and INT<4> in the *Allowed Global Shorts* list (when the MULTI\_FORMAT directive is 'ON').

## Unnamed Signals

If you have unnamed signals in your schematic, Design Entry HDL converts them to UNNAMED\_n, where n is a unique number.

**Note:** You can use unnamed nets with the NOT, CONCAT, and MERGE symbols only if you specify the size of the net with a SLASH symbol.

## Syntax

The syntax of the Unnamed net is as follows:

```
UNNAMED_11_CAPACITOR_I57_1 (UNNAMED_$Page_$PartName_$Path_$PinName)
```

When you backannotate the property values to the schematic, variables such as, \$page, \$part\_name, \$path, and \$pinname are substituted with the actual values. Therefore, the netname appears as UN\$11\$CAPACITOR\$I57\$1.

This is a shorter way of displaying an unnamed netname in Design Entry HDL. However, the long name will still be written to the netlist.

**Note:** It is not possible to configure the unnamed nets algorithm such that shorter unnamed nets are generated. The best way of configuring net names is to name the nets as per your convenience.

#### Netname length

When an unnamed net exceeds the net name length limit, the netname is truncated and the truncated value is written to the packager netlist. However, no warning is generated as the netnames are system generated and are automatically truncated while generation.

For user-defined long signal names, the information is written to the `PSTPROP.dat` file in the packaged directory whenever a net name is truncated. The `PNN` property defines the truncated physical net name generated by Packager.

#### Signal Concatenations

Signal concatenations allow you to merge a group of signals, ports, or signal aliases into a group. You can then route this group of signals to ports with a single wire.

You can create signal concatenations in Design Entry HDL schematics either textually or graphically (with the `CONCAT` symbols in the Standard library).

For more information, see [CONCAT symbols](#) (standard libraries).

#### To concatenate signals, ports, or signal aliases textually

- Attach a signal name to a wire and type the name of the two signals separated by a colon (:).

When the `MULTI_FORMAT` directive is 'ON', you can also use a comma (,) or an ampersand (&) to concatenate two signals. If you set the `MULTI_FORMAT` directive to 'OFF', ampersands and commas do not represent concatenation and have no special meaning in a signal name. For more information, see [Naming Signals](#) on page 195.

A concatenated signal of this type must be connected to a pin of the same width.

For example, if you have two signals `hi_addr(15:0)` and `lo_addr(15:0)` you can attach the following signal name to a wire.

```
hi_addr(15:0):lo_addr(15:0)
```

The wire now represents a 32-bit signal.

For vectored signals, VHDL and Verilog have leftmost and rightmost bits. For the 32-bit signal example above, the leftmost bit is `hi_addr(15)` and the rightmost bit is `lo_addr(0)`.

You can also create concatenations that contain more than two signals. For example, the following concatenation creates a 21-bit signal, assuming that `b` is a scalar signal. The leftmost bit is `a(0)`, and the rightmost bit is `c(0)`.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

```
a(0 to 9):b:c(9 downto 0)
```

In addition to regular signals, a concatenation can include ports from the entity for this architecture as well as aliases for other signals. The output of a concatenation can be named. You can also feed the result of one concatenation (or slice) into the input of another concatenation.

**Note:** You can use unnamed nets with CONCAT symbols only if you size the net using the SLASH body.

#### To concatenate signals, ports, or signal aliases graphically

1. In Design Entry HDL, choose *Component – Add*.

Part Information Manager appears.

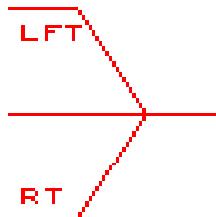
2. Select standard from the *Library* list in the search pane.
3. Select a CONCAT symbol from the *Cells* list.

For more information, see [CONCAT symbols](#) (standard libraries).

1. Click in the Design Entry HDL drawing area to place the symbol.
2. Close Part Information Manager.
3. Wire the signals you want to concatenate to the pins on the left side of the CONCAT symbol. You can also attach signal names directly to the pins.
4. Wire the right pin of the CONCAT symbol to the instance to which you want to connect the concatenation.

#### Example: Concatenating 3 Signals

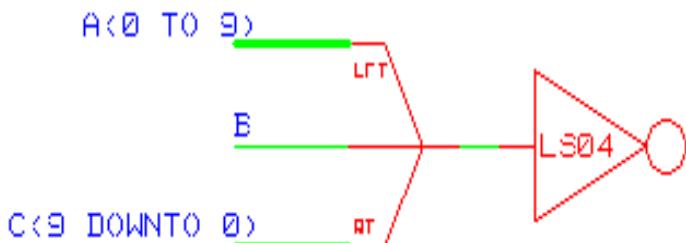
1. Add the CONCAT3 symbol from the Standard library.



- 2.** Wire the signals you want to concatenate to the left pins. In this example, the signals are the vectored signal  $a(0 \text{ to } 9)$ , the scalar signal  $b$ , and the vectored signal  $c(9 \text{ downto } 0)$ .



- 3.** Wire the right pin to the instance to which you want to connect the concatenation.



The output of the following concatenation is the 21-bit signal:

$a(0 \text{ to } 9) \& b \& c(9 \text{ downto } 0)$

You can cascade the output of one concatenation into the input of another. The output of the above CONCAT3 symbol can be connected to the input of another CONCAT symbol.

## Signal Replication

To construct a signal replication, use either textual or graphical concatenations, or a combination of both. For example, if you want a 10-bit wide concatenation of the signal GND, do one of the following:

- Name the signal.

GND : GND

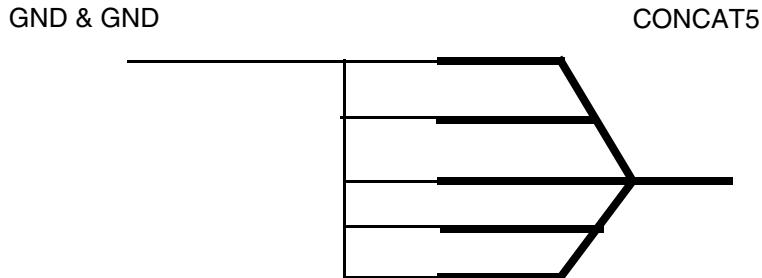
- Use the CONCAT10 symbol.

For more information, see [CONCAT symbols](#) (standard libraries).

## Allegro Design Entry HDL User Guide

### Creating a Schematic

- Use a combination of text and graphics.



The output of one concatenation can also be attached to the input of another concatenation. For example, if the output of the CONCAT5 symbol shown above is attached to the input of a CONCAT4, the resulting width of the signal will be 40 ( $2 \times 5 \times 4$ ).

## Merge Symbols

Use a MERGE symbol in the Standard library to combine several signals into a single vectored signal, or separate a vectored signal into a number of separate signals. With MERGE symbols, you can draw a vectored signal (a bus) as a single wire in parts of the drawing, and as several signals in other parts of the drawing.

There are nine MERGE symbols in the Standard library: 2 MERGE, 3 MERGE, 4 MERGE, 5 MERGE, 6 MERGE, 7 MERGE, 8 MERGE, 9 MERGE, and 10 MERGE1. Use 2 MERGE to merge two signals, 3 MERGE to merge three signals, and so on. Each MERGE symbol has four versions: two for merging signals and two for separating a vectored signal. Versions 1 and 2 have inputs on 0.2-inch centers and versions 3 and 4 have inputs on 0.1-inch centers.

You can use a MERGE symbol as a “demerger” by using a different version of the symbol. Versions 2 and 4 of each MERGE symbol are used to separate a vectored signal into several signals.

**Note:** Design Entry HDL processes designs faster if you use BIT TAP symbols instead of MERGE symbols to slice signals.

For more information on BIT TAP symbols, see [BITTAP](#).

You can also create new MERGE symbols by copying the `HDL_CONCAT=TRUE` property from one of the MERGE symbols in the Standard library to the new symbol.

## Using Merge Symbols

### **To use a MERGE symbol to merge signals**

1. In Design Entry HDL, choose *Component – Add*.

Part Information Manager appears.

2. Select standard from the *Library* list in the search pane.

3. Select a MERGE symbol from the Cells list. The component is attached to your cursor.

To merge two signals, select the 2 MERGE symbol. To merge three signals, select the 3 MERGE symbol, and so on.

4. Move the cursor to the Design Entry HDL drawing area, but do not click in it.

5. Select version 1 or 3 of the MERGE symbol.

To select the Version, right-click and choose *Version*. The next version of the component is displayed. Repeat this step until the version you want is displayed.

If you want Design Entry HDL to only display those versions of the component that match its PACK\_TYPE, you can set the List\_Valid\_Versions\_Metadata directive to ON in the .cpm file.

**Note:** Versions 1 and 3 of a MERGE symbol are for merging signals; versions 2 and 4 are for separating vectored signals.

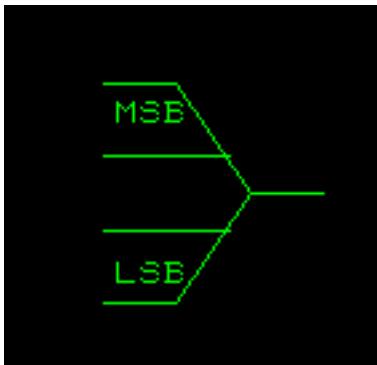
6. Click in the Design Entry HDL drawing area to place the symbol.

7. Close Part Information Manager.

8. Connect the signals you want to merge to the left pins of the MERGE symbol.

### Example

Version 3 of the 4 MERGE symbol below is used to merge four signals:



### To use a MERGE symbol to separate a vectored signal

1. In Design Entry HDL, choose *Component – Add*.

Part Information Manager appears.

2. Select *standard* from the *Library* list in the search pane.

3. Select a MERGE symbol from the Cells list. The component is attached to your cursor.

To separate a vectored signal into two signals, select the 2 MERGE symbol. To separate a vectored signal into three signals, select the 3 MERGE symbol, and so on.

4. Move the cursor to the Design Entry HDL drawing area, but do not click in it.

5. Select version 2 or 4 of the MERGE symbol.

To select the Version, right-click and choose *Version*. The next version of the component is displayed. Repeat this step until the version you want is displayed.

If you want Design Entry HDL to only display those versions of the component that match its PACK\_TYPE, you can set the List\_Valid\_Versions\_Metadata directive to ON in the .cpm file.

**Note:** Versions 2 and 4 are for separating vectored signals; versions 1 and 3 are for merging signals.

6. Click in the Design Entry HDL drawing area to place the symbol.

7. Close Part Information Manager.

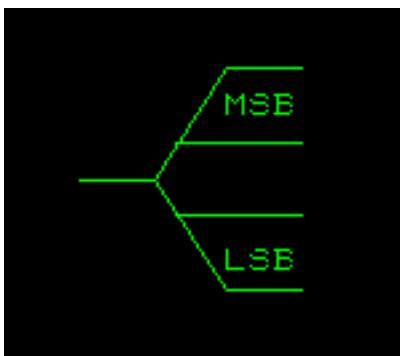
8. Connect the signal you want to separate to the left pin of the MERGE symbol.

#### 9. Name the output signals.

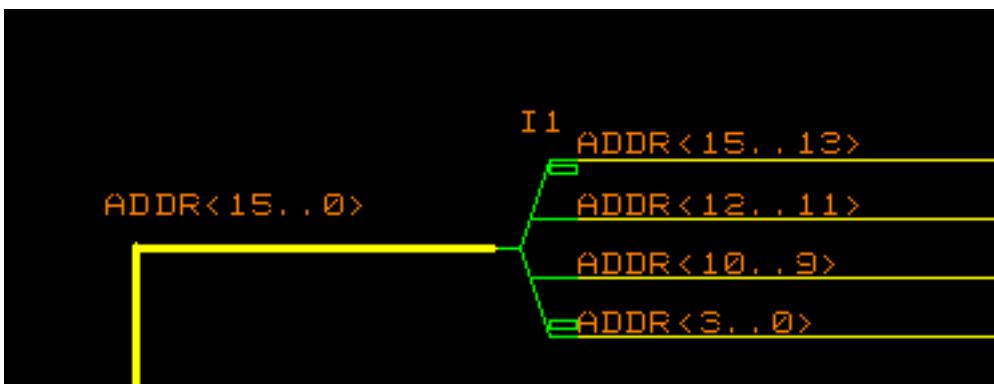
For example, to separate a bus `addr<15..0>` into four signals, name the output signals `addr<8..0>`, `addr<10..9>`, `addr<12..11>`, and `addr<15..13>`.

#### **Example**

Version 4 of the 4 MERGE symbol below is used to separate a vectored signal into four signals.



The following example shows how the 4 MERGE symbol given above is used to separate a vectored signal into many signals.



#### **Rules for Using MERGE symbols**

- The width of the output net must match the sum of the widths of the input nets. The following assumptions apply:
  - If the output is not named (that is, its width is not specified), the width of the output net is assumed to be the sum of the widths of the input signals.

- ❑ If some input nets are not named (that is, their width is not specified), each unnamed input net is assumed to be 1-bit wide.
- If the input nets are unnamed and the output net is also unnamed, you must specify either the widths of all the input nets or the width of the output net. You can use a SLASH symbol to specify the width of an unnamed signal. For more information on the SLASH symbol, see [Specifying the Size of Nets](#) on page 208.
- If one or more input pins are unconnected, the width of the output net must be greater than or equal to the sum of the input widths, assuming a width of 1 for each unconnected pin.
- The name of the output pin must be the highest alphanumeric value of all the pins on the symbol. For example, if the input pins are named AA, DD, and FF, the output pin cannot be named BB.
- Versions 3 and 4 of 2 MERGE, 4 MERGE, 6 MERGE, 8 MERGE, and 10 MERGE have off-grid outputs. Do not use them if you need to have all pins on the grid.

## Signal Slices (Bit and Part Selects)

In VHDL, a slice is a way to reference specific bits of a vectored signal, port, or signal alias. In Verilog, slices are called “bits” and “part selects.”

You can create slices in Design Entry HDL schematics either textually or graphically (using the SLICE symbol in the Standard library). You can also use the `tap` command in the Design Entry HDL Console window to create slices.

### To create a slice textually

- Specify the bits you want to slice in the signal name.

For example, if you have a signal `addr<31:0>`, and you want to reference its leftmost bit, attach the signal name `addr<31>` to a wire or a pin. If you want to reference the leftmost three bits, attach the signal name `addr<31:29>` to a wire or a pin.

**Note:** If you set the MULTI\_FORMAT directive to ‘OFF’, you must be careful about the syntax you use to specify the width of the signal. See [Naming Signals](#) on page 195 for more information.

### To create a slice graphically

1. In Design Entry HDL, choose *Component – Add*.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

Part Information Manager appears.

2. Select standard from the *Library* list in the search pane.
3. Select SLICE from the *Cells* list.
4. Click in the Design Entry HDL drawing area to place the symbol.

If you want to create several slices, continue clicking until all the slice symbols are placed.

If you placed a single SLICE symbol,

- a. From the *Text* menu, choose *Attributes*.
- b. Click on the edge of the straight part of the SLICE symbol to display the *Attributes* dialog box.
- c. Change the value of the *BN* property to the bit number you want to tap.
- d. Click *OK* to save the changes and close the *Attributes* dialog box.

**Note:** The value of the *BN* property can be a range specification to tap multiple bits.

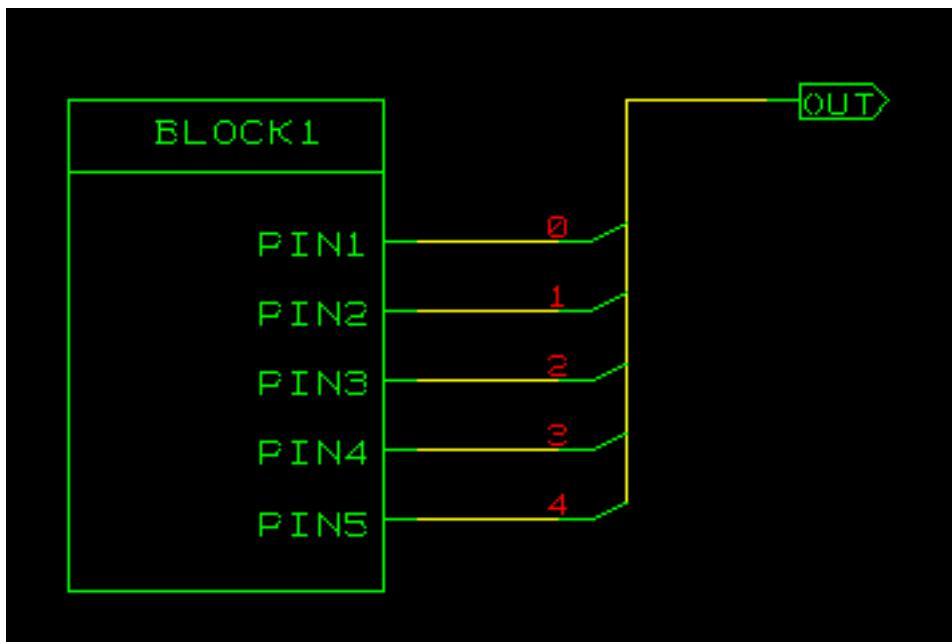
If you added more than one SLICE symbol,

- a. In Design Entry HDL, choose *Wire – Bus Tap Values*.  
The *Bus Tap Range* dialog box appears.
  - b. Specify the Most Significant Bit (*MSB*), Least Significant Bit (*LSB*), and the increment between them.
  - c. Click *OK*.
  - d. Draw a line across the SLICE symbols you placed on the schematic. The slices get numbered from the *MSB* to the *LSB*.
5. Wire the bent part of the SLICE symbols to a vectored signal and the straight part to the pin to which you want to connect the SLICE.

Typically, you do not name the wire on the straight part of the SLICE because the SLICE provides the name for the wire.

### Example

In this example, five SLICE symbols were placed on the schematic, and the *Wire – Bustap Values* menu option in Design Entry HDL was used to number the BN property on each SLICE.



### To slice multiple bits

1. Add a SLICE symbol from the Standard library.
2. From the *Text* menu, choose *Attributes*.
3. Click on the edge of the straight part of the SLICE symbol to display the *Attributes* dialog box.
4. Change the value of the BN property to a range specification. For example, you can set the value to 1 to 10 or 10:1 or size:1.

### Rules for Using SLICE Symbols

- You cannot slice a concatenation of signals.
- You can set the value of the BN property on a SLICE to a range specification, for example, BN=1 to 10.

## Setting the Verilog Logic Type for Ports and Signals

The default Verilog logic type for ports and signals is `WIRE`. You can change the default Verilog logic type for a project. Examples of other Verilog types for ports and signals include `WAND` and `WOR`.

**Note:** Verilog does not support the use of abstract data types such as floating points and integers.

The Verilog logic type is determined by the `VLOG_NET_TYPE` property. With this property, you can choose the Verilog logic type at the following levels:

- [Setting the Verilog Logic Type for All Ports and Signals in All Drawings of a Project](#)
- [Setting the Verilog Logic Type for All Ports and Signals in a Drawing](#)
- [Setting the Verilog Logic Type for a Specific Port](#)
- [Setting the Verilog Logic Type for a Specific Signal](#)

The Verilog logic type you select for an individual port or signal has precedence over the logic type you specify for the drawing, which in turn has precedence over the logic type you set for the project.

**Note:** Verilog allows a signal of type `WIRE` to be connected to ports on instances of different types.

### **Setting the Verilog Logic Type for All Ports and Signals in All Drawings of a Project**

You can specify the default Verilog logic type for all the ports and signals in all drawings of a project. You can change these defaults for a project.

You can override the default Verilog logic type specified for the project by specifying the Verilog logic type for all ports and signals in a specific drawing. You can further override the Verilog logic type specified for a drawing by specifying the Verilog logic type for individual ports and signals.

#### ***To set the default Verilog logic type for a project***

1. In Design Entry HDL, choose *Tools – Options*.

The *Design Entry HDL Options* dialog box appears.

2. Select the *Output* tab.

Ensure that the *Create Netlist* check box is selected.

3. Click the *Options* button next to the *Verilog* check box.

The *Verilog Netlist* dialog box appears.

4. In the *Default Net Type* text box, type the Verilog logic type you want to use for all the ports and signals in the design. The default type is `WIRE`. Examples of other Verilog types for ports and signals include `WAND` and `WOR`.

5. Click *OK* to save the changes.

6. Click *OK* to close the *Design Entry HDL Options* dialog box.

### Setting the Verilog Logic Type for All Ports and Signals in a Drawing

You can override the default Verilog logic type specified for the project by specifying the Verilog logic type for all ports and signals in a specific drawing. Specify the Verilog logic type for all ports and signals in a drawing by attaching the `VLOG_NET_TYPE` property to a `VERILOG_DECS` symbol.

For more information on the `VERILOG_DECS` symbol, see [VHDL DECS and VERILOG DECS Symbols](#).

#### ***To set the Verilog logic type for all ports and signals in a drawing***

1. Add a `VERILOG_DECS` symbol from the Standard library to the first page of the drawing.
2. Choose *Text – Attributes* and click on the `VERILOG_DECS` symbol to display the *Attributes* dialog box.
3. Click *Add*.
4. Type `VLOG_NET_TYPE` in the *Name* text box and type the Verilog logic type in the *Value* text box.

The value of the `VLOG_NET_TYPE` property can be `WIRE`, `WAND`, `WOR`, or any other legal Verilog type.

5. Click *OK* to save the changes and close the *Attributes* dialog box.

**Note:** The `VLOG_NET_TYPE` property on an individual port or signal has precedence over the `VLOG_NET_TYPE` property on a `VERILOG_DECS` symbol, which in turn has precedence over the Verilog logic type specified for the project.

#### Setting the Verilog Logic Type for a Specific Port

You can set the Verilog logic type for each port individually or as a default for all the ports of a symbol. The type is declared with the `VLOG_NET_TYPE` property. Because Verilog does not support abstract data types, the Verilog logic type of ports cannot be an abstract data type.

##### ***To declare the Verilog logic type of a port***

1. In Design Entry HDL, choose *Text – Attributes*.
2. Click on a pin on the port to display the *Attributes* dialog box.
3. Click *Add*.
4. Type `VLOG_NET_TYPE` in the *Name* text box and type the Verilog logic type in the *Value* text box.  
The value of the `VLOG_NET_TYPE` property can be `WIRE`, `WAND`, `WOR`, or any other legal Verilog type.
5. Click *OK* to save the changes and close the *Attributes* dialog box.

**Note:** The `VLOG_NET_TYPE` property on an individual port has precedence over the `VLOG_NET_TYPE` property on a `VERILOG_DECS` symbol, which in turn has precedence over the Verilog logic type specified for the project.

#### Setting the Verilog Logic Type for a Specific Signal

You can set the Verilog logic type for each signal. The type is declared with the `VLOG_NET_TYPE` property.

##### ***To set the Verilog logic type for a signal***

1. In Design Entry HDL, choose *Text – Attributes*.
2. Click on the signal to display the *Attributes* dialog box.
3. Click *Add*.
4. Type `VLOG_NET_TYPE` in the *Name* text box and type the Verilog logic type in the *Value* text box.  
The value of the `VLOG_NET_TYPE` property can be `WIRE`, `WAND`, `WOR`, or any other legal Verilog type.
5. Click *OK* to save the changes and close the *Attributes* dialog box.

**Note:** The `VLOG_NET_TYPE` property on an individual signal has precedence over the `VLOG_NET_TYPE` property on a `VERILOG_DECS` symbol, which in turn has precedence over the Verilog logic type specified for the project.

#### **To assign a type Supply 0 or Supply 1 to power and ground symbols**

1. In Design Entry HDL, choose *Text – Attributes*.
2. Click on the symbol or on a pin of the symbol to display the *Attributes* dialog box.
3. Click *Add*.
4. Type `VLOG_NET_TYPE` in the *Name* text box and type the Verilog logic type in the *Value* text box.  
The value of the `VLOG_NET_TYPE` property can be `WIRE`, `WAND`, `WOR`, or any other legal Verilog type.
5. Click *OK* to save the changes and close the *Attributes* dialog box.

For VHDL, the `VHDL_INIT` property should be attached to the power or ground symbol or to its pin with a value of `1` or `0` respectively. This results in the power signal getting assigned that value in the VHDL created by Design Entry HDL.

## Setting the VHDL Logic Type for Ports and Signals

The default VHDL logic type for all ports and signals in Design Entry HDL schematics is `STD_LOGIC` (for scalar ports and signals) and `STD_LOGIC_VECTOR` (for vectored ports and signals). You can change these defaults for a project.

Examples of other VHDL logic types you can use for ports and signals include `BIT` and `BIT_VECTOR`. VHDL also lets you declare a signal or port as an abstract data type such as a floating point number or integer. See [Abstract Data Types in VHDL](#) on page 241 for more information about abstract data types and the restrictions on their use.

The VHDL type of a port or signal is determined by the `VHDL_SCALAR_TYPE` and `VHDL_VECTOR_TYPE` properties. With these properties, you can choose the VHDL logic type for ports and signals at the following levels:

- [Setting the VHDL Logic Type for All Ports and Signals in All Drawings of a Project](#)
- [Setting the VHDL Logic Type for All Ports and Signals in a Drawing](#)
- [Setting the VHDL Logic Type for a Specific Port](#)
- [Setting the VHDL Logic Type for a Specific Signal](#)

The VHDL logic type you select for an individual port or signal has precedence over the logic type you specify for a drawing, which in turn has precedence over the logic type you set for the project.

**Note:** You can also set the initial value of a signal with the `VHDL_INIT` property.

For more information, see [Setting the Initial Value of a Signal](#) on page 206.

### Setting the VHDL Logic Type for All Ports and Signals in All Drawings of a Project

You can specify the default VHDL logic type for all the ports and signals in all drawings of a project. You can change these defaults for a project.

You can override the default VHDL logic type specified for the project by specifying the VHDL logic type for all ports and signals in a specific drawing. You can further override the VHDL logic type specified for a drawing by specifying the VHDL logic type for individual ports and signals.

#### **To set the VHDL logic type for a project**

1. In Design Entry HDL, choose *Tools – Options*.

The *Design Entry HDL Options* dialog box appears.

2. Select the *Output* tab.

Ensure that the *Create Netlist* check box is selected.

3. Click the *Options* button next to the *VHDL* check box.

The *VHDL Netlist* dialog box appears.

4. In the *Vector Type* text box, enter the VHDL logic type you want to use for the vectored ports and signals in the design. The default type is `STD_LOGIC_VECTOR`.

5. In the *Scalar Type* text box, enter the VHDL logic type you want to use for the scalar ports and signals in the design. The default type is `STD_LOGIC`.

6. Click *OK* to save the changes.

7. Click *OK* to close the *Design Entry HDL Options* dialog box.

### Setting the VHDL Logic Type for All Ports and Signals in a Drawing

You can override the default VHDL logic type specified for the project by specifying the VHDL logic type for all ports and signals in a specific drawing. Specify the VHDL logic type for all the

ports and signals in a drawing by attaching the VHDL\_SCALAR\_TYPE and VHDL\_VECTOR\_TYPE properties to a VHDL\_DECS symbol.

For more information on the VERILOG\_DECS symbol, see [VHDL DECS and VERILOG DECS Symbols](#).

#### **To set the VHDL type for a drawing**

1. Add a VHDL\_DECS symbol to the first page of the schematic.
2. Choose *Text – Attributes*.
3. Click on the VHDL\_DECS symbol to display the *Attributes* dialog box.
4. Add the VHDL\_SCALAR\_TYPE and VHDL\_VECTOR\_TYPE properties as below:
  - Add the VHDL\_SCALAR\_TYPE property if the drawing has only scalar ports.
  - Add the VHDL\_VECTOR\_TYPE property if the drawing has only vectored ports.
  - Add the VHDL\_SCALAR\_TYPE and VHDL\_VECTOR\_TYPE properties if the drawing has both scalar and vectored ports.

The value of the VHDL\_SCALAR\_TYPE property can be STD\_LOGIC, BIT, or any other legal VHDL scalar type. The value of the VHDL\_VECTOR\_TYPE property can be STD\_LOGIC\_VECTOR, BIT\_VECTOR, or any other legal VHDL vector type.

5. Click *OK* to save the changes and to close the *Attributes* dialog box.

**Note:** The VHDL\_SCALAR\_TYPE or VHDL\_VECTOR\_TYPE property on an individual port or signal has precedence over the property on a VHDL\_DECS symbol, which in turn has precedence over the VHDL logic type specified for the project.

#### **Setting the VHDL Logic Type for a Specific Port**

You can set the VHDL logic type for each port individually or as a default for all the ports of a symbol. The type is declared with the VHDL\_SCALAR\_TYPE and VHDL\_VECTOR\_TYPE properties.

The VHDL logic type can be an abstract data type. See [Abstract Data Types in VHDL](#) on page 241 for more information about abstract data types and the restrictions on their use.

**Note:** The VHDL\_SCALAR\_TYPE or VHDL\_VECTOR\_TYPE property on an individual port has precedence over the property on a VHDL\_DECS symbol, which in turn has precedence over the VHDL logic type specified for the project.

#### **To declare the VHDL logic type of a port**

1. In Design Entry HDL, choose *Text – Attributes*.
2. Click on a pin on the port to display the *Attributes* dialog box.
3. Add the `VHDL_SCALAR_TYPE` and `VHDL_VECTOR_TYPE` properties as below:
  - If the symbol has only scalar ports, add the `VHDL_SCALAR_TYPE` property.
  - If the symbol has only vectored ports, add the `VHDL_VECTOR_TYPE` property.
  - If the symbol has both vectored and scalar ports, add both the `VHDL_SCALAR_TYPE` and `VHDL_VECTOR_TYPE` properties.

The value of the `VHDL_SCALAR_TYPE` property can be `STD_LOGIC`, `BIT`, or any other legal VHDL scalar type. The value of the `VHDL_VECTOR_TYPE` property can be `STD_LOGIC_VECTOR`, `BIT_VECTOR`, or any other legal VHDL vector type.

4. Click *OK* to save the changes and to close the *Attributes* dialog box.

**Note:** The `VHDL_SCALAR_TYPE` or `VHDL_VECTOR_TYPE` properties attached to pins of ports have precedence over the `VHDL_SCALAR_TYPE` or `VHDL_VECTOR_TYPE` properties attached to the origin of the symbol.

#### **Setting the VHDL Logic Type for a Specific Signal**

#### **To set the VHDL logic type for a signal**

1. In Design Entry HDL, choose *Text – Attributes*.
2. Click on the signal to display the *Attributes* dialog box.
3. Add the `VHDL_SCALAR_TYPE` and `VHDL_VECTOR_TYPE` properties as below:
  - If the signal is scalar, add the `VHDL_SCALAR_TYPE` property.
  - If the signal is vectored (a bus), add the `VHDL_VECTOR_TYPE` property.

The value of the `VHDL_SCALAR_TYPE` property can be `STD_LOGIC`, `BIT`, or any other legal VHDL scalar type. The value of the `VHDL_VECTOR_TYPE` property can be `STD_LOGIC_VECTOR`, `BIT_VECTOR`, or any other legal VHDL vector type.

4. Click *OK* to save the changes and to close the *Attributes* dialog box.

## Specifying Ranges for Ports, Signals and Aliases

Range specifications are used in VHDL and Verilog to declare the widths of vectored ports, signals, and aliases and to create slices of these objects. When you create Design Entry HDL schematics, you can use VHDL or Verilog syntax to specify a range.

Follow these conventions for range specifications:

- Enclose range specifications in angle brackets <>. You can also use ( ) or [ ].  
**Note:** You cannot use ( ) or [ ] if you set the MULTI\_FORMAT directive to 'OFF'.
- Specify a descending range with a colon (:), ellipsis (..), or either of the strings `downto` or `DOWNTO`.  
**Note:** If the left and right bounds of a range specification are constant integers and the right integer is greater than the left, the colon (:) is interpreted as specifying an ascending range.
- Specify ascending ranges with a colon (:), ellipses (..), or either of the strings `to` or `TO`.

### Examples

The following are examples of legal range specifications in Design Entry HDL schematics:

<10 downto 0>	11-bit descending range
<10..0>	11-bit descending range
<0 to 10>	11-bit ascending range
<10:0>	Colon (:) is the same as <code>downto</code>
<0:10>	Colon (:) also works like <code>to</code>
<size-1:0>	Parameterized descending range
<0 to size-1>	Parameterized ascending range

The following examples are illegal in Design Entry HDL schematics

:

<10 to 0>	Illegal. <code>to</code> must be an ascending range.
<0 downto 10>	Illegal. <code>downto</code> must be a descending range



If you specify the port range in a user-defined package, you should ensure that the width of the ports in the symbol match with the port range specified in the package.

## Unconstrained Ranges for Ports, Signals, and Aliases

While VHDL lets you create arrayed objects with unconstrained bounds, Design Entry HDL does not support unconstrained ports, signals, or aliases. If you want an unconstrained range for an object, use a parameterized range for the object.

### Example

For example, in VHDL you may have the following port (a vectored port with an unconstrained range) on an entity:

```
entity CPU is
  port (
    io_addr: out std_logic_vector;
    io_busy: in std_logic;
  );
end cpu;
```

For Design Entry HDL, you can change the above entity declaration to the following:

```
entity CPU is
  generic (size: positive);
  port (
    io_addr: out std_logic_vector (size - 1 downto 0);
    io_busy: in std_logic;
  );
end cpu;
```

Architectures that instantiate this CPU entity can generate the value for the SIZE property by using the pre-defined LENGTH attribute for the signal attached to the io\_addr port.

Similarly, if you want an unconstrained range for a signal, ADDR, in a schematic, declare it as a parameterized range as below:

```
ADDR (size-1:0)
```

## Resolved Types and Resolution Functions

In VHDL, if a signal has multiple drivers, you must define a resolution function to resolve the signal conflict. You can declare a resolved signal in two ways in VHDL—

- The signal declaration can refer to a resolved type.

- The signal declaration can specify a resolution function and an unresolved type.

However, in Design Entry HDL, only the first option is possible. If you want to use a resolved signal, you must reference an existing resolved type when you declare the signal type.

## Type Conversion

If you need to connect a signal of one type to a port of another type, use a type conversion function that will result in the correct VHDL output. There are some restrictions on using type conversion functions with abstract data types.

**Note:** Verilog does not use type conversion functions; in Verilog, you can connect a signal of the type `wire` to ports of other types. While generating Verilog text, Design Entry HDL ignores the `VHDL_IN_CONVERT` and `VHDL_OUT_CONVERT` properties that are used on the schematic.

### To specify type conversion functions in a schematic

- If the port is an input, attach this property to the pin:

`VHDL_IN_CONVERT = function_name`

- If the port is an output or a buffer, attach this property to the pin:

`VHDL_OUT_CONVERT = function_name`

- If the port is bi-directional, attach these properties to the pin:

`VHDL_IN_CONVERT = function_name1`

`VHDL_OUT_CONVERT = function_name2`

For example, if a pin `IO1` in a schematic is connected to a signal `INT` and has the following properties:

`VHDL_IN_CONVERT=MYLIB.PKG.FIN`

`VHDL_OUT_CONVERT=MYLIB.PKG.FOUT`

the VHDL text generated for the instance of the symbol has the following in its port map clause:

`MYLIB.PKG.FOUT(IO1) => MYLIB.PKG.FIN(INT)`

**Note:** Attach the `VHDL_IN_CONVERT` and `VHDL_OUT_CONVERT` properties to pins of component instances in your schematic, not to the `IMPORT` or `EXPORT` port symbols.

#### Restrictions on Using Type Conversion Functions

If you have an object (for example, a signal) of an abstract data type connected to an object (for example, a port) of a bit-oriented type, do not specify a type conversion function using the `VHDL_IN_CONVERT` and `VHDL_OUT_CONVERT` properties. Instead, create an entity that has two ports, one of each type. This entity performs the type conversion.

However, if both objects are non-vectorized types, or both objects are vectored types and have the same number of elements, you can use the `VHDL_IN_CONVERT` and `VHDL_OUT_CONVERT` properties.

#### Abstract Data Types in VHDL

VHDL supports a variety of data types. For example, it is possible in VHDL to have a signal or a port declared as an abstract data type. Examples of an abstract data type include a floating-point number, an integer, or a record made up of a set of data types. These abstract types are different from types such as `BIT`, `BIT_VECTOR`, `STD_LOGIC`, and `STD_LOGIC_VECTOR` because their correspondence to the hardware implementation is not explicitly stated.

Design Entry HDL supports the use of abstract data types through the `VHDL_SCALAR_TYPE` and `VHDL_VECTOR_TYPE` properties on schematics and by referencing these types within entity declarations.

**Note:** Verilog does not support the use of abstract data types. If you are using Verilog, do not use abstract data types.

#### Restrictions on the Use of Abstract Data Types

A signal of one type can be connected to a port of another type if a type conversion function is specified with the `VHDL_IN_CONVERT` and `VHDL_OUT_CONVERT` properties. However, if you have an object (for example, a signal) that is an abstract data type connected to another object (for example, a port on a component) that is a bit-oriented type, you should not use a type conversion function. Instead, create an entity that has two ports, one of each type. This entity performs the type conversion.

However, the `VHDL_IN_CONVERT` and `VHDL_OUT_CONVERT` properties work correctly if both objects are non-vectorized types or both objects are vectored types and have the same number of elements.

## Using Iterated Instances

If you use iterated instances, you can replicate parts without building them as parameterized models. When you create an iterated instance, Design Entry HDL automatically expands the iterated instance into multiple instances when it generates VHDL and Verilog representations of the schematic. The number of instances you want to generate must always be a constant; it cannot be a parameter.

You cannot have iterated instances of parts that have parameterized port widths. All signals attached to an iterated instance must have a fixed width. If the width of a signal matches the width of the pin, every generated instance has that signal attached. If the width of a signal is greater than the width of the pin to which it is attached, Design Entry HDL automatically attaches the correct bits of the signal to the correct pins.

To use the iterated instance feature, add a `PATH` property to an instance. The `PATH` property specifies the number of instances you want. For example, to generate 16 instances of a part, add the following property to the part:

```
PATH = I3<15..0>
```

The generated instance labels in Verilog and VHDL are:

```
I3_GEN_15  
I3_GEN_14  
I3_GEN_13  
...  
I3_GEN_0
```

When the `MULTI_FORMAT` directive is ON', you can also set the `PATH` properties in one of the following three ways:

- `PATH = I3(15..0)`
- `PATH = I3[15..0]`
- `PATH = I3[15:0]`

### ***Example***

The design `MYPART` has a net `A(X)` and a `DEFINE` body with the following properties:

```
X_FIRST = 1, X_STEP = 8, and X_SIZE=32
```

The value of `X` in the first instance is 1, in the second 9, in the third 17, and in the fourth 25. Therefore, when `MYPART` is used, it is instantiated four times.

## Saving a Design

Design Entry HDL displays a \* sign in the title bar to show that the current page needs to be saved.

### Note:

When you save the design, Design Entry HDL writes the current design on the disk. The \* sign disappears from the title bar.

Before saving the design, Design Entry HDL automatically runs all the checks that are normally run when you choose *Tools – Check*. Design Entry HDL also checks for connectivity errors on other pages in the design.

To save an existing drawing:

- Choose *File – Save*.

To save an existing drawing with a new name:

1. Choose *File – Save As*.
2. In the *View Save As* dialog box that appears, highlight the existing drawing name in the *Cell* box, and type a new drawing name.
3. Click *Save*.

To save a new drawing:

1. Choose *File – Save As*.
2. In the *View Save As* dialog box that appears, type a drawing name in the *Cell* box.

Design Entry HDL appends .SCH.1.1 to the file name that you specify. For example, if the file name you enter is MEMORY, Design Entry HDL names the drawing MEMORY.SCH.1.1. (Design Entry HDL assumes version 1 and page 1 of the drawing.)

3. Click *Save*.

If any errors are found on the current page, Design Entry HDL reports them.

Design Entry HDL also checks for connectivity errors on all the pages in the design and reports errors, if any.

### ***To view the errors, open the Markers control window***

- Choose *Tools – Markers*.

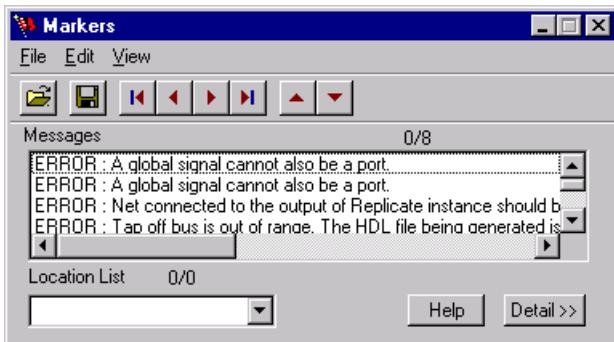
## Allegro Design Entry HDL User Guide

### Creating a Schematic

or

- Click the Markers Controls button in the *Markers* toolbar.

Click Yes to view the HDL-Direct errors in the Markers window.



Click on an error to find it on the schematic. Design Entry HDL highlights the area where the error occurred.

When you save a schematic, Design Entry HDL does not copy over the parts used in the schematic to the local database. Design Entry HDL is a by-reference editor that references all parts in the schematic from various libraries that reside at the reference or local area.

## Working With Existing Designs

This section describes the following:

- [Opening a Drawing](#) on page 244
- [Recovering a Drawing](#) on page 245
- [Reverting to the Previous Saved Version of a Drawing](#) on page 245

### Opening a Drawing

1. Choose *File – Open*.
2. Select a cell you want to open.
3. Click on the cell to expand. The expanded cell displays all the views in it.
4. Select the view you want to open and click *Open*. You can also double-click on the schematic view (sch\_n) to view the pages and double-click on a page to open it in Design Entry HDL.

## Allegro Design Entry HDL User Guide

### Creating a Schematic

---

Design Entry HDL opens schematic and symbol files. Other views (Verilog and VHDL) are opened based on the editor registered for these views in Project Manager.

**Note:** Plus signs (+) indicate there are lower level listings. The + changes to a minus (-) when there are no more lower level listings. For schematics, you choose the page number of the schematic that you want to open.

You can also start Design Entry HDL from the Project Manager to access your drawings. See the [Project Manager User Guide](#) for instructions on starting Design Entry HDL from that entry point. The steps shown here for opening a drawing still apply.

## Recovering a Drawing

To recover drawings that were being edited when Design Entry HDL or your system crashes, do the following:

1. Choose *File – Recover*.
2. In the file browser that appears, navigate to the `./temp/xxxnedtmp` directory where Design Entry HDL places undo log files.

Every time you start Design Entry HDL, a temporary directory is created in the `<project_directory>/temp` directory. An undo log file for each drawing is stored in this directory. By default, `xxnedtmp` is the name of the temporary directory. If the `xxnedtmp` directory already exists, a `xxnedtmp1` directory is created. If these two directories already exist, `xxnedtmp2` is created, and so on. The name of the undo log file for the first drawing edited is `undo1.log`. The second drawing's undo log file is `undo2.log`, and so on.

3. Select the undo log file for the drawing you want to recover and click *Open*.  
Design Entry HDL gives the recovered drawing a unique name (for example, `RECOVER1.SCH.1.1`). The recovered drawing is only saved in memory, not on disk.
4. Choose *File – Save As* to save the drawing with a different name.

## Reverting to the Previous Saved Version of a Drawing

- Choose *File – Revert*.

Design Entry HDL displays the last saved version of the current drawing.

# **Allegro Design Entry HDL User Guide**

## Creating a Schematic

---

---

# Working with Wires

---

This section describes the procedures for working with wires in Design Entry HDL.

## About Signals and Connectivity

It is important to identify each of the primary inputs and outputs of the circuit and other important signals with a name. Signal names identify signals on the drawing. Signals with the same name are interpreted as the same signal. This is how Design Entry HDL connects signals across multiple pages of a drawing.

Signal names also let you enter additional information:

- |                 |   |
|-----------------|---|
| Assertion level | Describes the active state of the signal when asserted. By convention, a signal is active high for positive logic and active low for negative logic. An asterisk * and '-' represents active low. Two signals with the same name but different assertion levels are not the same signal.<br><br>For example, Design Entry HDL treats A* and – A as low-asserted pins. |
| Signal bits     | Signals can have a single bit or multiple bits. The bit portion of the signal name is called the bit subscript and is always enclosed in angle brackets, like this: <3..0>.<br><br>A signal without a signal bit is called a scalar. A signal with a bit subscript can be a scalar or a vectored signal.  |
| Properties      | Describe characteristics of the signal, control how the compiler interprets the signal, or conveys physical information.  |

Design Entry HDL handles signal names as properties. For example, attaching a signal called BUS ENABLE to a wire is equivalent to attaching a property SIG\_NAME=BUS ENABLE to that wire. In the symbol, the SIG\_NAME properties are understood as PIN\_NAME properties and can only be attached to pin connections.

The names you attach to the signals in the drawing are written into the connectivity file that Design Entry HDL creates when you save the drawing.

## About Bus Taps

Design Entry HDL provides several different bus taps for use in schematics. These bus taps are in the Standard Library.

The most convenient way to tap buses is to choose *Wire – Bus Tap*. You can choose *Tools – Options* and specify the tap to use in the Graphics tab.

You can use other tap symbols (`tap.body`, `bustap.body`, `msbtap.body`, and `lsbtap.body`) or create your own tap symbol.

**Note:** For guidelines for creating tap symbols, see the [Guidelines for Creating Tap Symbols](#) in the *Using the Standard Library Symbols* chapter of *Allegro Design Entry HDL Reference Guide*.

When you add a tap using *Wire – Bus Tap*, the BN property is added to the bus tap. Design Entry HDL understands that if you have a bus named `<20..5>` and you attach a tap to it with the BN property set to 7, then you are tapping bit 7, not bit 12.

## About Bus Names

Design Entry HDL supports several bit numbering syntax conventions. Because the signal name syntax affects library parts and many design tools, a single site must use the same syntax system wide. Bit subscripts can use two dots (..) or a colon. Bit ordering can be most significant bit to least significant bit (msb to lsb) or vice versa.

Bus Name	Associated Signal Name
<code>A&lt;3..0&gt;</code>	<code>A&lt;3&gt;, A&lt;2&gt;, A&lt;1&gt;, A&lt;0&gt;</code>
<code>A&lt;0..3&gt;</code>	<code>A&lt;0&gt;, A&lt;1&gt;, A&lt;2&gt;, A&lt;3&gt;</code>
<code>A&lt;0&gt;</code>	<code>A&lt;0&gt;</code>
<code>A&lt;7..0:2&gt;</code>	<code>A&lt;6&gt;, A&lt;4&gt;, A&lt;2&gt;, A&lt;0&gt;</code>

## Drawing a Wire Manually

To draw a wire without naming it, do the following:

1. Select one of the following options:
  - Choose *Wire – Draw*.
  - Right-click on the component where you want to add the wire, and choose *Add Wire* from the pop-up menu.
2. Click a pin on a component.
3. To change the orientation of the wire as you draw it, right-click and choose *Orientation* from the pop-up menu.
4. Each time you choose *Orientation* you can change the bend of the wire.
5. Click again wherever you want the wire to bend, or click a pin on another component.

To name a wire when you draw it, do the following:

1. Select one of the following options:
  - Choose *Wire – Draw*.
  - Right-click and choose *Signal Name* from the pop-up menu.
2. Type a signal name in the *Signal Name* box.
3. Click *OK*.
4. Click wherever you want the wire to bend, or click a pin on another component.

### Tips for Drawing Wires

---

To...	Do this...
End a wire at a pin, dot, or other wire	Click left
Snap a wire to the nearest pin	Click <b>Ctrl+right</b>
Change the bend of the wire	Click or right-click and choose <i>Orientation</i> from the pop-up menu.
End a wire in a free space	Double-click at the final point.

## Auto-Routing a Wire

1. Choose *Wire – Route*.
2. Click the edge of a component, then click the edge of another component.

## Stretching a Wire

1. Choose *Edit – Move*.
2. Click a wire end and stretch the wire to the desired length.

## Bending a Wire

1. Choose *Wire – Draw*.
2. Begin drawing a wire.
3. Right-click and choose *Orientation* from the pop-up menu.
4. The bend of the wire changes from orthogonal to diagonal. You can continue to cycle through different wire bends by choosing *Orientation* from the pop-up menu
  - Each time
  - Once, then press Ctrl+Left

## Splitting a Wire

1. Choose *Edit – Split*.
2. Click on a wire and move the cursor down or up.  
Design Entry HDL displays the wire you are working with as red.
3. Double-click.  
Design Entry HDL displays one of the wire ends as red, indicating you can work with it separately.

## Snapping a Wire to the Nearest Pin

1. Choose *Wire – Draw*.

2. Press *Ctrl* + right-click.

Design Entry HDL draws a wire starting at the closest pin.

3. Press *Ctrl* + click right to snap the other end of the wire to the nearest pin.

## Naming a Signal

To name an existing wire, do the following:

1. Choose *Wire – Signal Name*.

The *Signal Name* dialog box appears.

2. Type one or more signal names on separate lines.

3. Select the wires you are naming in the same order you entered them in the *Signal Name* dialog box.

To name a wire when you draw it, do the following:

1. Select one of the following options:

- Choose *Wire – Draw*.
- Right-click and choose *Signal Name...* from the pop-up menu.

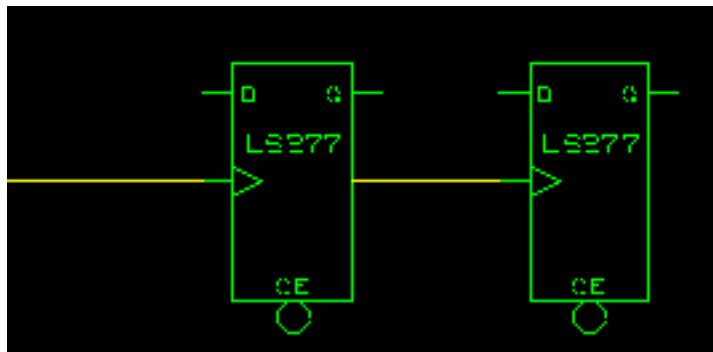
2. Type a signal name in the *Signal Name* box.

3. Click wherever you want the wire to bend, or click a pin on another component.

## Wiring Bus-Through Pins

1. Locate bus-through pins.
2. Choose *Wire – Draw*.
3. Click a component at the location across from the input pin, and connect the wire to an input pin on another component.

#### Example



## Marking Wire Connections

1. Choose *Wire – Dot/Connection*.
2. Click a wire intersection.

## Naming Signals on a Bus

1. Choose *Wire – Bus Name*.
2. Type a name in the *Bus Name* box.
3. Specify *MSB* (most significant bit), *LSB* (least significant bit), and *Increment*.
4. Click above the first wire.  
Design Entry HDL attaches a flexible line to the cursor.
5. Move the cursor so that the line crosses all the taps and click again.

The MSB value is placed on the tap closest to the first location you click, and the LSB value is placed on the tap closest to the second location you click.

#### Example

Say you want to name a 7-bit bus. You might specify:

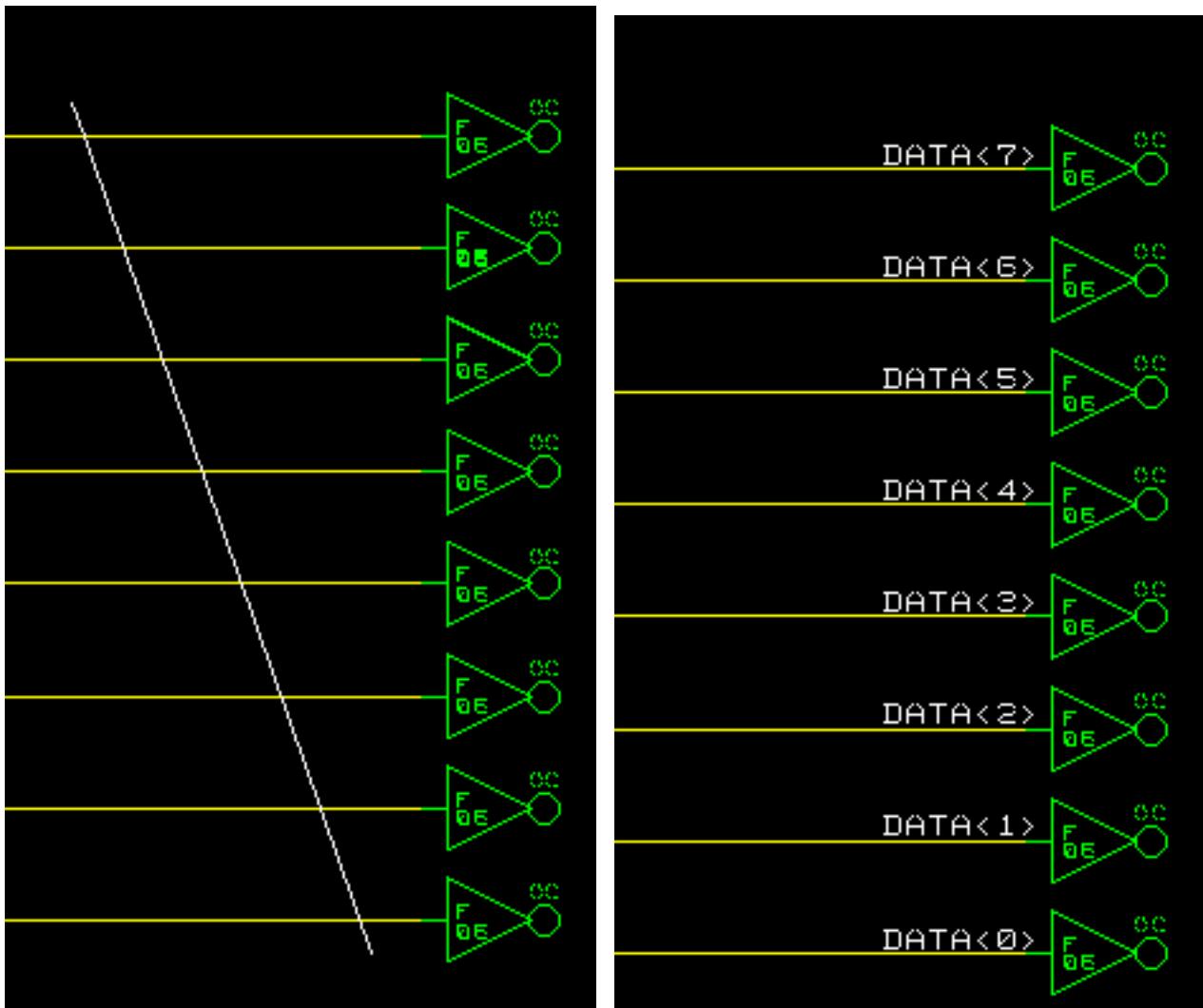
- data as the bus name
- 7 as the most significant bit
- 0 as the least significant bit

## Allegro Design Entry HDL User Guide

### Working with Wires

- 1 as the increment

In this example, the first click is above the top bit.



#### Before

The second click is below the bottom bit. Design Entry HDL draws a line between the two points. <7> is placed on the tap closest to the first location you click and <1> on the tap closest to the second location you click.

#### After

Then the bus names and values appear.

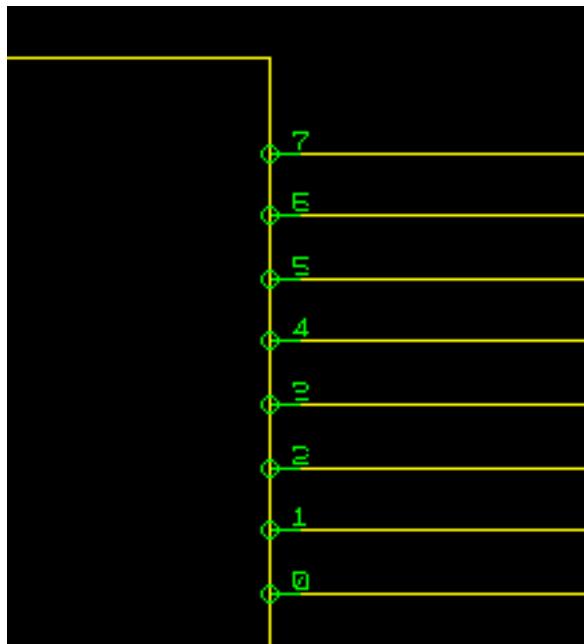
## Specifying a Tap Symbol

1. Choose *Tools – Options*.
2. In the *Symbols* box on the *Graphics* tab, type the name of the tap symbol you want to use in the *Tap Symbol* box.

## Attaching Values to Bus Taps

1. Choose *Wire – Bus Tap Values*.
2. Specify *MSB* (most significant bit), *LSB* (least significant bit), and *Increment*.
3. Click above the first wire.  
Design Entry HDL attaches a flexible line to the cursor.
4. Move the cursor so that the line crosses all the taps and click again.

The MSB value is placed on the tap closest to the first location you click, and the LSB value is placed on the tap closest to the second location you click.



## Changing Wire Thickness and Pattern

Choose one of the following in the *Wire* menu.

Thick      Makes the wire thick to indicate a bus.

Thin      Is the normal thickness for a wire.

Pattern...      Lets you choose from a variety of wire patterns.

# **Allegro Design Entry HDL User Guide**

## Working with Wires

---

---

# Working with Libraries and Components

---

Design Entry HDL includes extensive analog and digital libraries, and simulation models that you can use on your schematic pages. These libraries support design entry, simulation, timing, test and physical layout—a complete solution for designing digital, analog and mixed signal systems.

- For more information on digital libraries and simulation models, see the [Allegro Design Entry HDL Libraries Reference](#).
- For more information on analog libraries and simulation models, see the PSpice Simulator documentation.

## About the Standard Library

Cadence provides a Design Entry HDL library of standard components that lets you define and control signals in designs. These components include merge bodies for merging signals and tap bodies for tapping bits from buses. Other special parts contained in the Standard Library are NOT bodies and differently sized drawing borders.

Although the components in the Standard Library can be used for any of the supported design types, many of them are created especially for structured designs.

For more information, see *Using the Standard Library Symbols* in *Allegro Design Entry HDL Reference Guide*.

## Working with Libraries

This section describes the procedures for working with libraries.

- [Adding New Libraries](#) on page 258
- [Browsing Libraries](#) on page 258
- [Adding Libraries to the Search Stack](#) on page 259

- [Removing Libraries from the Search Stack](#) on page 259
- [Defining Library Search Order](#) on page 260

## Adding New Libraries

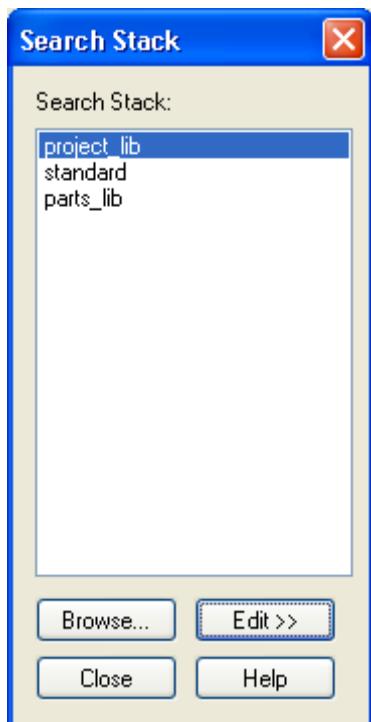
Available libraries are defined in the `cds.lib` file. To add new libraries, you must edit the `cds.lib` file. Do this using *Setup* in the Project Manager.

**Note:** See the [\*Project Manager User Guide\*](#) for more information.

## Browsing Libraries

1. Choose *File – View Search Stack*.

The *Search Stack* dialog box appears, showing the list of active libraries.



2. Select a library and double-click on it, or click *Browse*.

The Part Information Manager dialog box displays the list of components for the library you specify.

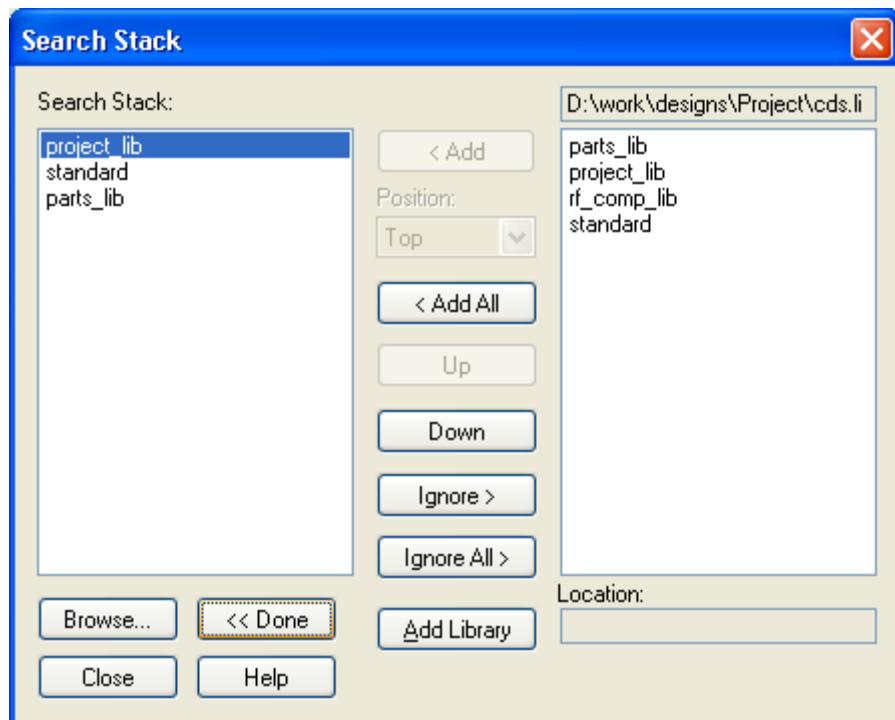
## Adding Libraries to the Search Stack

1. Choose *File – View Search Stack*.

The *Search Stack* dialog box appears, showing the list of active libraries.

2. Click *Edit >>*.

The *Search Stack* dialog box expands to display the libraries installed in your `cds.lib` file.



3. Select a library from the list of available libraries on the right.
  4. Optionally, specify *Top* or *Bottom* in the *Position* box to tell Design Entry HDL where to place the library in the active libraries list.
  5. Click *< Add*.
- The library you specify is added to the list of active libraries.
6. Click *<< Done*.

## Removing Libraries from the Search Stack

1. Choose *File – View Search Stack*.

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

---

The *Search Stack* dialog box appears showing the list of active libraries.

2. Click *Edit >>*.

The *Search Stack* dialog box expands to display the libraries installed in your `cds.lib` file.

3. Select a library in the active libraries list on the left and click *Ignore >*.

4. Click *Yes* in the confirmation box.

The library is removed from the list.

**Note:** You can add the library back to the list of active libraries.

## Defining Library Search Order

### **To define a library search order as you add libraries**

1. Choose *File – View Search Stack*.

The *Search Stack* dialog box appears showing the list of active libraries.

2. Click *Edit >>*.

The *Search Stack* dialog box expands to display the libraries installed in your `cds.lib` file.

3. Select a library.

4. Specify *Top* or *Bottom* in the *Position* box to instruct Design Entry HDL where to place the library in the active libraries.

5. Click *< Add*.

The library you specify is added to the list of active libraries.

6. Click *<< Done*.

### **To redefine the entire library search order**

1. Expand the *Search Stack* dialog box.

2. Press `Ctrl + left` and select each library from the active libraries list on the left.

3. Click *Ignore >*.

4. Click *Yes* in the confirmation box.

5. Press `Ctrl + left` and select libraries in the desired search order in the list of available libraries on the right.
  6. Click *< Add*.
- Libraries are listed in the Search Stack in the order in which you have added them.
7. Click *<< Done*.

## Working with Components

This section describes the procedures for working with components.

- [Browsing the Component List](#) on page 262
- [Creating Design Entry HDL Parts](#) on page 262
- [Creating a Symbol in Design Entry HDL](#) on page 263
- [Creating Entity Declarations from Symbols](#) on page 265
- [Creating the chips.prt File](#) on page 268
- [Creating a Part Table File](#) on page 270
- [Adding a Component](#) on page 270
- [Modifying Components](#) on page 276
- [Replacing a Component](#) on page 276
- [Breaking Up a Component](#) on page 278
- [Changing Pin States on a Component](#) on page 279
- [Choosing a Version of a Component](#) on page 279
- [Mirroring Components or Blocks](#) on page 279
- [Changing the Orientation of Components or Text](#) on page 280
- [Sectioning a Component](#) on page 280
- [Swapping Pins on a Component](#) on page 287
- [Ways to Determine if a Component Has Bus-Through Pins](#) on page 287
- [Deleting a Library Component \(Cells, Views, and Files\)](#) on page 287
- [Creating a Page Border Symbol](#) on page 288

## Browsing the Component List

To browse logical components one library at a time

1. Click *Component – Add*.

The Part Information Manager dialog box appears.

2. Scroll the library list in the search pane. You can view individual cells for each selected library.

## Creating Design Entry HDL Parts

You can create parts in Design Entry HDL or PCB Librarian. If you do not have PCB Librarian installed, create parts using the procedures and guidelines in this section.

If you have installed PCB Librarian, you can use it to create parts for use in Design Entry HDL designs.

Each Design Entry HDL part is a collection of views. In the Lib:cell.view structure, a library and a cell (parts) are directories. Under each part, there is a directory for a view type. Each view directory contains a file that defines the view.

The following illustrates the directory structure that has the files and directories that define a part - ls00.



In this figure, `ls ttl` is the library name, `ls00` is the part name (cell level directory), and the directories underneath contain the view files for `ls00`.

To create a Design Entry HDL part, do the following:

1. Create a directory with the part name `ls00`.
2. Create the following directories underneath the `ls00` directory:

- chips
  - part\_table
  - sym\_1
3. Create a symbol in Design Entry HDL.
  4. Save the symbol view files created in Design Entry HDL under the sym\_1 directory.
  5. Create a `chips.prt` file.
  6. Save the `chips.prt` file under the `chips` directory.
  7. Create a part table file.

## Creating a Symbol in Design Entry HDL

To create a new symbol, you should be in the symbol view.

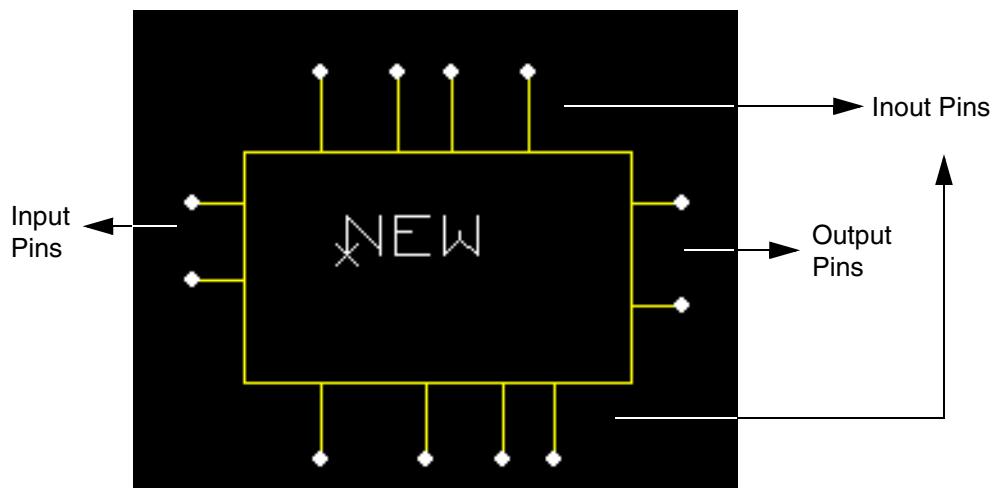
When you create a symbol, ensure that the schematic grid setting is compatible with the symbol grid setting. For DE-HDL to be able to place a symbol on the grid, the schematic grid setting should be compatible with the symbol grid setting. If the grid settings are incompatible, the pin-pitch (distance between the pins of a component) would not be compatible with the schematic grid. This can result in the symbol being placed off grid.

To create a symbol, do the following:

1. In Design Entry HDL, choose *File – Open*.
2. From the *Library* drop-down list, select the library in which the new part is to be added.
3. In the *Cell* field, specify the new symbol name.
4. From the *View* drop-down list, select *Symbol*.
5. Specify Version as 1 and click *Open*.
6. Choose *Wire – Draw*.
7. Draw a symbol shape.
8. Choose *Wire – Draw* to draw pin stubs on the symbol.
9. Choose *Wire – Dot* for adding a pin to the symbol.

Selecting *Wire – Dot* adds a dot to the symbol. This dot can be added on the edge of the pin stubs to represent a pin.

**Note:** By default, Design Entry HDL treats a pin as an input, output, or an inout pin depending on which side of the symbol it is attached. By default, a pin that is on the left of a symbol is an input pin. A pin attached to the right of the symbol is an output and pins on the top and the bottom of a symbol are inout pins. You can change the default properties by adding the `vhdl_mode` or `vlog_mode` properties to the pins and assigning them the desired values. For example, to use a pin that is to the right of a symbol as an input pin, add the `vlog_mode` property to the pin with a value of `IN`.



10. The next step is to name the pin. Choose *Wire – Signal Name*.
11. In the *Signal Name* dialog box, specify the pin name and click on the dot representing the pin. The name is attached to the pin.

Alternatively, you can add the `pin_name` property with the pin name as the property value for each pin. To do this, choose *Text – Property*. Enter `pin_name` in the *Property Name* field and the pin name in the *Property Value* field.

12. Choose *Text – Notes*.

Add pin names and attach the pin names to the respective pins.

**Note:** This step is required so that the pin names are visible when you instantiate the symbol in a schematic.

13. Choose *File – Save*.

**Note:** You can create multiple versions of a symbol. The second version of a symbol can be created only after you have created the first version of a symbol. You can create a new symbol with same name and assign the Version as 2. You can also save the existing symbol as Version 2 and then make modifications to version 2. To save the existing symbol with a different version, choose *File – Save As*. Specify the version as 2 and

click **Save**.

For information on guidelines to follow while creating symbols, see [Allegro Design Entry HDL Libraries Reference](#).

### Symbol Naming Conventions

Follow the following Design Entry HDL rules while creating symbols:

- Symbol names must be legal Verilog and VHDL names.
- If the pins have the same base name, they are part of the same VHDL or Verilog port. For example, pins `SEL(1)` and pins `SEL(0)` represent the single port `SEL(1:0)` in the entity declaration.

The following are some examples of pin names:

Name	Description
A	single-bit pin
<code>SEL(1:0)</code>	two-bit wide pin
<code>SEL(1)</code>	one bit of a two-bit wide vectored port
<code>SEL(0)</code>	one bit of a two-bit wide vectored port
<code>I(size-1:0)</code>	parameterized pin width
<code>I(SIZE-1:0)</code>	SCALD style parameterized pin width

### Creating Entity Declarations from Symbols

If the parts you are using do not have an entity declaration in the design library, you can use Design Entry HDL to automatically generate entity declarations from symbols. This section describes the properties you can add to the symbol to ensure that an accurate entity declaration is generated. Typically, you make these properties invisible in the symbol view.

You can add properties for:

- [Declaring VHDL Generics or Verilog Parameters](#)
- [Declaring Port Modes](#)
- Declaring the VHDL Logic Type of Ports

For more information, see [Setting the VHDL Logic Type for Ports and Signals](#) on page 234.

- Declaring the Verilog Logic Type of Ports

For more information, see [Setting the Verilog Logic Type for Ports and Signals](#) on page 231.

- [Declaring Libraries](#)

- [Declaring Use Clauses](#)

### Declaring VHDL Generics or Verilog Parameters

To define VHDL generics or Verilog parameters, attach the following property to the origin of the symbol.

`GENERICn=name:type`

where `n` is a unique number, `name` is the name of the generic parameter, and `type` is the generic parameter type.

### Declaring Port Modes

For every port in your symbol, attach the `VLOG_MODE` or `VHDL_MODE` property on one of the pins of the port.

**Note:** If a port has several pins, you need to attach the property on only one of the pins.

***To declare the port mode in Verilog, attach one of the following properties:***

`VLOG_MODE=INPUT`  
`VLOG_MODE=OUTPUT`  
`VLOG_MODE=INOUT`  
`VLOG_MODE=BUFFER`  
`VLOG_MODE=LINKAGE`

***To declare the port mode in VHDL, attach one of the following properties:***

`VHDL_MODE=IN`  
`VHDL_MODE=OUT`  
`VHDL_MODE=INOUT`  
`VHDL_MODE=BUFFER`  
`VHDL_MODE=LINKAGE`

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

---

If you want to read the value of an OUT port inside an architecture, do either of the following:

- Declare the port as an INOUT.
- Use behavioral assignments.

#### ***How port modes are determined when you save a symbol***

When you save a symbol, the port mode of the ports on the symbol is determined as below:

1. If the VLOG\_MODE or VHDL\_MODE property is attached to a port on the symbol, the value of the property is used to determine the port mode of the port.
2. If neither the VLOG\_MODE nor the VHDL\_MODE property is attached to a port on the symbol, the port mode for the port on the symbol will be determined from the `chips.prt` file as below:
  - If the BIDIRECTIONAL=TRUE property is attached to a pin, the port mode is INOUT
  - Else if the OUTPUT\_TYPE property with any (or no) combination of INPUT\_LOAD and OUTPUT\_LOAD properties are attached to a pin, the port mode is OUTPUT
  - Else if only the INPUT\_LOAD property is attached to a pin, the port mode is INPUT
  - Else if only the OUTPUT\_LOAD property is attached to a pin, the port mode is OUTPUT
  - If both the INPUT\_LOAD and OUTPUT\_LOAD properties are attached to a pin, and the OUTPUT\_TYPE or BIDIRECTIONAL=TRUE property is not attached to the pin, the port mode, cannot be determined from the `chips.prt` file. The port mode will be determined by steps 3 or 4 below.

**Note:** If both the OUTPUT\_TYPE and BIDIRECTIONAL=TRUE properties are attached to a pin, the BIDIRECTIONAL property takes precedence over the OUTPUT\_TYPE property and the port mode is INOUT.

3. If the VLOG\_MODE or VHDL\_MODE property is not attached to ports on the symbol, and if the `chips.prt` file does not exist, the port mode of ports on the symbol will be determined from the schematic. For example, if the signal A on the schematic is connected to an OUTPORT symbol, the port mode for pin A on the symbol will be declared as OUT.
4. If neither the VLOG\_MODE nor the VHDL\_MODE property is attached to a port on the symbol, and if both the schematic and the `chips.prt` file do not exist, the port mode of the port on the symbol will be determined by Design Entry HDL using its internal algorithms.

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

---

**Note:** Cadence recommends that you use the VLOG\_MODE and VHDL\_MODE properties to declare port modes when you are creating a symbol in Design Entry HDL. This ensures that the port mode of the ports on the symbol are declared as per your requirements.

### Declaring Libraries

To generate library clauses from a Design Entry HDL symbol view:

- Attach the following property to the origin of the symbol:

```
LIBRARYn = libname
```

where *n* is a unique number and *libname* is the name of the library.

#### **Example**

```
LIBRARY1 = ieee
```

### Declaring Use Clauses

To generate use clauses from a Design Entry HDL symbol drawing view:

- Attach the following property to the origin of the symbol:

```
USEn = libname
```

where *n* is a unique number and *libname* is the name of the library.

#### **Example**

```
USE1 = IEEE_VITAL_PRIMITIVES.ALL
```

### Creating the chips.prt File

The chips.prt file is used by Packager-XL to associate pin numbers and names in your part. This file can be created using a text editor like vi or Windows Notepad. Given below is a sample chips.prt file with descriptions (marked #) on sections:

```
FILE_TYPE=LIBRARY_PARTS;
# This is the header. This line identifies the type of the file.
TIME=' COMPILED ON THU JAN 10 14:52:02 1991 ';
# This is just a comment.
primitive '74LS00','74LS00_DIP';
# There could be multiple primitives. The basic primitive name in this case is
```

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

74LS00. Adding an DIP specifies the PACK\_TYPE as DIP. There are other PACK\_TYPE values like SOIC, BG, FG etc. You can specify the pin name-number assignment for multiple primitives in one section. You need to specify a different primitive when the pin name-number assignment is different from the basic primitive.

```
pin
'B'<0>:
# This is the name of the pin. In this case, B<0> represents an element of a vector
pin. All pins and the pin numbers are to be written in this file.
INPUT_LOAD='(-0.4,0.02)';
PIN_NUMBER='(13,10,5,2)';
# The 4 pin numbers represent the pin numbers in each of the 4 sections of the
device.
PIN_GROUP='1';
'A'<0>:
INPUT_LOAD='(-0.4,0.02)';
PIN_NUMBER='(12,9,4,1)';
PIN_GROUP='1';
'-Y'<0>:
OUTPUT_LOAD='(8.0,-0.4)';
PIN_NUMBER='(11,8,6,3)';
end_pin;
body
POWER_PINS='(VCC:14;GND:7)';
# Name:pin number; name2:pin_number2.....
FAMILY='LSTTL';
PART_NAME='74LS00';
BODY_NAME='LS00';
DEFAULT_SIGNAL_MODEL='SN74LS00N TI';
JEDEC_TYPE='DIP14_3';
CLASS='IC';
TECH='74LS';
end_body;
end_primitive;
# You can enter the second primitive after end_primitive.This is typically done for
different pack_types.
primitive '74LS00_SOIC';
pin
'B'<0>:
INPUT_LOAD='(-0.4,0.02)';
PIN_NUMBER='(13,10,5,2)';
PIN_GROUP='1';
'A'<0>:
INPUT_LOAD='(-0.4,0.02)';
PIN_NUMBER='(12,9,4,1)';
PIN_GROUP='1';
'-Y'<0>:
OUTPUT_LOAD='(8.0,-0.4)';
PIN_NUMBER='(11,8,6,3)';
end_pin;
body
POWER_PINS='(VCC:14;GND:7)';
FAMILY='LSTTL';
PART_NAME='74LS00';
BODY_NAME='LS00';
DEFAULT_SIGNAL_MODEL='SN74LS00D TI';
JEDEC_TYPE='SOIC14';
CLASS='IC';
TECH='74LS';
end_body;
end_primitive;
END.
```

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

For more information on the `chips.prt` file, see the [Allegro Design Entry HDL Libraries Reference](#).

## Creating a Part Table File

The part table file associates a logical part with physical parts having varying physical properties. Each row in a part table corresponds to a physical part.

You can create a part table file (`.ptf`) using any text editor.

Given below is a sample part table file:

```
FILE_TYPE = MULTI_PHYS_TABLE;
PART '74F08'
CLASS = IC
:PACK_TYPE(OPT='SOIC') ,PKG(OPT='SOIC') = JEDEC_TYPE, PART_NUMBER, COST, STATUS;
SOIC      , SOIC (1)      = SOIC14      , CDN0000-48 , .83 , PREF
DIP       , DIP  (2)      = DIP14_3     , CDN0001-48 , .47 , NONPREF
LCC       , PLCC20 (3)    = PLCC20      , CDN0003-48 , .91 , NONPREF
END_PART

PART '74F138'
CLASS = IC
:PACK_TYPE(OPT='SOIC') ,PKG(OPT='SOIC') = JEDEC_TYPE, PART_NUMBER, COST;
SOIC      , SOIC (1)      = SOIC16      , CDN0000-38 , .93
DIP       , DIP  (2)      = DIP16_3     , CDN0001-38 , .77
LCC       , PLCC20 (3)    = PLCC20      , CDN0003-38 , .9
END_PART

PART '74F244'
CLASS = IC
:PACK_TYPE(OPT='SOIC') ,PKG(OPT='SOIC') = JEDEC_TYPE, PART_NUMBER, COST;
SOIC      , SOIC (1)      = SOIC20W     , CDN0000-45 , .93
DIP       , DIP  (2)      = DIP20_6     , CDN0001-45 , .87
LCC       , PLCC20 (3)    = PLCC20      , CDN0003-45 , .71
END_PART
END.
```

For more information on part table files, see the [Allegro Design Entry HDL Libraries Reference](#).

## Adding a Component

1. Choose *Component – Add*.

Part Information Manager appears.

2. Select a library from the *Library* list in the search pane.

Design Entry HDL displays the components in the library you select.

3. Select a component. The component attaches to the cursor.

4. Click on the drawing to place the component.

You can continue placing components until you choose another menu item or select *Done* from the pop-up menu.

#### ***To add a component with physical information,***

1. Choose *Component – Add* to select a component.
2. Choose *PPT Options* from the pop-up menu that appears when you right-click on a PPT row in the Part Information Manager window.

The Property Options dialog box appears.

**Note:** See [Defining Physical Property Options](#) on page 188 for more information on defining physical property options.

## **Locking Components**

If you are working with critical components in a schematic block, or when multiple designers work on a design, you might not want any changes to a component. In such cases, you can identify the component as locked so as not to allow any changes.

Placement, editing, wiring, and property changes are not allowed on locked components.

If required, users other than the designer who locked the component can unlock the component.

To lock a component, do the following:

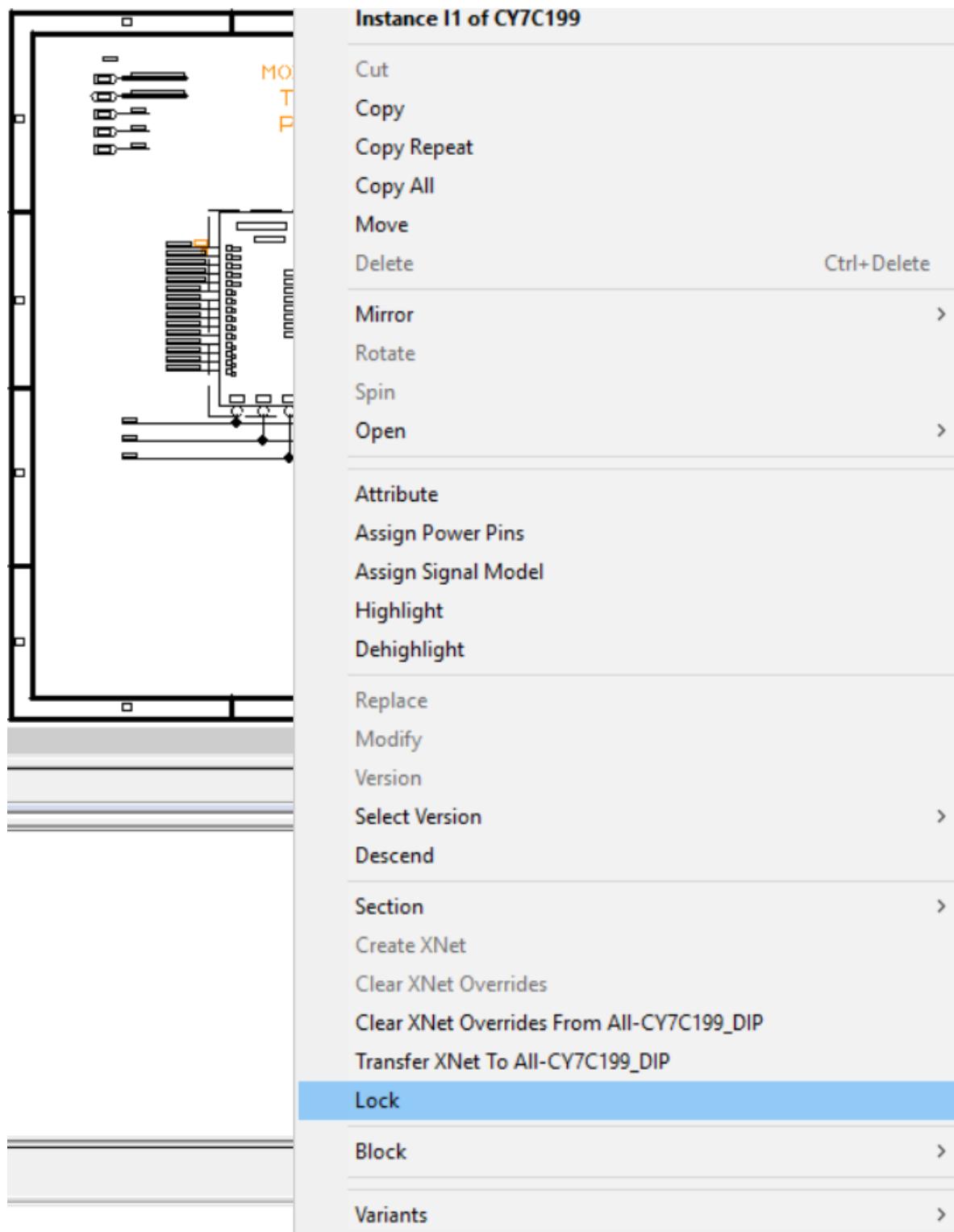
1. Select the component that you want to lock.

**Note:** You cannot lock components in a read-only schematic.

# Allegro Design Entry HDL User Guide

## Working with Libraries and Components

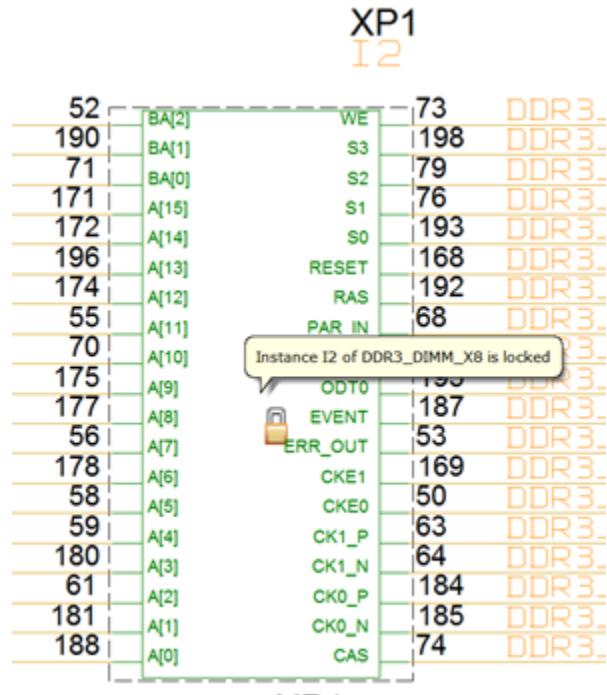
2. Right-click and select *Lock*.



# Allegro Design Entry HDL User Guide

## Working with Libraries and Components

The component is locked.



## Compatible Footprints Check

When modifying or replacing components, you can use a directive, ALLOW\_FOOTPRINT\_COMPATIBILITY\_CHECK, in the START\_CONCEPTHDL...END\_CONCEPTHDL section to define whether DE-HDL should check for footprint compatibility between components.

You can define three values for this directive:

- always - DE-HDL will always check for footprint compatibility for all components in the design
- inst - define the JEDEC\_TYPE\_CHECK property for symbols or instances that should be checked for footprint compatibility. The value of this property can be set as 1, ON, or TRUE. When modifying or replacing components, only those instances that have this property will be checked for compatible footprints.
- disable - DE-HDL will not check whether components have matching footprints

You can also define compatible footprints using a file named cjedectype.txt.

When searching for compatible footprints, DE-HDL finds compatible footprints by matching the source component and target component JEDEC types and ATL\_SYMBOLS.

DE-HDL first searches the instance property for JEDEC and ALT\_SYMBOLS, then PTF properties and then the properties in chips.prt. If you have a cjedectype.txt file in the SITE area, DE-HDL also checks this file for compatible footprints.

## Using Compatible JEDEC\_TYPES

DE-HDL allows you to define components with compatible JEDEC\_TYPES, that is, components which have different JEDEC\_TYPES but which can be replaced with each other and thereby occupy the same footprint on the board.

You can define compatible JEDEC\_TYPES by creating a file named cjedectype.txt. To create the cjedectype.txt file, copy the cjedectype.txt file located at <your\_install\_dir>/share/cdssetup and paste it under the cdssetup directory in the SITE area, which is at the same level as the project file.

An example of the cjedectype.txt file is as follows:

```
#####
# File for jedec type compatibility
#####
```

```
#  
C200901_010 C200901_011 C200901_011 C200901_011 C200901_012 C200901_013;  
ADDAMS_CAP ADDAMS_CAPC ADDAMS_CAPD;  
CC0603 CC1812 CC1812 CC1825 CC1206 CC1812;  
RC1206 RC0603 RC2010;
```

This example defines four sets of compatible JEDEC\_TYPES. Each set of compatible JEDEC\_TYPES is defined in a single line ending with a semi colon (;). Compatible JEDEC\_TYPES are separated by a space.

## How compatible JEDEC\_TYPES Work?

Assume that you have a component with the RefDes U1 and the JEDEC\_TYPE DO\_35\_NP. Also assume that this component has two compatible JEDEC\_TYPES: DO\_35\_NP and DO\_35\_NP2.

If you try to change the value of U1, DE-HDL searches for its compatible JEDEC\_TYPES in the cjeedectype.txt file. If compatible JEDEC\_TYPES for the selected component exist, a dialog box with a list of the available compatible JEDEC\_TYPES is displayed. Since U1 has compatible JEDEC\_TYPES, a dialog box displays the following message:

*Compatible footprints for "DO\_35\_NP" are: DO\_35\_NP, and DO\_35\_NP2.*

Click *OK* to continue changing the value of U1. The Part Table Filter dialog box is displayed. The JEDEC\_TYPE column displays the symbol \* signifying that all available JEDEC\_TYPES for the component are displayed. You can now choose any available part and change the value of U1.

Choosing only those parts that correspond to compatible JEDEC\_TYPES is recommended.

If you choose a component that does not have any compatible JEDEC\_TYPE and try to change its value, the Part Table Filter dialog box will display only those components that have the same JEDEC\_TYPE.

## Modifying Components

### ***To modify a single component***

1. Choose *Component – Modify*.
2. Select a component whose physical properties you want to modify.

The *Modify Component* dialog box appears with the filter set to the current physical property values in the component.

3. Click *Reset Filters* to display all rows in the part table file.
4. Select the desired row of physical properties to attach to the component you want to modify.

**Note:** If you want to see all the PPT rows by default, set the following directive in the .cpm file:

MODIFY\_QUICK\_RESET\_FILTERS 'ON'

or

MODIFY\_QUICK\_RESET\_FILTERS 'TRUE'

You can continue selecting and modifying components until you choose another menu command or select *Done* from the pop-up menu.

You can modify the physical properties of all components in a group if they are the same logical components.

### ***To modify a group of components***

1. Choose *Group – Components – Modify*.
2. Select a row in the *Physical Part Filter* dialog box.

The physical properties of all the components in the group are replaced with the row you select in the *Physical Part Filter*.

## Replacing a Component

1. Choose *Component – Replace*.
- Part Information Manager appears.

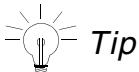
2. Select a component.
3. Click on the component in the schematic to replace it.

The component is replaced with version 1 of the component that you selected in Part Information Manager.



***Some properties, such as LOCATION and \$PN, may be lost when you use the Component - Replace command. Use Tools - Global Update - Global Component Change to replace components if you do not want to lose backannotated or user-defined properties.***

**Note:** If you want to replace the component with another version of the component, double-click the selected component in Part Information Manager and select another version in the *Version* field.



If you want a component to be replaced or modified only with a list of specific components from the library, set the directive `ALLOWED_ALTERNATE_PART_PROP` to a part property name, such as `PART_NUMBER` and `VALUE`, on the basis of which a component can be replaced. For more information, refer to the [ALLOWED\\_ALTERNATE\\_PART\\_PROP](#) section of *Allegro Front-End CPM Directive Reference Guide*.

If you are in the pre-select mode in Design Entry HDL, you can replace multiple components by doing the following:

1. Use *Ctrl+click* or *SHIFT+click* to select multiple components.
2. Choose *Component – Replace* to display Part Information Manager.
3. Select the component that should replace all the components.

#### ***To replace a component along with its physical properties***

1. Choose *Component – Replace*.  
Part Information Manager appears.
2. Select a component from the *Library*.
3. Right-click on a PPT row in the Part Information Manager window and choose *PPT Options*.

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

---

The *Property Options* dialog box loads the PPT file for the selected component.

4. Make the required changes in the *Property Options* dialog box and click *OK*.

**Note:** See [Defining Physical Property Options](#) on page 188 for more information on defining physical property options.

5. Click on an existing component in the schematic to replace it.

You can continue replacing components until you choose another menu item or right-click to choose *Done*.

#### **To replace components in a group**

1. Set the current group.
2. Choose *Component – Replace*.

The *Replace Component* dialog box appears.

3. Select the component that should replace all components in the current group.

If you want to replace the components in the group with a component along with its physical properties, do the following:

- a. Right-click on a PPT row in the Part Information Manager window and choose *PPT Options*.

The *Property Options* dialog box appears.

- b. Select the appropriate row of physical properties from the *Property Options* dialog box and click *Close*.

All the components in the current group are replaced with version 1 of the component that you selected in the *Replace Component* dialog box.

## **Breaking Up a Component**

1. Choose *Component – Smash*.
2. Click a component in your drawing.
3. Select the discrete pieces that made up the component.

## Changing Pin States on a Component

1. Choose *Component – Bubble Pins*.
2. Click a pin.

**Note:** If the pins are part of a bubble group, you can choose *Bubble Pins* to convert the component from one form to another.

## Example of Converting a Component from One Form to Another

For example, a NOT body is defined with both the `BABBLED` and `BUBBLE_GROUP` properties attached:

```
BABBLED=(B)  
BUBBLE_GROUP=(A | B)
```

Because `BABBLED=(B)`, pin B is bubbled when the component is initially added to a drawing. If you choose *Component – Bubble Pins* and click either pin A or B, the attached `BUBBLE_GROUP` property specifies that pin A is now the bubbled pin and pin B the un-bubbled pin.

## Choosing a Version of a Component

1. Choose *Component – Version*.
2. Click a component in your drawing to display the next version.
3. Continue clicking on the component to view all the versions until the original version is displayed again.

**Note:** You can also run the `version` command using this stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#) on page 131.

## Mirroring Components or Blocks

1. Choose *Edit – Mirror*.
2. Click a component or a block.

## Changing the Orientation of Components or Text

To rotate a component when adding it to the schematic:

1. Choose *Component – Add*.
2. Select a component to add.
3. Right-click and choose *Rotate* from the pop-up menu.
4. Choose *Rotate* from the pop-up menu continuously to rotate the component another 90 degrees each time.

To rotate a component that has already been placed in the schematic:

1. Choose *Edit – Rotate*.
2. Click a component.
3. Continuously clicking on the component rotates it another 90 degrees.

**Note:** When you rotate a component, the associated property text appears either vertically along the left side of the component or horizontally above the component.

To spin a component:

1. Choose *Edit – Spin*.
2. Click a component.
3. Continuously click on the component to spin it again.

**Note:** When you spin a component, the associated property text spins around with the component.

## Sectioning a Component

1. Enlarge the drawing so that the component you want to section is clearly visible.
2. Choose *Component – Section*.
3. Click a component.

Each time you click, you select a different section of the physical component, and different pin numbers are displayed. The section assignment is removed each time you cycle through all the available sections.

When you section a component, the following properties are added on the component:

- SEC

Assigns a logical component to a particular section within a physical part.

- SEC\_TYPE

Identifies the package type in the *chips.prt* file used to get pin number assignments when sectioning a part.

## Sectioning Multiple Components

You can assign pin numbers to multiple logical part instances simultaneously using the *Component – Section* menu. This helps you avoid sectioning each part instance individually. If you section the part instances incorrectly, you can also unsection multiple part instances simultaneously. Pin numbers can be alphanumeric.

To section multiple components, do the following:

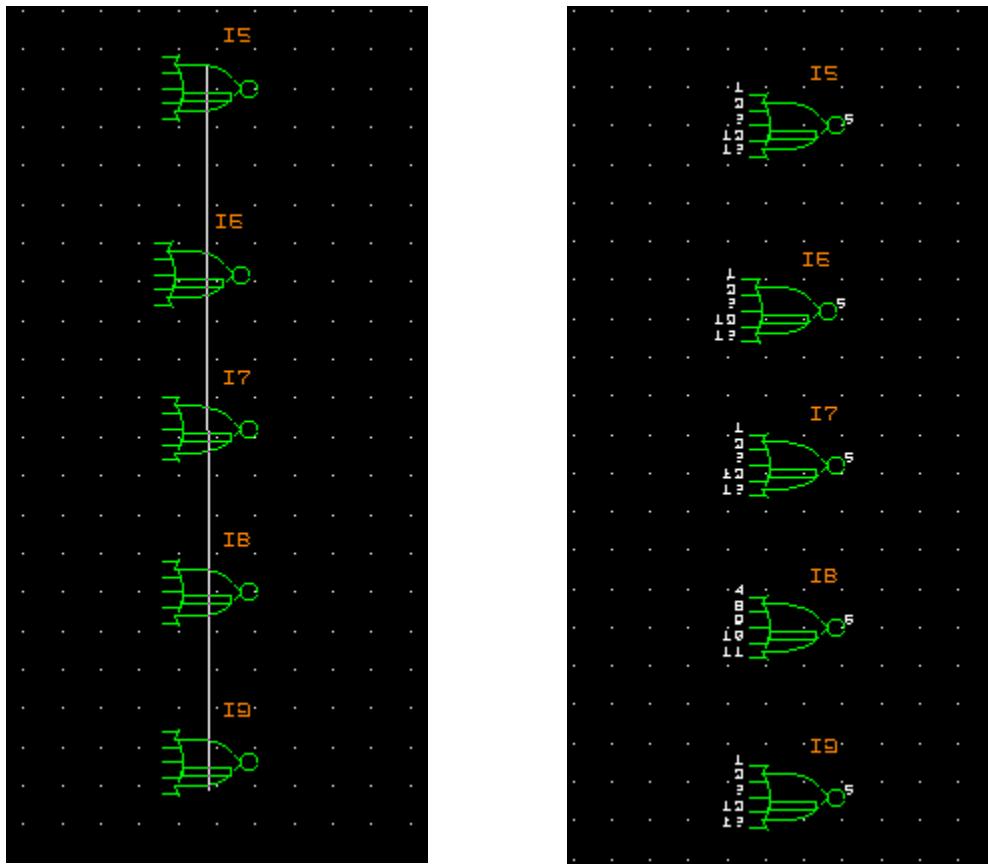
1. Choose *Component – Section – Multiple Sections*.

You will be prompted to select the components to section.

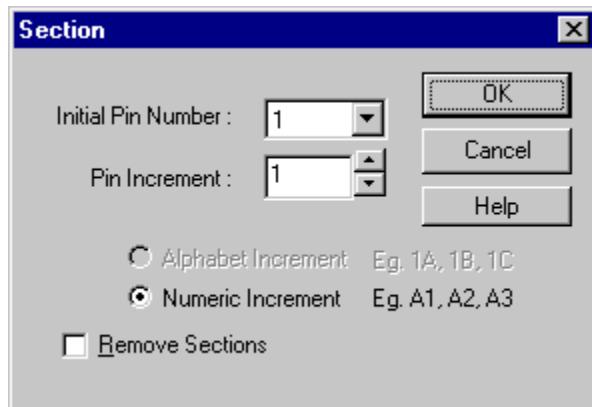
## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

2. Select the first component and drag the mouse to the last part instance, drawing a line with the mouse.



**Note:** You must ensure that all the part instances you want to section are covered by a single line.



3. Specify the starting pin number in the *Initial Pin Number* text box of the Section dialog box.

## Allegro Design Entry HDL User Guide

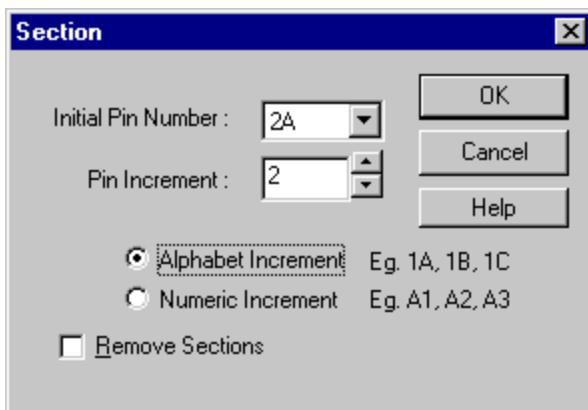
### Working with Libraries and Components

4. Select a number by which you want to increment the subsequent pin numbers, in the Pin Increment spin box.

For example, for a part F153, if you specify 4 as the pin increment, 4 pins will be skipped while sectioning the second part instance, I19 and all the subsequent part instances.

5. Specify whether you want to assign an alphabet or numeric increment.

- For components that have alphabetic or alphanumeric pin numbers, you can select the *Alphabet Increment* option. The pin numbers will increment alphabetically. For example, 1A, 1B, 1C, and so on. This option is enabled only for pins with alphanumeric pin numbers.



- If you select the *Numeric Increment*, the pin numbers will increment numerically. For example, A1, A2, A3, and so on.

Selected part instances will be sectioned with pin numbers as defined.

## Examples

The following examples explain the behavior of the sectioning command for components that have a single pin per section and multiple pins per section.

### Single Pin Per Section

Consider the example of a design with three instances of a resistor, Res, with the varying values for the initial pin number and pin increment:

#### Case 1

- Initial pin number = 1
- Pin increment = 1

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

- Resulting pin numbers = (1,2), (1,2), (1,2)

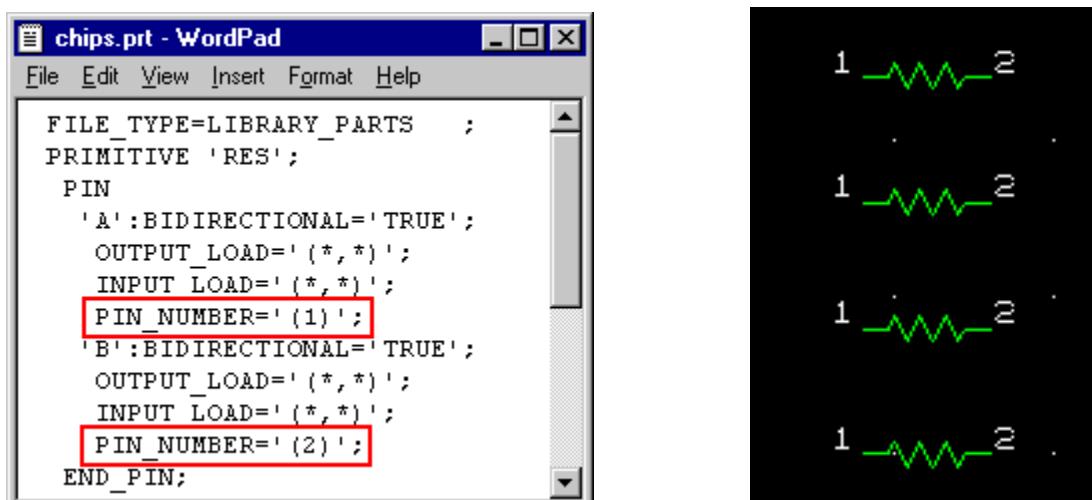
#### Case 2

- Initial pin number = 1
- Pin increment = 2
- Resulting pin numbers = (1,2), (1,2), (1,2)

#### Case 3

- Initial pin number = 1
- Pin increment = 3
- Resulting pin numbers = (1,2), (1,2), (1,2)

Thus, irrespective of the increment you specify, in a component with a single pin per section, the resulting pin numbers assigned will remain the same.



#### Multiple pins per section

Consider the following examples to understand sectioning for components with multiple pins per section.

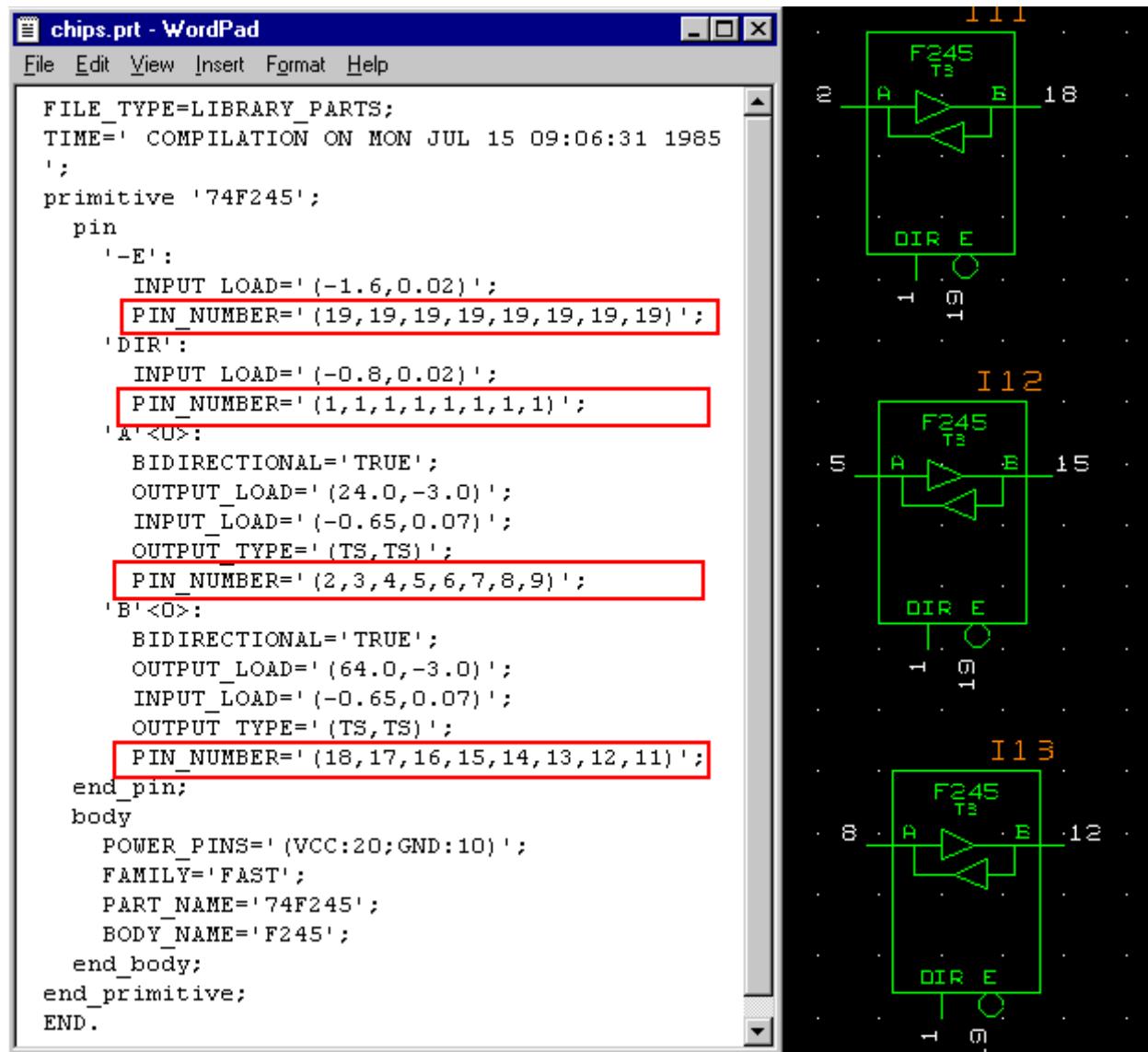
#### Case 1

- Initial pin number = 2
- Pin increment = 3

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

- Result - The first pin, pin A in the first instance of the component F245, will be assigned the initial pin number of 2. For the same pin in the next instance, three pin numbers (since the pin increment is specified as 3) will be skipped (2, 3, and 4) and pin number 5 will be assigned. Similarly, the same sequence will be followed for other pins.



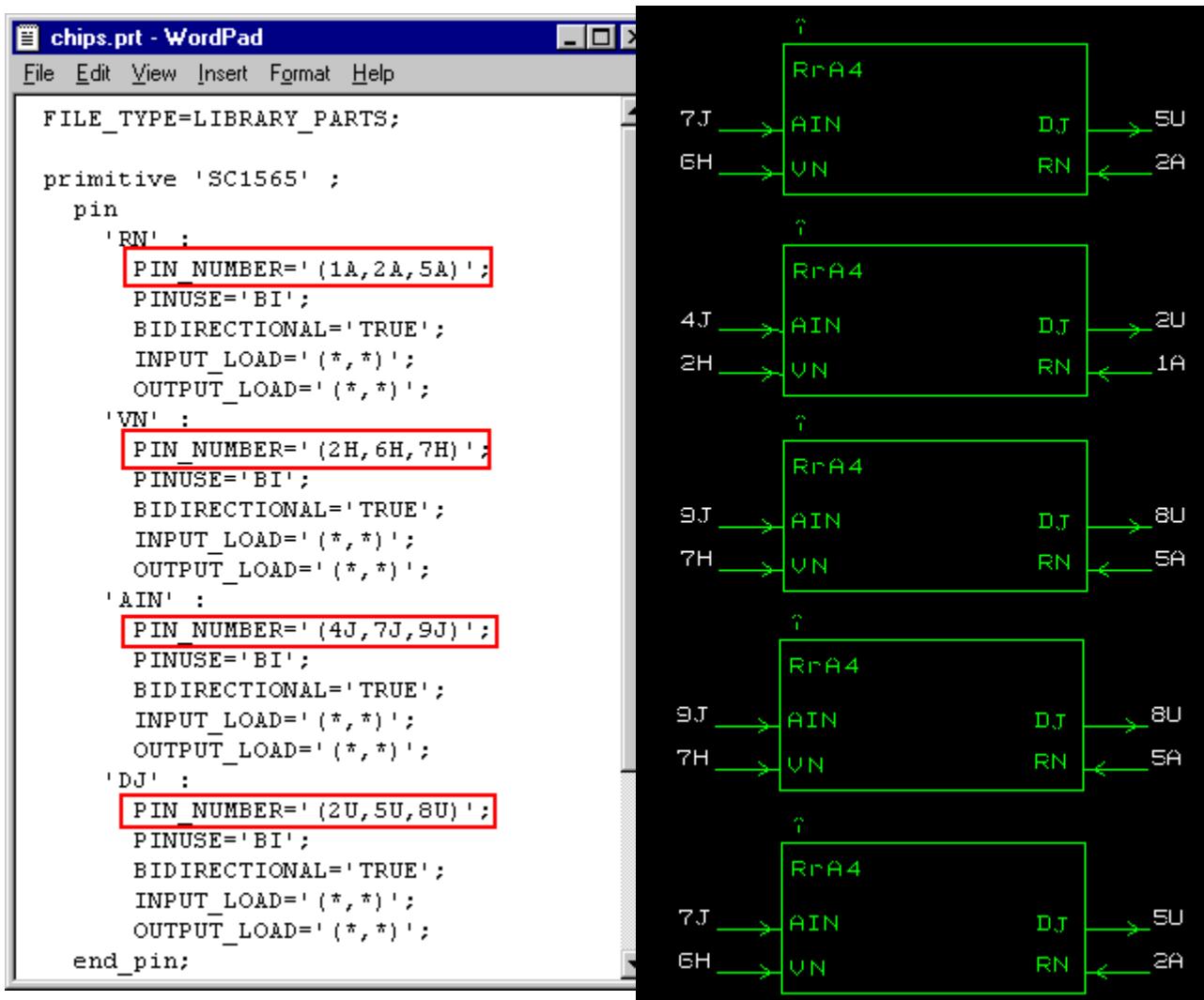
### Case 2

- Initial pin number = 2A
- Pin increment = 2
- Result - The first pin, pin RN in the first instance of the component RrA 4, will be assigned the initial pin number of 2A. For the same pin in the next instance, two pin numbers (since

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

the pin increment is specified as 2) will be skipped (2A and 5A) and pin number 1A will be assigned. Similarly, the same sequence will be followed for other pins.



## Unsectioning Components

After sectioning various part instances, if you find that you have sectioned them incorrectly, you can revert to the original state by unsectioning the part instances. To unsection multiple part instances, do the following:

1. Choose *Component – Section – Multiple Sections*.
2. Select the first component and drag the mouse to the last part instance, drawing a line with the mouse.

3. Select the *Remove Sections* check box in the Section dialog box.

All the selected part instances are unsectioned.

## Swapping Pins on a Component

A component must be sectioned before you can swap pins on it.

1. Choose *Component – Swap Pins*.
2. Click the two pins you want to swap.

**Note:** Properties attached as a result of swapping pins can only be deleted or moved, not changed. You should not change the `PN` property. After swapping, `$PN` becomes the hard property `PN`.

Swapping pins with the `HAS_FIXED_SIZE` property in Design Entry HDL stores pin names as `$PN` (soft property) instead of `PN` (hard property) resulting in conflicting values in the Occurrence Property File (OPF). Therefore, it is recommended that you first close Constraint Manager, if it is running, and then run Export Physical after swapping pins.

When swapping pins on a component with the `ALLOW_PINTEXT_SWAP` directive added to the `.cpm` file, pin text is also swapped along with the pin name. For more information about the `ALLOW_PINTEXT_SWAP` directive, refer to the [ALLOW\\_PINTEXT\\_SWAP](#) section of Allegro Front-End CPM Directive Reference Guide.

## Ways to Determine if a Component Has Bus-Through Pins

- Choose *Display – Pins* to display an asterisk at the location of every pin.
- Choose *Display – Pin Names* to display the pin names for the component.
- Bus-through pins have the same name as the corresponding visible pin.
- Look at the symbol view of the component to see if the component is defined with a bus-through pin.

## Deleting a Library Component (Cells, Views, and Files)

1. Choose *File – Remove*.
2. In the scroll area of the *View – Remove* dialog box that appears:
  - a. Select a cell to delete the entire cell.

- b. Click + next to the cell name to expand the hierarchy, and select a view to delete.
  - c. Click + again to expand the hierarchy, and select a page to delete.
3. Click *Remove*.



***You must not delete cells, views or files from Windows Explorer or DOS command prompt. This can create problems in the design.***

## Creating a Page Border Symbol

The first step while creating any design is to add a page border. You can have a design without page borders, but it is a good design practice to add page borders. Page borders provide a convenient way of documenting information such as the date, the design name, the page number, the engineer's name, the company logo and so on, on the schematic. Design Entry HDL allows you to specify the default page border that you want to be used automatically every time you create a schematic page. For more information, see [Setting Automatic Page Borders](#) on page 90.

Page borders are required when you cross reference a design. When you plot a schematic, it is often difficult to trace the location of a signal or instances of a part. CRefer traces the signals and parts in a schematic and annotates the location of each one in text reports. CRefer writes the page number and the location of the part or signal in relation to the page border.

The Cadence Standard library provides six standard page borders—A SIZE PAGE to F SIZE PAGE—that you can use in your design.

This section describes the procedures for customizing a page border in the Standard library or creating a page border of your own.

- [Customizing a Page Border in the Standard Library](#) on page 288
- [Creating a Page Border of Your Own](#) on page 290

### Customizing a Page Border in the Standard Library

Cadence recommends that you customize a page border in the Standard library instead of creating a page border of your own. This is because page borders are created by drawing wires and adding notes, and it is time consuming to create a page border of your own.

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

---

To customize a page border in the Standard library, do the following:

1. Create a new project using Project Manager.
2. Choose *Tools – Design Entry HDL* in Project Manager to start Design Entry HDL.
3. In Design Entry HDL, choose *File – Open*.  
The *View Open* dialog box appears.
4. Select *Standard* library in the *Library* drop-down list.  
The list of components in the library are displayed.
5. Select the page border that you want to customize.  
The page border name is displayed in the *Cell* field.
6. Select *Symbol* from the *View* drop-down.
7. Click *Open* to open the symbol for the page border in Design Entry HDL.
8. Choose *File – Save As*.  
The *View Save As* dialog box appears.
9. From the *Library* drop-down list, select the library in which you want to save the page border.
10. Specify the name of the page border in the *Cell* field.
11. Click *Save*.
12. Make the necessary changes in the page border. For example, you can do the following:
  - Choose *Wire – Draw* to add boxes for placing notes, or add your company logo by drawing wires. For example, the Cadence logo in the CADENCE A SIZE PAGE page border symbol in the Standard library was created by drawing wires.
  - Choose *Text – Note* to add notes, URLs, copyright information, non-disclosure information and so on.
  - Add custom text. For more information, see [Adding Custom Text on Page Borders](#) on page 291.
13. Choose *File – Save* to save the changes.  
Maintain the page border symbol in a reference library so that other users can use the page border.

You can now use the page border symbol on your schematic pages. If you want to cross reference a design that uses the page border, define the page border in the `cref.dat` file located at `<your_install_dir>/share/cdssetup/creferhdl/`.

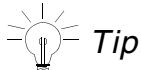
## Creating a Page Border of Your Own

1. Create a new project using Project Manager.
2. Choose *Tools – Design Entry HDL* in Project Manager to start Design Entry HDL.
3. Run the following command in the Design Entry HDL console window:

```
edit <page_border_name>.sym.1.1
```

Design Entry HDL creates a symbol drawing named `<page_border_name>.sym.1.1` and places the ORIGIN symbol from the Standard library on the drawing. The ORIGIN symbol is placed at coordinates (0, 0) on the drawing.

**Note:** You should not move the ORIGIN symbol from this location on the drawing.



*Tip*  
You can choose *Display – Coordinate* and click on the ORIGIN symbol to display the coordinates in the Design Entry HDL console window

4. Choose *Text – Attributes* and click on the ORIGIN symbol.  
The *Attributes* dialog box appears.
5. Click *Add*.
6. Type `COMMENT_BODY` in the *Name* field.
7. Type `TRUE` in the *Value* field.
8. Click *OK* to close the *Attributes* dialog box.
9. Choose *Wire – Draw* to draw the page border.
10. Choose *Text – Note* to add zones on the page border. The zones are used by CRefer to display the location of schematic objects in the CRefer text reports. For more information, see [Creating Zones on Page Borders](#) on page 292.
11. Choose *Wire – Draw* to add boxes for placing notes or add your company logo by drawing wires. For example, the Cadence logo in the CADENCE A SIZE PAGE page border symbol in the Standard library was created by drawing wires.
12. Choose *Text – Note* to add notes, URLs, copyright information, non-disclosure information and so on.

13. Add custom text. For more information, see [Adding Custom Text on Page Borders](#) on page 291.

14. Choose *File – Save* to save the changes.

Maintain the page border symbol in a reference library so that other users can use the page border.

You can now use the page border symbol on your schematic pages. If you want to cross reference a design that uses the page border, define the page border in the `cref.dat` file located at `<your_install_dir>/share/cdssetup/creferhdl/`.

For more information on creating zones on page borders, refer to the topic *Working with the Cref Data File* in the [Allegro Design Entry HDL Utilities User Guide](#).

### Adding Custom Text on Page Borders

You can add custom text in page borders to display page numbers, design information, cross referencing information and so on, on the schematic pages. For more information, see [Working with Custom Text](#) on page 377.

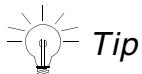
**Note:** You must attach the custom text to the ORIGIN symbol on the page border.

The page border symbol displays the format string for the custom text. When the page border is instantiated on a schematic page, the values of custom variables are substituted. For example, add the following custom text on the page border symbol:

Page <CON\_PAGE\_NUM>

When the page border is instantiated on a schematic page, the custom variable `CON_PAGE_NUM` will take its actual value on each page. For example, Page 1 or Page 2.

See [Adding Custom Text](#) on page 383 for the procedure for adding custom text.



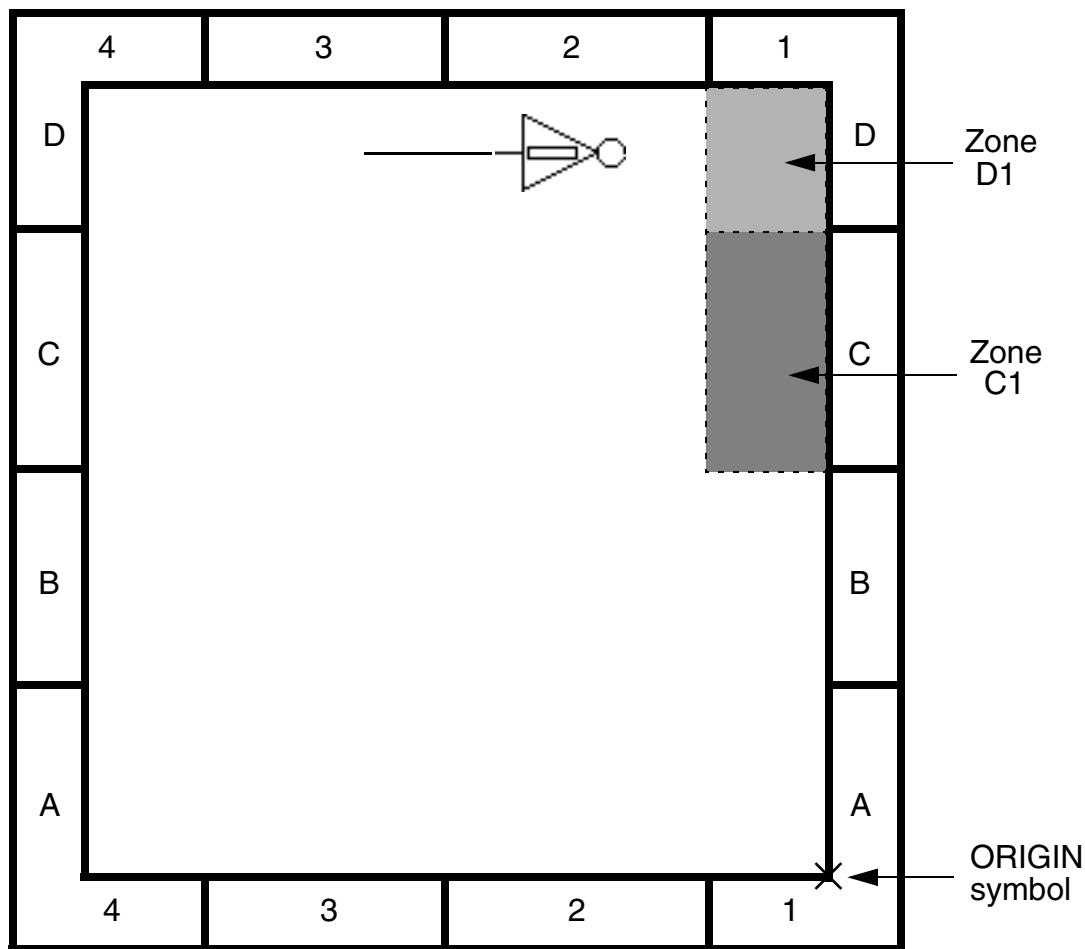
To quickly locate the origin of a page border symbol, run the following console window commands:

```
find origin  
next
```

The origin of the page border symbol is highlighted.

## Creating Zones on Page Borders

You can create zones on page borders as shown in the following figure.



The zones are used by CRefer to display the location of schematic objects in the CRefer reports. For example, the Crefparts report will display the location of the `1s04` component in the schematic above as:

```
<value of LOCATION property> 74LS04 <cell_name> [ <page_number>D2 ]
```

For example, if the `1s04` component that has the `$LOCATION=U1` property is located in zone D2 on a schematic page `ANALOG_IO.SCH.1.8`, the Crefparts report will display the location of the `1s04` component as:

```
U1 74LS04 ANALOG_IO [ 8D2 ]
```

# Using Component Revision Manager

## Overview

Creating and finalizing a schematic can be a time-consuming process subject to modifications at various stages of design. Some modifications may occur due to changes made in the reference library cells. Often, these changes are unpredictable, and are difficult to incorporate into your schematic. Importantly, the process to update and synchronize your schematic manually can be iterative and can impact your design timelines.

To help you have up-to-date library cells in your designs when changes occur in the reference libraries, use the Component Revision Manager. This helps you do the following:

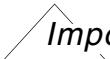
- Receive notifications for differences between the schematic and reference library cells.
- Update a schematic with modified reference library cells in your design.
- Highlight the schematic cells that are different from reference library cells.
- Save the differences between the schematic and library cells.

## How Component Revision Manager Works

Whenever you save your design, Allegro Design Entry HDL creates the `metadata` folder in your design project folder. The comparison between the metadata of your schematic cells and the library cells is the key behind identifying the differences. Make sure that you have enabled the creation of metadata in Allegro Design Entry HDL. For information on how to configure metadata creation preferences, see [Setting Preferences for Metadata Creation](#).

**Note:** The location of reference or local libraries is defined in the `cds.lib` file of your design project.

**Note:** If a schematic cell does not contain any metadata, Component Revision Manager does not check it for differences with the library cell.



### *Important*

Make sure that your metadata has been generated in the latest SPB version. Cadence strongly recommends that you do not use metadata created using earlier versions.

## Getting Started with Component Revision Manager

The Component Revision Manager opens when you do the following:

- Launch Allegro Design Entry HDL
- Edit a page of your schematic

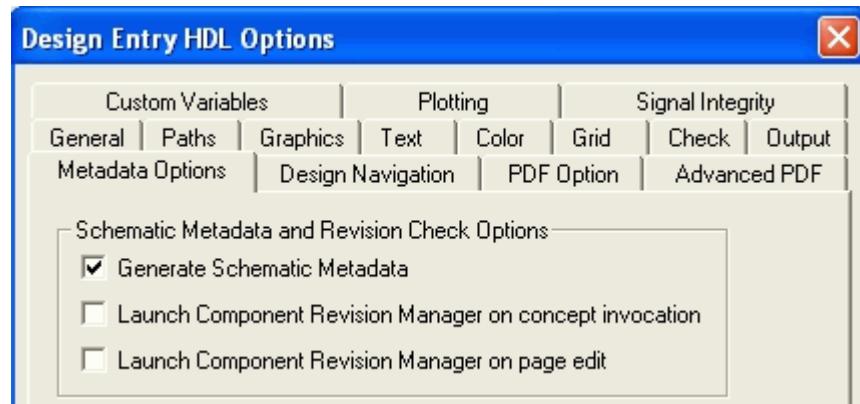
**Note:** You do not have any user interface controls (commands or button) to launch the *Component Revision Manager* window. It automatically launches when differences exist between the metadata of your schematic and library cells.

### Setting Preferences for Metadata Creation

Before you start using Component Revision Manager, make sure that Allegro Design Entry HDL has been configured to create metadata for your design projects.

1. Choose *Tools – Options*. The *Design Entry HDL Options* dialog box appears.
2. Select the *Metadata Options* tab.

**Figure 7-1 Design Entry HDL Options dialog box with Metadata Options tab selected**



3. To generate schematic-related metadata in Allegro Design Entry HDL, select the *Generate Schematic Metadata* check box in the *Schematic Metadata and Revision Check Options* section.

**Note:** The metadata creation, by default, is set to off. Use the `GENERATE_SCH_METADATA 'ON'` directive in the cpm file to enable metadata creation, by default.

4. To ensure that your schematic is automatically checked for differences between the library cells and schematic cells when:

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

---

- a. You launch Allegro Design Entry HDL, select the *Launch Component Revision Manager on Design Entry HDL Invocation* check box in the *Schematic Metadata and Revision Check Options* section.
  - b. You edit a page, select the *Launch Component Revision Manager On page Edit* check box in the *Schematic Metadata and Revision Check Options* section.
5. Click *OK*.

**Note:** By default, the ability to check for differences between the schematic and library cells is set to off. Alternatively, you can use the following directives (to be specified in cpm file) to control the default revision check behavior.

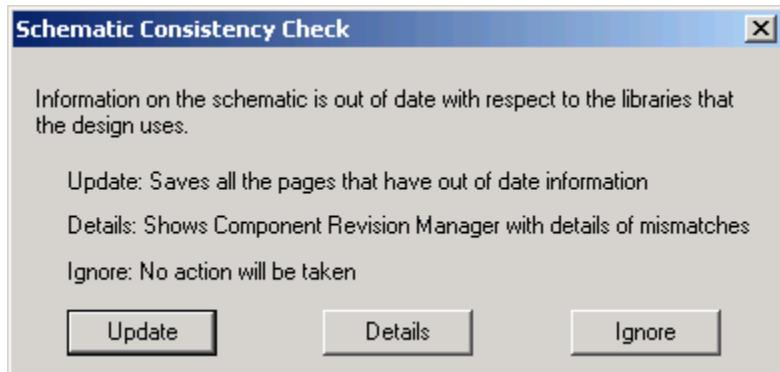
Directive	Description
SYNC_ON_STARTUP 'ON'	Checks for differences between the schematic and library cells at Design Entry HDL startup.
SYNC_ON_PAGE_EDIT 'ON'	Checks for differences between the schematic and library cells every time you move from one page to another in design.

**Note:** After enabling metadata creation, make sure that you run the `hier_write` command (by entering it in the console window of Design Entry HDL) on the design. This will update the metadata for the existing components, and will create metadata for the newly added components, if any. This ensures that the Component Revision Manager detects the differences between the schematic and library cells correctly.

### Checking Schematic Consistency

As soon as you open a design project or edit a page of your schematic, Allegro Design Entry HDL identifies the obsolete cells existing in your schematic or page, and the *Schematic Consistency Check* dialog box appears with the message that differences exist between the schematic cells and the library cells.

**Figure 7-2 Schematic Consistency Check dialog box notifies you of the differences between the schematic and reference library cells**



The *Schematic Consistency Check* dialog box contains the following buttons:

- *Update*: Click *Update* to save all the pages in your schematic with the up-to-date library cell information without viewing the differences.
- *Details*: Click *Details* to see the differences in the *Component Revision Manager* window.
- *Ignore*: Click *Ignore* to retain the differences as it is, and continue to working with the design project.

## Using Component Revision Manager

The *Component Revision Manager* window appears when you click *Details* in the *Schematic Consistency Check* dialog box. It helps you perform the tasks that involve:

- Highlighting an Instance
- Updating Design
- Saving Component Revision Details

## Changing Views in the Schematic Cells pane

You can arrange the schematic cell information according to the various views of a schematic. The available views are: page-wise, instance-wise, and block-wise.

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

#### Page-wise View

- In the *Component Revision Manager* window, choose *Option – Navigation Option – Page(s)* to view the schematic cells (in the *Schematic Cells* pane) on a per-page basis. In this view, only the page number (where a cell is located) of the design appear under the *Page(s) Impacted* header column. The page details follow the syntax: @ <library name> . <block name> (view name) : page <page number> .

**Figure 7-3 Page-wise view of the Schematic Cells pane**

Sch. Version	Status	Page(s) Impacted
*	*	*
5.0.0	Major Update	@sync1_lib.sync1(sch_1):page1
4.0.0	Major Update	@sync1_lib.sync1(sch_1):page2

#### Instance-wise View

- In the *Component Revision Manager* window, choose *Option – Navigation Option – Instance(s)* to view the schematic cells (in the *Schematic Cells* pane) on a per-instance basis. In this view, the instance name of a cell appears under the *Instance(s) Impacted* header column. The page details follow the syntax: @ <library name> . <block name> (view name) : page <page number> \_ <instance number>.

**Figure 7-4 Instance-wise view of the Schematic Cells pane**

Sch. Version	Status	Instance(s) Impacted
*	*	*
3.2.0	Minor Update	@sync1_lib.sync1(sch_1):page1_i3
3.2.0	Minor Update	@sync1_lib.sync1(sch_1):page1_i4
3.2.0	Minor Update	@sync1_lib.sync1(sch_1):page1_i7
3.2.0	Minor Update	@sync1_lib.sync1(sch_1):page1_i8

#### Block-wise View

- In the *Component Revision Manager* window, choose *Option – Navigation Option – Block(s)* to view the schematic cells (in the *Schematic Cells* pane) on a per-block basis. For example, your schematic can contain blocks such as top, bottom, or low. In this view, the block name (to which the schematic cell belongs) appears under the *Block(s) Impacted* header column.

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

**Figure 7-5 Block-wise view of the Schematic Cells pane containing a block named “sync1”**

Sch. Version	Status	Block(s) Impacted
*	*	*
4.0.0	Major Update	sync1

**Note:** If the schematic contains blocks that are added from a “read only” location, the corresponding cell rows in the right pane have a gray background.

### Highlighting an Instance

Highlighting an instance lets you view the placement of schematic cells on the schematic. To do so:

1. Select a row in the *Schematic Cells* pane.
2. Right-click on the row. The pop-up menu appears. Choose *Highlight Instance*. Alternatively, choose *Option – Highlight Instances*.

**Figure 7-6 Choosing Highlight Instance highlights an instance on the schematic**

Sch. Version	Status	Page(s) Impacted
*	*	*
5.0.0	Major Update	@sync1_lib.sync1(sch_1):page1

Update Design  
Highlight Instance

3. The schematic cell highlights on the schematic in red.

**Note:** Highlighting a row (in either page-wise or block-wise view of the *Schematic Cells* pane) can result in highlighting more than one schematic cell.

### Updating Design

Updating a design automatically resolves all the existing differences between the reference library cells and the schematic. The complete schematic is updated with all the cells in the reference library. To update the design, do the following:

1. Select a row in the *Schematic Cells* pane.
2. Right-click on the row. The pop-up menu appears. Choose *Update Design*. Alternatively, choose *Option – Update Design*.

**Figure 7-7 Choosing Update Design updates all the instances of the schematic**

Sch. Version	Status	Page(s) Impacted	
*	*	*	
5.0.0	Major Update	@sync1.lib.sync1(sch_1):page1	<b>Update Design</b>
4.0.0	Major Update	@sync1.lib.sync1(sch_1):page2	Highlight Instance

3. As soon as the design is updated, all the rows representing schematic cells are removed from the *Component Revision Manager* window.

**Note:** Schematic cells of read-only blocks are not updated.

**Note:** For more information, see *Allegro Design Workbench Library Revision Manager User Guide*.

### Saving Component Revision Details

Component Revision Manager also lets you save differences between the schematic cells and the reference cells in an ASCII text file named `syncStatus.txt`. To save the file, choose *File – Save*.

## Allegro Design Entry HDL User Guide

### Working with Libraries and Components

**Figure 7-8 Sample syncStatus.txt file**

```
FILE_TYPE = SYNC_DIFF
{ Command Line Sync Engine run on 09-Jun-2005 AT 19:24:32 }

Cell Name = 'as27'
Cell Version in Library = '1.1.0'

    Cell Version used in Schematic = '1.0.0'
    Status = 'Minor Update'
    Instance(s) Impacted :-
        Instance = '@parts_lib1.blk_ro(sch_1):page1_8p'
*****
|_
Cell Name = 'f153'
Cell Version in Library = '2.0.0'

    Cell Version used in Schematic = '1.0.0'
    Status = 'Major Update'
    Instance(s) Impacted :-
        Instance = '@parts_lib1.blk_ro(sch_1):page1_i15'
*****
Cell Name = 'pres'
Cell Version in Library = '2.0.0'

    Cell Version used in Schematic = '1.0.0'
    Status = 'Major Update'
    Instance(s) Impacted :-
        Instance = '@parts_lib1.blk_ro(sch_1):page2_i121'
*****
```

## Setting Default View

Component Revision Manager also lets you set the default view of the *Schematic Cells* pane. You can do so using the NAVIGATION\_OPTION directive in the cpm file. The following table lists the ways in which you can use this directive.

**Table 7-1**

Using the directive...	Lets you...
NAVIGATION_OPTION 'page'	Show the occurrences of schematic cells (in the <i>Schematic Cells</i> pane) on a per-page basis. When you specify this directive, only the page number (where the cell is located) of the design appears under the <i>Page(s) Impacted</i> header column.
NAVIGATION_OPTION 'instance'	Shows the occurrences of schematic cells (in the <i>Schematic Cells</i> pane) on a per-instance basis. When you specify this directive, the page number and the instance name of the cell appears under the <i>Instance(s) Impacted</i> header column.
NAVIGATION_OPTION 'block'	Shows the occurrences of schematic cells (in the <i>Schematic Cells</i> pane) on a per-block basis. For example, your schematic may contain blocks such as top, bottom, or low. When you specify this directive, the block name (to which the schematic cell belongs) appears under the <i>Block(s) Impacted</i> header column.

## Hiding and Displaying Details Pane

By default, the Details pane is visible to you.

1. To hide the pane, choose *Option – Details Window*.
2. To display the pane, choose *Option – Details Window*, again.

## Exiting Component Revision Manager

To quit Component Revision Manager, choose *File – Close*.

## **Allegro Design Entry HDL User Guide**

### Working with Libraries and Components

---

---

# Working with Properties and Text

---

This section describes the procedures for working with properties and text in Design Entry HDL.

## About Properties

Properties (also called attributes) are used to convey information about a design. There are three types of Design Entry HDL properties.

- System properties - these are system-assigned properties that are attached to wires (nets) or pins.
- Schematic properties - these are user-assigned properties that can be attached to components, wires (nets), or pins.
- Symbol properties - these are librarian-assigned properties that are attached to a part through the part symbol drawing, the `chips.prt` file, or the part table file (PPT). Definitions of the Design Entry HDL properties include these classifications. In addition, you can use extensions to Design Entry HDL properties.

Properties consist of a name and value. Using the *Text – Property Display* menu command, you specify whether Design Entry HDL displays the property name alone, the property value alone, both, or neither. A property name can combine letters, numbers, and underscores, but the first character must be alphabetic. A property value can include space and punctuation marks.

The following types of property value entries are supported:

25oct98 10:31:46.03

(size + 4) / 5 + 35 MOD A

This is a long property value

@#\$%\*( ) { } [ ] < >

**Note:** The maximum permissible length for a property name is 31 characters and that for a property value is 255 characters.

While Design Entry HDL does not interpret most properties (it passes them to other system tools), it does interpret these properties:

- LAST\_MODIFIED
- PIN\_NAME
- SIG\_NAME
- Properties added by *Tools – Back Annotate*, *Component – Section*, and *Component – Swap Pins*
- PATH

## Locking Properties

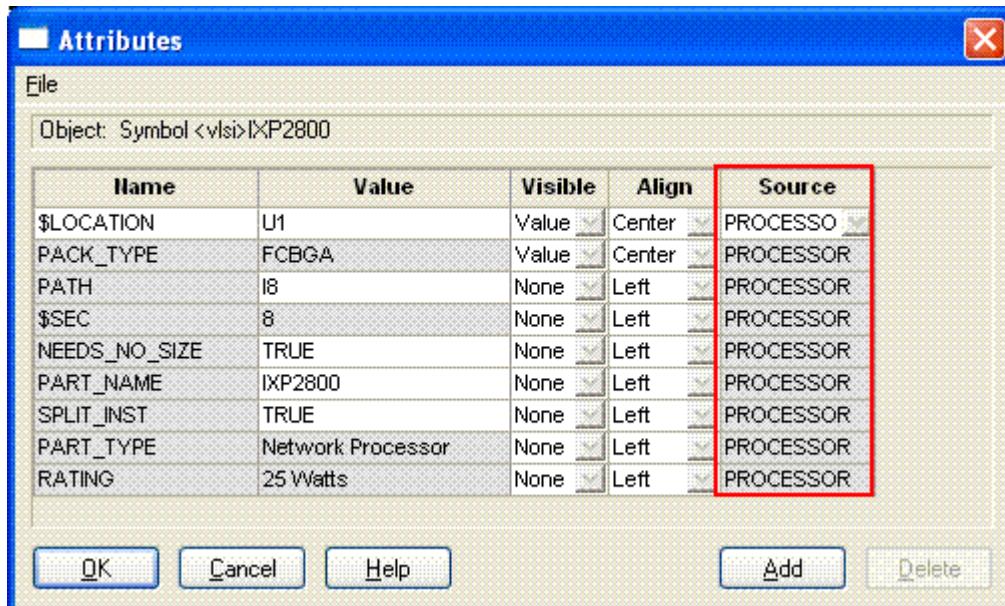
In DE-HDL, key properties cannot be edited or deleted. As a result, properties locked for property names and values are key properties. You cannot edit key properties in the schematic through any of these — the Attributes dialog box, change command, text editor, scripts, or the Global Modification dialog box.

Injected properties can be edited and deleted, so properties that are not locked are injected properties. However, injected properties that are symbol properties can be edited but cannot be deleted. As a result, these properties are only locked for the property name.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

In the Attributes dialog box, the key property names and values are grayed out. For more information about the Attributes dialog box, see [Master and In-context Properties](#) on page 321



You can see that the names and values of key properties for the symbol — PART\_TYPE, and PACK\_TYPE — are non-editable (grayed out). The values of injected properties — LOCATION, and PART\_NAME — are editable.

Now, if you want to change a physical component, you do not change the key property in the schematic. Instead, you select another physical component from Part Information Manager. This ensures that the components on the schematic are always synchronized with those in the libraries.

While the properties are locked by default, you can use the ALLOW\_PROPERTY\_LOCKING directive to control the locking and unlocking of the key properties in the schematic. The ALLOW\_PROPERTY\_LOCKING is set in the START\_CONCEPTHDL section of the .cpm file. By default, it is set to ON, which means the properties are locked. You can set it to OFF to unlock the properties in the schematic.

## Copying Properties

You can copy most properties on the drawing. Default properties and properties that you add are automatically included in copies made of components. If you change a default property on a component, the property on a copy of the component also changes. For more information on copying properties see [Master and In-context Properties](#) on page 321

Properties you cannot copy are:

- pin properties
- wire properties
- unnamed signals
- PATH properties

## Moving Properties

Design Entry HDL moves properties with the object to which they are attached, or you can move properties independently. You cannot move the PATH property between drawings.

## Adding Properties

### ***To add one property at a time***

1. Choose *Text – Property*.
2. In the *Property* dialog box, enter a name in the *Property Name* box and a value in the *Property Value* box.
3. Click *OK*.
4. Click the object to which you are attaching the property.
5. Click near the object to indicate where to display the property information.

As the default, Design Entry HDL displays only the property value. Choose *Text – Property Display* to modify how properties are displayed.

**Note:** You can also run the *property* command using this stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#) on page 131.

You can also add a property using the *property* command.

**Note:** To propagate the new properties to the board, you must define the properties in PCB Editor as well.

1. Choose *Setup – Property Definitions* option in PCB Editor.

The *Define User Properties* dialog box opens.

2. Define the properties so that PCB Editor adds them to the board for the schematic. For more details on how to use the *Define User Properties* dialog box, click the *Help* button in the dialog box.

#### **Adding properties using the Attributes drop-down list**

Design Entry HDL provides you a list of valid properties that you can add to any part or net on a schematic. This feature saves you from accidentally misspelling the names of Design Entry HDL-supported properties and running into errors later. These properties are available in the form of a drop-down list in the Attributes dialog box, which you use to assign additional properties to a part or a net. If you select a part instance, a list of approved part instance properties is displayed. Similarly, if you select a net, a list of net properties is displayed. For information on adding properties in release 16.5 and onwards, see [Master and In-context Properties](#) on page 321

#### **To add properties together using the Attributes drop-down list**

1. Select the part or the net with which you want to associate a new property.
2. Choose *Text – Attributes*

Alternatively, you can choose *Attribute* from the popup menu which appears when you right-click a part ([Figure 8-1](#) on page 308) or a net ([Figure 8-2](#) on page 309). This displays the Attributes dialog box.

3. Click the *Add* button.

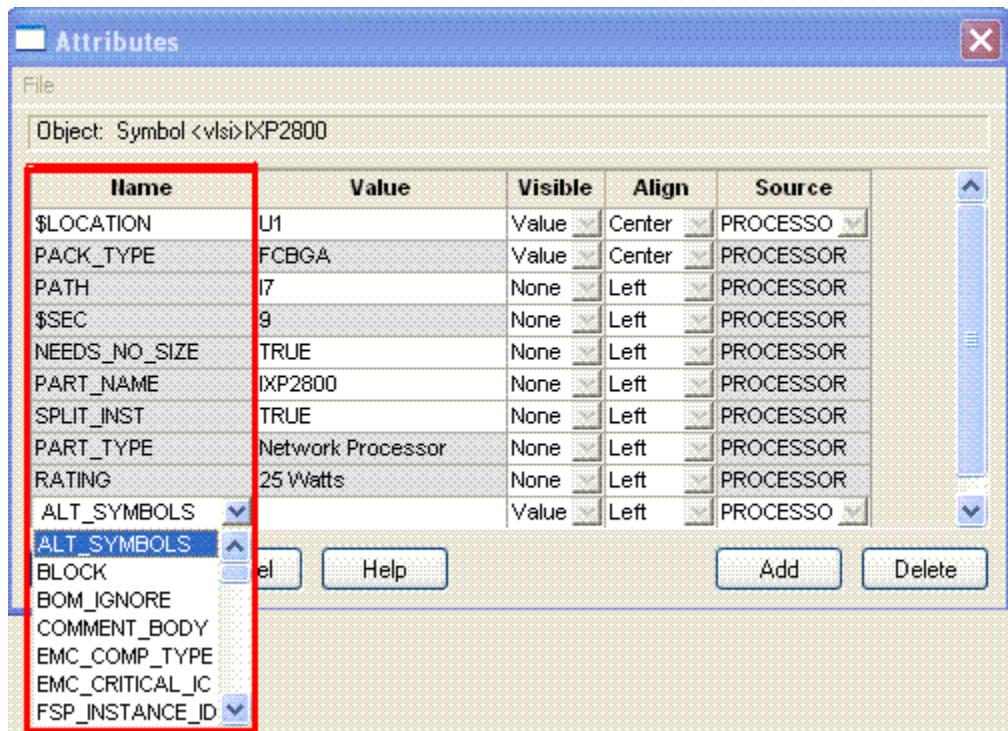
A new row is added to the existing list of properties.

4. Click in the *Name* box.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

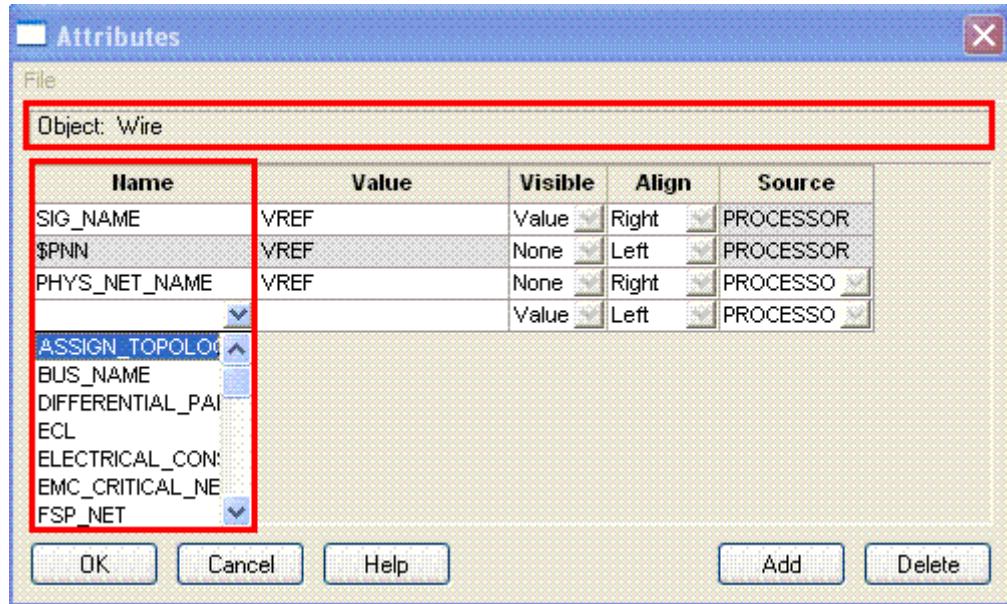
A drop-down list is displayed. Depending on your selection, the drop-down list shows properties you can assign to the selected part or net.



**Figure 8-1**

## Allegro Design Entry HDL User Guide

### Working with Properties and Text



**Figure 8-2**

1. Browse through the list of properties and select the property that you want to attach to the part or the net.
2. Specify other details, such as value, visibility, and alignment details and click OK.

The property is added to the selected part or net.

**Note:** To propagate a new property to the board, you must define the property in PCB Editor as well.

1. Choose the *Setup – Property Definitions* option in PCB Editor.

The *Define User Properties* dialog box opens.

2. Define the property so that PCB Editor adds it to the board for the schematic. For more details on how to use the *Define User Properties* dialog box, click the *Help* button in the dialog box.

3. Displaying and Modifying Property Attributes

To display property attributes

1. Choose *Text – Attributes*.

#### 2. Select an object.

The object you select is highlighted, and the *Attributes* dialog box displays attributes for the object.

**Note:** You can also run the `attribute` command using this stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#) on page 131.

#### **To display net properties from net synonyms**

##### 1. .

##### 2. Choose *Text – Attributes*.

##### 3. Select the net.

If you need to find the net in your design, use *Tools – Global Navigate* to find a net that spans the design hierarchy or multiple pages in a design.

The net is highlighted. The *Attributes* dialog box displays the object name and a list of properties and property values, as well as the canonical name of the net synonym for which the properties were defined.

#### **To modify a property**

##### 1. Click a property in the *Attributes* dialog box.

##### 2. Highlight a property name or value and type a new name or value, or adjust property visibility and alignment as needed.

##### 3. Click *OK* for modifications to appear on the drawing, or click *Cancel*.

#### **To delete a property**

##### 1. Click a property in the *Attributes* dialog box.

##### 2. Click *Delete*.

The property is deleted from the *Attributes* dialog box.

##### 3. Click *OK* to delete.

## Making an Attribute File

1. Display the *Attribute* dialog box.
2. In the *Attribute* dialog box, choose *File – Load Attributes*.
3. Select a file in the directory browser that appears and click *Open*.

Properties already on the object are listed in the *Attributes* dialog box first, followed by properties from the attribute file you loaded.

4. Delete unnecessary properties from the *Attributes* dialog box.
5. Change the values of properties and display options if necessary.
6. In the *Attribute* dialog box, choose *File – Save Attributes*.
7. Specify a new filename in the *File* box of the *Save As* window that appears.



Click *Cancel* to close the *Attribute* dialog box if you do not want to apply these properties to the object you selected to open the dialog box.

## Adding a URL

Design Entry HDL recognizes a URL in the schematic. If a property value or a note in the schematic is a URL, Design Entry HDL shows it as an active Internet link to the corresponding website.

You can add a URL in the following ways:

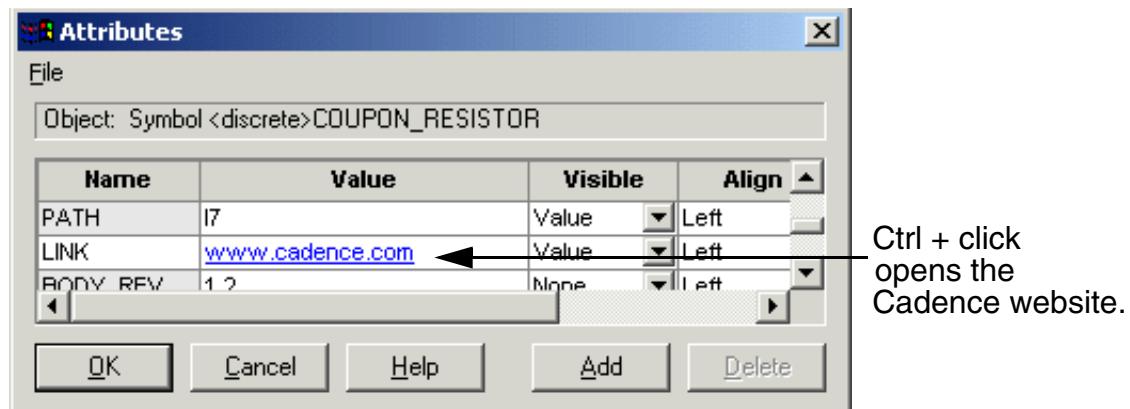
- as a property value in the *Attributes* dialog box
- as a note on the schematic
- as a property value in the *Physical Part Filter* dialog box

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

#### Example 1

Ctrl + click (LMB) on the following property value opens the Cadence website.



#### Example 2

Double-clicking on the following note opens the Cadence PCB website.



#### Example 3

For a component COUPON\_RESISTOR, add a property in its .ptf file as follows:

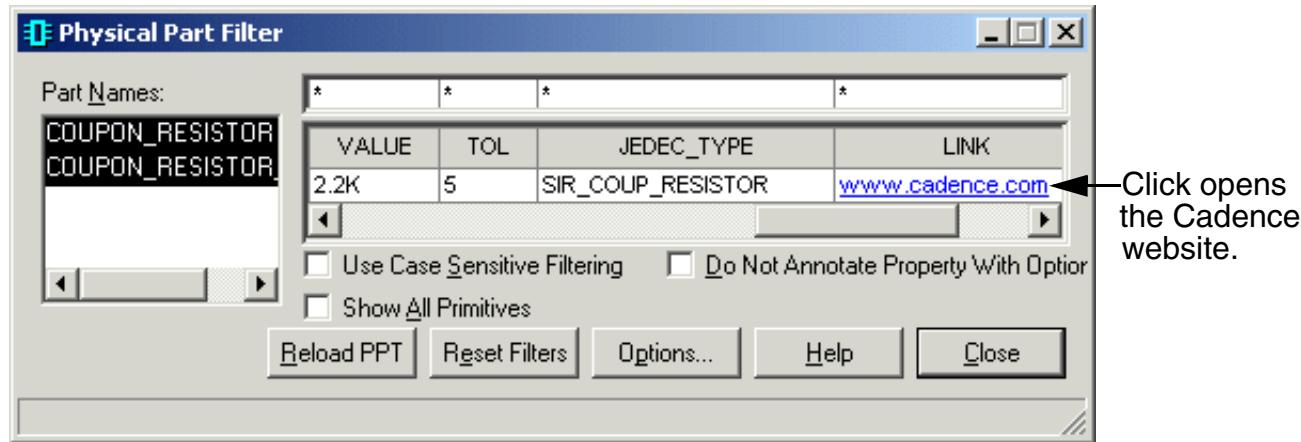
```
:VALUE TOL POWER PACK_TYPE = PART_NUMBER VALUE TOL JEDEC_TYPE LINK;  
2.2K 5% 1/10W SMT = 067-222-050 2.2K |5 SIR_COUP_RESISTOR www.cadence.com
```

Property LINK = www.cadence.com

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

When COUPON\_RESISTOR is added in physical mode, Design Entry HDL recognizes the property LINK as a URL as shown below:



## Adding Text

### To add individual text lines

1. Choose *Text – Note*.

The *Note* dialog box is displayed.

2. Type text in the *Notes* section of the dialog box.

You can enter several strings before placing them on the drawing.

3. Enable *Queue* to place text lines in the order you enter them. Enable *Select* to choose a text line and place it (the text you select for placement is highlighted in the *Note* dialog box).

**Note:** You can also run the *note* command using this stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#) on page 131.

### To add text from a file

1. Choose *Text – File*.

The *Open* dialog box appears.

2. Navigate to the text file you want and highlight the file name.
3. Click *Open*.
4. Click in a blank space in the drawing.

To include a blank line, type a space on the line in the file before adding it to the drawing.

The line is otherwise ignored when the file is added to the drawing. Tabs and non-printable characters in the file are displayed as a # sign.

**Note:** You can also use the following command to attach a note from file in the schematic:

```
filenote <filename> (<x>,<y>)
```

Save the command in a .scr file, if needed.

## Modifying Text

### **To modify single text items**

1. Choose *Text >Change*.
2. Select the text to be modified.  
**Note:** You can select multiple text items.
3. Use the arrow keys to position the cursor in the text line.
4. Press any character key to add text, press *Del* or *Backspace* to delete a character, press *Ctrl+K* to delete text from the cursor to the end of the line, press *Enter* or *Return* to move to the next text selection, or press *Esc* to end changes. Alternatively, you can use the pop-up menu to display the text change editor.
5. Choose *Done* from the pop-up menu when you have finished making modifications.

**Note:** You can also run the *change* command using this stroke pattern:



For more information on strokes and a list of available stroke patterns, see [Running Commands with Strokes](#) on page 131.

#### **To modify text on grouped objects**

1. Choose *Group – Set Current Group* to specify a group.

**Note:** You can highlight the current group to emphasize which group you are working with (*Group – Highlight [x]*).

2. Choose *Group – Text Change [x]*.

**Note:** A message might display with a reminder that you cannot change section properties.

3. Right-click and choose *Editor* from the pop-up menu.

4. Edit text in the text change editor (*Ctrl+E*) and save (*File – Save* or *File – Save As*) before exiting the editor (*File – Exit*).

Text is picked from the canvas in the order of right to left and top to bottom.

5. Right-click and choose *Done* from the pop-up menu.

#### **Resizing Text**

##### **To increase text size**

1. Choose *Text – Increase*.
2. Click the text line whose size you want to increase.

##### **To decrease text size**

1. Choose *Text – Decrease*.
2. Click the text line whose size you want to reduce.

#### **Setting the Text Size**

##### **To set the text size**

1. Choose *Text – Set Size*.

The *Text Set Size* dialog box appears.

2. Enter the text size in inches.

3. Click *OK*.
4. Click a note, a URL, a property, or a group to change the text size.

Design Entry HDL changes the text size to the size you have specified.

**Note:** Design Entry HDL accepts the text size between 0.009 inches and 1.74 inches.

## Changing the Text Editor

1. Choose *Tools – Options*.
2. Select the *Text* tab.
3. In the *Text Change Editor* box, specify the editor you want to use.
4. Click *OK*.

## Adding Port Names from the Corresponding Symbol

1. Choose *Text – Port Names*.
2. Click in the drawing.
3. Pin names from the symbol are listed.

## Swapping Notes or Properties

1. Choose *Text – Swap*.
2. Click a note or property.
3. Click a second note or property.
4. Design Entry HDL swaps the text line in one location on the drawing with the text line in the other location.

## Reattaching a Property from One Object to Another

1. Choose *Text – Reattach*.
2. Click a property to be reattached.
3. Design Entry HDL draws a line from the property to the current cursor position.
4. Click an object that will be the new attachment point for the property.

5. If required, choose *Edit – Move* to move the property closer to its new attachment point.

## Specifying Property Display for a Specific Object

1. Choose *Text – Property Display – Name/Value/Both/Invisible*, depending on whether you want to
  - Display only the property *Name*
  - Display only the property *Value*
  - Display both the property name and the value.
  - Make properties *Invisible*
2. Select a property.

## Specifying Property Display for Objects Globally

The Global Property Visibility feature helps you control the name and/or value visibility of a property globally. You no longer need to change the visibility of a specific property for each component, net, or individually. You can make a property with a specific value visible globally across the various pages of a design. For example, for all the resistors used in a design with the SIGNAL\_MODEL property set to DEFAULT\_RESISTOR\_22OHM\_2\_1, you can make the property visible or invisible in the entire design. You can also control the scope of display of a property based on whether you want to make a specific property visible on a page, a module, or the entire design. You can even define a page-range in which you want the property to be visible.

To change the visibility of a property:

1. Choose *Text – Global Property Display*.

The Global Property Visibility Change dialog box is displayed. Use this dialog box to change the visibility of a property.

**Note:** The *Text – Global Property Display* menu command is disabled in the Occurrence Edit mode.

2. Specify the name of the property in the Property Name field.

For example, to change the visibility of the SIGNAL\_MODEL property, type SIGNAL\_MODEL. You can also use a combination of a few letters in the property name and an asterisk instead of typing the complete property name.

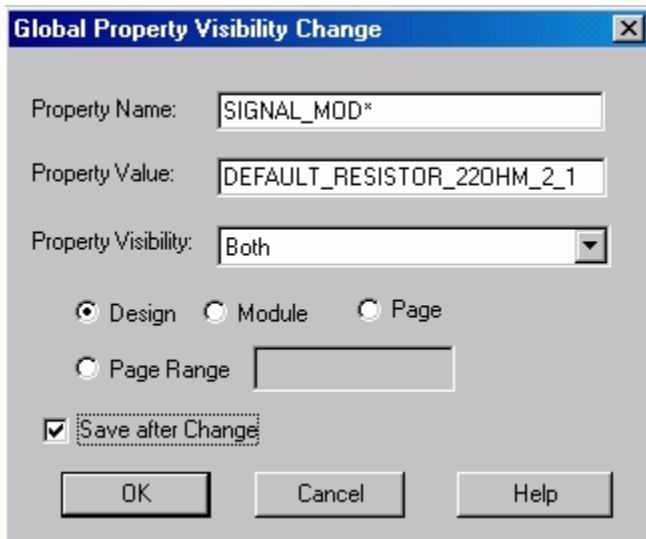
## Allegro Design Entry HDL User Guide

### Working with Properties and Text

- Specify the value of the property that should match in order for the property to be displayed in the Property Value field.

For example, to change the visibility of all the resistors with the SIGNAL\_MODEL property set to DEFAULT\_RESISTOR\_22OHM\_2\_1, type *DEFAULT\_RESISTOR\_22OHM\_2\_1* in the Property Value field.

- Select the required visibility from the Property Visibility field. You can choose from Invisible, Name, Value, and Both.
- Define the scope of the visibility by selecting the relevant option. You can choose from Design, Module, Page, and Page Range.
- Select the *Save after Change* check box if you want to save the changes to the design after changing the visibility.



- Click OK.

## Specifying Visibility of Pin Properties

Design Entry HDL lets you specify the default visibility of pin properties. When you add a component on the schematic, the visibility of the properties on the pins of the component is controlled by the visibility option you have selected.

To set the default visibility for pin properties, do the following:

- Choose *Tools – Options*.

The *Design Entry HDL Options* dialog box appears.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

2. Select the *Text* tab.
3. Select the pin property visibility option.

Select	To
<i>Invisible</i>	<p>Make all pin properties invisible when you add a component in your schematic.</p> <p>The pin property will be invisible in your schematic even if the visibility of the property on the pin on the symbol for the component is set to <i>Name</i>, <i>Value</i> or <i>Both</i>.</p>
<i>Defined By Component</i>	<p>Make all pin properties visible or not depending on how the property visibility is defined on the pin on the symbol for the component.</p>

4. Click *OK* to save the changes.

## Specifying Visibility of Pin Numbers

Pin numbers are assigned to the pins of a component in a design when you package the design. You may not want the pin numbers on some components to be displayed on the schematic because it clutters the schematic. You may also want the pin numbers on some components to be always visible by default because you want to plot the pin numbers on such components.

The value of the `$PN` property on a pin is the pin number for the pin. Design Entry HDL allows you to specify the default visibility of pin numbers on a component in the schematic by adding the `$PN=?` and `$PN=#` properties on the pins on the symbol for a component. For more information, see the following sections:

- [Making Pin Numbers Visible by Default](#) on page 320
- [Making Pin Numbers Invisible by Default](#) on page 320

### Making Pin Numbers Visible by Default

To make the pin numbers on a component visible by default, add the `$PN=?` placeholder property on the pins on the symbol for the component.

Before you add the component on the schematic, set the *Pin Property Visibility* option to *Invisible* in the *Text* tab of the *Design Entry HDL Options* dialog box.

After you package the design, the pin numbers will be visible on the schematic in the Occurrence Edit mode. When you run *Tools – Back Annotate*, the pin numbers will be visible on the schematic in the Hierarchy and Expanded mode.

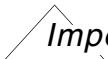
After you package the design, the `$PN=?` property becomes `$PN=<pin_number>`. If you now change the visibility of `$PN` property on the component to *None*, the pin number will not be visible anymore.

### Making Pin Numbers Invisible by Default

To make the pin numbers on a component invisible by default, add the `$PN=#` placeholder property on the pins on the symbol for the component.

Before you add the component on the schematic, set the *Pin Property Visibility* option to *Invisible* in the *Text* tab of the *Design Entry HDL Options* dialog box.

After you package the design, the pin numbers will remain invisible on the schematic. The `$PN=#` property becomes `$PN=<pin_number>`. If you now change the visibility of `$PN` property to *Value*, the pin number will become visible on the schematic.



#### Important

Setting `$PN = #` does not set the pin numbers to invisible when the part is sectioned using the *Section* command. When you set `$PN=#`, whenever packager is run and the package information is backannotated on the schematic, `$PN` setting is evaluated before updating the pin number on the schematic symbol instance.

If the value is set to `#`, the pin number is not annotated. However, when you manually section a part, `$PN` setting is not evaluated before updating the pin number on the schematic symbol instance. Therefore, pin number is annotated regardless of the value you set for `$PN`.

## Master and In-context Properties

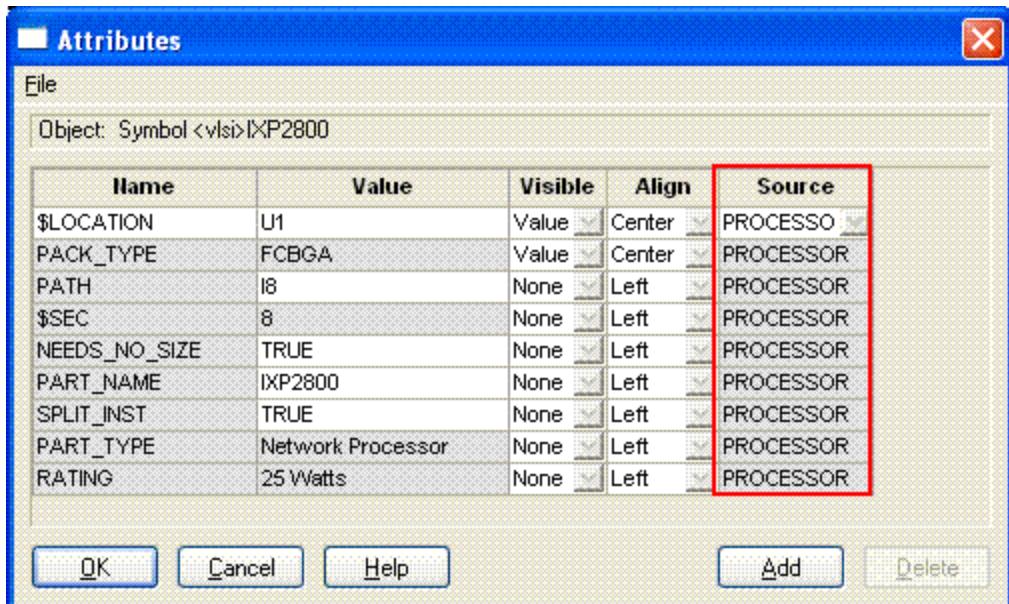
Master and in-context properties are directly stored in the block on which the object exists, or at the root (top) level design in context of which the block has been instantiated.

When you store the property in the block itself, it is visible in all the instances of the block and any change to the property in the block is propagated to all the instances of the block. If you want to store property changes only on a particular instance or occurrence of a block, the property needs to be stored in the root (top) level design. This property is stored on the instance in context of the root (top) level design.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

You can also choose the location for storing property changes. The location where the property is stored is known as the **Source** of the property and it is visible in the Attribute form. Source displays the name of the block where the property is stored.



When a property is added to a block, the default source for that block/root property is selected. This default is selected based on the type of the block being added (flat, non-replicated, or replicated). You can change the source by clicking on the drop-down list and selecting another source block/root. Depending on the source, properties are called occurrence properties or block-level properties.

Block-level changes are changes in the property value that are stored in the block directly. These changes are visible in all instances of the block.

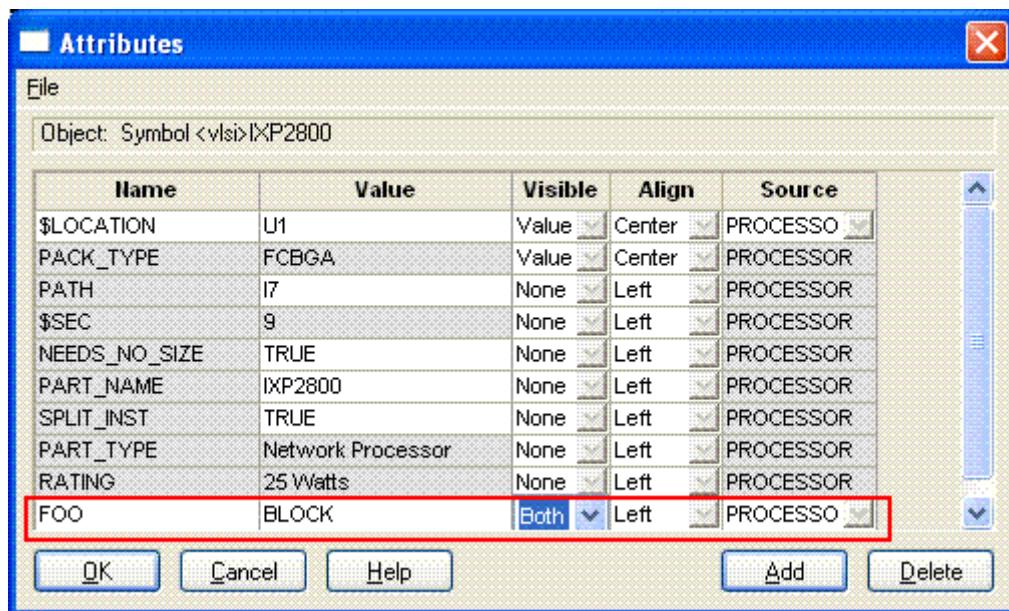
Occurrence property changes are changes in the property value that are stored in the root (top) level design. These properties are added to a specific instance (occurrence) of the block and are visible only on that specific instance of the block.

The next section explains how to add block-level and occurrence properties. For the purpose of this document, a sample design example is used, where Processor is the root design. The design consists of two blocks—RAMBUS\_CHANNEL and QDR\_CHANNEL. RAMBUS\_CHANNEL is instantiated only once in this design and QDR\_CHANNEL is instantiated four times.

## Adding Properties to a Flat Design or Root (Top) Level Design

Any property that you add in a flat design or root (top) level design is saved as a block property by default. As there is only one design level, the property is stored only in that design and you do not have an option to save the property as an occurrence property.

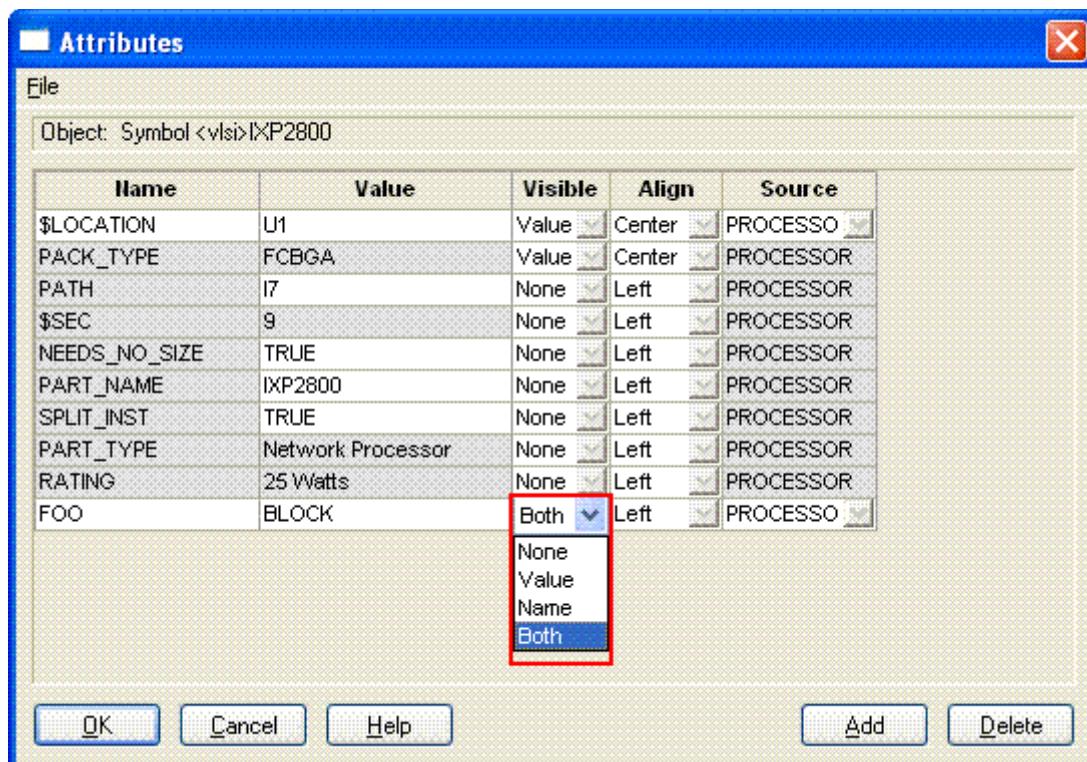
In the following figure, a new property, FOO, is added with BLOCK as the value. It is saved with PROCESSOR as the Source, that is, as a block property. The Source column drop-down has only one value—PROCESSOR.



After you add the new property, you can choose to view the name, or value, or both the name and value, on the page.

## Allegro Design Entry HDL User Guide

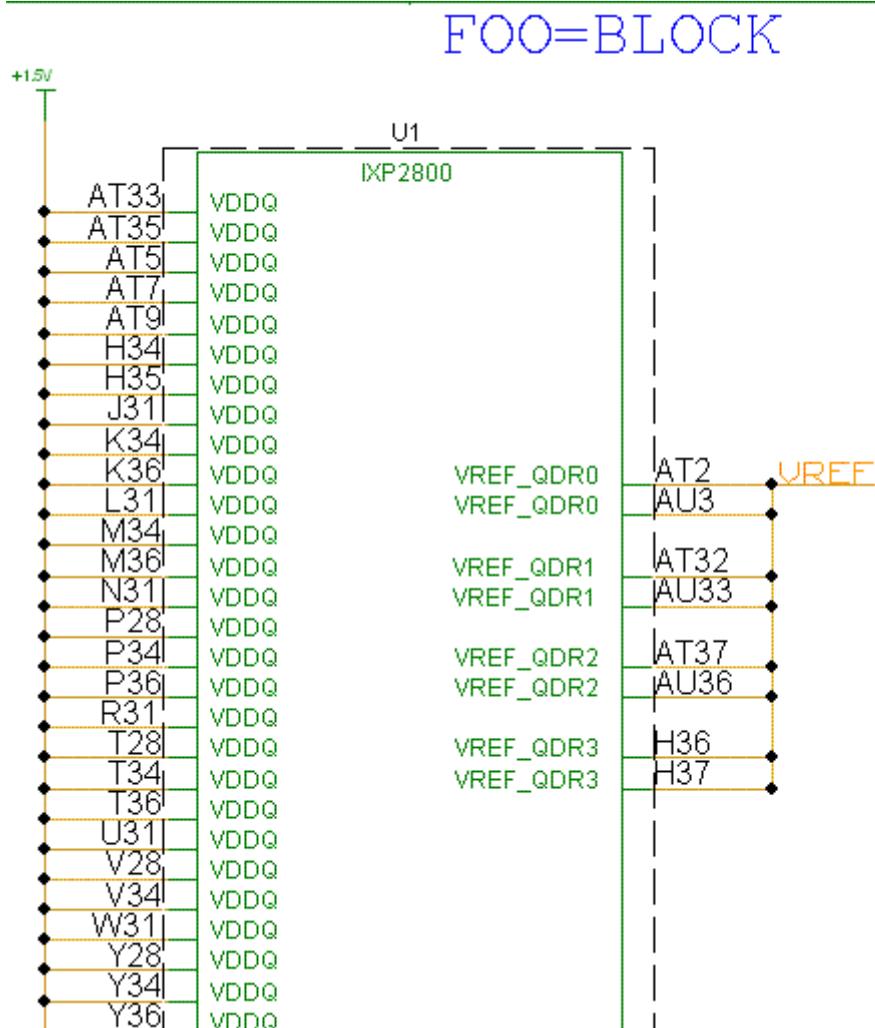
### Working with Properties and Text



In the following figure, you can see the name and value displayed on the page.

# Allegro Design Entry HDL User Guide

## Working with Properties and Text

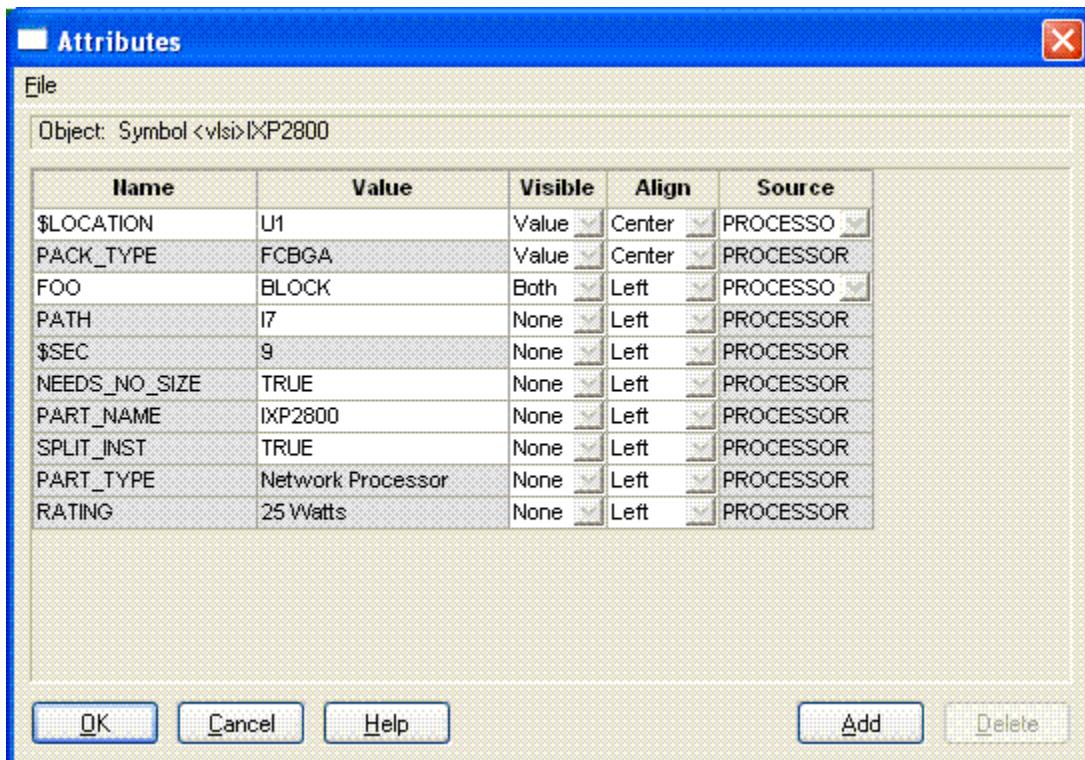


## Allegro Design Entry HDL User Guide

### Working with Properties and Text

To view attributes and add a property to a flat or root (top) level design, do the following:

1. In the design, view attributes on any of the components in the Processor block.

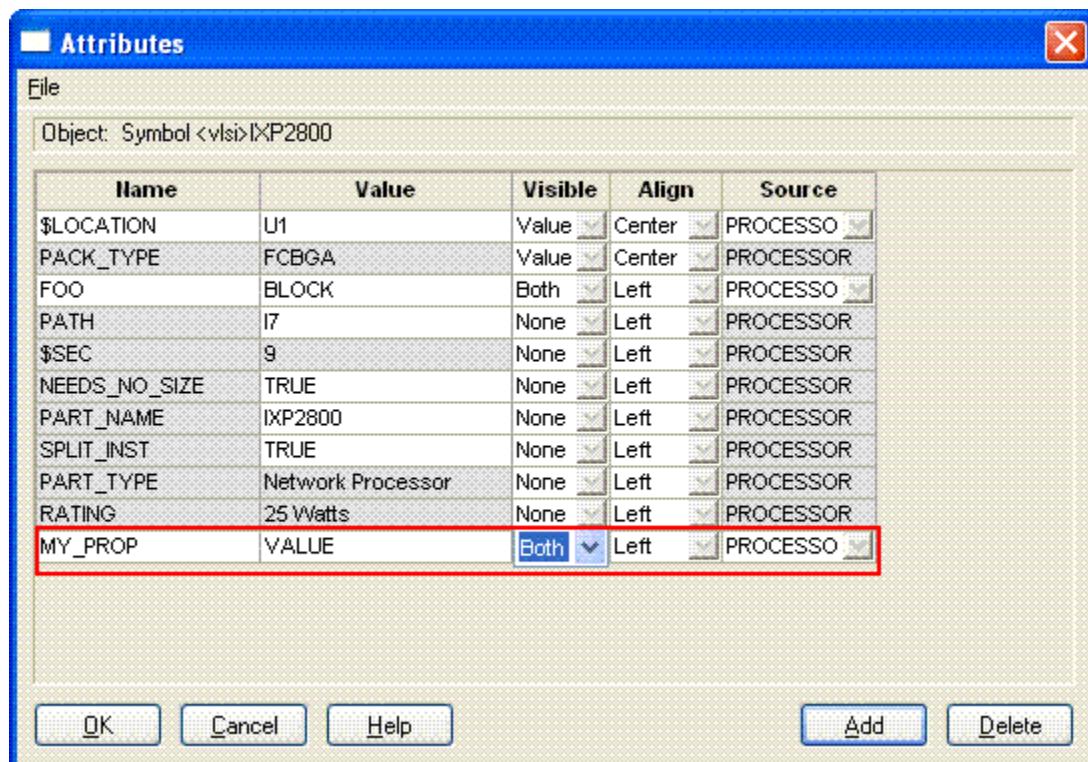


Note the new column for the Source of the property. Also, note that all the properties have the source set as Processor, which is the root (top) level design as well as the current block name.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

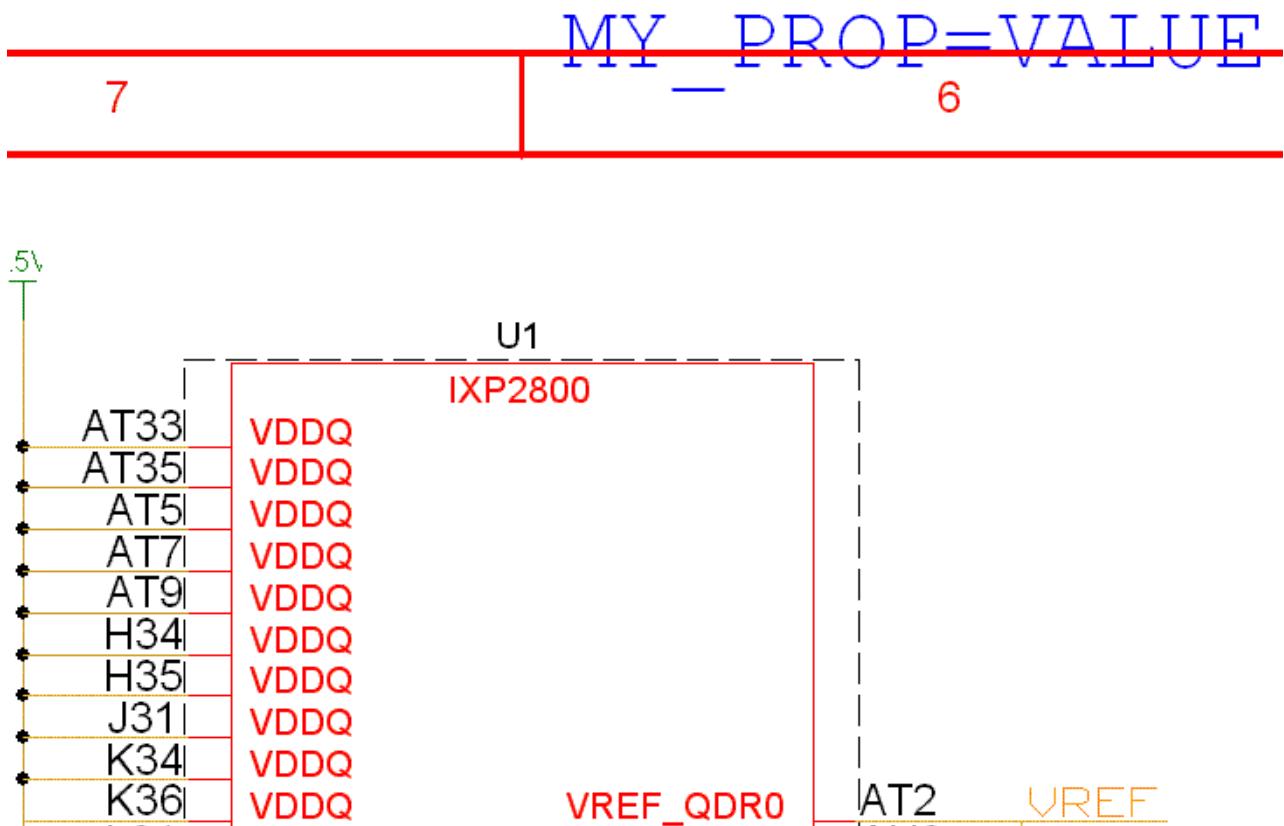
2. Click the *ADD* button to add a new attribute.



## Allegro Design Entry HDL User Guide

### Working with Properties and Text

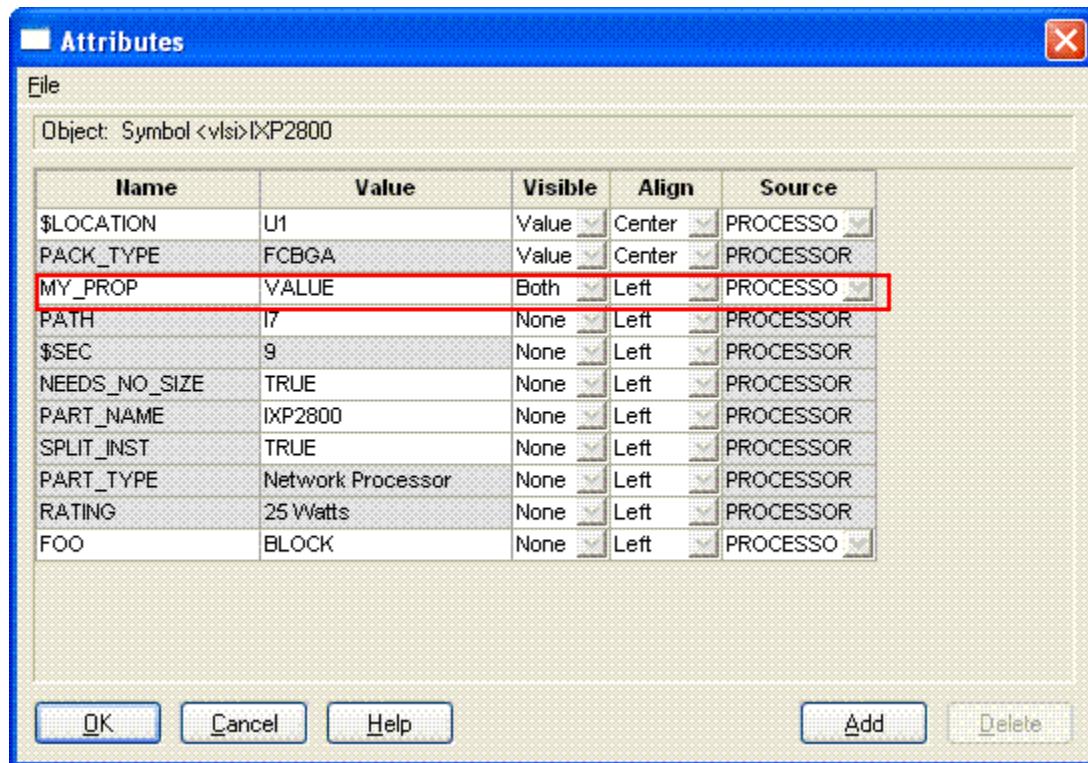
3. Click *OK* to add the property.



## Allegro Design Entry HDL User Guide

### Working with Properties and Text

Open the attribute form again on the same instance. The property is added.



Save the schematic sheet.

## Adding Properties in Non-Replicated Designs

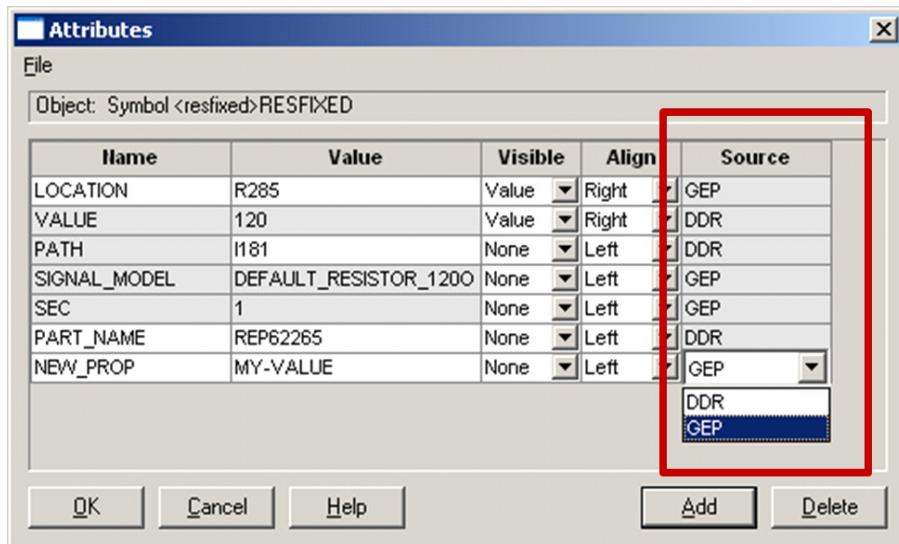
A non-replicated design is a design block that has been instantiated only once in the current design. The design block has only one occurrence. When you add a property to a non-replicated flat design, the property is saved as a block property by default. As there is only one instance of the block, storing the property as a block or occurrence is the same. Therefore, you can store the property value directly in the block by default.

It is possible that you would like to add this property only on this occurrence (instance) of the block. But, as you proceed with the design, you might have to add more instances of this block and you might want to retain this property only for this instance of the block. In this case, you can change the source of the property from the default value of Block to Occurrence. This can easily be done by changing the default source from the block name to the root (top) level

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

design name. The drop-down list for the Source column provides all the possible design names where the property can be stored.



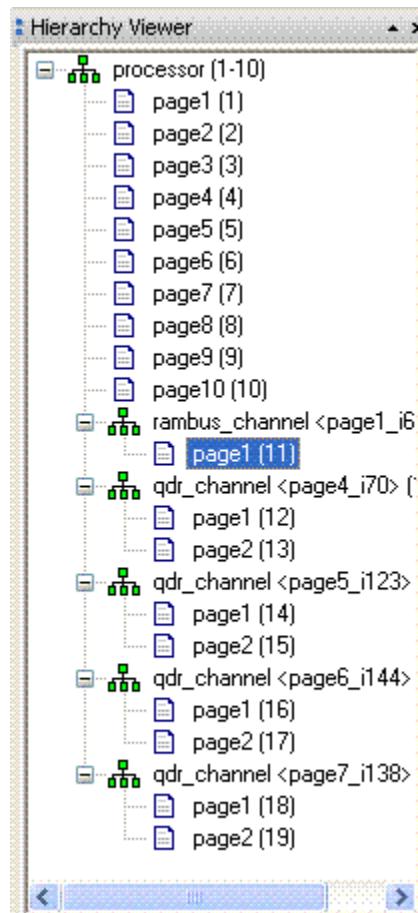
The design contains one instance of the RAMBUS\_CHANNEL block and therefore, it becomes the non-replicated design block.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

To view attributes and add a property to a non-replicated design, do the following:

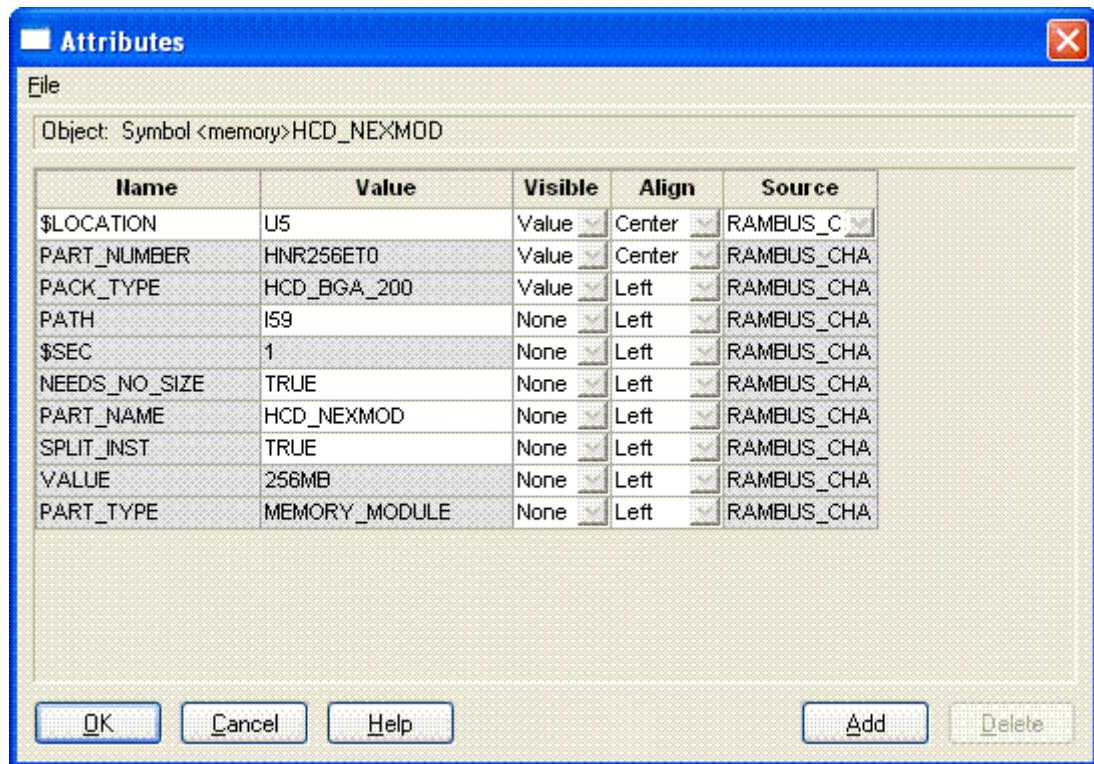
1. Select *page1* of the rambus\_channel block from the hierarchy viewer.



## Allegro Design Entry HDL User Guide

### Working with Properties and Text

2. Select an instance on the schematic page and open the attribute form.

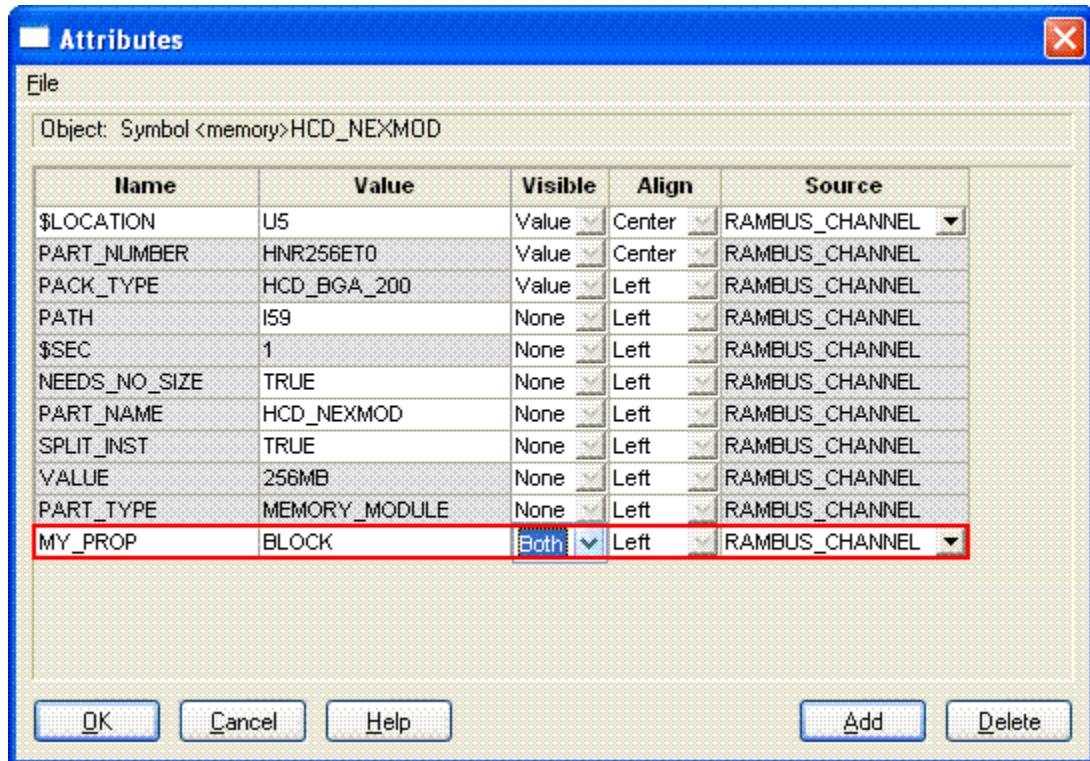


All the attributes of the instance are displayed. Note that all the attributes have the source set to RAMBUS\_CHANNEL, that is, the same block name. This means that all the property values are from the block directly.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

3. Click the *Add* button on the attribute form to add a new property.

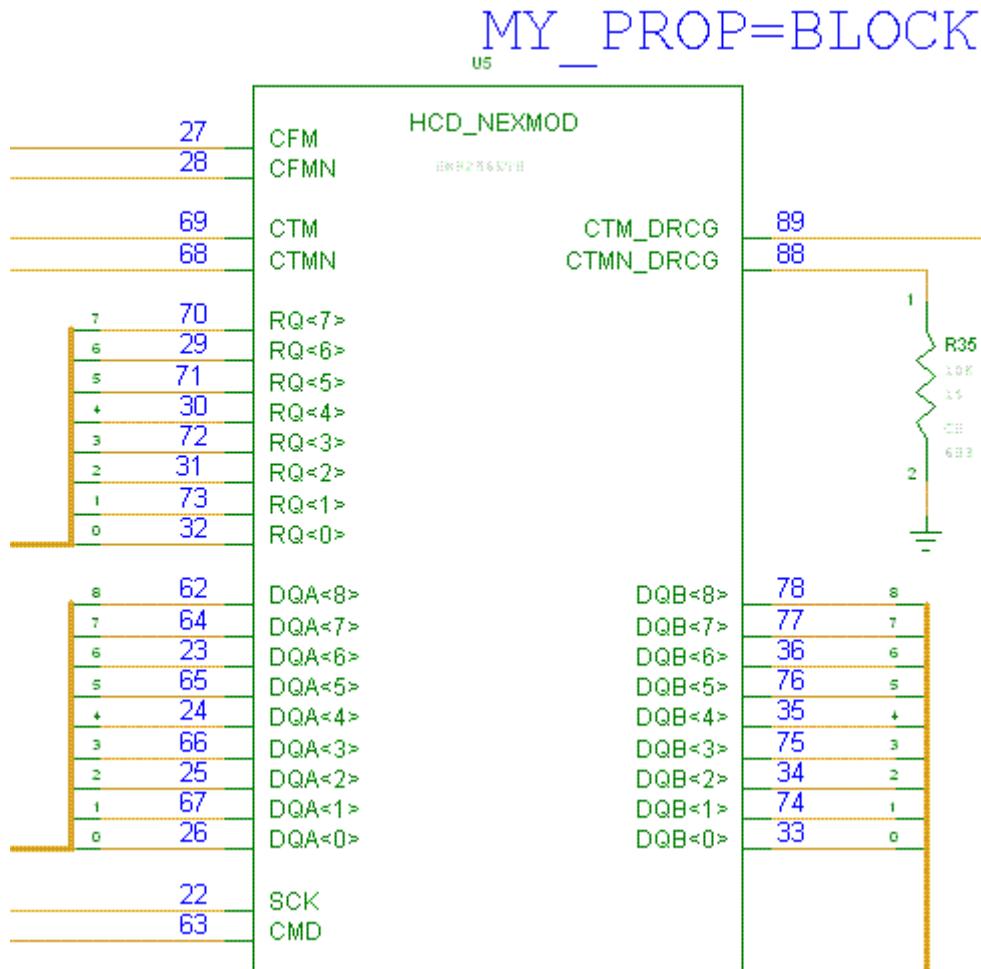


By default, the source for the property is RAMBUS\_CHANNEL, that is, the same block name. This means that for non-replicated designs, the default source of the property is the master or the block itself.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

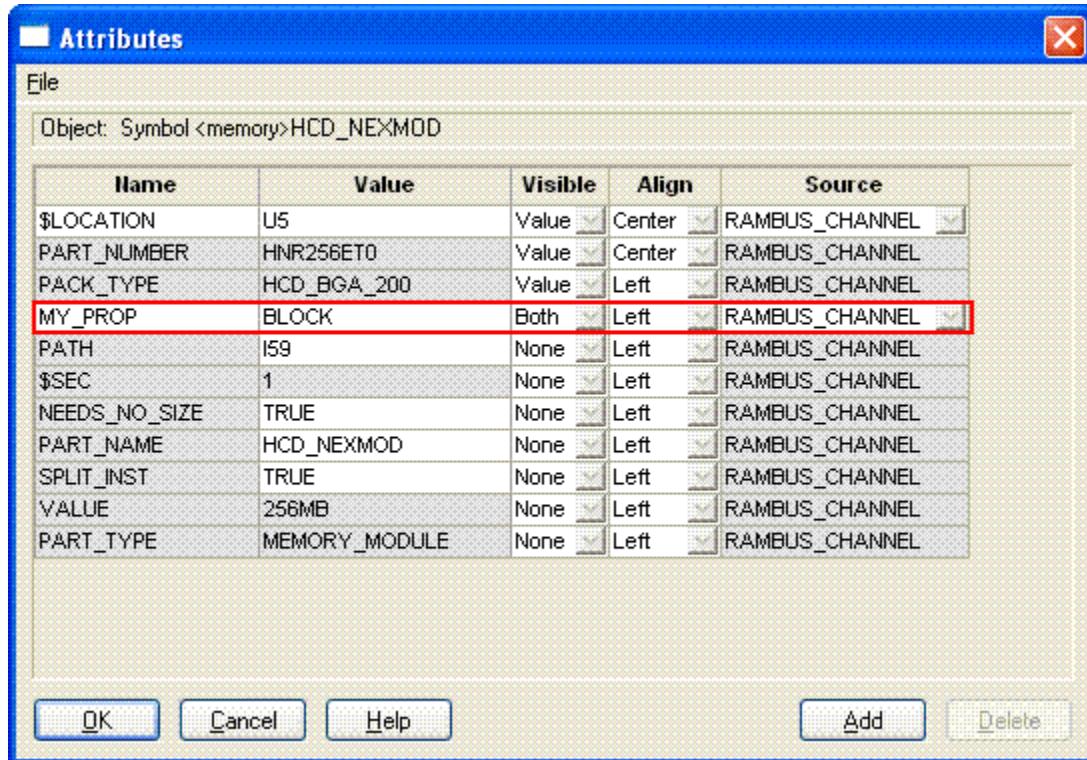
- Click **OK** to add the property.



## Allegro Design Entry HDL User Guide

### Working with Properties and Text

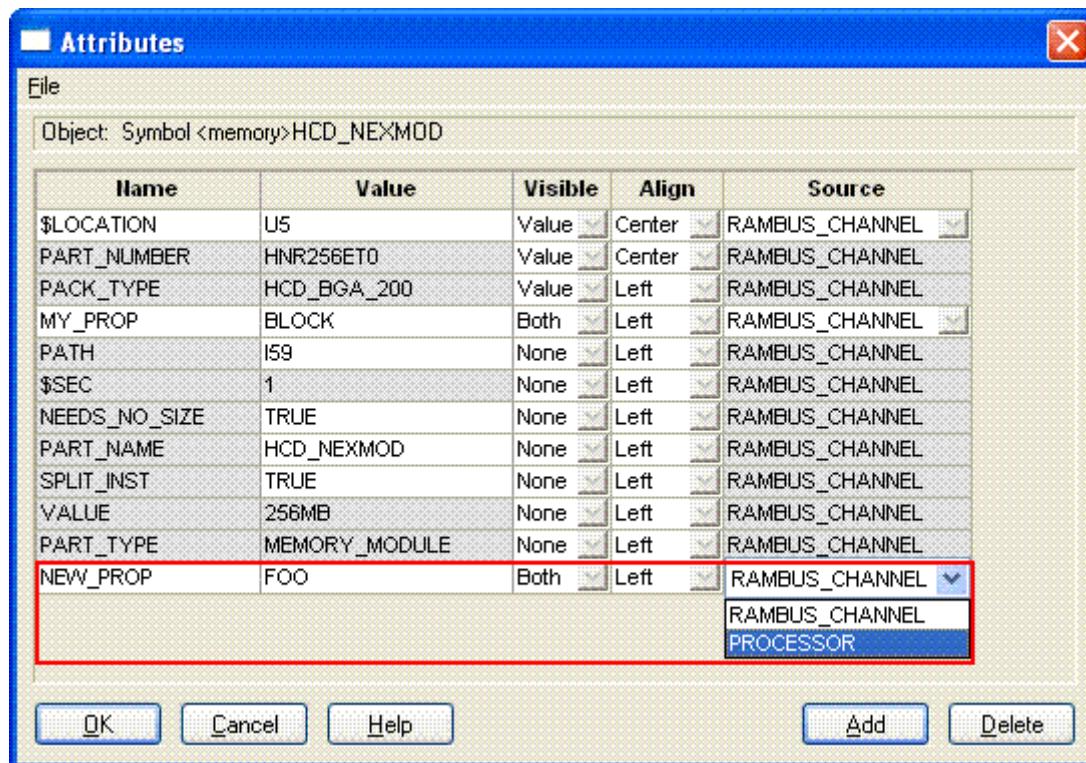
5. Select the attributes of the same instance again. The newly added property is visible with the source as RAMBUS\_CHANNEL.



## Allegro Design Entry HDL User Guide

### Working with Properties and Text

6. Click *Add* to add another new property.



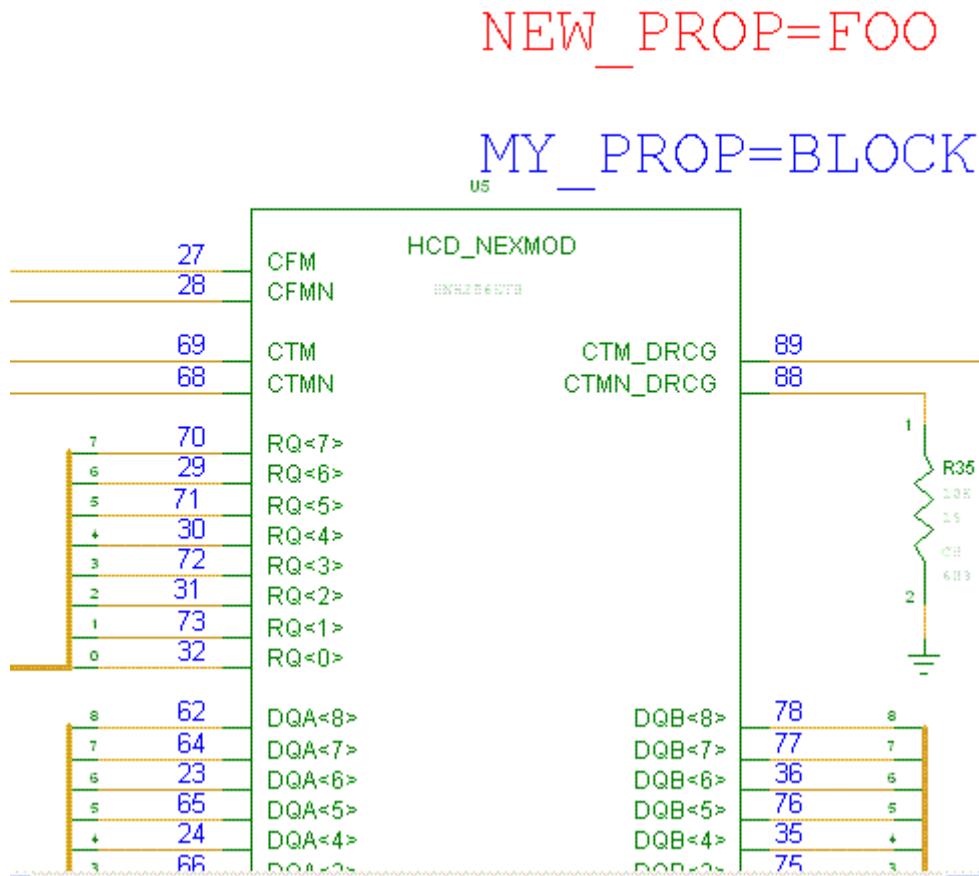
By default, the source for the property is RAMBUS\_CHANNEL, that is, the same block name. If you click the combo box in the Source column, you will see a drop-down list displaying the different options for setting the source. The other option in the source column is for the block PROCESSOR, which is the root (top) level design.

7. Select the source as PROCESSOR.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

Selecting the source as PROCESSOR will save the property in the root (top) level design. You can choose where you would like to place the property.



- Save the schematic sheet.

### Adding Properties in Replicated Designs

A replicated design is a design block that has been instantiated multiple times in the current design. The design block has more than one instance (occurrence) of it. When you add a property to a replicated design, it is added as an Occurrence property by default.

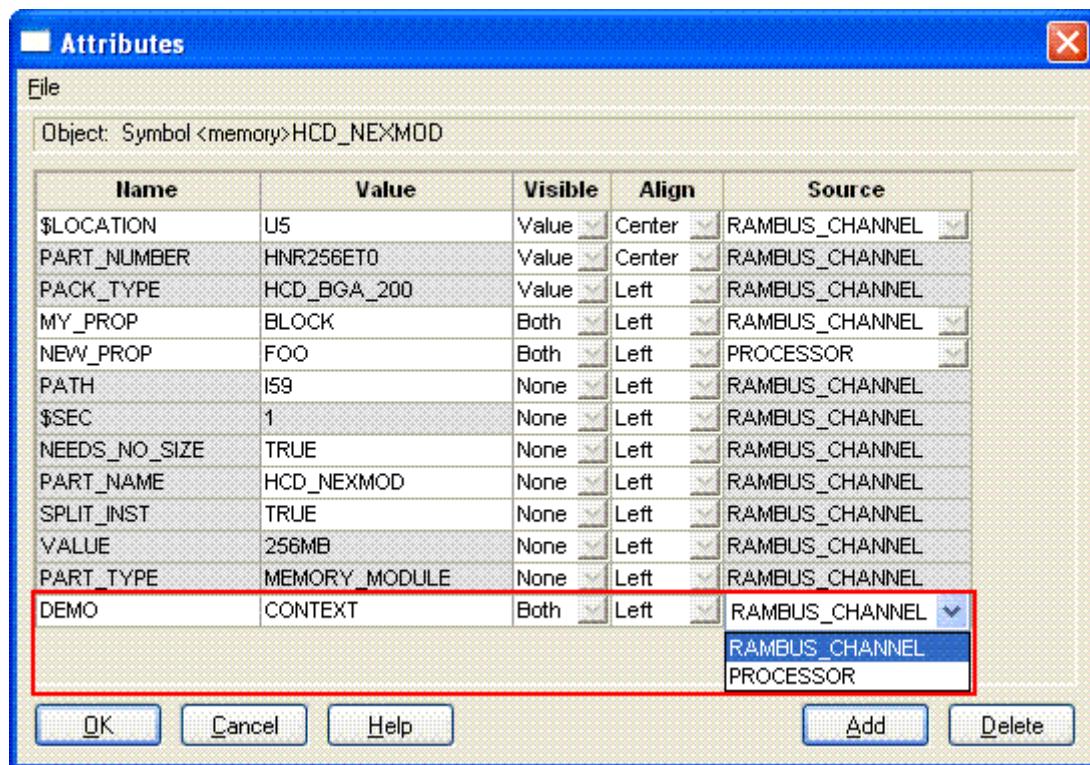
Since there are multiple instances of the block, the property being added is considered for the specific instance (occurrence) of the block. As a result, the property value is stored in the root (top) design by default. This property is only available on this instance of the block and not on the other instances.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

**Note:** When you add an occurrence property on a vectored pin or aliased bus, the property will not be visible in the Attribute form. It will only be visible on individual bits of the pin or bus by selecting the Show Index check box.

It is possible that you would like to add this property to all the instances of a block. You can change the source of the property from the default value of Occurrence to Block to enable this. This can easily be done by changing the default source from the root (top) level design name to the block name. The drop-down for the Source column provides all the possible design names where the property can be stored.

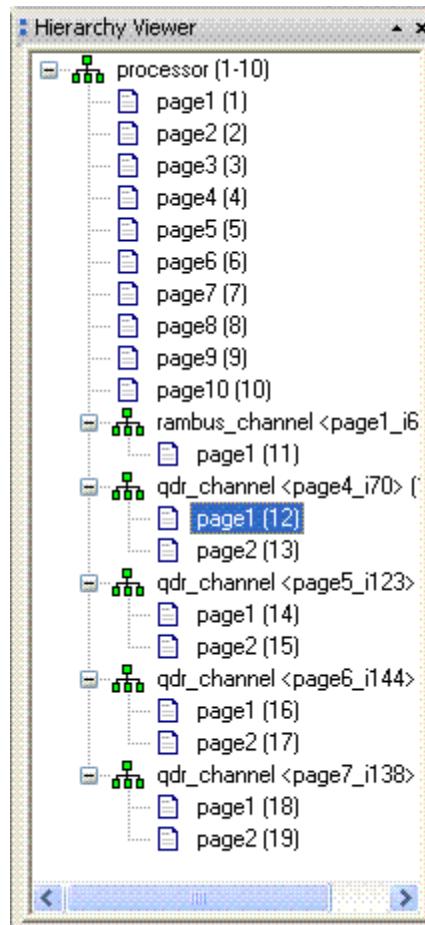


## Allegro Design Entry HDL User Guide

### Working with Properties and Text

Do the following to view attributes and add a property on a replicated design:

**Note:** The design block is the replicated block because it contains four instances of the QDR\_CHANNEL block.

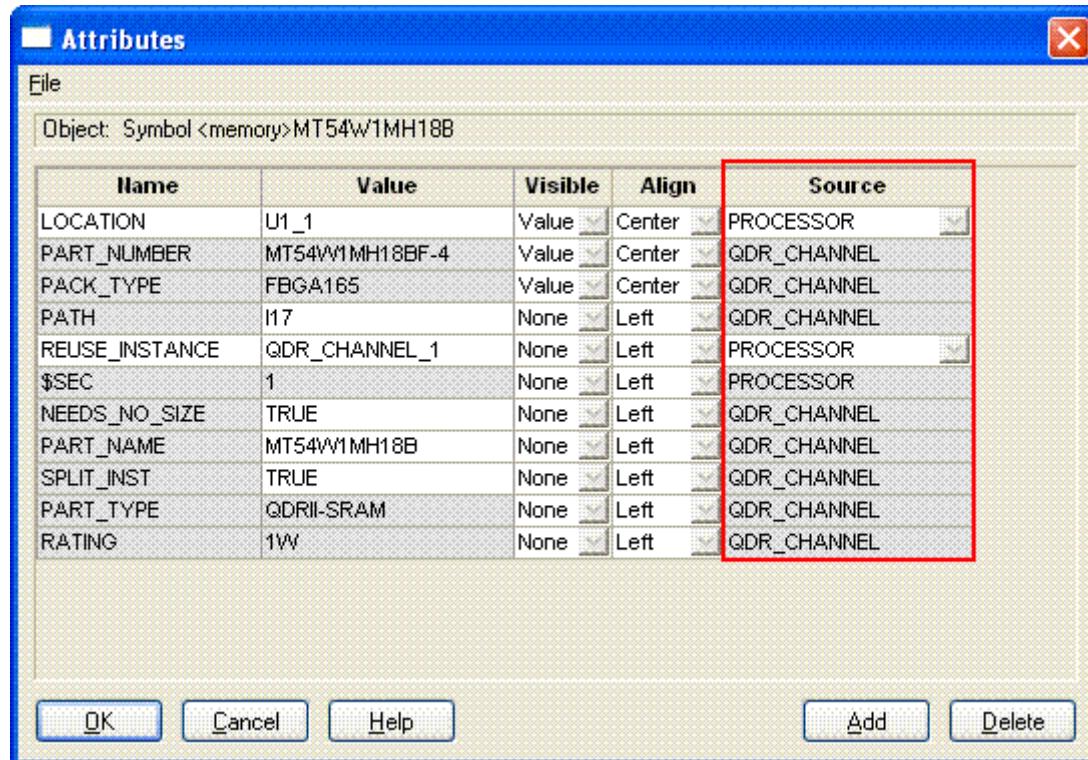


1. Select *page1(12)* of the first QDR\_CHANNEL block.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

2. Select an instance on the schematic page and open the Attribute form.



All the attributes of the instance are displayed. Note that some of the attributes have the source set to QDR\_CHANNEL, i.e., the same block name, while for some of the attributes, the source is set to PROCESSOR, i.e., the root (top) level block name. This indicates that some of the property values are stored in the block itself, while others are stored in the root (top) level block.

Note that all the packaging-related attributes, such as Location (RefDes), Section and REUSE\_INSTANCE are stored in PROCESSOR, that is, the root (top) level block name.

Since the packaging of this block is with regard to the root (top) level design, the properties are stored at that level. Also note that for this instance of the QDR\_CHANNEL block, the LOCATION property has the value U1\_1 and the REUSE\_INSTANCE property has the value QDR\_CHANNEL\_1.

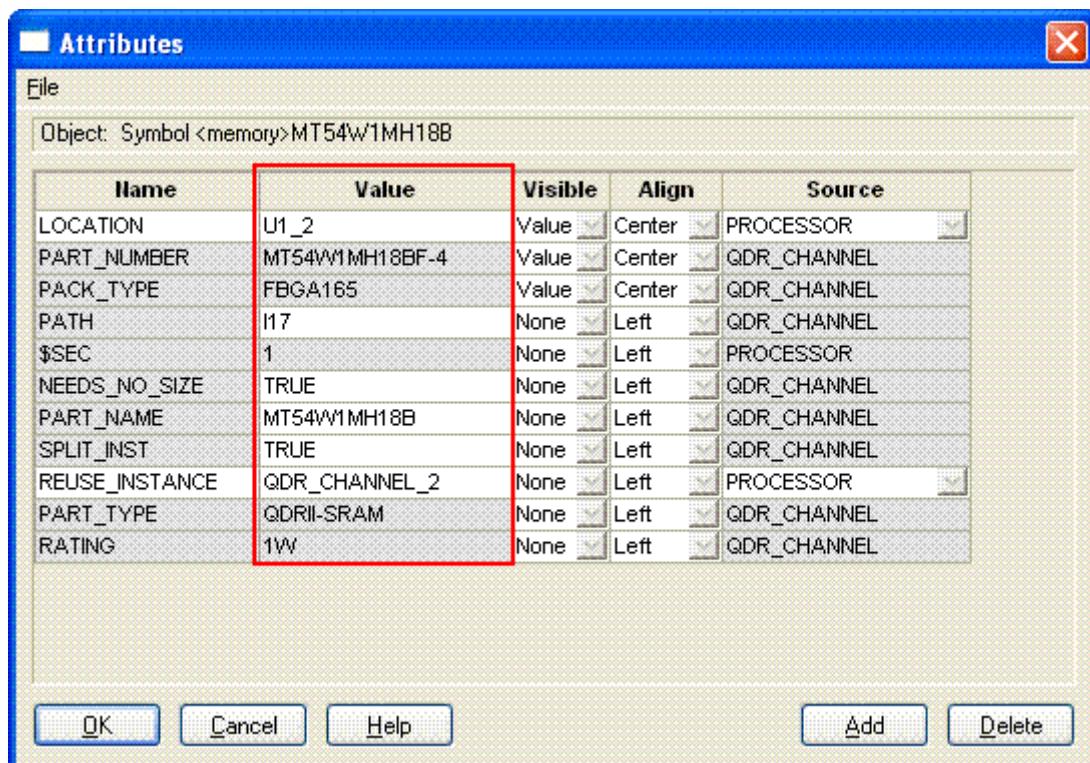
All the other attributes are stored in the block itself and therefore, the source is displayed as QDR\_CHANNEL.

3. Select page1 of the second QDR\_CHANNEL block.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

Note that because the schematic sheet is the same as in the first instance of the QDR\_CHANNEL block, the zoom level on this schematic sheet is preserved.



4. Select the same instance as selected earlier, and open the Attribute form.

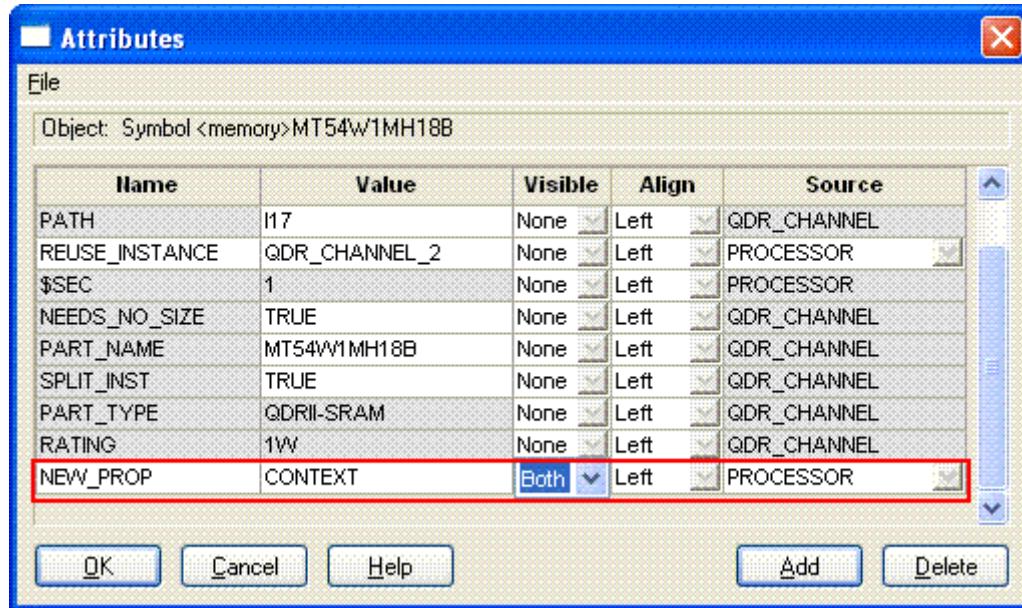
Note that all the attributes with the source as QDR\_CHANNEL have the same value as the previous case. Attributes with PROCESSOR as the source have different values. As these attributes are stored in the root (top) level design, the values of these attributes are specific to this instance of the QDR\_CHANNEL block.

Also note that for this instance of the QDR\_CHANNEL block, the LOCATION property has the value U1\_2 and the REUSE\_INSTANCE property has the value QDR\_CHANNEL\_2. Similarly, all the instances of the QDR\_CHANNEL block will have different packaging properties, which will be stored in the root (top) level design.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

5. Click *Add* on the Attributes form to add a new property.

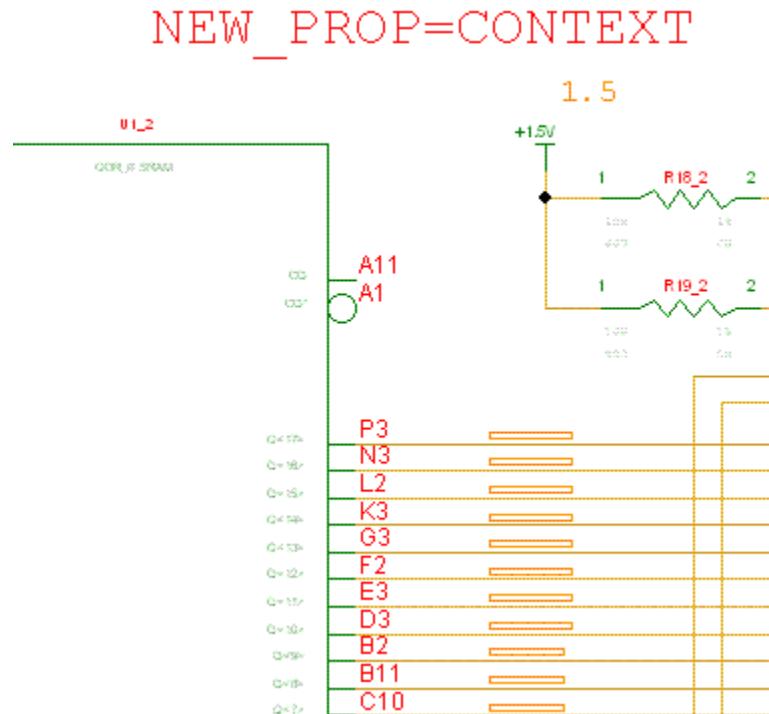


By default, the source for the property is PROCESSOR, that is, the root (top) level block name. This indicates that for replicated designs, the default source of the property is the In-context or the root (top) level block.

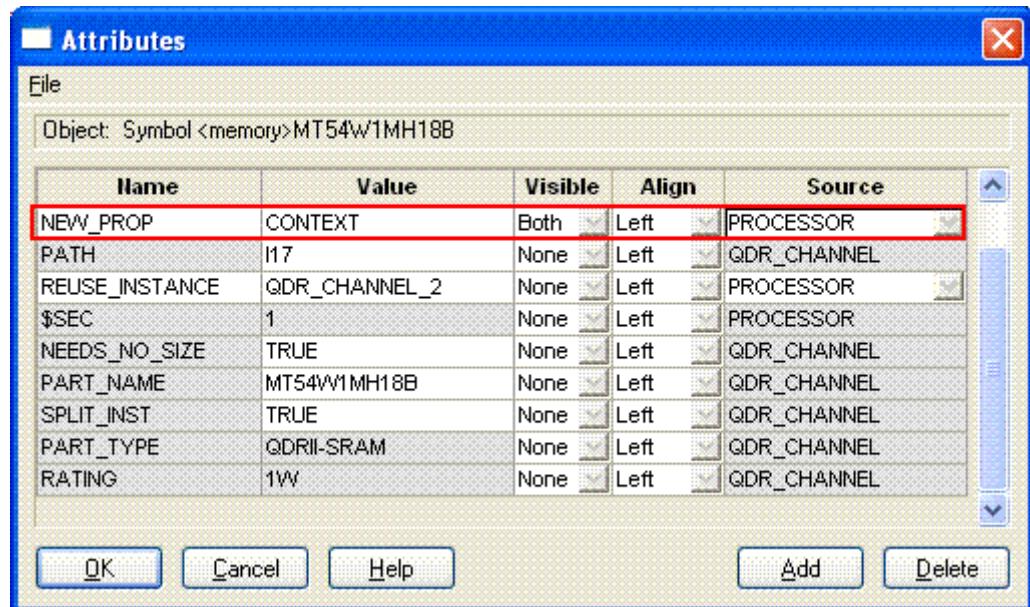
# **Allegro Design Entry HDL User Guide**

## Working with Properties and Text

6. Click *OK* to add the property.



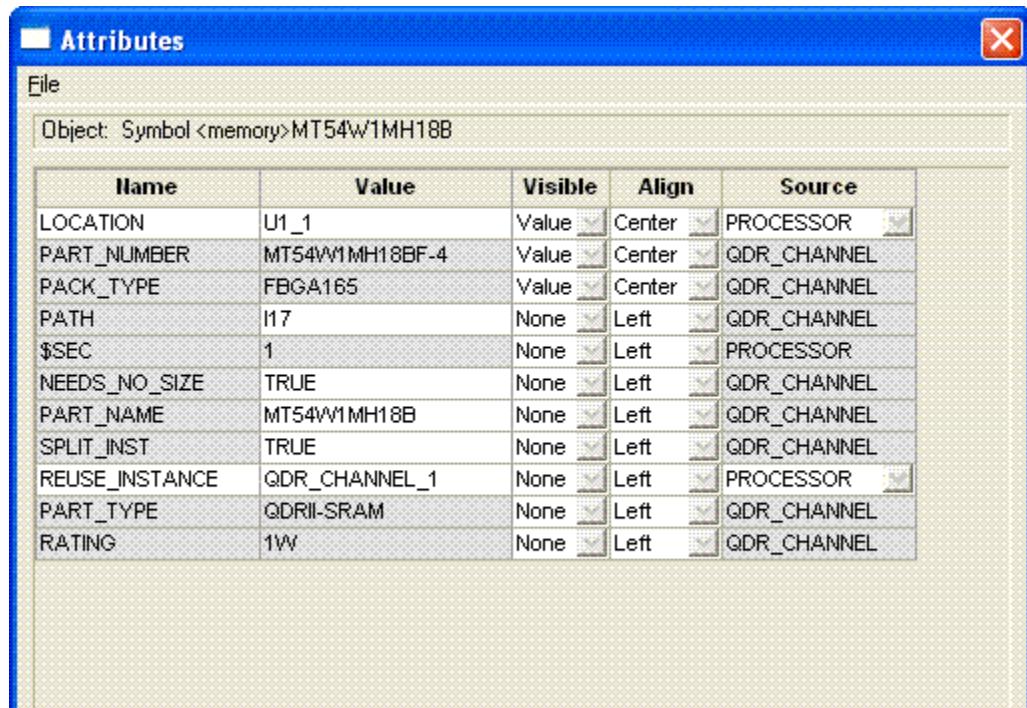
7. Select the attributes of the same instance again and you will notice that the newly added property is visible with QDR\_CHANNEL as the source.



## Allegro Design Entry HDL User Guide

### Working with Properties and Text

8. Select the previous block instance of the QDR\_CHANNEL block again and open the Attributes form for the same instance.

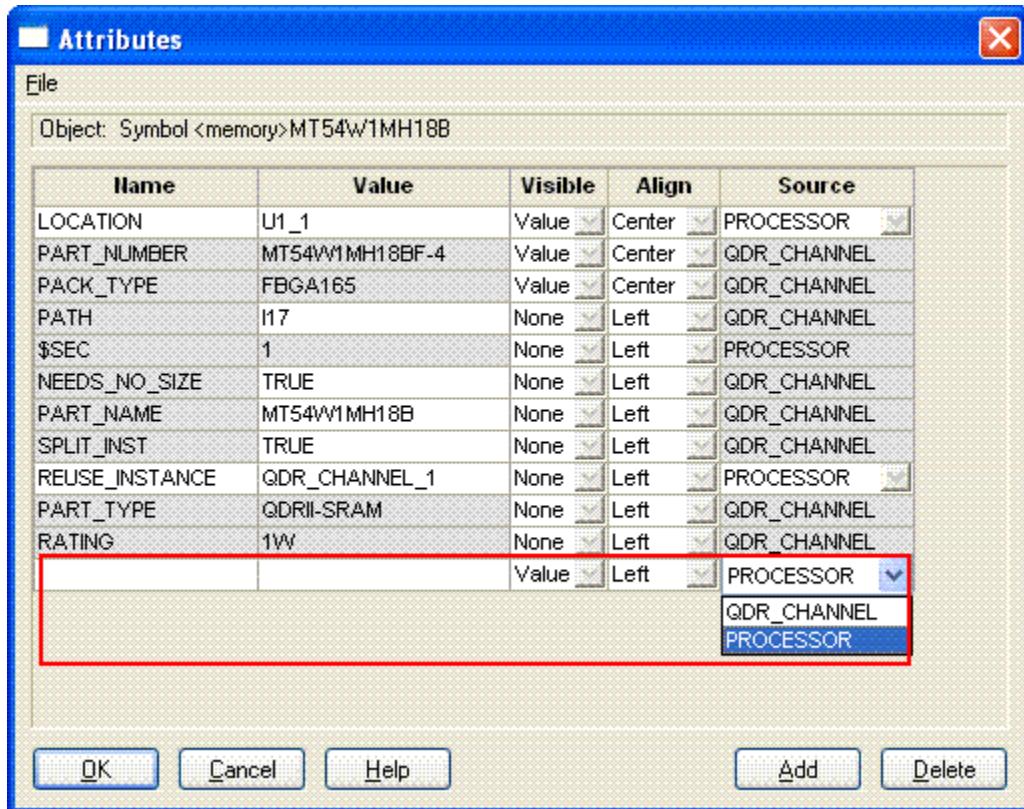


Notice that the NEW\_PROP property with the CONTEXT value is not visible in this block instance.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

- Click the Add button on the Attribute form to add a new property.

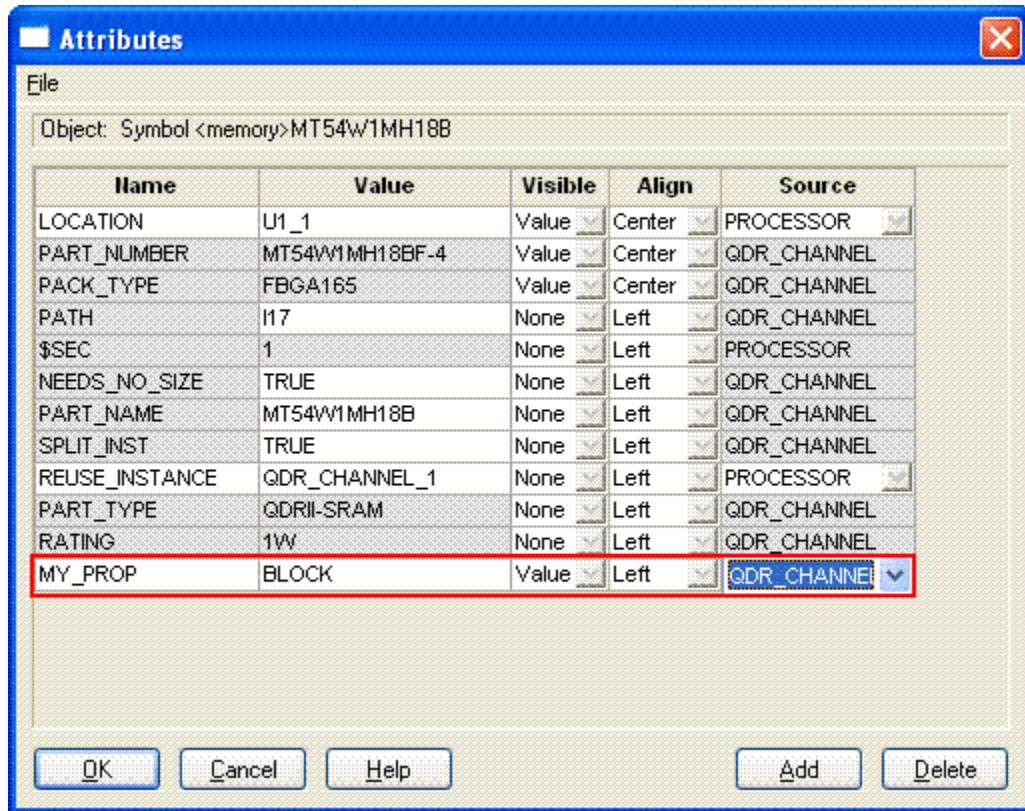


By default, the source for the property is PROCESSOR, that is, the root (top) level block name. Click on the combo box in the Source column and a drop-down list with the different options for setting the source is displayed. The other option in the Source column is for the QDR\_CHANNEL block, which is the same block itself.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

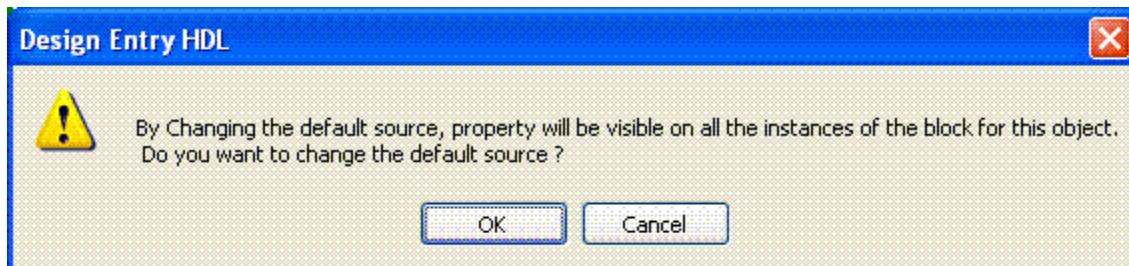
10. Select the source as QDR\_CHANNEL.



Selecting QDR\_CHANNEL saves the property in the block itself. Therefore, you have the flexibility to select where you want to place the property.

11. Click OK.

A warning message is displayed.



12. Click OK.

The property is added to the schematic sheet.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

#### 13. Save the schematic.

As this property is added as a block-level property, it will be available at all the instances of the QDR\_CHANNEL block.

The following table provides a quick summary of how you can save properties:

Design	Default Property Selected	Option to Save
Flat design	Block-level property	Block level
Non-replicated design	Block-level property	Block level or Occurrence
Replicated design	Occurrence property	Block level or Occurrence

## Editing Existing Properties

On editing an existing property value, the source with which the property was added is retained. When a block-level property is edited, it is saved as a block-level property. An occurrence property is edited and saved as an occurrence property.

While editing properties, If you want to change the source of a block-level property to a context property, you can only change it if you have edited or changed the value. if the value is the same, the property is considered a block-level property. This is applicable for all components, pins, and net properties.

**Note:** In-context section assignment cannot be done once a master section property has been assigned by Packager-XL.

All packaging properties are edited as occurrence properties. When a bus property is added to a whole bus in the .dcf file, it is added to the bus object. Editing a bus property for individual bits or a partial bus applies changes to the complete bus.

If a property is applied to a partial bus, it is applied to individual bits in the .dcf file. If you need to change property values for certain bits of a bus, then you need to apply them as overrides on the bus bits. The voltage property can only be edited in the block to which it is added for global nets. If you want to edit the voltage property in some other block, you can set the visibility value then edit the property.

#### *Important*

While editing a property, if you copy all then drag the object on the same page, occurrence properties are also copied for all nets and components. However, when pasting after a cut, copy, or copy all operation will only copy block level properties anywhere on the same sheet or different sheets or across the hierarchy.

There are some properties that can only be added as block-level properties. You cannot add these properties as Occurrence properties. Properties that can only be added as block-level are:

- HAS\_FIXED\_SIZE
- SIG\_NAME
- VOLTAGE
- DIFFERENTIAL\_PAIR
- NEEDS\_NO\_SIZE
- PLUMBING\_BODY
- SIGNAL\_MODEL
- PAGE\_BORDER
- SIZE
- HDL\_
- CDS\_
- XR
- \$XR

Occurrence editing is disabled for:

- Properties on Global nets
- Properties on Unnamed nets
- Properties which are not propagated through netlist:
  - smenubody
  - scommentbody
  - splumbingbody

- sflagbody
- sportbody
- spinnamebody
- sdrawingbody
- sdefinebody

## Deleting Properties

When you delete block-level or Occurrence properties, they are deleted from the source. However, if you have an Occurrence and block-level property on a component, pin or net, and you delete the Occurrence property, the property row from the Attribute form is not deleted. The block-level property is rippled to that row.

If you want to delete the block-level property, first click OK in the Attribute form then save the design. Reopen the Attribute form and delete the block-level property.

## Case-Sensitivity in Property Names and Values

In the HDL environment, case sensitivity of property names and values is supported in the front-to-back flow. You can define properties in the schematic, PPT and `chips.prt` files. They can also be fed back from the board.

**Note:** In Design Entry HDL, all property names and values are automatically uppercased. To change this behavior,

1. From the *Tools* menu in Design Entry HDL, select *Options – Text*.
2. Deselect the *Upper-Case Input* check box.

You can now enter lower or mixed case property names and property values. They will not be automatically uppercased by Design Entry HDL and will be displayed in the schematic in the same case that they are entered.

By default, for all such properties

- the name of the property is uppercased by the tools
- the case of the property value is preserved.

You can change this default behavior for specific properties by attaching specific attributes to them in the `cdsprop.paf` file, which is located at

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

<your\_inst\_dir>/share/cdssetup

The `cdsprop.paf` is a text file. You can edit this file to make necessary changes to the case sensitivity of property names and values. You can place this file in the project directory if the attributes have to be applied only to that project.

To indicate that the case of a property name should be preserved, use the keyword `preservename`. To indicate that the case of a property value should not be preserved, use the keyword `uppercasevalue`.

Example:

```
alt_symbols: uppercasevalue
\TimingVersion\ :preservename
```

In the example, `ALT_SYMBOLS` is assigned the keyword `uppercasevalue`. If you now specify the property, its value will always be uppercased.

The `TimingVersion` property is specified within backslashes (\) since the `cdsprop.paf` file is in the VHDL namespace. (In VHDL, backslashes are used to denote that the properties are case-sensitive.

`TimingVersion` is assigned the keyword `preservename`. So the property name will not be uppercased to `TIMINGVERSION`.

## Case Insensitive Property Values

There are certain properties whose values should always be uppercased because they do not support case-sensitive values. These properties are assigned the keyword `uppercasevalue` in the default `cdsprop.paf` located in the installation hierarchy  
<your\_inst\_dir>/share/cdssetup.

The properties defined in `cdsprop.paf` whose values should always be uppercased are:

- `ALT_SYMBOLS`
- `BODY_NAME`
- `CDS_NAME_OF_PART`
- `CDS_LOCATION`
- `CDS_PN`
- `GROUND_NETS`
- `JEDEC_TYPE`

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

- LOCATION
- \$LOCATION
- MERGE\_NC\_PINS
- MERGE\_POWER\_PINS
- NC\_PINS
- PACK\_TYPE
- PACK\_SHORT
- PART\_NAME
- PIN\_NUMBER
- POWER\_GROUP
- POWER\_PINS
- PN
- \$PN
- SD\_SUFFIX\_SEPARATOR
- SUBDESIGN\_SUFFIX

**Note:** If you remove these assignments from the `cdsprop.paf` file, Packager-XL generates a warning message and these property values will continue to be treated as case-insensitive.



You can get an updated list of properties from the `cdsprop.paf` file located in your installation hierarchy `<your_inst_dir>/share/cdssetup`.

### Case Insensitive Property Names

There are some properties whose names are always treated as case-insensitive. These property names are always uppercased.

The following is a list of case-insensitive properties:

- ALT\_SYMBOLS
- BIDIRECTIONAL

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

- BODY\_NAME
- BODY\_TYPE
- CDS\_LONG\_PART\_NAME
- CDS\_NAME\_OF\_PART
- CDS\_PARENT\_CHIPS\_PHYS\_PART
- CDS\_PARENT\_PPT
- CDS\_PARENT\_PPT\_PART
- CDS\_PARENT\_PPT\_PHYS\_PART
- CDS\_PHYS\_NET\_NAME
- CDS\_PRIM\_FILE
- CDS\_SEC
- CDS\_LOCATION
- CDS\_PN
- GROUP
- HAS\_FIXED\_SIZE
- INPUT\_LOAD
- JEDEC\_TYPE
- LOCATION
- \$LOCATION
- MERGE\_NC\_PINS
- MERGE\_POWER\_PINS
- NC\_PINS
- NO\_BACKANNOTATE
- OUTPUT\_LOAD
- OUTPUT\_TYPE
- PACK\_TYPE
- PACK\_IGNORE

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

- PACK\_SHORT
- PART\_NAME
- PATH
- PINCOUNT
- PIN\_GROUP
- PIN\_NUMBER
- POWER\_GROUP
- POWER\_PINS
- PN
- \$PN
- PRIM\_FILE
- ROOM
- SEC
- \$SEC
- SD\_SUFFIX\_SEPARATOR
- SIZE
- SUBDESIGN\_MASTER
- SUBDESIGN\_SUFFIX
- WIRE\_GATE
- XY

**Note:** If you assign any of these properties as case-sensitive (by assigning the keyword preservename) in the `cdsprop.paf` file, Packager-XL will generate a warning message and continue to treat these as case-insensitive.

## Case Sensitivity and PPTs

In Design Entry HDL, when you select a part with physical information, Design Entry HDL compares the key property name on the Design Entry HDL canvas and the key property name in the physical part table file. This comparison is case-insensitive.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

For example, if the key property name in the PPT header is abcd and the property name on the schematic instance is 'ABCD', the match will still take place.

For properties that have the keyword uppercasevalue in `cdsprop.paf`, the comparison for the value is always case-insensitive. For example, the PACK\_TYPE property has the keyword uppercasevalue in `cdsprop.paf`. Let us suppose that a PPT file contains the following key property and value

```
PACK_TYPE=dip
```

The schematic instance may contain DIP, Dip, or dip, and the value will be still matched with the PACK\_TYPE value in the PPT file.

For properties that do not have the keyword uppercasevalue (i.e. the values are to be treated case-sensitive), the comparison for the key property values between the schematic instance and the PPT can be done case-sensitively or case-insensitively. The default is case-insensitive matching. To change this to case-sensitive matching, you can select an option Perform Case Sensitive Row Match provided in the Part Table tab of Project Manager setup.

Example:

Consider PPT rows of the following type

```
:ABCD = EFGH;  
'EFGH' (1) = 'pqr1'  
'efgh' (2) = 'pqr2'  
END_PART
```

If the schematic instance has the property value EFGH and you have not selected the *Perform Case Sensitive Row Match* option, Packager-XL will match the value on the schematic instance with the values in both the rows. An error message is also generated if a unique match is not found.

If you have selected the *Perform Case Sensitive Row Match* option, the match will be carried out with only the first row.

Once the row is matched, to send the PPT properties in the pst files, the algorithm is the same (the case of the values is preserved and the names are uppercased). To override these defaults, you can use the keyword uppercasevalue/preservename in the `cdsprop.paf` file.

## Constructing PPTs with Case-Sensitive Values

If there are rows in a PPT that contain the key property values that differ only in case, you must specify explicit subtype names for each row.

Example:

Consider PPT rows of the following type

```
:ABCD      = EFGH      ;
'EFGH' (1)    = 'pqr1'
'efgh' (2)    = 'pqr2'
END_PART
```

In this example, the rows are named explicitly using 1 and 2 in the brackets following the key property values. The rows can be named also by using a unique ~<string> in each of the brackets.

**Note:** If an explicit naming is not done (empty brackets or ! in the brackets), the PPT will not be usable and will not load by any of the tools. This restriction exists because the physical part name resulting from any PPT row has to be unique after uppercasing it.

## Assigning Power Pins

You use the *Assign Power Pins* dialog box to view existing properties and add new properties to components. The following properties can be viewed in the *Assign Power Pins* dialog box:

- POWER\_GROUP
- POWER\_PINS
- NC\_PINS
- MERGE\_NC\_PINS
- MERGE\_POWER\_PINS

When you edit these properties in the dialog box, Design Entry HDL creates new POWER\_GROUP, POWER\_PINS, and NC\_PINS properties for the power pins. For details on how the POWER\_PINS and POWER\_GROUP properties work together, refer to the “Preparing your Schematic for Packaging” chapter of the *Packager-XL User Guide*.

## Viewing Properties on Power Pins

The number of power pins and NC pins in the component is determined by the POWER\_PINS, MERGE\_POWER\_PINS, NC\_PINS, and MERGE\_NC\_PINS properties on it.

- If a MERGE\_POWER\_PINS property exists on the instance, the number of power pins in the component increases.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

The `POWER_PINS` property in the `.ptf` file is first overlaid on the `POWER_PINS` property in the `chips.prt` file. The resulting `POWER_PINS` property is then combined with the `MERGE_POWER_PINS` property on the instance.

- If a `MERGE_NC_PINS` property exists on the instance, the number of NC pins in the component increases.

The `NC_PINS` property in the `.ptf` file is first overlaid on the `NC_PINS` property in the `chips.prt` file. The resulting `NC_PINS` property is then combined with the `MERGE_NC_PINS` property on the instance.

The pin names are determined by the `POWER_PINS` and `POWER_GROUP` properties on the component. If there is a `POWER_GROUP` property in the `.ptf` file, it is combined with the `POWER_PINS` properties in `chips.prt` to obtain the pin names. To ignore the `POWER_GROUP` property in the `.ptf` file, you can set the `SCH_POWER_GROUP_WINS_OVER_PPT` in the `.cpm` file of the project. For details on how the directive works, refer to [Example 3](#).

**Note:** You cannot edit the occurrence property for `POWER_PINS` and `NC_PINS`.

**Note:** It is recommended that you add the `Power_group` property as a schematic property using the Power Group dialog.

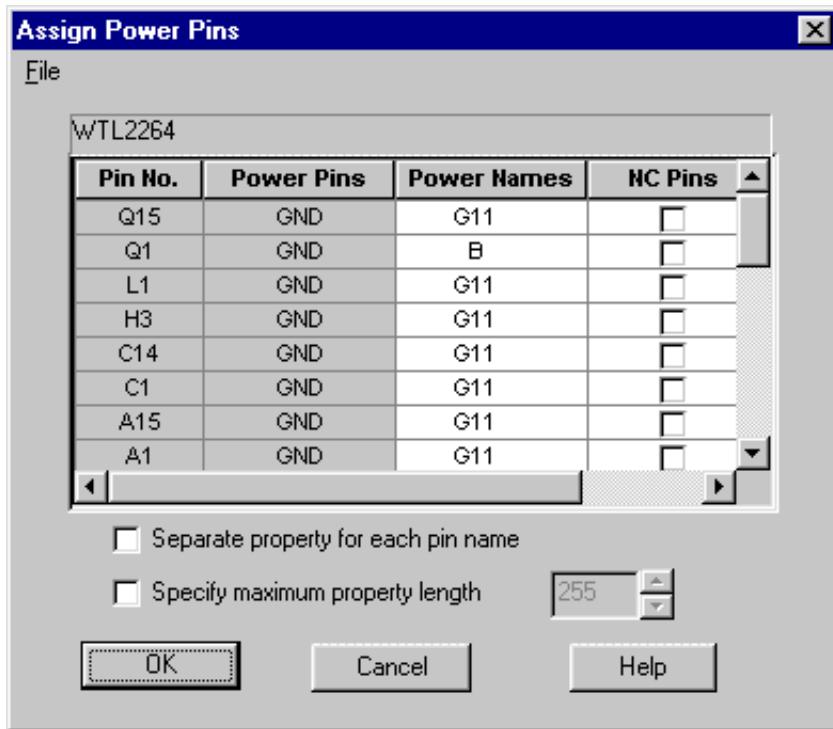
For further details on how the `POWER_PINS` and `POWER_GROUP` properties work together, refer to the “Preparing your Schematic for Packaging” chapter of the [Packager-XL User Guide](#).

The *Assign Power Pins* dialog box can be opened by choosing the *Text – Assign Power Pins* menu option and clicking on the component (post-select mode) or by clicking the right menu button on the selected component and choosing *Assign Power Pins* from the pop-up

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

menu. The *Assign Power Pins* dialog box reads the properties existing on the power and NC pins of a component and appears as follows:



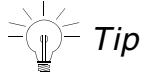
The first two columns, *Pin No.* and *Power Pins*, contain the power pin numbers and power pin names read from the `chips.prt` and `.ptf` file of the component. The power pin numbers and power pin names in the `.ptf` file are overlaid on those in the `chips.prt` file. These columns are always disabled. So, the pin numbers and pin names in the `chips.prt` and `.ptf` file cannot be changed.

If you have added the component using the Physical Part Filter, the `PACK_TYPE` property specified then is used. Or else, if the design is packaged, the `PACK_TYPE` property in the `chips.prt` file determines the primitive to be read from the file or else the default primitive is read. The power pins from the specified primitive appear in the *Assign Power Pins* dialog box.

The third column, *Power Names*, contains the power pin names specified on the instance of the component. For a pin, the value of this field is determined as follows:

- If no `POWER_PINS/NC_PINS/POWER_GROUP` property exists on the instance, this column is exactly the same as the *Power Pins* column (which means that the power pin name on the instance is the same as that in the `chips.prt/.ptf` file).
- If a `POWER_PINS` property exists on the instance, its value overrides the value of the `POWER_PINS` property in the `chips.prt/.ptf` file.

- If a `POWER_GROUP` property also exists on the instance, the power pin name is specified by the value of the `POWER_GROUP` property.



#### *Tip*

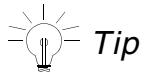
You can control the way properties created through the Assign Power Pins dialog box are displayed on the schematic. In the *Text* tab of the Design Entry HDL Options dialog box, set the *Power Property Visibility* as *Invisible*, *Name*, *Value* or *Both*.

#### **To assign a new power name to the pin, do the following:**

Select a signal from the list of available global signals in the drop-box or enter a signal name of your own choice.

The fourth column *NC Pins* specifies the NC pins for the component. The value of this column is determined as follows:

- The property on the instance gets first priority, and overrides the value on the symbol and the `chips.prt/.ptf` files.
- The property on the symbol gets second priority and overrides the value in the `chips.prt/.ptf` files.
- The property in the `.ptf` file gets third priority and overrides the value in the `chips.prt` file.
- The property value in the `chips.prt` gets last priority.



You can sort each column alphanumerically by clicking on the heading of the column. For example, to sort according to *Power Names*, click on the heading *Power Names*.

#### **Length of Property Value**

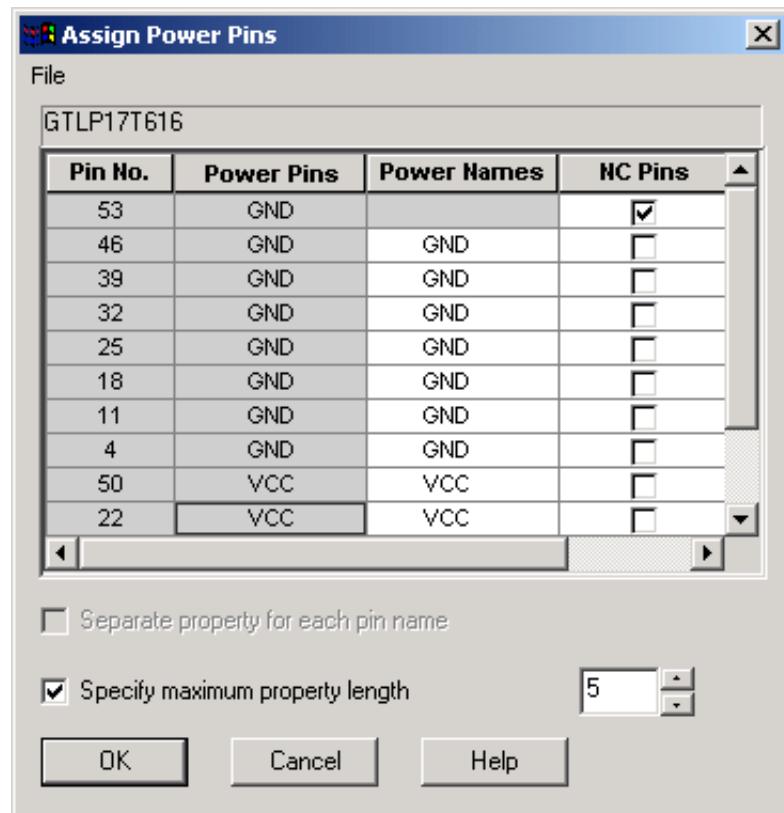
The two check boxes, *Separate property for each pin name* and *Specify maximum property length*, are used to control the maximum length of a property value annotated by the Assign Power Pins dialog box. By default, the maximum length of a property value in Design Entry HDL is 255 characters.

You can use the *Specify maximum property length* option to specify the maximum number of characters in the property. Design Entry HDL then splits the property such that the number of characters in each property does not exceed the specified length.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

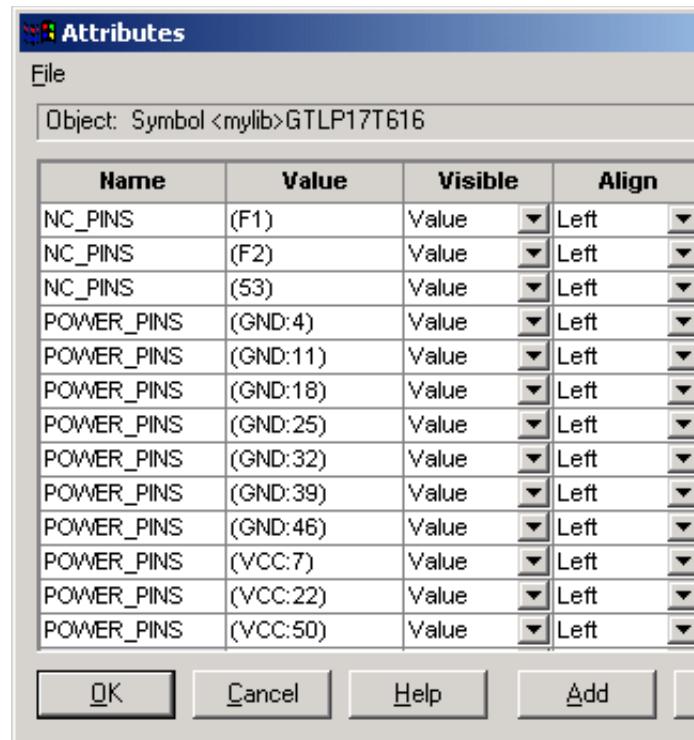
For example, if the following case, the maximum length of property value is specified as 5 characters:



## Allegro Design Entry HDL User Guide

### Working with Properties and Text

So, different POWER\_PINS properties are created for the instance as follows:

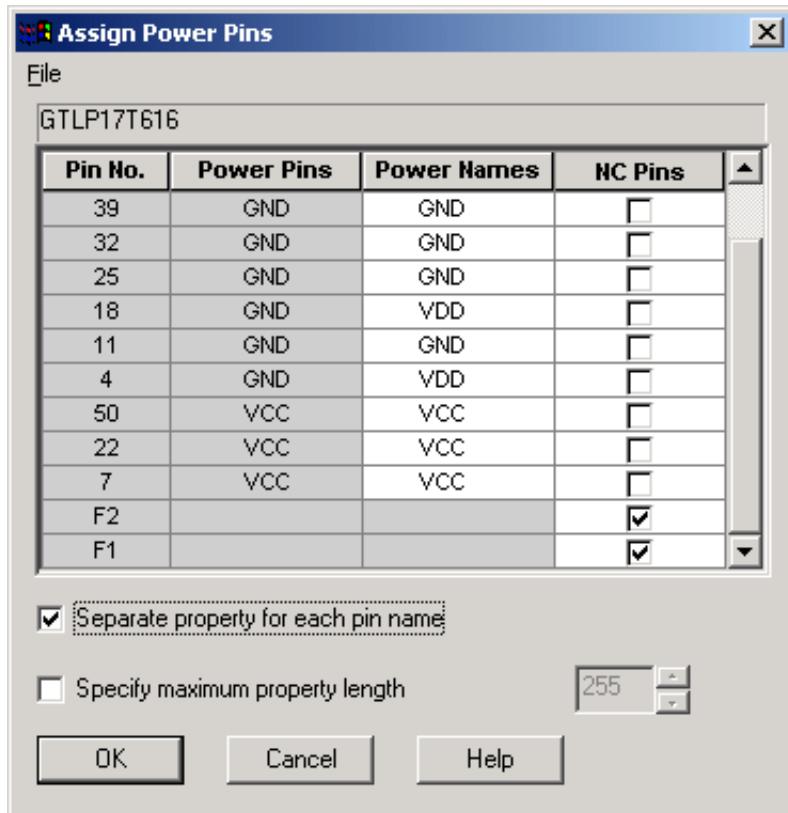


If *Separate property for each pin name* is selected, Design Entry HDL creates a new POWER\_PINS property for a pin name.

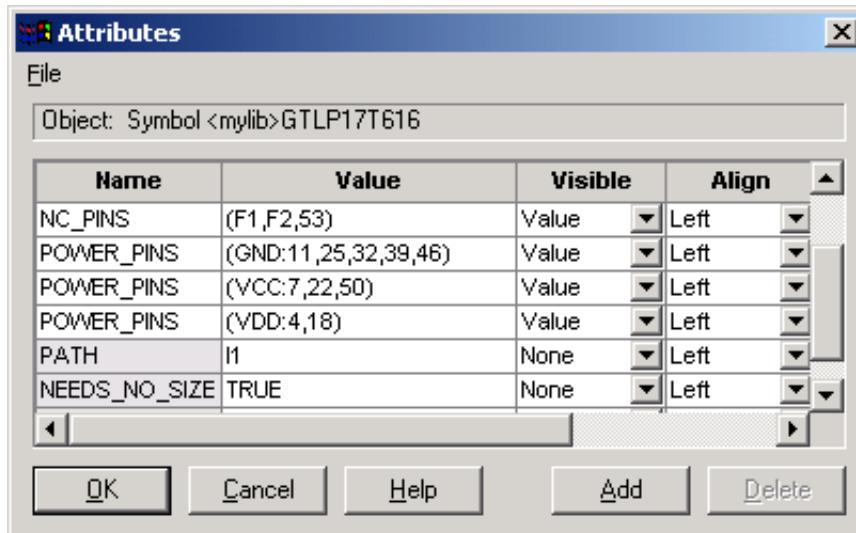
## Allegro Design Entry HDL User Guide

### Working with Properties and Text

For example, in the following case, there are 3 different power source names, GND, VDD, and VCC.



So, 3 different POWER\_PINS properties, one per name, are created as follows



## Using the Assign Power Pins Dialog Box

### Example 1

Consider an instance of component WTL1163 that has the following properties:

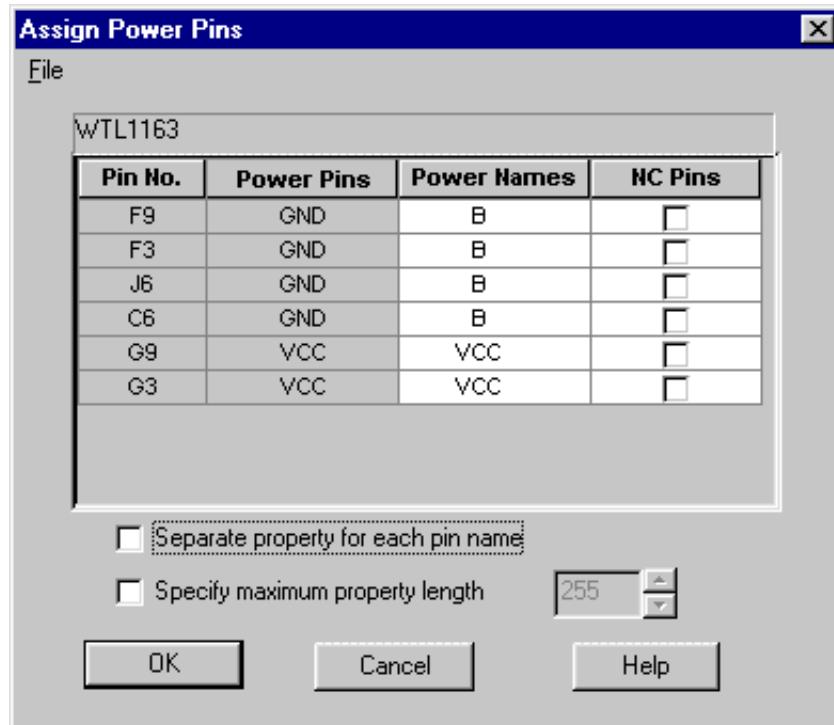
In chips.prt:

```
POWER_PINS = (GND:F9,F3, J6, C6; VCC: G9,G3)
```

On the instance:

```
POWER_GROUP = (GND = B)
```

The *Assign Power Pins* dialog box appears as follows:

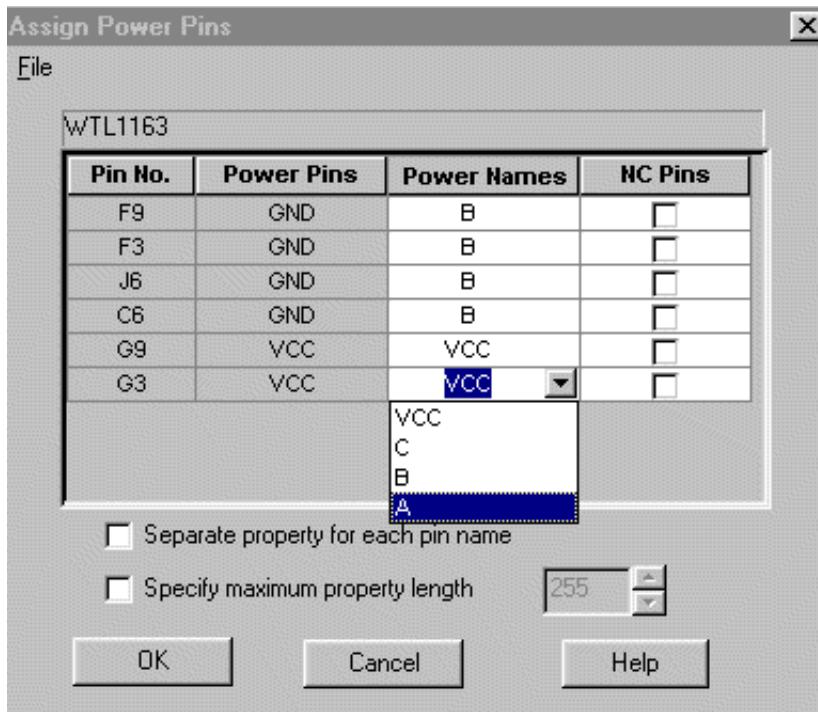


## Allegro Design Entry HDL User Guide

### Working with Properties and Text

**To change the power source for pin G3 to global signal A, do the following:**

Select A from the list of available global signals in the drop box as follows



Now, the properties on instance of component WTL1163 are as follows:

In chips.prt:

```
POWER_PINS = (GND:F9,F3, J6, C6; VCC: G9,G3)
```

On the instance:

```
POWER_GROUP = (GND = B)
```

```
POWER_PINS = (A:G3;GND: J6, C6, F9, F3; VCC: G9)
```

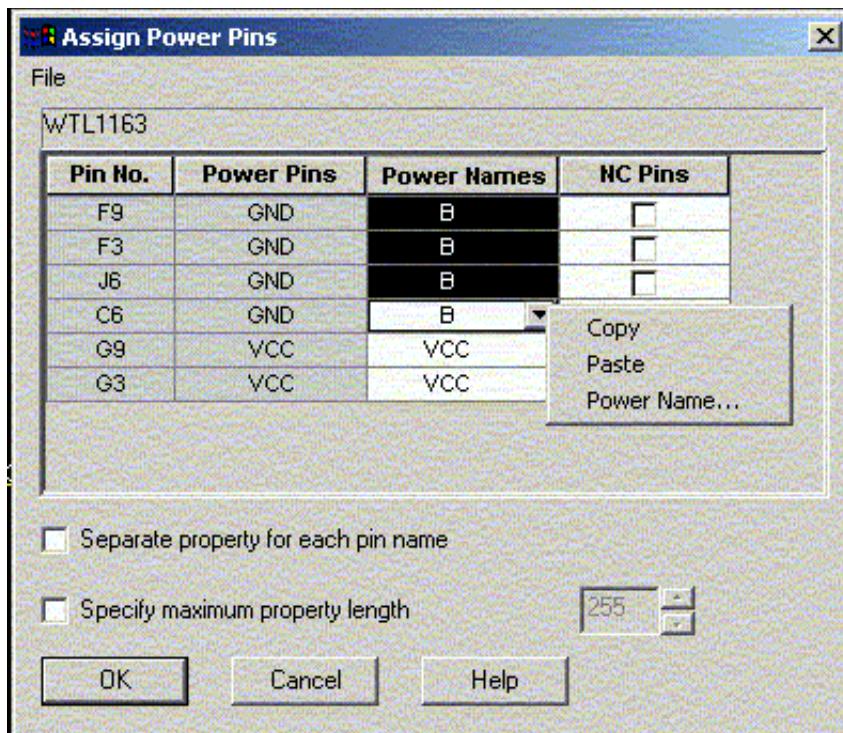
**To change the power source for pins F9, F3, J6, and C6 to global signal C,**

1. Select the *Power Names* entries for pins F9, F3, J6, and C6 by holding the *Shift* key down and pressing an arrow key or by dragging the left mouse button.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

2. Right-click to bring up the pop-up menu as follows



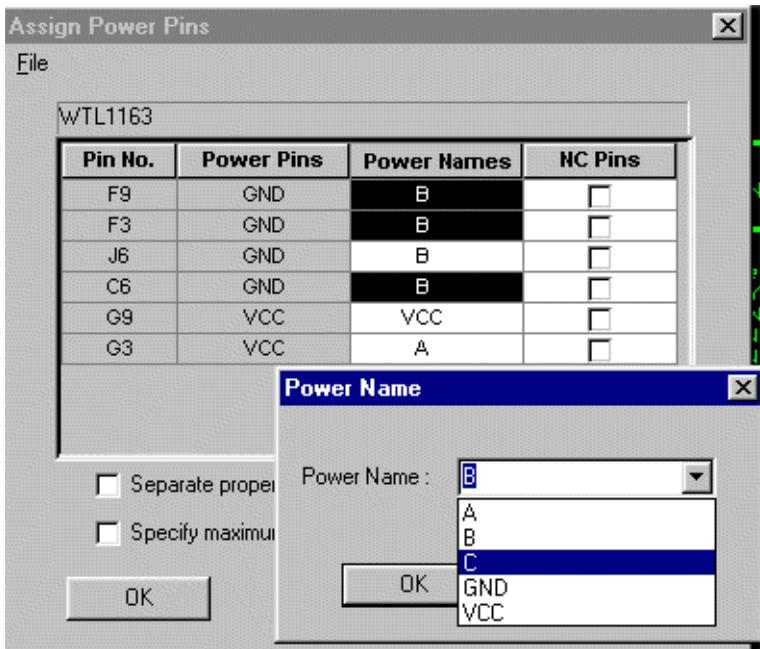
3. Choose *Power Name*.

The *Assign Power Pins* dialog box appears.

## Allegro Design Entry HDL User Guide

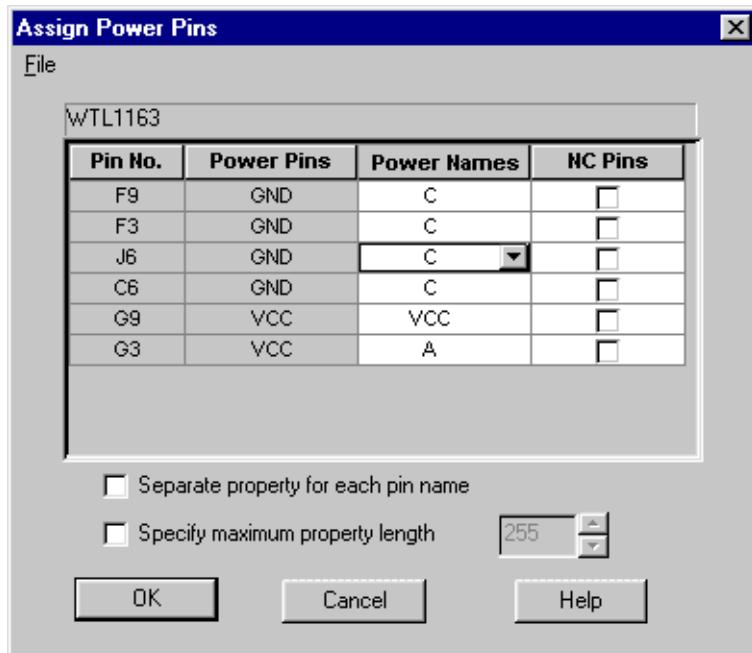
### Working with Properties and Text

4. Select C from the list of available global signals as follows



5. Click *OK*.

The *Power Name* dialog box closes and the power name for pins F9, F3, J6, and C6 changes to C as follows



## Allegro Design Entry HDL User Guide

### Working with Properties and Text

Now, the properties on instance of component WTL1163 are as follows

In chips.prt:

```
POWER_PINS = (GND:F9,F3,J6,C6;VCC:G9,G3)
```

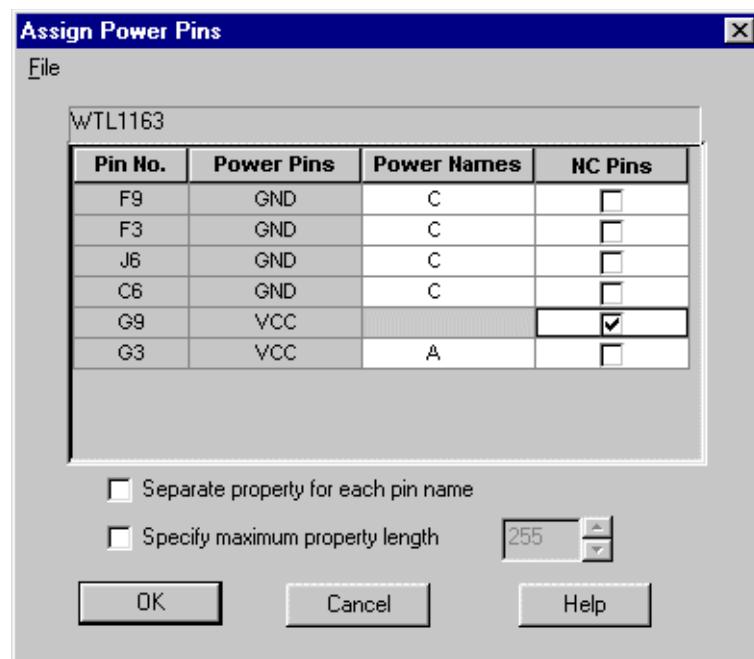
On the instance:

```
POWER_GROUP = (GND=C)
```

```
POWER_PINS = (A:G3;GND:F9,F3,J6,C;VCC:G9)
```

**To make power pin G9 an NC pin,**

- Select the check box in the *NC Pins* column as follows



### Example 2

Consider an instance of component INTERFACE2 that has the following properties:

In chips.prt:

```
POWER_PINS = (GND:13;VDD:12;VCC:11)
```

```
NC_PINS = (14,15,16)
```

On the symbol:

```
POWER_GROUP = (VCC= VCC1)
```

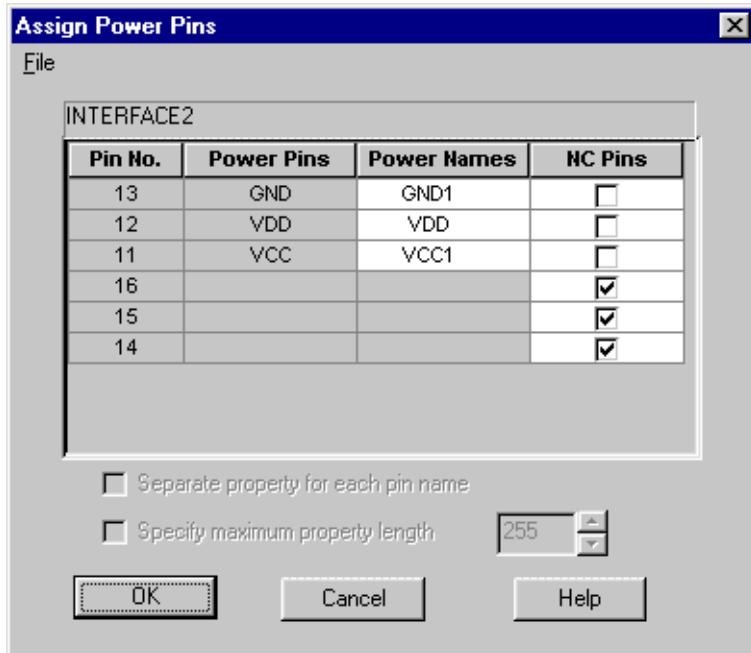
## Allegro Design Entry HDL User Guide

### Working with Properties and Text

On the instance:

```
POWER_GROUP = (GND=GND1)
```

The *Assign Power Pins* dialog box appears as follows



The check boxes *Separate property for each pin name* and *Specify maximum property length* are disabled because a property exists on the symbol of the component, which cannot be deleted.

### Example 3

Consider a component that has the following properties:

In chips.prt:

```
POWER_PINS = (VCC:1, 2, 3)
```

In .ptf:

```
POWER_GROUP = (VCC=VCC1)
```

On the instance:

```
POWER_GROUP = (VCC=VCC2)
```

If the SCH\_POWER\_GROUP\_WINS\_OVER\_PPT directive is not set in the .cpm file, the POWER\_GROUP property in the .ptf file combined with the POWER\_PINS property in the chips.prt file gives POWER\_PINS =(VCC1:1, 2, 3). This POWER\_PINS property together

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

with the `POWER_GROUP` property on the instance results in an error because there is no power name `VCC`.

Now, if you set the `SCH_POWER_GROUP_WINS_OVER_PPT` directive in the `.cpm` file, the `POWER_GROUP` property in the `.ptf` file is ignored. The `POWER_PINS = (VCC:1, 2, 3)` property combined with the `POWER_GROUP = (VCC=VCC2)` forms the `POWER_PINS = (VCC2:1, 2, 3)` property.

### Example 4

Consider an instance of component WTL1163 that has the following properties:

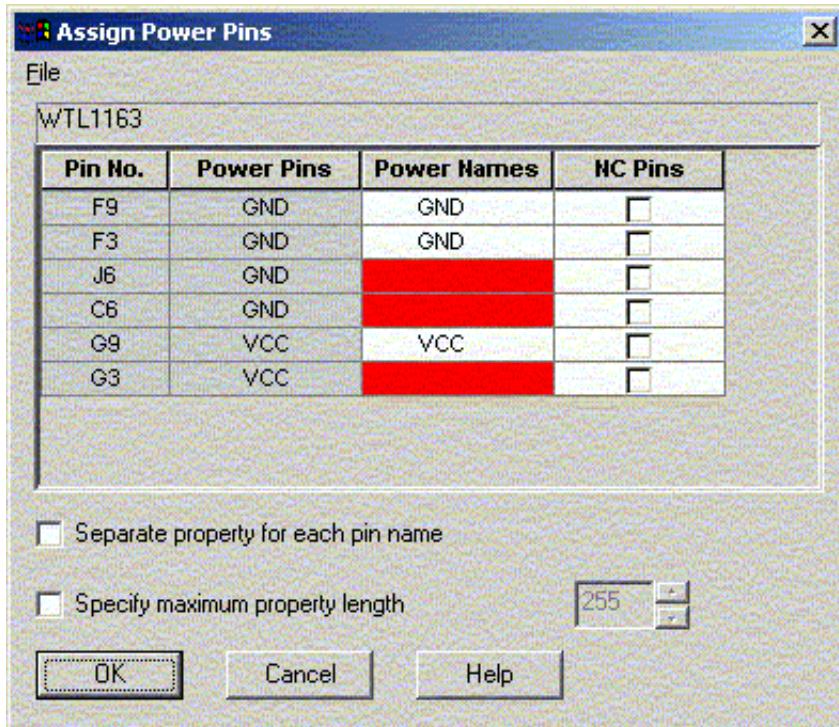
In `chips.prt`:

```
POWER_PINS = (GND:F9, F3, J6, C6; VCC: G9,G3)
```

On the instance:

```
POWER_PINS = (GND:F9, F3; VCC: G9)
```

The *Assign Power Pins* dialog box appears as follows



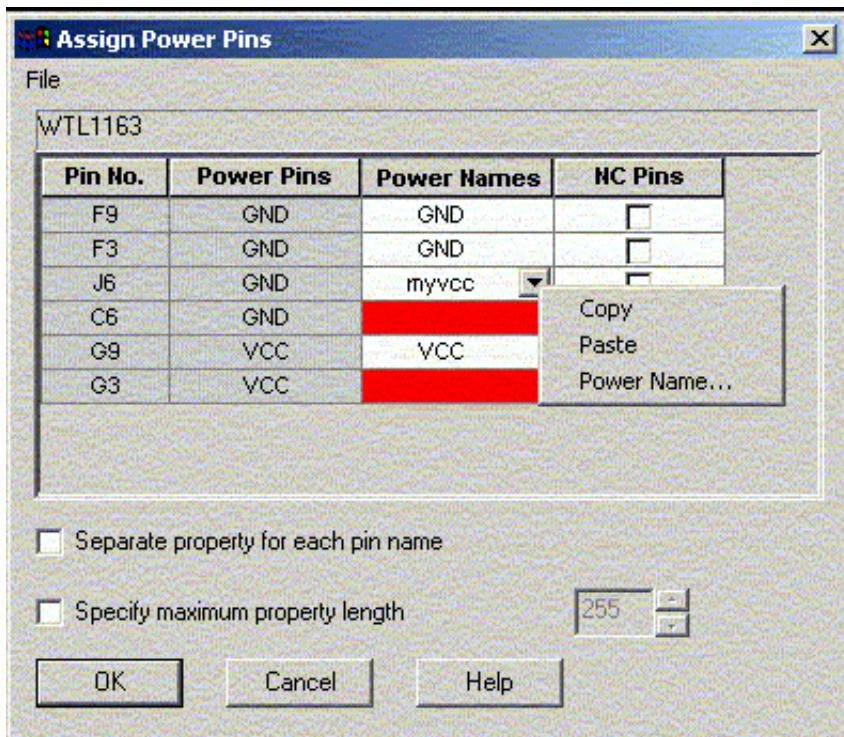
The fields in red indicate that pins that were assigned as power pins in the `chips.prt` file are not defined as power pins on the schematic. You must assign global signals to these pins.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

**To assign global signal MY\_VCC to pins J6, C6, and G3,**

1. In the *Power Names* field for pin J6, type MY\_VCC.
2. Right-click to bring up the popup menu as follows



3. Choose *Copy*.
4. Right-click on *Power Names* field for pin C6 and choose *Paste* from the popup menu.
5. Repeat step 4 for pin G3.
6. Click *OK*.

### Controlling the Overwriting of POWER\_PINS Property

It can happen that you are creating your schematic and the parts being used in the schematic in parallel. While designing the schematic, you are also assigning properties to power and NC pins of components. In this case, the `POWER_PINS` properties on the instance will override the properties written in the `chips.prt` file. But, if you want the properties in the `chips.prt` file to take priority, use the `ALLOW_POWER_PINS` directive.

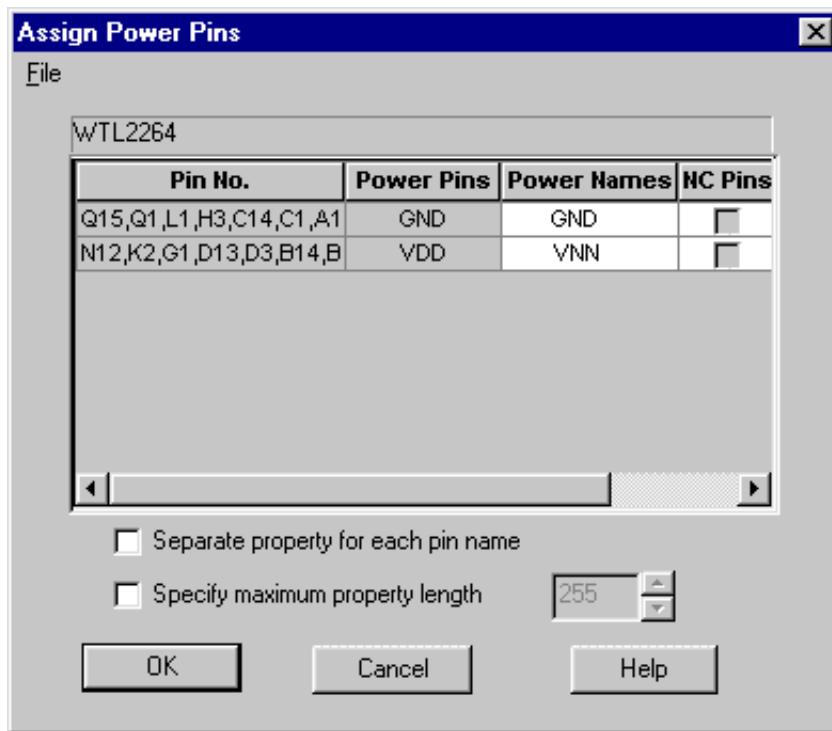
The `ALLOW_POWER_PINS` directive can be set in the `.cpm` file (or `site.cpm` file). It controls the overwriting of existing `POWER_PINS` property on an instance. By default, the directive

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

ALLOW\_POWER\_PINS is set to ON. If it is set to OFF, the POWER\_PINS property cannot be changed from the *Assign Power Pins* dialog box.

When the directive is set to OFF, Design Entry HDL reads only the POWER\_PINS properties from the chips.prt file and POWER\_GROUP property from the instance. In case POWER\_PINS and NC\_PINS properties are present on the instance, an error message is flagged that because the ALLOW\_POWER\_PINS directive is set to OFF, POWER\_PINS, NC\_PINS, MERGE\_POWER\_PINS, and MERGE\_NC\_PINS will not be read from or assigned to the instance. Then, the *Assign Power Pins* dialog box appears.



All the pins that have the same power source appear in one row. This means that you can change only the POWER\_GROUP property on the instance of the component.

### Assigning Power Pins to a Group of Components

You can use the *Assign Power Pins* dialog box to assign properties to a group of components also. The components in the group should satisfy the following conditions:

- belong to the same physical part
- have the same POWER\_PINS and POWER\_GROUP properties

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

Using a group to assign properties to all sections of a split part is particularly important because all the sections must have the same properties on their power and NC pins.

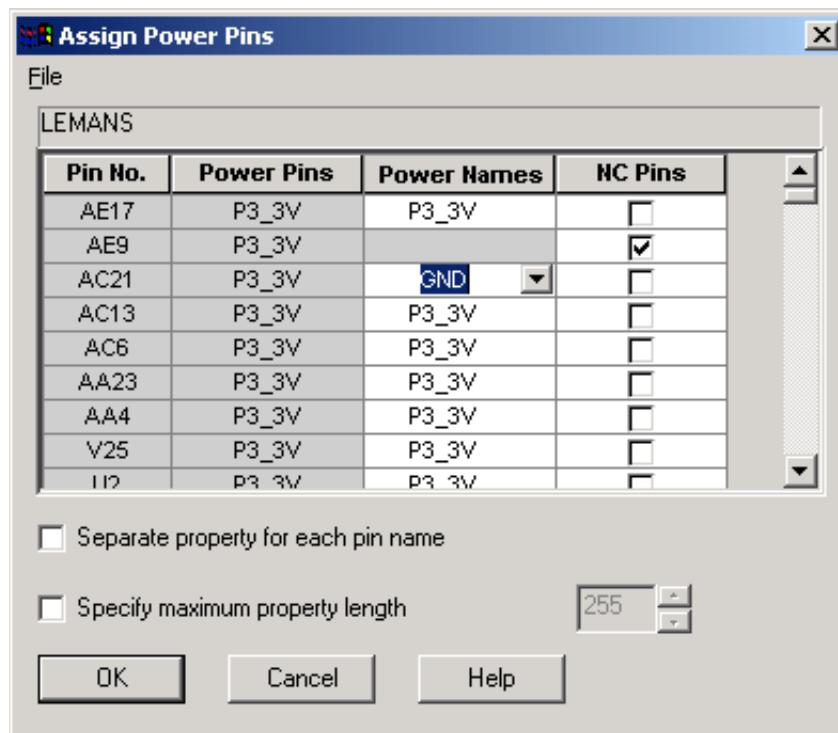
#### **Example**

Let us suppose that you have three sections of a split part on a page of a schematic. You want to make pin AE9 an NC pin, and change the power name of pin AC21 to GND for all sections.

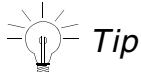
To edit the properties on these sections together, do the following

1. Create a group that consists of the three sections.
  2. Choose *Group – Assign Power Pins[A]*.
- The *Assign Power Pins* dialog appears.
3. For pin AE9, select the *NC Pins* check box.
  4. For pin AC21, select the GND name in the *Power Names* column.

The *Assign Power Pins* dialog box appears as follows:



5. Click *OK*.



*Tip*

If you have more instances of the component on other pages of the design, use the *Load Attributes* and *Save Attributes* options of the *Assign Power Pins* dialog box.

### Saving the Properties

If you want to assign the same properties to components across several pages of your design, you can use the *Load* and *Save* options of the *Assign Power Pins* dialog box. The properties for a component are saved in an attribute file, which has the following name:

`<library name>_<component name>.att`

***To load a file containing properties on power pins of a component,***

- Choose *File – Load* in the *Assign Power Pins* dialog box.

***To save a file containing properties on power pins of a component,***

- Choose *File – Save* in the *Assign Power Pins* dialog box.

### Subtype Names in POWER\_GROUP Property

A subtype name existing in a `POWER_GROUP` property does not appear in the *Assign Power Pins* dialog box. However, it is appended to the `POWER_GROUP` property when it is written on the instance even if changes made in the *Assign Power Pins* dialog box result in modification of the property. If the changes made in the *Assign Power Pins* dialog box result in deletion of the `POWER_GROUP` property, the subtype name information is lost.

For details on subtype names, refer to the *Preparing your Schematic for Packaging* chapter of the [\*Packager-XL User Guide\*](#).

## Working with Text Macros

This section describes the procedures for working with text macros in Design Entry HDL.

## About Text Macros

You use text macros to globally replace a string of characters with another. A text macro is a text template that represents variable information that can be used in different places. When the information changes value, you need to change only the macro definition.

Text macros are used for defining global information that is needed in many places. A text macro consists of a name (identifier) and a definition.

The rules for naming a text macro are as follows:

- It can consist of letters, digits and the underscore character only.
- It can start only with a letter.
- It cannot exceed 31 characters.

A text macro definition represents a character string up to 255 characters in length.

When you run Packager-XL, it replaces occurrences of each text macro with the strings it represents. For example, the text macro CDS can represent the string Cadence Design Systems. The process of replacing the text macros with the strings of characters they represent is called text macro expansion. In the current implementation, text macros can only be used in properties on instances. Packager-XL expands the text macro placed within a property value.

## How to use Text Macros

Text macros need to be identified within the property value with the '%' character.

For example:

PROP1 = 'W=%WIDTH, L=%LENGTH'

**Note:** The presence of the two text macros WIDTH and LENGTH in the property value is flagged with the '%' character. Packager-XL only expands the identifier following the '%' character. The comma marks the end of the macro identifier WIDTH and the end of string marks the end of the macro identifier LENGTH. In this example, if width was defined as 2 and length as 3, the above property would be expanded as

PROP1 = 'W=2, L=3'

You can use a space, a comma or an end of string character to separate one macro identifier from another. If the text macro is to be immediately followed by text (that is by any character acceptable as an identifier), enclose it in quotes.

For example:

PROP1 = 'This property value is % 'TM'ed.'

The text macro TM is identified by the quotes.

Text macros within property values can include parameters, but they cannot have embedded text macros (nested macros). If they do appear, they are ignored.

#### **Where to Define Text Macros**

There are two places to define macros

- on individual drawings using the DEFINE symbol or \parameter or \param on a hierarchical block
- in a text file

##### **Default Text Macro File**

No default text macro file is provided by Cadence.

##### **SCALD Compatibility**

The text macro file in HDL architecture is not compatible with SCALD.

#### **Defining Text Macros on a Drawing Using the DEFINE Symbol**

Text macros are defined in a drawing using DEFINE symbols. The DEFINE symbol is a part of the standard library located at <your\_inst\_dir>/share/library/standard.

To define a text macro for a drawing, add the DEFINE symbol and use the PROPERTY command or use the *Attribute* dialog box in Design Entry HDL to attach the property to the DEFINE symbol. The PROPERTY command expects a name/value pair separated by a space. The name/value pair corresponds to the identifier/definition of the macro.

For example, if you add to the DEFINE symbol CDS = 'CADENCE DESIGN SYSTEMS' and attach the property MY\_PROP = %CDS on an instance in the schematic, Packager-XL will interpret "CDS" as the macro identifier and "CADENCE DESIGN SYSTEMS" as the macro definition and accordingly substitute CDS with CADENCE DESIGN SYSTEMS in the property value. This property will appear in the Packager-XL output file (pstxprt.dat) as MY\_PROP = CADENCE DESIGN SYSTEMS.

There is no limit to the number of macros you can add to a DEFINE symbol or the number of DEFINE symbols you can add to a drawing. A text macro that is defined on a particular

drawing using the DEFINE symbol is operative within that drawing and all other drawing (modules) within its hierarchy.

**Note:** 16.5 release onwards, DEFINE bodies are supported at a design level and do not process page level properties if they differ in value. In case of multiple DEFINE bodies on multiple pages, use DEFINE body on the page with the lowest physical page number.

## Defining Text Macros Using \PARAMETER or \PARAM

Packager-XL allows you to pass values of macros down to one level by defining macros using \parameter or \param. To define a text macro using \parameter or \param, suffix the term \parameter or \param to the property value string.

Both \param or \parameter (case insensitive) are treated as potential text macros. All properties that you define using \PARAM or \PARAMETER are written into the viewprops.prp file for the block where they are defined.

Text macro substitution takes place only when Packager-XL is run in the forward mode.

**Note:** Text macros is not supported for LOCATION property.

### Example

Consider the example of a block CIRCUIT1, which has two instances ICA and ICB. The VCC pin on ICA must be connected to 5V power supply and the VCC pin on ICB must be connected to 3V power supply. The property VOLTAGE = 'VALUE1'V is attached to VCC pin of the block CIRCUIT1. If the property VALUE1 = 5\param is attached to block instance ICA and the property VALUE1 = 3\param is attached to block instance ICB, then running Packager-XL will cause the property VOLTAGE to have values 5V and 3V for ICA block instance and ICB block instance, respectively.



Do not use E\PARAM (case insensitive) as an attribute value. Doing so results in an error. This is because the letter e is treated as an integer and not as a string. Any other single letter (a through z), or a combination of letters, is treated as a string during Verilog generation.

For any string, the defparam statement is written as:

```
defparam page1_i1.prop = "string"; (with parameter value in quotes)
```

While, for any numeric parameter value, the defparam statement is written as:

```
defparam page1_i1.prop = 121; (parameter value without quotes)
```

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

In case of property value being e\param, the quotes are not written:

```
defparam page1_i1.prop = e;
```

In case using E\PARAM is unavoidable, use the following recommended syntax:

```
Vlog_param1= <prop name>:TYPE
```

```
<prop name> = VALUE
```

For example, if you require SFX = E\PARAM, you specify the following:

```
vlog_param1=SFX:string
```

```
SFX=E
```

## Defining a Text Macro in a File

You can define text macros that are known globally in all modules in a text macro file. When you define a global text macro in a text file, the macro cannot be overridden. A macro defined here overrides macros defined using the DEFINE symbol and using the \parameter option.

### Name and Location

The reserved name for text macro file is cdsprop.tmf.

**Note:** The text macro file loaded by Packager-XL from <your\_inst\_dir>/cdssetup/cdsprop.tmf. Currently, this file is empty.

The search is done in the following sequence:

- current working directory
- ~ (home)
- CDS\_SITE/share
- <your\_inst\_dir>/cdssetup

### Syntax

The text macro file contains a list of macro identifiers and associated definitions. A text macro specification is defined within one line in the file and has the following syntax:

```
macro_identifier = 'macro_definition'
```

where macro\_identifier is expressed in the VHDL name space and macro\_definition is a string enclosed in quotes.

**Note:** The space and tab characters are always ignored outside tokens. Comments are allowed anywhere outside a token, and they begin with “#” until the end of the line.

For example, in a schematic that has two blocks within it, BLOCK1 and BLOCK2, a property MY\_LOC=5\PARAMETER is attached to BLOCK1 with the ls04 instance inside it having property LOCATION=U%'MY\_LOC'10. BLOCK2 has property MY\_LOC=8\PARAMETER and instance ls32 inside it has property LOCATION=U%'MY\_LOC'20 attached to it. A DEFINE symbol inside BLOCK2 contains the macro MY\_LOC=6. Now, create the cdsprop.tmf file in the project directory, which has macro defined as MY\_LOC=3.

When you save the schematic and package the design, global substitution of the macro takes place at all levels of the hierarchy with the macro value defined in text macro file. Thus, the property on ls04 appears as LOCATION=U310 and the property on ls32, which is inside block2, appears as LOCATION=U320.

## Working with Custom Text

Custom text is context-specific text that you can attach to the origin of a symbol or to objects on a schematic.

Custom text is different from notes and comments because of the following two reasons:

- Custom text is attached to objects
- Custom text can be context-specific

Custom text is specified in the form of two strings:

- Format string

The format string specifies the format in which the actual custom text is to be displayed. It may contain Custom Variables and environment variables.

- Display string

The display string contains the substituted values of the custom and environment variables. This is the string that actually appears on the schematic.

Custom text is made context-specific by adding Custom Variables and environment variables to it.

## Custom Variables

Custom variables are Design Entry HDL variables that take values depending on where they are placed. They are of the following two types:

- Inbuilt Design Entry HDL variables

The values for custom variables that are defined within Design Entry HDL are supplied by Design Entry HDL itself. The case of the values is the same as used by Design Entry HDL. For example, in Design Entry HDL, the design names and library names are in lowercase. So, the value of an inbuilt custom variable, say CON DESIGN LIB, would also be in lowercase. These variables are not visible in the Design Entry HDL Options dialog box as you cannot change their values.

- User-defined variables

You can define new custom variables for storing and displaying information such as COMPANY NAME and AUTHOR. Design Entry HDL writes the user-defined custom variables for a project in the .cpm file. You can define these variables and their values in the Design Entry HDL Options dialog box. For more information on defining new custom variables, see [Defining New Custom Variables](#) on page 382.

As per requirement, you can define custom variables specific to your site (in the site.cpm file) or specific to your home (in your home's site.cpm file) or specific to your project (in the project's .cpm file).

Custom variables make the plots of cross-referenced schematics more illustrative and easy to use. Design Entry HDL provides some pre-defined variables whose values are substituted by CRefer. These are available in cross-referenced schematics. Custom text is not visible in HPF plots.

## Examples of Custom Text

### **Example 1**

Using two inbuilt custom variables, <CON\_PAGE\_NUM> and <CON\_TOTAL\_PAGES>, specify the format string as:

```
This is page <CON_PAGE_NUM> of <CON_TOTAL_PAGES>
```

If the above custom text is added on page 1 of a 10-page schematic, it appears as:

```
This is page 1 of 10
```

and if it is placed on page 5 of a 10-page schematic, it appears as-

```
This is page 5 of 10
```

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

Hence the name custom text, which means that the text is displayed depending on which page it is added on.

#### **Example 2**

Using a user-defined custom variable AUTHOR, specify the format string as:

The author of this design is <AUTHOR>

If you have defined a variable AUTHOR = Bob, the display string is:

The author of this design is Bob

For details on defining new variables, please see [Defining New Custom Variables](#).

#### **Example 3**

Using an environment variable <CONCEPT\_INST\_DIR>, specify the format string as:

The software is located in <\$CONCEPT\_INST\_DIR>

If the value of CONCEPT\_INST\_DIR is set to C:\Programs\Cadence, the display string is:

The software is located in C:\Programs\Cadence

## Inbuilt Design Entry HDL Variables

### **Drawing-Specific Variables**

The values of these variables are different across different drawings.

- CON DESIGN LIB  
Name of the design library
- CON DESIGN NAME  
Name of the design
- CON DESIGN VIEW  
Currently open view of the design. It is one of the schematic views
- CON PAGE NUM  
Current page of the drawing in a cell. It is different for each page of the drawing.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

---

This variable does not take into account the pages in all the cells used in a hierarchical design. For more information, see [The CON\\_PAGE\\_NUM and CON\\_TOTAL\\_PAGES Custom Text Variables](#) on page 488.

#### ■ CON\_TOTAL\_PAGES

Total number of pages in a cell. The value of CON\_TOTAL\_PAGES will be the value of the highest page number you have assigned for a page in the cell. For example, if you have four pages in a cell with numbers 1, 2, 3, and 6, the value of CON\_TOTAL\_PAGES is 6.

This variable does not take into account the pages in all the cells used in a hierarchical design. For more information, see [The CON\\_PAGE\\_NUM and CON\\_TOTAL\\_PAGES Custom Text Variables](#) on page 488.

#### ■ CURRENT\_DESIGN\_SHEET

Current page of the drawing in a hierarchical design. For more information, see [The CURRENT DESIGN SHEET and TOTAL DESIGN SHEETS Custom Text Variables](#) on page 489.

#### ■ TOTAL\_DESIGN\_SHEETS

Total number of pages in a hierarchical design. For more information, see [The CURRENT DESIGN SHEET and TOTAL DESIGN SHEETS Custom Text Variables](#) on page 489.

## ***Global Variables***

The value of these variables is constant across all pages of the schematic.

#### ■ CON\_ROOT\_LIB

Library to which the root design belongs

#### ■ CON\_ROOT\_NAME

Name of the root design

#### ■ CON\_ROOT\_VIEW

View of the root design

#### ■ CON\_PROJ\_NAME

Name of the current project file

#### ■ CON\_PROJ\_FULLNAME

Absolute path of the current project file

#### ***Parent Variables***

The value of these variables is set when you descend into the hierarchy or edit using the canonical name of the design. These variables are not set for root designs.

- CON\_PARENT\_NAME  
Name of the parent design
- CON\_PARENT\_CNAME  
Canonical name of the parent design. The value is substituted in accordance with the Design Entry HDL option of displaying lib, cell, view.
- CON\_PARENT\_LIB  
Name of the library to which the parent of the current design belongs
- CON\_PARENT\_VIEW  
View of the parent design

#### ***CRef Variables***

The value of these variables is set in the `schcref_1` view when the design is cross-referenced with the option *Create Flattened Schematic*.

- CREF\_TO\_LIST  
Defines where the pages for the blocks are located in the cross-referenced flattened design.
- CREF\_FROM\_LIST  
Defines where the pages are coming from in a flattened design
- CREF\_ORIG DESIGN NAME  
Defines the original design name
- CREF\_ORIG PAGE  
Defines the original page number
- CREF\_ORIG VIEW  
Defines the view of the original design

## Using Custom Text

Custom text can be added to the symbol of a page border. To add custom text to symbols, attach it to the origin. The symbol then displays the format string. When the page border is instantiated, the values of custom variables are substituted. For example, if you have the `Page <CON_PAGE_NUM>` custom text on the page border symbol then instantiate the page border, the custom variable `CON_PAGE_NUM` take its actual value on each page. For example, Page 1 or Page 2.

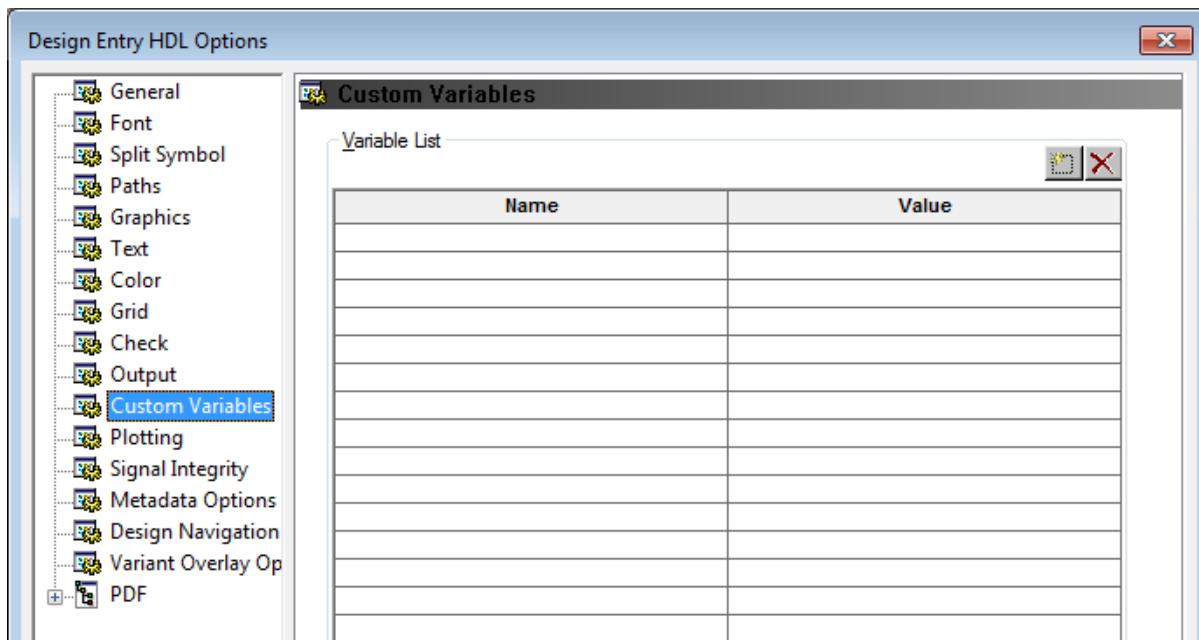
### Defining New Custom Variables

You can define your own variables to use in a design project. These variables can be defined using the *Design Entry HDL Options* dialog box.

To define new variables, do the following:

1. Choose *Tools – Options*.

The *Design Entry HDL Options* dialog box appears.



2. Click on *Custom Variables*.
3. Enter the name of the variable.
4. Enter the value of the variable.

## Allegro Design Entry HDL User Guide

### Working with Properties and Text

The value is constant for all pages of the design.

For example, you can enter AUTHOR as the name of the variable and SMITH as the value.

You can define more variables by clicking on the *Add* button (  ).



You cannot leave the value of the variable blank. The variable is deleted from the list if no value is specified.

5. Click *Apply*.

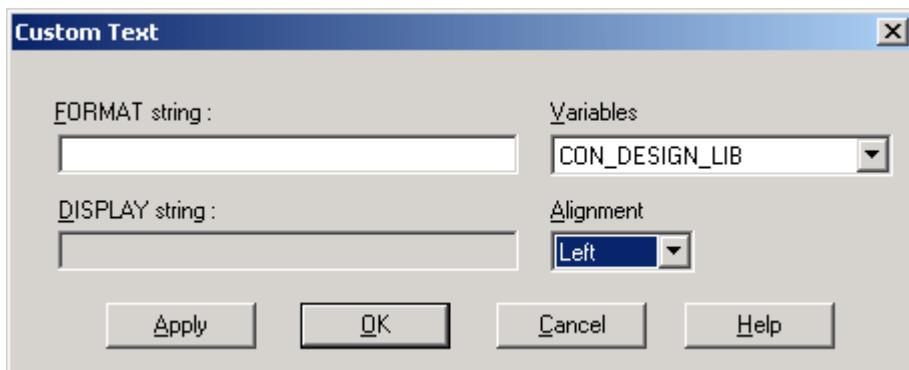
6. Click *OK*.

Design Entry HDL adds the custom variable to your design project.

### Adding Custom Text

1. Choose *Text – Custom Text*.

The Custom Text dialog box appears.



2. Enter the format string for the custom text.

3. Select the variable from the Variables drop-down list or type your own custom variable. You can also add an environment variable.

The variable is added to the format string. The *DISPLAY string* field displays this line with the current value of the variable.

If you want to add an environment variable to the format string, precede it with an \$ sign. The current value of the environment variable is displayed in the *Display string* field.

4. Select the *Alignment* of the text as *Left*, *Center*, or *Right*.

#### 5. Click *Apply*.

The values of the custom variables in the *Display string* are updated depending upon the current page and the custom text is attached to the cursor.

#### 6. Click on an object to attach custom text to it.

**Note:** If you are adding custom text on a symbol, click on the origin of the symbol to attach the custom text.

#### 7. Click again to place the custom text at the location where you want it to be displayed.

#### 8. Repeat steps 4 to 7 to add the same text on different pages of the design.

#### 9. Click *OK*.

The custom text is attached to the cursor.

#### 10. Right-click and select *Done*.

**Note:** You can add multiple custom text to the same object.

### Case-Sensitivity in Custom Text

You have the option of defining the default casing for custom text using *Design Entry HDL Options – Text* and checking the *Upper-case Input* box. While setting a variable, if this check box is checked, the variable name is stored in uppercase. When adding a variable to custom text, the text is automatically converted to upper case.

### Modifying Custom Text

#### 1. Choose *Text – Change*.

#### 2. Click on the custom text you want to modify.

The *Custom Text* dialog box appears with the format string highlighted.

#### 3. Edit the text in the *FORMAT* string field.

#### 4. Click *OK*.

The custom text is modified.

**Note:** You can move, copy, rotate, spin, and delete custom text.

---

# Working with Block Designs

---

This section describes the procedures for using blocks to create hierarchical designs in Design Entry HDL.

## About Blocks

Block diagrams let you create and edit symbols for top-level drawings. The symbols can then be replaced with functional designs.

All blocks have the property BLOCK=TRUE attached to the origin of the block.

These conventions apply to blocks:

- Show inputs on the left side of the block.
- Show outputs on the right.
- Any signal going through the top or bottom of a block defaults to an INOUT.

**Note:** Even if you have a BLOCK=TRUE property on a symbol, you cannot edit it. Design Entry HDL does not support this feature currently.

## About View Generation in Hierarchical Designs

Genview lets you generate a design view from an existing view. A design can be represented by these views:

- Schematic (SCH)
- Symbol (SYM)
- VHDL
- Verilog

You can generate views:

- Top down, where top-level symbol drawings are converted into VHDL or Verilog templates.
- Bottom up using a schematic or VHDL or Verilog text to create a symbol drawing.

## Generating Views for Top-Down Design

After creating a top-level block diagram, you can create the corresponding VHDL or Verilog template. You can view or edit the templates by adding properties to the origin of a symbol that was created using *Tools – Generate View*. You can also add pin properties to change the default mode and type of ports in the VHDL or Verilog template.

## Generating Views for Bottom-Up Design

In a bottom-up design, you can create a symbol from a VHDL or Verilog template.

Values for VHDL and Verilog properties are obtained from the template file. Using the template lets you create a symbol with minimal editing and reduces errors from pin name mismatching.

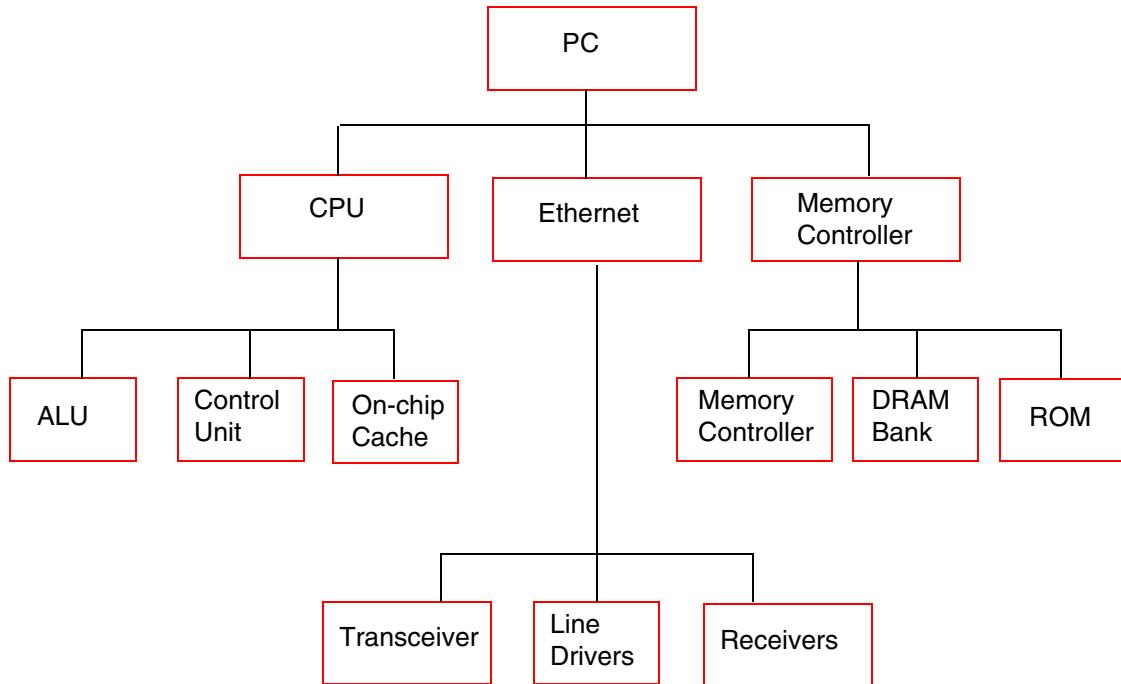
**Note:** You will get the following error if the xcon file is present at the top in the master.tag file. to resolve the error manually edit the file.

"Genview supports only SCHEMATIC/SYMBOL/VERILOG/VHDL as input type.  
.xcon not recognised."

## Creating Hierarchical Designs

A hierarchical design is a large and complex design divided into sub designs. Each of the sub designs can be further divided into sub designs. For example, if you have a design called PC that contains sub-designs CPU, Ethernet, and Memory Controller, PC (the top-level design) is called a hierarchical design.

### Example of a Hierarchical Design



The hierarchical design method is typically followed for large and complex designs. These designs are divided into individual modules where each module represents a logic function.

To create a hierarchical schematic in Design Entry HDL, you can choose either of the following methods:

- Top Down method
- Bottom Up method

### Top Down Method

In the Top Down method, you first create the top-level drawing (PC in this case). In the top-level drawing, you can add blocks that represent individual modules. In the case of PC, the top-level drawing will have three blocks:

- CPU
- Ethernet
- Memory Controller

After creating the top-level drawing with the necessary blocks, you create the lower level schematics and save them as cells. These schematics should have the same names as those of the blocks in the top-level schematic.

For example, If the blocks in the schematic PC are named CPU, Memory, and Ethernet, the lower level schematics should be named CPU.SCH.1.1, Memory.SCH.1.1, and Ethernet.SCH.1.1 respectively. These names will ensure that Design Entry HDL links the schematics with the blocks. When you double-click on the block CPU, Design Entry HDL descends to CPU.SCH.1.1.

To create the hierarchical design

1. Create a top-level schematic. For this example, call it PC.SCH.1.1
2. Add three blocks to PC.SCH.1.1.

Name the blocks CPU, MEMORY, and ETHERNET. Choose Block – Rename to rename the blocks.

3. Choose *File – New* to create a schematic for CPU.

4. Add the following blocks:

- ALU
- CU
- OCC

5. Choose *File – Save As...* to access the *View Save As* dialog box.

6. In the *Library* field, choose *pc\_lib*.

7. In the tree view, select CPU.

8. Select *Schematic* in the *View* field.

9. Click *Save*.

The schematic you created for the CPU block is saved as CPU.SCH.1.1.

Similarly, you can create schematics using *File – New* and save them using *File – Save As* for ALU, CU, and OCC in CPU.SCH.1.1. After completing these tasks, use *File – Return* to return to the top-level drawing, PC.SCH.1.1. In PC.SCH.1.1, You can double-click on the CPU block to descend through the hierarchy.

You can complete the design PC by creating blocks and schematics for all levels in the design.

## Bottom Up Method

In the Bottom Up method, you can create a low-level schematic first. For the design PC again, you can create the schematic drawings for ROM, DRAM Bank, and Memory Controller. Name the drawings ROM.SCH.1.1, DRAM.SCH.1.1, and MEM.SCH.1.1, respectively.

You then create the schematic for a higher-level drawing; for example, Memory Controller. Name the schematic Memory.SCH.1.1. In Memory.SCH.1.1, create three blocks and name them (*Block – Rename*) ROM, DRAM, and MEM. After saving the blocks, you can descend to MEM.SCH.1.1 by double clicking on block MEM in Memory.SCH.1.1.

## Adding a Block

To add a block using the Design Entry HDL block name

1. Choose *Block – Add*.
2. Click where you want to place the block, move the cursor diagonally, and click again.  
**Note:** If you are zoomed too far in on the drawing, Design Entry HDL warns that blocks must have a minimum width and height. Zoom out to place the block.
3. Design Entry HDL assigns the name BLOCKn. You can change block names at any time by choosing *Block – Rename*.

To add a block and name it yourself

1. Choose *Block – Add*.
2. Right-click and choose *Block Name...* from the pop-up menu.
3. Type a name in the *Block Name* box.
4. If you enter the name of an existing block, a copy of the specified block attaches to the cursor. Click where you want to place the block.

If you enter a unique name, click where you want to place the block, move the cursor diagonally, and click again.

**Note:** If you are zoomed too far in on the drawing, Design Entry HDL warns that blocks must have a minimum width and height. Zoom out to place the block.

## Renaming a Block

1. Choose *Block – Rename*.

2. Enter a block name in the *Block Name* box.
3. With the new block name attached to the cursor, click the block that you want to rename.

**Note:** If you specify an existing block name, Design Entry HDL asks if you want to overwrite the existing block. Choose *Yes* or *No*.

### Resizing a Block

1. Choose *Block – Stretch*.
2. Click a corner or side of the block that you want to stretch, move the cursor to resize the block, and click again.

### Wiring Blocks

To manually draw a wire between blocks

1. Choose *Block – Draw Wire*.
2. Click the edge of one block.
3. Click wherever you want the wire to bend, or click the edge of another block.

**Note:** If no block pins exist where you want to add a wire, Design Entry HDL adds pins and names them  $\text{PIN}_n$ . You can change this name at any time by choosing *Block – Rename Pin*.

### Tips for Wiring Blocks

- Click left to end a wire at a pin, dot, or other wire.
- Click left twice at the final point to end a wire in a free space.

Click **Ctrl+left** and continue clicking to change the bend of the wire.

To auto-route a wire between blocks

1. Choose *Block – Route Wire*.
2. Click the edge of one block and then click the edge of another block.

**Note:** If no block pins exist where you want to add a wire, Design Entry HDL adds pins and names them  $\text{PIN}_n$ . You can change pin names at any time by choosing *Block – Rename Pin*.

**Note:** You can also run the `route` command using this stroke pattern:

#### Mirroring Components or Blocks

1. Choose *Edit – Mirror*.
2. Click a component or block.

#### Displaying Block Properties

All blocks have a BLOCK=TRUE property attached to the block's origin. By default, this property is not displayed. This property distinguishes a block component from a body component.

To display the BLOCK=TRUE property

1. Display the console window, and enter the command `display both`.
2. Select a block.

The BLOCK=TRUE property appears near the origin of the block.

#### Adding Block Pins

1. Choose *Block – Add Pin*.
2. Choose the type of the pin you want to add.  
The *Block Pin Add* dialog box appears.
3. Type one or more pin names on separate lines.
4. Click the edge of the block in the same order that you entered pin names.

Design Entry HDL adds the pins where you specify.

**Note:** To toggle the pin type before you place the pin on the block, right-click and choose *Change Mode*. Alternately, press *Ctrl* and click the left mouse button in a two-button mouse or click the middle mouse button in a three-button mouse.

#### Renaming Block Pins

1. Choose *Block – Rename Pin*.  
The *Block Pin Rename* dialog box appears.
2. Type one or more pin names on separate lines.
3. Select existing pins that you want to rename.

Design Entry HDL changes the name of the pins you select.

#### **Deleting Block Pins**

1. Choose *Block – Delete Pin*.
2. Click the pins you want to delete.

**Note:** Click the pin, not the pin name.

#### **Moving Block Pins**

1. Choose *Block – Move Pin*.
2. Click the block pin that you want to move.
3. Click the pin's new location.

**Note:** You cannot move a pin across components.

### **Using Read-only Blocks in Your Design**

You can create a reusable block using Design Entry HDL and PCB Editor and maintain it in a reference library. You can then use the block as a read-only block in other designs.

#### **Navigating the Drawing Hierarchy**

To view a block diagram from the top-level schematic, do the following:

1. Choose *File – Edit Hierarchy – Descend*.
2. Click a block in the schematic.  
Design Entry HDL descends into the symbol view.
3. Continue descending the drawing hierarchy by repeating steps 1 and 2.

#### **Example**

If you have vlog\_rtl, sch\_1, and sym\_1 views of the drawing and wish to descend into them when you double-click on the top level drawing, set the following environment variable.

```
Setenv CONCEPT_DESCEND_EDIT_LIST vlog_rtl, sch_1, sym_1
```

After setting this environment variable, when you double-click on the drawing, Design Entry HDL searches for the vlog\_rtl view and displays it. If this view is not present, Design Entry HDL displays the sch\_1 view.

#### ***To ascend the drawing hierarchy from a lower level block diagram***

1. Choose *File – Edit Hierarchy – Ascend*.
2. Continue ascending the drawing hierarchy by repeating steps 1 and 2.

#### ***To return to the previous drawing***

- Choose *File – Return*.

**Note:** You can view a list of the drawings that Design Entry HDL will return and the order in which they will be accessed by choosing *Display – Return*.

**Note:** Release 15.7 onwards, UNC path names are supported for Genview.

### **Generating a Design View**

1. Choose *Tools – Generate View*.

The *Genview* dialog box appears.

2. Specify the source view in the lib.cell:view format.

You can also specify a Verilog or VHDL source file from which you want to generate the view.

- Select *Verilog* in the *Type* drop-down list if you have selected a Verilog source file.
- Select *VHDL* in the *Type* drop-down list if you have selected a VHDL source file.

3. If the source is a file, select the destination library where you want Design Entry HDL to create the destination cell.

If the source is a view, the destination library is the same as the library for the source view.

4. Select the view that you want to generate in the *View* drop-down list.

5. In the *Type* drop-down list, select the type of the view you have selected in the *View* drop-down list:

Select the type	If you have selected the following view
Schematic	sch_1
Symbol	sym_1
VHDL	vhdl_1
Verilog	vlog_1

6. Select the *Retain Graphics* check box if you want to retain the placement of pins that already existed on the graphic for the symbol.

For example, suppose that the symbol view already exists. If you add or delete a pin in the source view or source file and regenerate the symbol view, the placement of the pins that already existed (pins that were not deleted in the source view or source file) on the symbol will be retained.

Cadence recommends that you use this option if you have already used the symbol on your schematic. This will ensure that the connectivity between a wire and a pin of the symbol on the schematic is not lost because the placement of the pin on the symbol does not change.

If you do not select this check box, the graphic for the symbol is regenerated and the pin placement is done by Design Entry HDL using its internal algorithms.

7. Select the *Split Vector Ports* check box if you want the vectored ports in the source view or source file to be split into multiple pins (representing each bit of the vectored port) on the symbol.

For example, if the source view or source file has a vectored port `DATA<3..0>`, the following four pins will be added on the symbol:

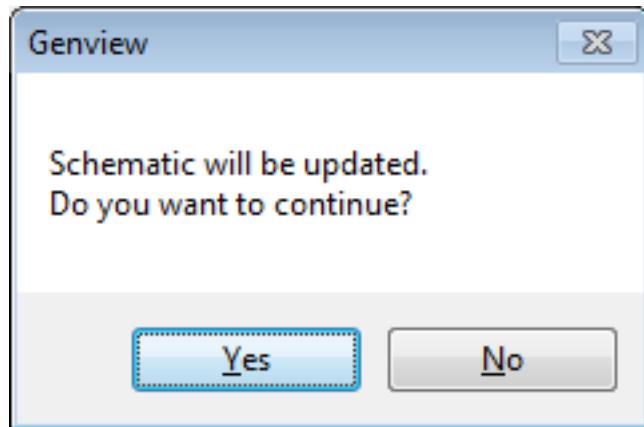
- DATA<3>
- DATA<2>
- DATA<1>
- DATA<0>

If this check box is not selected, the symbol will have a pin named `DATA<3..0>`.

8. Click *Generate*.

The *Output* field displays the results of the generate view process.

**Note:** Before updating the schematic view, a confirmation dialog box prompts you to continue with the updating the schematic.



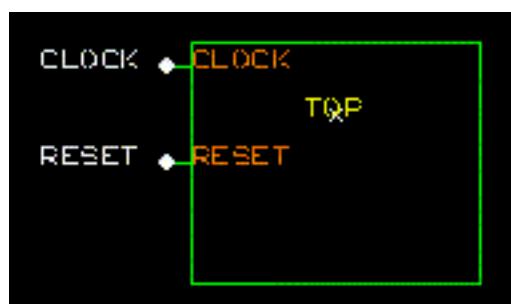
#### **Example of Using Retain Graphics and Split Vectored Ports Options**

The *Retain Graphics* and *Split Vectored Ports* options are explained below using an example.

Suppose you have a schematic `TOP.SCH.1.1` as illustrated:



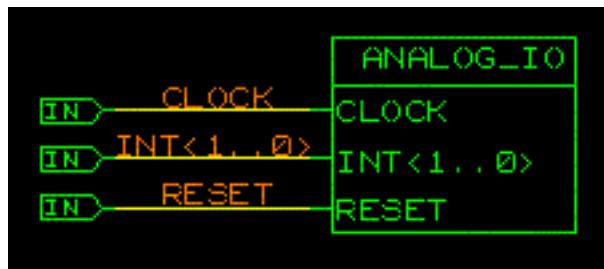
1. Generate the symbol for the schematic. A symbol named `TOP` will be created as below:



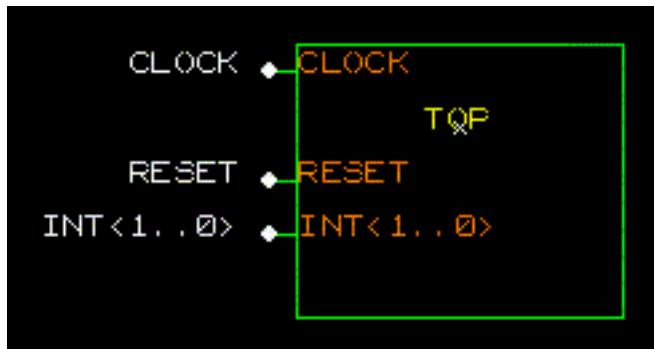
## Allegro Design Entry HDL User Guide

### Working with Block Designs

2. Add a pin, `INT<1..0>`, on the `ANALOG_IO` block and connect it to an input port `INT<1..0>` on the schematic:



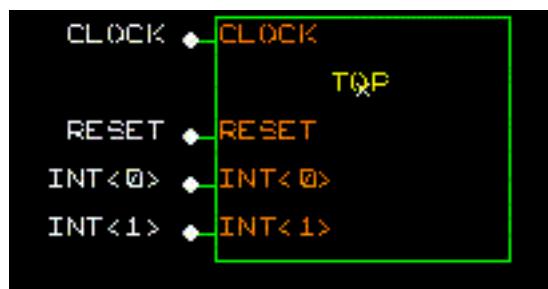
3. Generate the symbol for the schematic again with the *Retain Graphics* check box selected. The symbol `TOP` will be created:



Note that the placement of the `CLOCK` and `RESET` pins on the symbol has not changed.

Suppose you have instantiated the symbol `TOP` on some other schematic page and have connected a wire to the `CLOCK` pin on the symbol. The connectivity between the wire and the `CLOCK` pin is not lost now because the placement of the pin on the symbol has not changed.

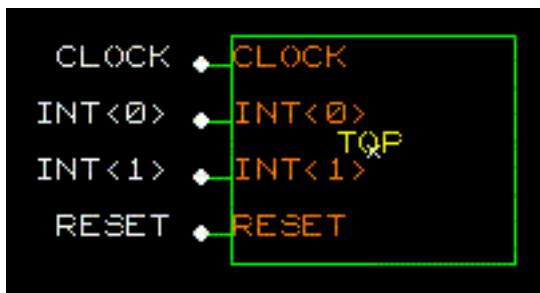
4. Generate the symbol for the schematic again with the *Retain Graphics* and *Split Vector Ports* check boxes selected. The `TOP` symbol will be created as follows:



Note that the vectored port, `INT<1..0>`, in the schematic has been split into two pins (representing each bit of the vectored port) — `INT<0>` and `INT<1>` — on the symbol. The vectored pin `INT<1..0>` is deleted from the symbol, and the pins `INT<0>` and `INT<1>` are added as new pins on the symbol.

Also note that the placement of the pins `CLOCK` and `RESET` on the symbol has not changed.

5. Generate the symbol for the schematic again with the *Retain Graphics* check box deselected and the *Split Vector Ports* check box selected. The symbol `TOP` will be created as follows:



Note that the placement of the pins on the symbol has changed. This is because the graphic for the symbol is regenerated when the *Retain Graphics* check box is not selected.

Suppose you have instantiated the symbol `TOP` on some other schematic page and connected a wire to the pin `CLOCK` on the symbol. The connectivity between the wire and the pin `CLOCK` is lost now because the placement of the pin on the symbol has changed.

### Adding a Symbol with Physical Part Information

1. Display Part Information Manager.
2. Select a library from the *Library* list in the search pane.
3. Select a component in the *Cells* list.

If physical information is available for that component, physical part names are listed in the Search Results pane. If no PPT is found for a part, you can place a symbol that serves as a placeholder for physical information which you can add later.

4. Select a part name.
5. Right-click and select *Add to Design* to add the cell in the schematic.

#### How does Genview behave when creating block symbols that should not include an entire bus in a port?

Genview extracts port information from the generated Verilog file. Verilog does not recognize partial buses and requires the entire bus to be declared as a port. As a result, using Genview you cannot create block symbols that include a partial bus in the port. When Genview creates a block symbol, all the bits are included on the port of the block symbol.



Manually changing the `pin_name` block symbol to the desired number of pins results in HDL error 267: *Port range specified in the schematic and symbol is different. Modify schematic/symbol to make port range same.* For example, this error occurs if the port range on the block schematic is `ADDRESS<17..0>` and the port range specified on the symbol is `ADDRESS<18..0>`.

**Workaround:** You must port all bits of a bus to the block symbol. If all the bits are not used, manually split the bus pins on the block symbol, so that you can annotate the unused pins with `PIN_TEXT`.

**Example:** A block schematic contains bus `ADDR<17..0>` connected to an IMPORT. The schematic also contains a net `ADDR<18>` that is connected separately but not included in the port.

When Genview creates the block symbol, all 19 bits are included in the vectored port when the intent is to only include `ADDR<17..0>` on this port of the block symbol. Keep the hierarchical schematic as is (all 19 bits of the address bus defined) and generate the view. Then, manually edit the hierarchical block symbol so that you split the bus on the symbol, show `ADDR<17..0>` as a hierarchical bus port, then add an additional single bit pin, `ADDR<18>`. Using `PIN_TEXT`, label this pin as `ADDR<18>` (grounded). That way, all the 19 pins are present in the block symbol and the MSB pin shows that it is connected to ground.

#### Does the REMOVE property work on blocks?

The REMOVE property is a simulation property and is ignored during netlisting. It applies only on components and not on blocks.

---

## **Working with Hierarchical Split Symbols**

---

Large hierarchical block symbols can become difficult to manage because of the substantial number of pins coming out of a single symbol. At times, such blocks cannot be placed on the standard page border either.

### **Overview – Hierarchical Split Symbols**

Allegro Design Entry HDL provides a solution to manage large hierarchical block symbols by splitting them into multiple split symbols. Instead of generating one difficult-to-manage large symbol, you can split the ports of a hierarchical block across multiple symbols. This support for hierarchical split symbols (referred to as HSS from this point on in the document) reduces the size of the block symbol. Additionally, the ports can be logically categorized and placed on different symbols to create symbols which can be placed across schematic sheets, especially near the circuitry to which they are connected.

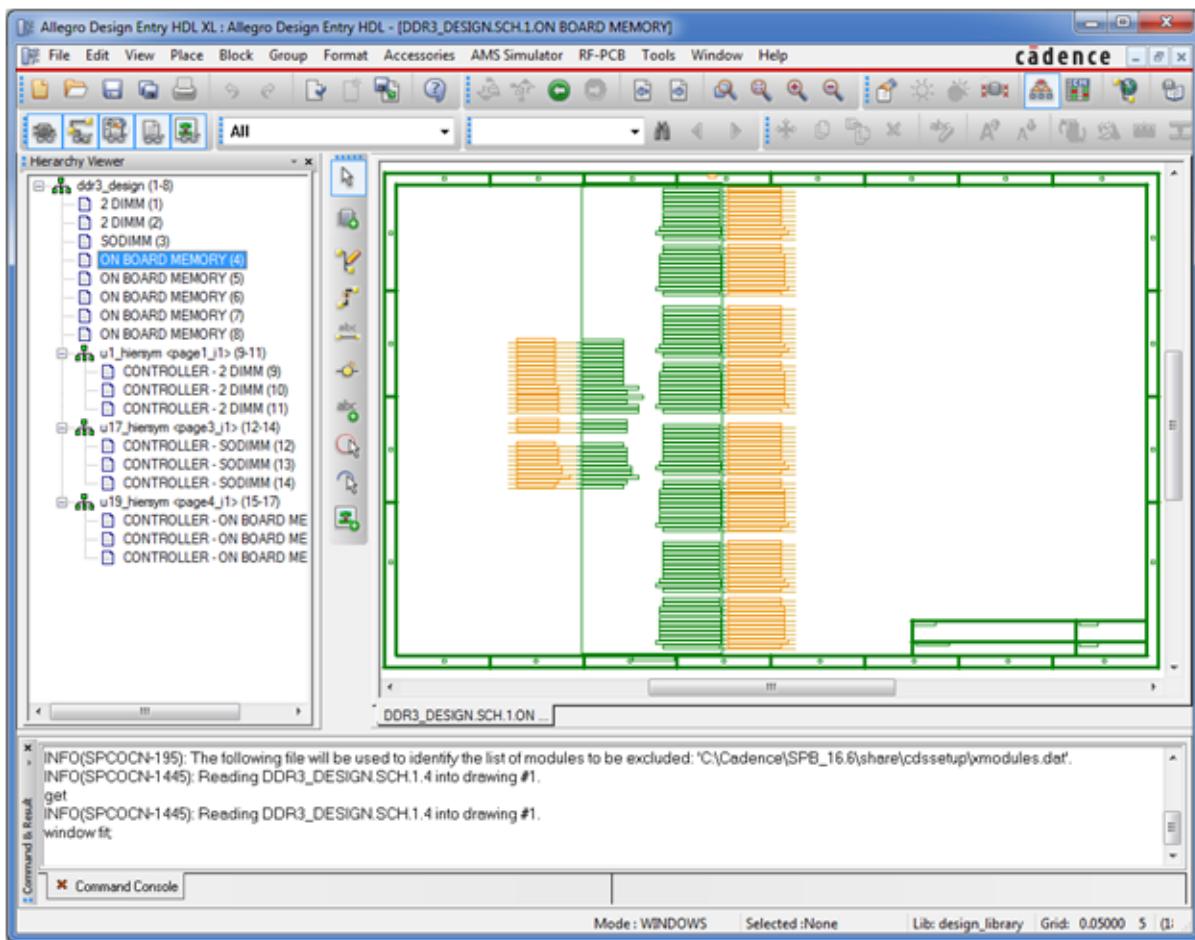
## Creating Split Symbols for Hierarchical Blocks

To effectively manage a block with multiple pins, you need to split the block into multiple smaller-sized symbols that can be placed within the page border. You first need to generate split hierarchical symbols for a hierarchical block. The ports of the block can then be logically categorized and split across different symbols. Splitting the blocks makes it easier to place them in the design.

To create a split symbol, do the following:

1. Launch Design Entry HDL.
2. Open the design consisting of blocks with multiple pins.

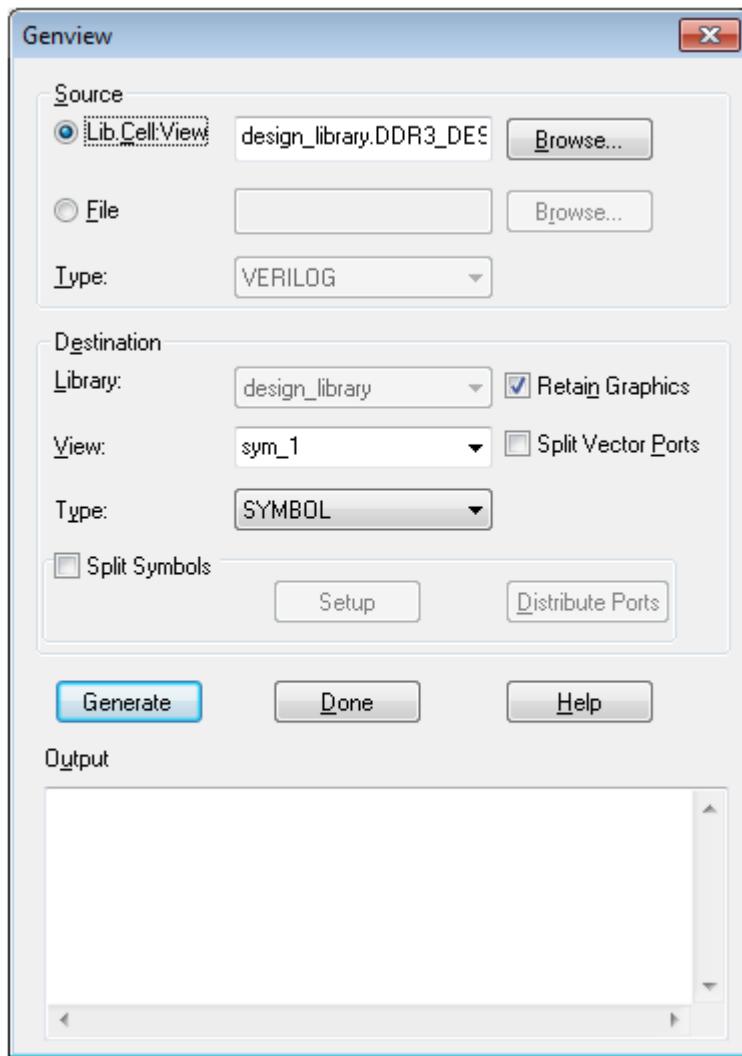
For example, in the following figure, the sheet ON BOARD MEMORY(4) contains a symbol with several pins. Note that the symbol does not completely fit on the B-sized sheet used.



## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

**3. Choose Tools – Generate View.**

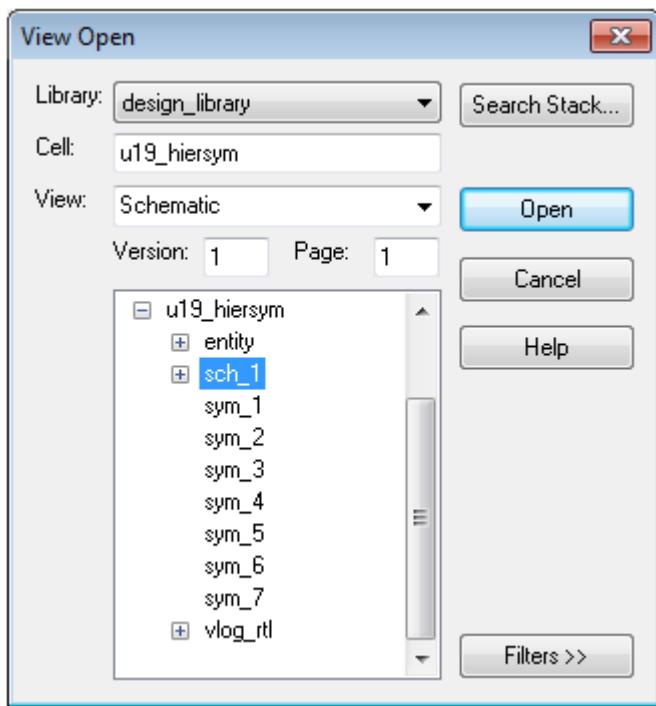


4. Select *Lib.Cell:View* as the Source option and click *Browse* to open the View Open dialog box.
5. Select *design\_library* as the Library.
6. Specify a cell name in the Cell field. For example, *u19\_hiersym*.
7. Select Schematic in the View field.
8. Specify 1 as the Version.

## Allegro Design Entry HDL User Guide

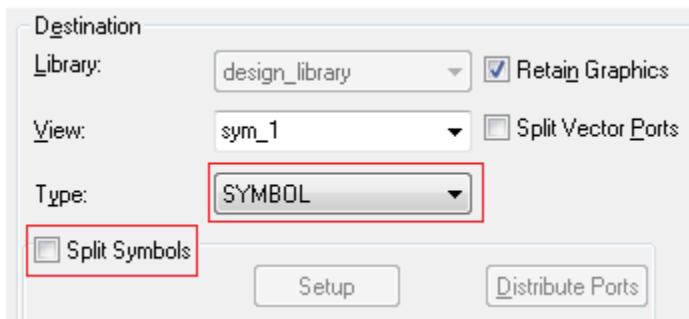
### Working with Hierarchical Split Symbols

You can alternatively select the page from the list.



**9. Click *Open*.**

In the Destination section, select Type as *SYMBOL*.



**10. To generate split symbols for this hierarchical block, select the *Split Symbols* check box.**

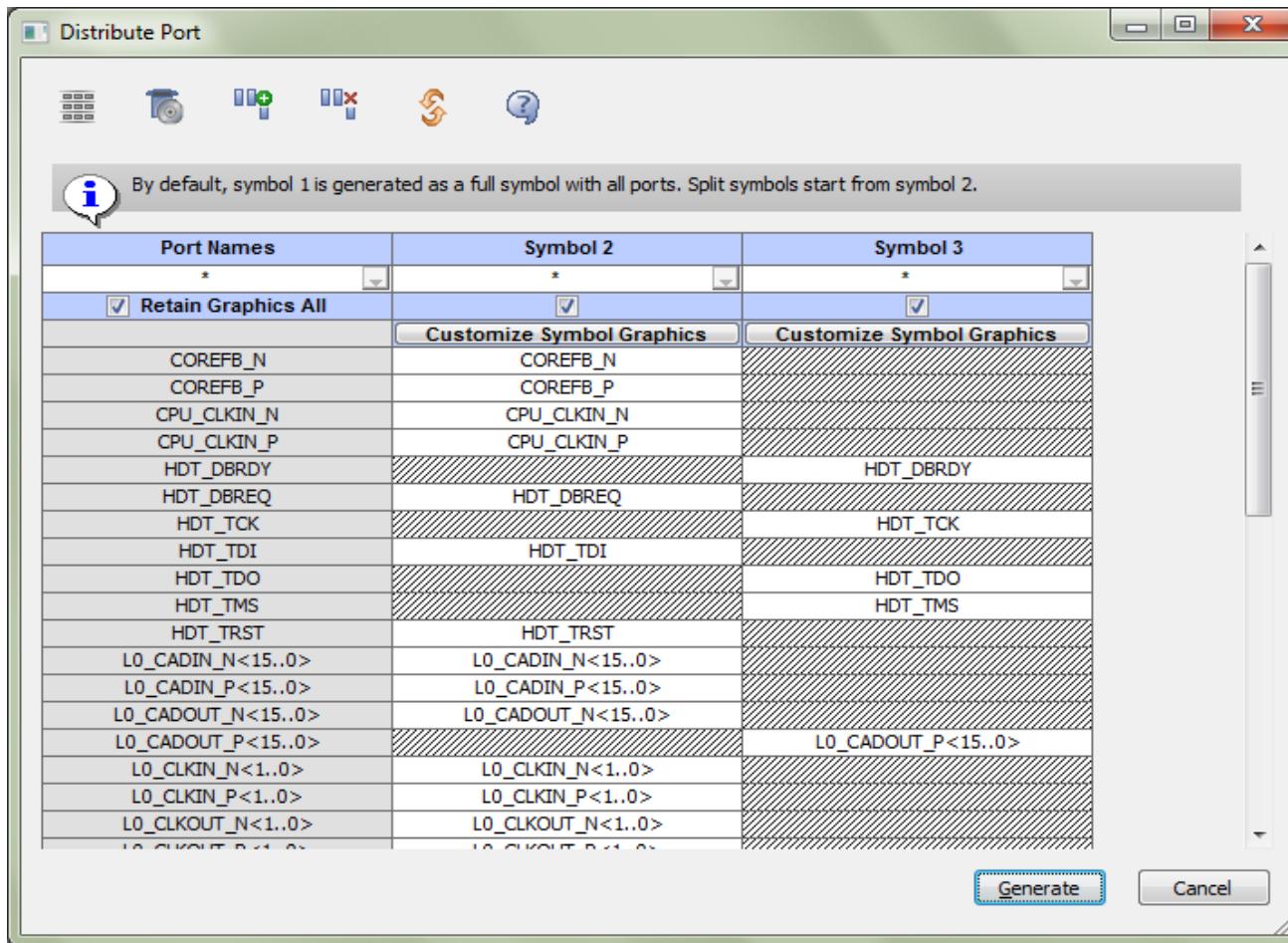
This enables the *Setup* and *Distribute Ports* buttons. As the next step, you need to distribute the ports of the hierarchical block across split symbols.

**11. Click the *Distribute Ports* button.**

## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

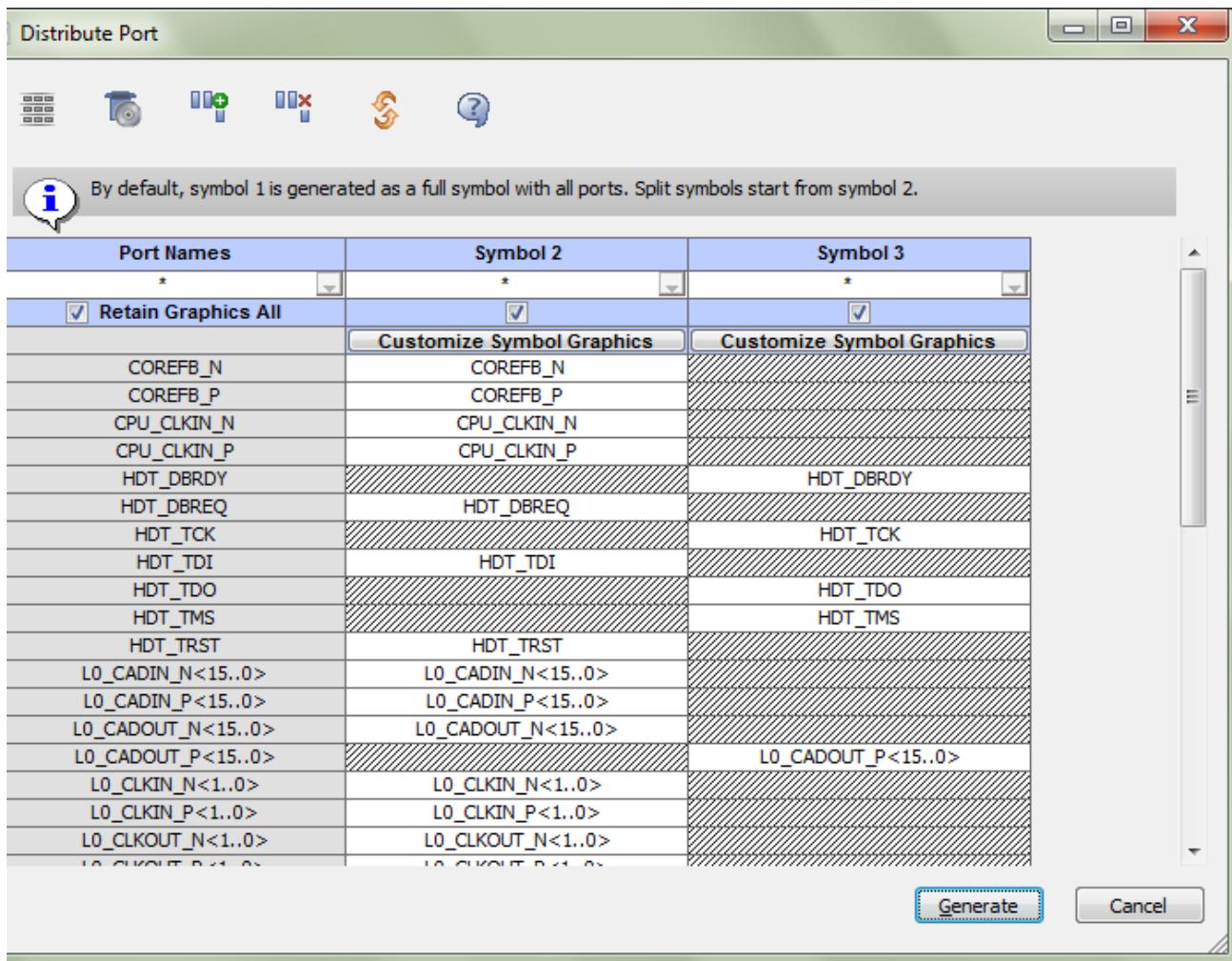
The Distribute Port dialog displays the current distribution of ports for the hierarchical block. Refer to the Distribute Port section of the Dialog Box Help chapter in *Allegro Design Entry HDL Reference Guide*.



## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

If split symbols exist for a block, the Port Name and the name of the split symbol are displayed in this dialog.



You can categorize the ports of the hierarchical blocks and use the categories to split the ports into various split symbols. For example, the ports can be divided into the following categories:

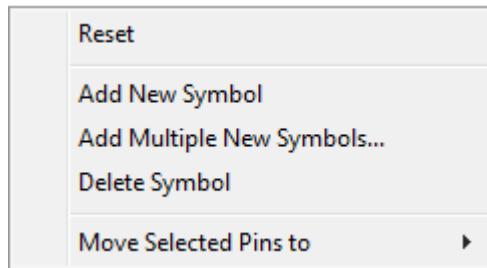
- Data signals
- Address bus
- Control signals

12. Select a set of ports of the same category and right-click.

## Allegro Design Entry HDL User Guide

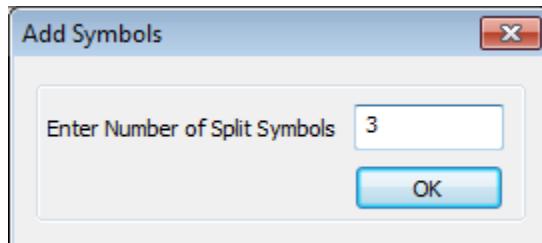
### Working with Hierarchical Split Symbols

13. Use the pop-up menu commands to add new split symbols for the hierarchical block. You can also move the selected ports to one of the split symbols using this menu. Additionally, you can change the location of ports.



14. Choose *Add New Symbol*.

A new column is added to the window representing the new symbol. If you choose the *Add Multiple New Symbols* command, you are prompted to specify the number of split symbols you want.



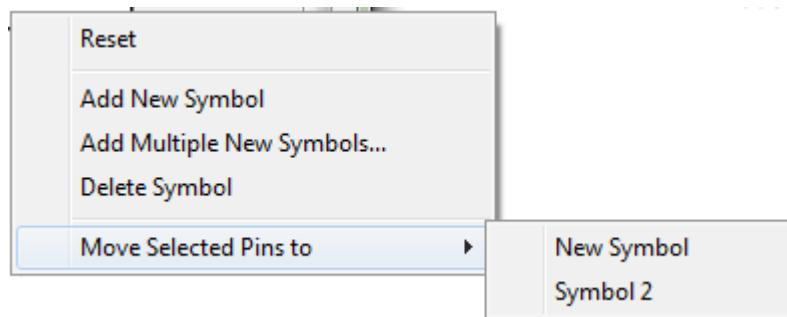
New columns are created corresponding to the number of split symbols you specify.

## Assigning Ports to Split Symbols

After creating the split symbols, you can assign pin ports to split symbols. You can move selected ports to a desired split symbol using two methods. You can either manually assign a port to a split symbol or use the auto distribution method.

To assign ports to a split symbol, do the following:

1. Select ports and right-click.
2. Choose the *Move Selected Pins to* menu command to move the selected ports to the desired split symbol.



Another method of distributing ports is using the auto distribute function available from the *Auto Distribute* (  ) tool bar icon.

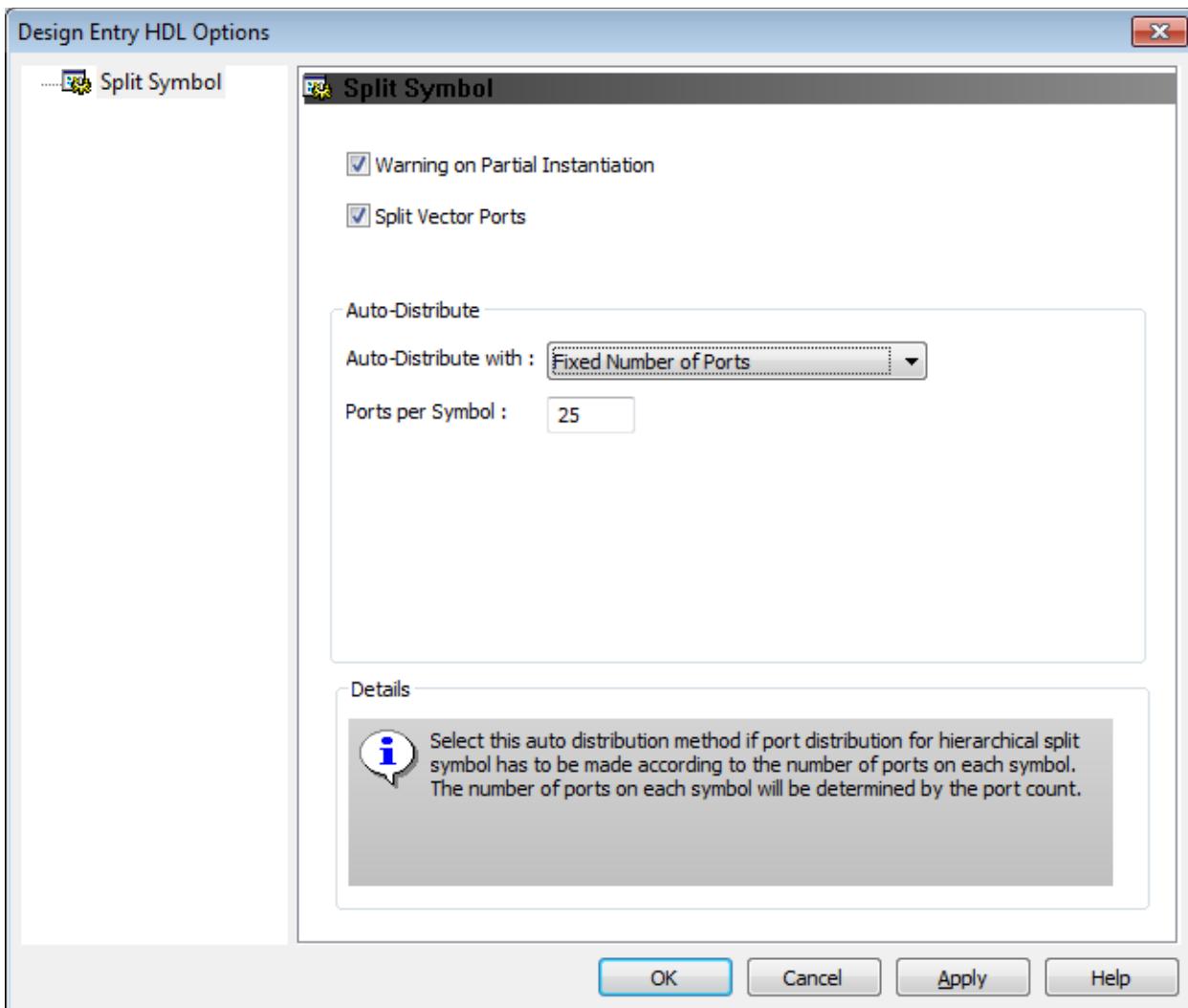
This function works on user-specified settings and distributes the ports across different split symbols. Before using the *Auto Distribute* command, you need to configure auto-distribution settings.

1. Click the *Settings* (  ) button on the tool bar.

## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

The *Split Symbol* page of the Design Entry HDL Options dialog is displayed. This is where you configure the auto-distribution settings. You can also access this dialog from the *Setup* button on the Genview dialog.



Note the *Warning on Partial Instantiation* option. If this option is selected, a warning message appears during save or check error operation when a hierarchical split symbol is partially instantiated, that is all the split symbols are not instantiated in the design.

**ERROR\_ON\_PARTIAL\_INSTANTIATION\_OF\_HSS** — While packaging a design, Packager-XL checks that all the interface nets of a block in a design are available as pins on the block symbol. When the block symbol is split, you may choose not to instantiate all the split symbols of the block in the design. This is partial instantiation. If you package a design where a split block is only partially instantiated, some of the pins of the interface nets might not be present in the design.

## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

---

By default, a design can be packaged without any problem even if the split block is partially instantiated. However, if you want to ensure that a design is packaged only when the split block symbols are completely instantiated, you can use the *ERROR ON PARTIAL INSTANTIATION OF HSS* directive.

**Auto Distribution** — You can decide on one of the following auto-distribution methods to place ports in the schematic:

- Fixed Number of Ports** - Use this method when split symbols have to be created based on the number of ports on each symbol. The number of ports on each symbol is determined by the port count you specify in the *Ports Per Symbol* field.
- Ports on Same Page** - Use this method to place all the ports on the same schematic sheet on the same symbol.

**Note:** If a net has Interface scope on one page and Local scope or is aliased on another page, the net with the Interface scope is honored. This port is used when creating split symbols using the auto-distribution method with the Ports on Same Page option.

- Pattern Based** - Use this method to place ports based on the port name pattern specified in the grid. Each row of the grid generates a separate hierarchical split symbol based on the pattern. Multiple patterns can be specified separated by a comma. For example, In\* or Pow\*.
- Property Based** - Use this method to place ports based on the property value specified in the Property Value field. Ports with the same property value are placed on the same symbol.

2. Based on your requirements, select the appropriate option from the *Auto-Distribute with* field.

For example, you can select the *Pattern Based* option and specify the following patterns:

\*w0\*, \*w1\*, \*w2\*, \*w3\*, \*a\*

3. Click *OK* to confirm.
4. Click the *Auto Distribute* button to distribute ports across split symbols.

The ports are distributed across split symbols based on the settings you specify in the Settings form.

The ports which do not follow any defined pattern are placed in the last split symbol. You can move ports from one symbol to another using the *Move Selected Pins to* command on the right-click pop-up menu.

## **Allegro Design Entry HDL User Guide**

### Working with Hierarchical Split Symbols

---

For better distribution of ports, you can change the location of pins from Left, Right, Top, and Bottom using the *Change Pin Location to* command. Changing the location of pins results in cleaner looking split symbols.

5. To delete an existing symbol, click the Delete Symbol icon.

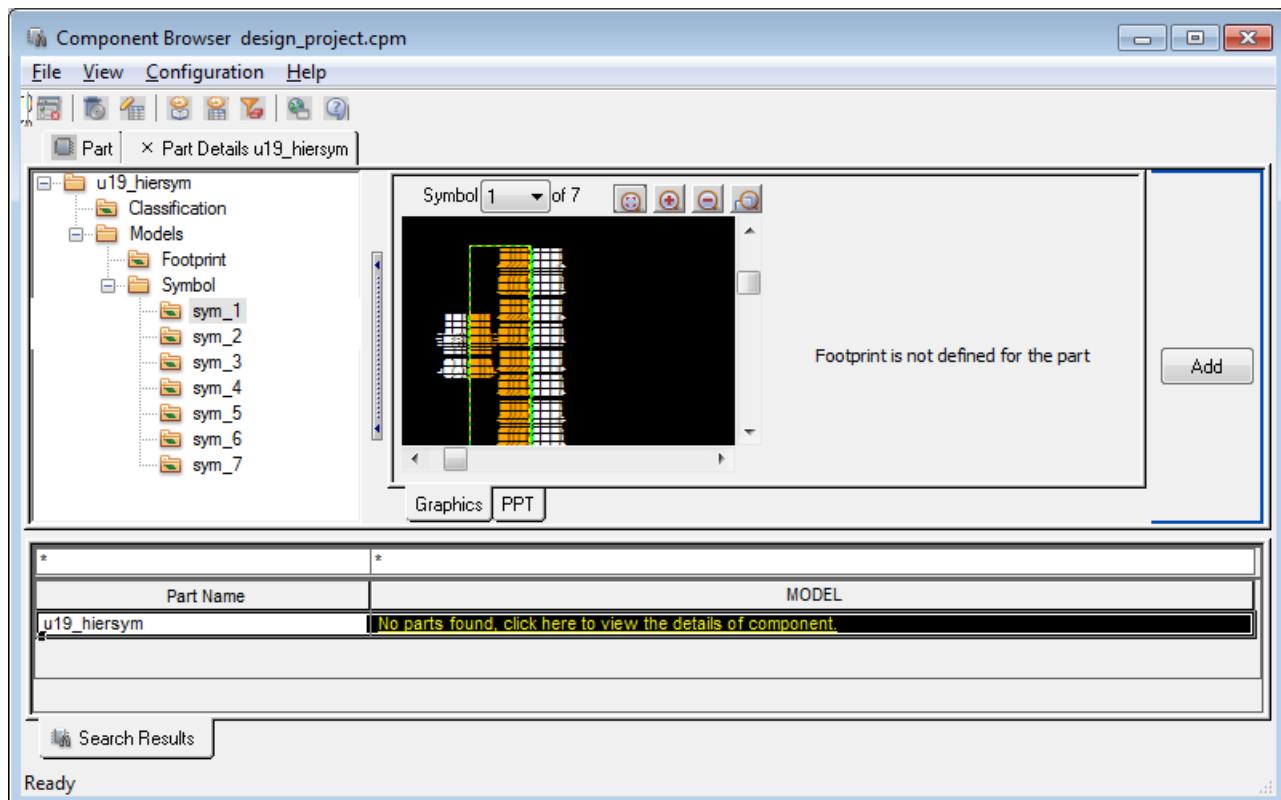
You are prompted to specify the split symbol number that you want to delete.

6. Click the *Generate* button to generate the split symbols for the hierarchical block.

When the symbol is generated, a message box appears that the completion of the operation.

## Using Hierarchical Split Symbols in a Design

You can instantiate HSS on a schematic canvas just as you would instantiate any other component using Part Information Manager.

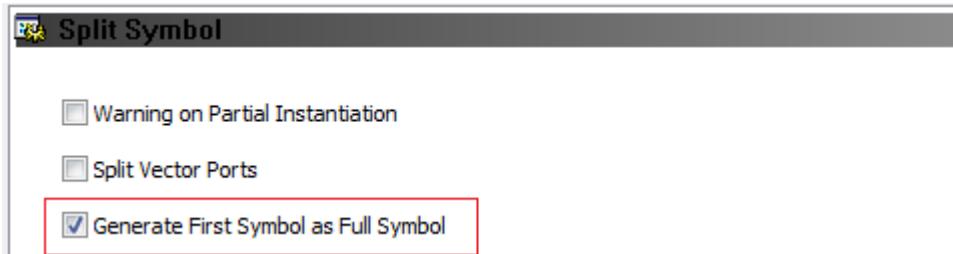


Note that the total number of symbols for the part is listed as 7, which is the number of symbols in which the symbol was split. Also note that in the figure, symbol 1 appears as the original symbol containing all the pins of the hierarchical block, as specified by the *Generate First Symbol as Full Symbol* option in the *Split Symbol* settings page of the Design Entry HDL Options dialog (*Tools – Options*).

## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

**Note:** The *Generate First Symbol as Full Symbol* option is only available in the Split Symbol settings page of the Design Entry HDL Options dialog available from the *Tools – Options* as this option is applicable at the design-level and not at the block-level.



This option can be set only the first time when the symbol is generated for a specific block, which does not already have a symbol view. If you have generated any symbol view for a specific block, for example generated sym\_1 as full symbol or split symbol, changing this option will not make any impact.

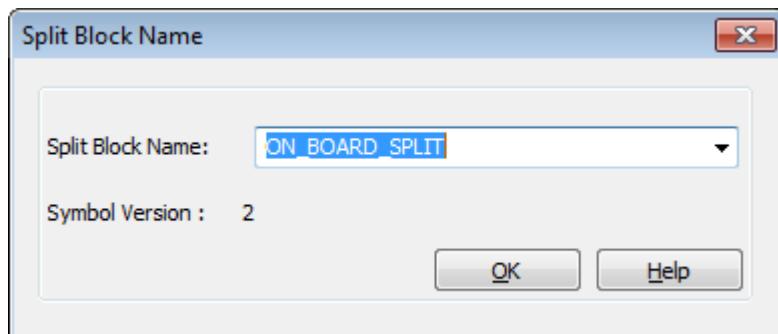
You can select the split symbol you want to instantiate from the drop-down list.



## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

When you place a split symbol on the canvas, a dialog box pops up.



As there are multiple split symbols for an HSS block, you need to provide a name to the split symbol instance at the time of placing it to bind all the split symbols together as a single block instance. The dialog box suggests a unique tool-generated name to uniquely identify the hierarchical block instance. You may want to change the name to any name of your choice. In case multiple instances of the same block are used in the design, a different name can be selected for the other instances of the block.

The split block name you specify in the Split Block Name dialog corresponds to the `SPLIT_BLOCK_NAME` property of the split symbols. While the symbol version number corresponds to the `SYM_NAME` property of split symbols.

**Note:** Changing the `SPLIT_BLOCK_NAME` property using the *Global Modify* feature is not recommended as it might result in data corruption.

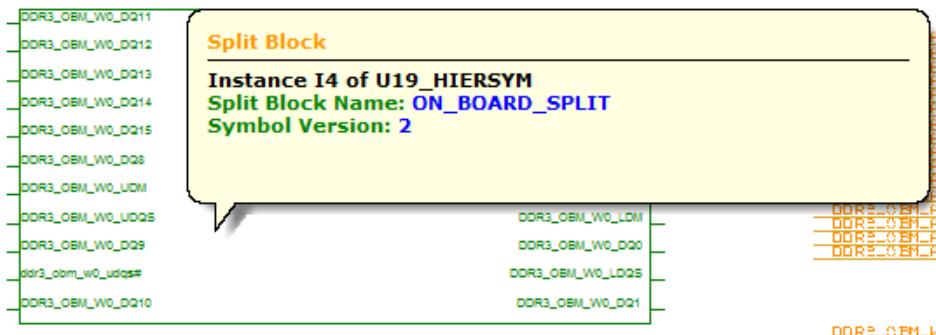
**Note:** The `SPLIT_BLOCK_NAME` and `SYM_NAME` properties are not searchable when you perform a quick search of only components, nets, or properties, or a combination of any of the three. These two properties are searched for only when all the filter options in the Find dialog are selected. This is because these properties are not written to the `.dcf` file.

**Note:** The `SPLIT_BLOCK_NAME` and `SYM_NAME` properties are stored in the `template.tsg` file located at `<your_install_dir>/share/cdssetup/concept/genview`. When working with hierarchical blocks and split symbols, ensure that the `SPLIT_BLOCK_NAME` and `SYM_NAME` properties are added to the `defProp` sub-section of the `template.tsg` file at the `$CDS_SITE` level.

## Allegro Design Entry HDL User Guide

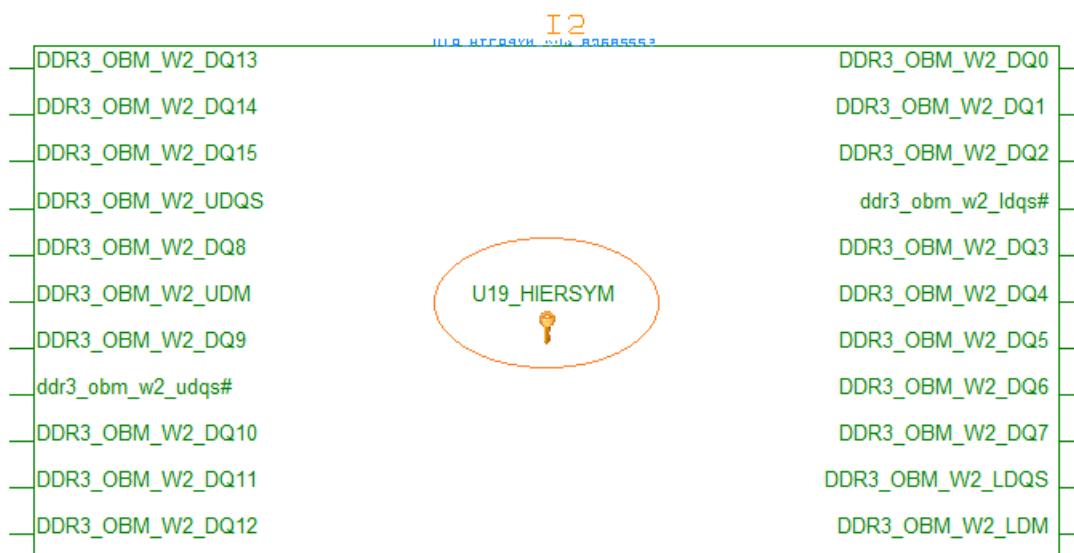
### Working with Hierarchical Split Symbols

The tool tip on the split symbol identifies the instance as a split block and provides the split block name and version.



### Pivotal Symbol

When you save the schematic, the first-added instance of the symbol appears with an icon visible at the center of the symbol. This graphic identifies this symbol as the primary instance. The instance identifier for this instance is used for identifying the hierarchical block.



When you perform operations such as move, cut, delete, replace, or version on the primary instance (also known as the *pivotal symbol*), the constraints and packaging data is lost. However, the same is not true for the other split symbols where the constraint and packaging data remains intact.

## Allegro Design Entry HDL User Guide

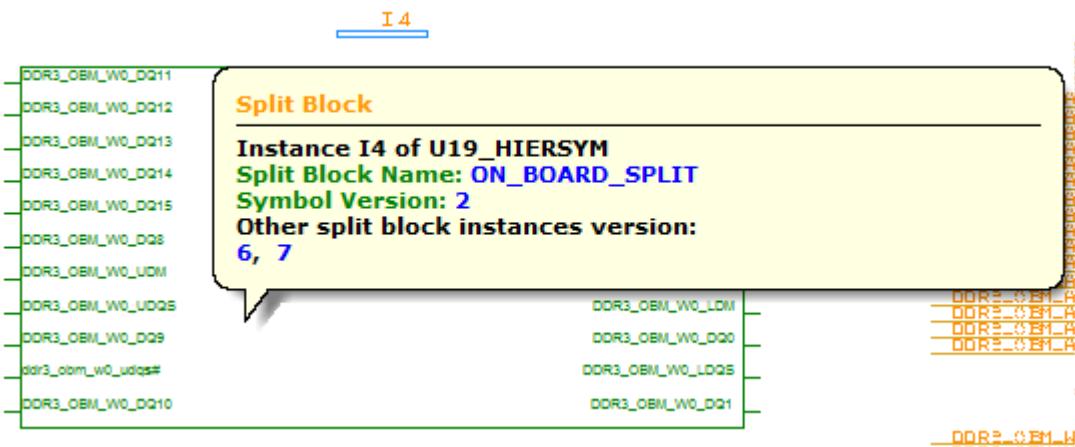
### Working with Hierarchical Split Symbols

**Note:** If the connectivity (xcon) or Constraint Manager database (dcf) files are removed, the information about the pivotal symbol is also lost. In such cases, a new pivot is generated by the tool.

Block properties, such as REF\_DES\_PATTERN, ROOM, and GROUP or user-defined properties can only be applied on a pivotal symbol, that is these properties can only be added or deleted from a pivotal symbol. These properties appear grayed on non-pivotal symbols.

### Identifying Split Block Instances

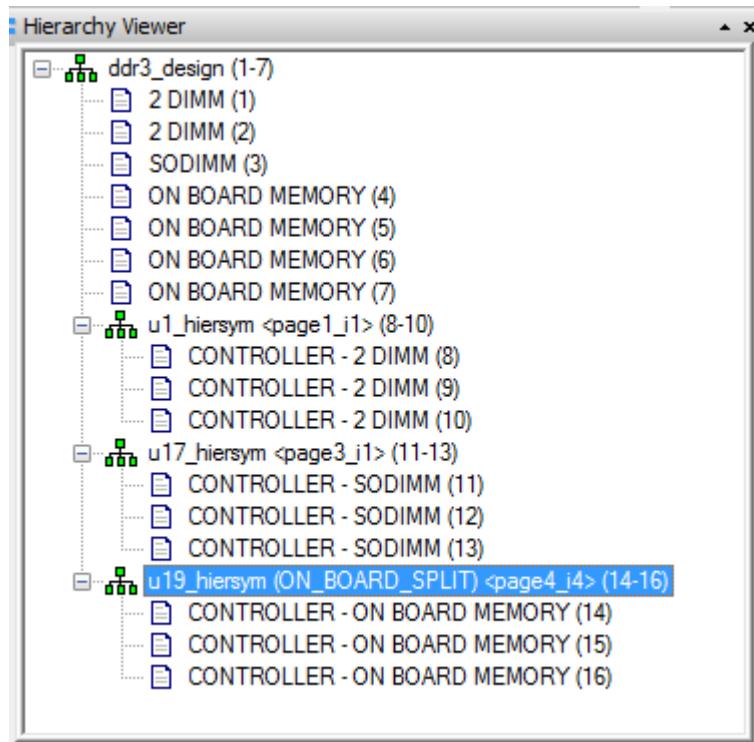
For all the split symbols which belong to the same split block name, the tool tip displays other split block instance versions instantiated in the schematic. The symbol version numbers are also hyperlinked and clicking on a version number navigates to the place where the specific split block symbol is instantiated.



## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

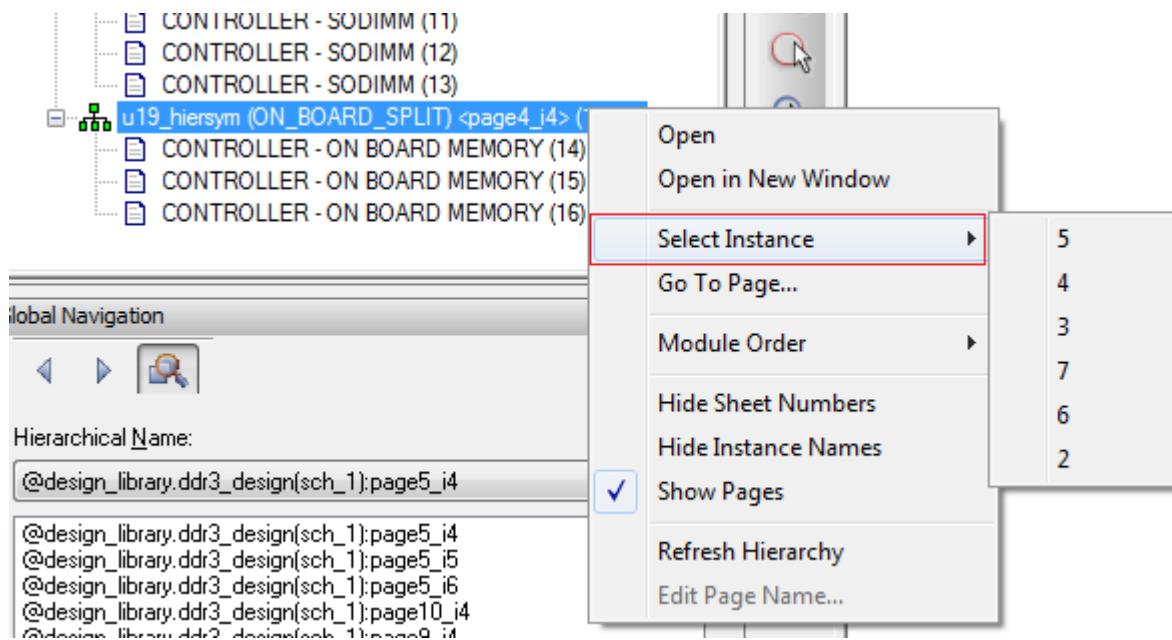
In the Hierarchy Viewer, the split block name is available to identify this instance as a split block instance:



## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

In the Hierarchy Viewer, you can select the split block name and from the right-click pop-up menu navigate to a specific instance of a split symbol.



**Note:** Port groups cannot be created on split symbols.

## Editing Split Symbols

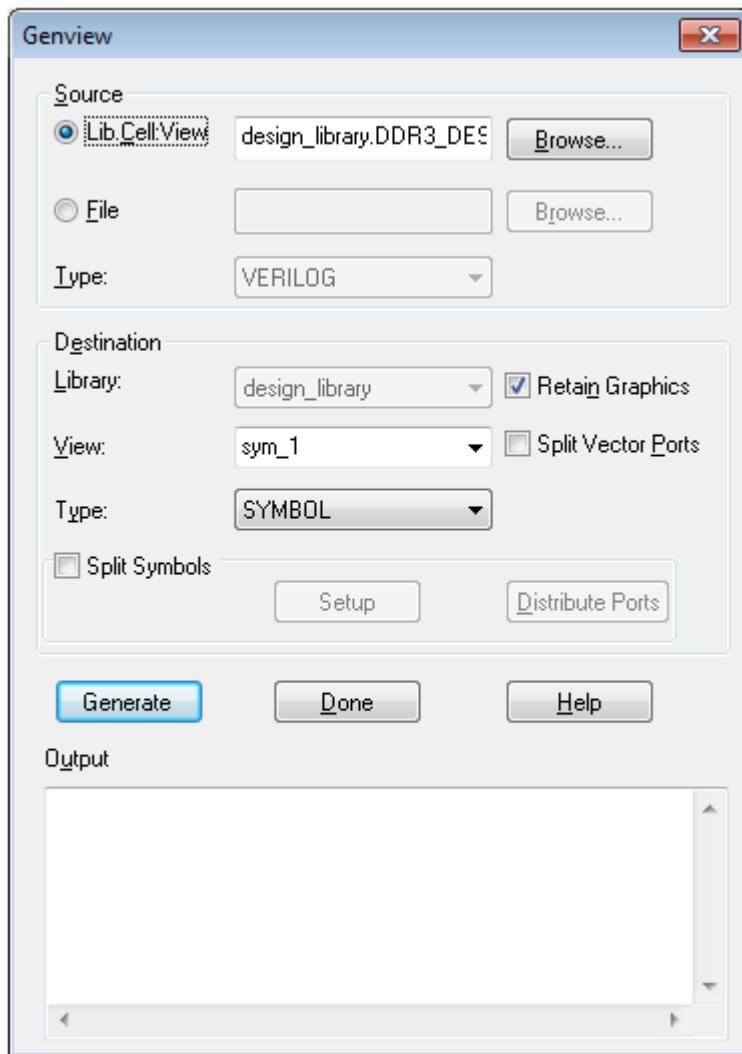
To edit a split symbol, do the following:

1. Launch Design Entry HDL.
2. Open the design or block whose symbols you want to edit.

## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

3. Choose *Tools – Generate View.*



4. Select the *Split Symbols* check box.

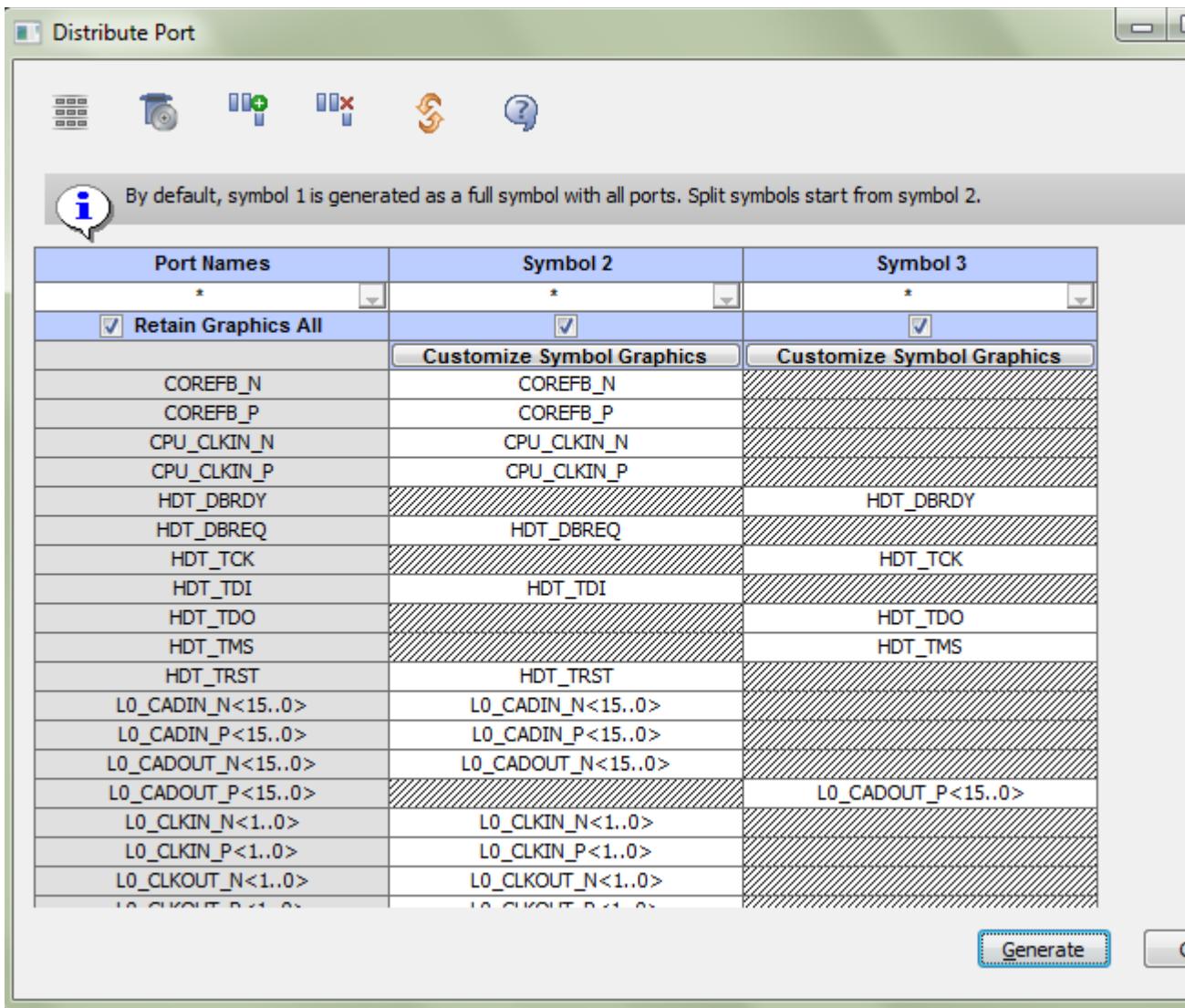
This enables the *Setup* and *Distribute Ports* buttons.

5. Click the *Distribute Ports* button.

## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols

The Distribute Port dialog displays the current distribution of ports for the hierarchical block. Refer to the Distribute Port section of the Dialog Box Help chapter in *Allegro Design Entry HDL Reference Guide*.



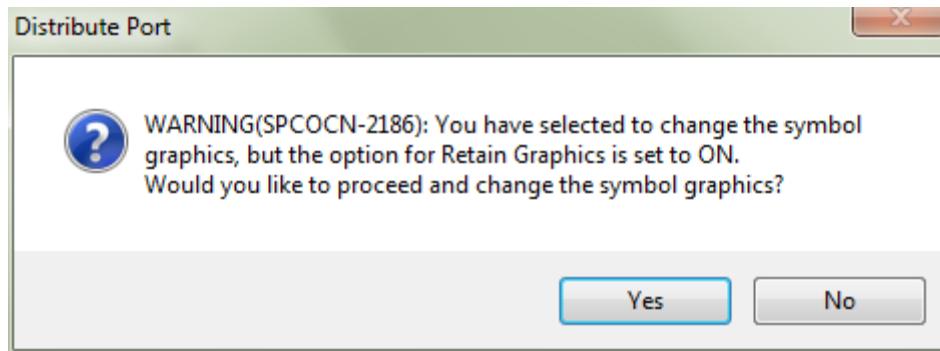
The *Retain Graphics All* check box is selected by default and indicates that the placement of pins that already exist on the graphics for the symbols should be retained when regenerating the symbols.

6. Select the symbol you want to edit and click the *Customize Symbol Graphics* button.

A warning message prompts you to proceed.

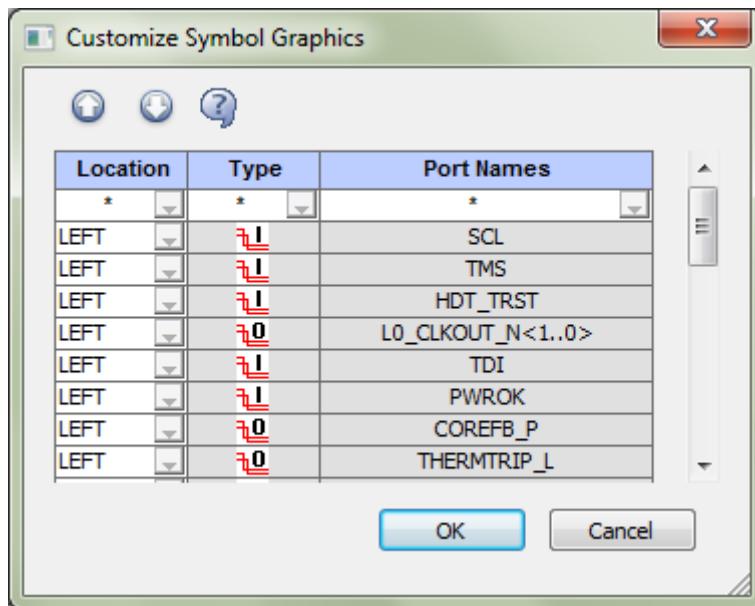
## Allegro Design Entry HDL User Guide

### Working with Hierarchical Split Symbols



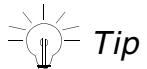
Click Yes to proceed.

The Retain Graphics All box for the selected symbol is automatically unchecked and the Customize Symbol Graphics dialog box opens.

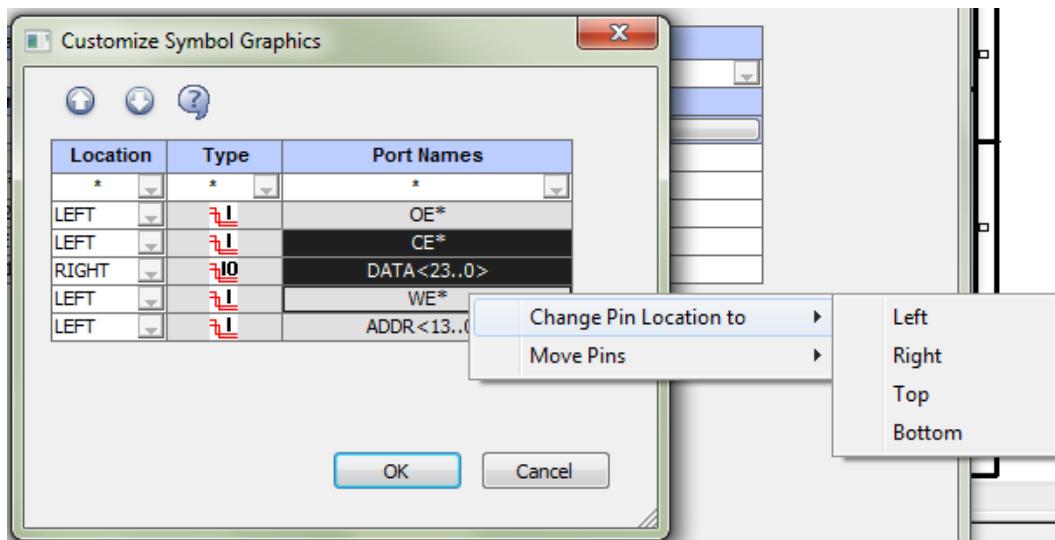


#### 7. Edit the symbol as required.

- Modify the location of the port on the split symbol in the Location column. The options are TOP, BOTTOM, LEFT, and RIGHT.
- Change the order of the port names on the split symbol by selecting the name and using the Move Row Up or Move Row Down toolbar buttons.



You can modify the location or the order of the port names on the split symbol for multiple ports at a time. To select a consecutive group of port names, click the first item, press and hold down the Shift key, and then click the last item. To select non-consecutive port names, press and hold down the Ctrl key, and then click each item that you want to select. Right-click on the selected items, choose *Change Pin Location to* or *Move Pins* and choose the required option.



You can also filter the results to be displayed using the column headers. For example, to view all left-aligned ports for the selected split symbol, select *LEFT* in the *Location* column header.

8. Click *OK* to close the Customize Symbol Graphics dialog box.
9. In the Distribute Port dialog box, click the *Generate* button to regenerate the split symbols.

**Note:** You can edit split symbols in two ways—by using the Customize Graphic Symbol dialog box or by opening the split symbol from the *File – Open* menu in the schematic canvas.



If you edit a split symbol in the Customize Graphic Symbol dialog box and later save the same symbol in the View Open dialog box (*File – Open*), changes made in the View Open dialog box may not reflect in the Customize Graphic Symbol dialog box.

For information on editing symbols, see “[Editing Symbols](#)” on page 624.

## **Allegro Design Entry HDL User Guide**

### Working with Hierarchical Split Symbols

---

# **Allegro Design Entry HDL User Guide**

## Working with Hierarchical Split Symbols

---

---

# **Working with Designs**

---

This section describes the following tasks:

- [Finding Nets and Cells in Your Design](#) on page 423
- [Finding Nets and Cells in Your Design](#) on page 423
- [Navigating Nets in Your Design](#) on page 425
- [Global Modification](#) on page 426
- [Part Manager](#) on page 447
- [Running Scripts](#) on page 460
- [Highlighting \(Cross-Probing\) Objects](#) on page 464
- [Applying Connectivity Changes from the Physical Design to Your Schematic](#)
- [Back Annotating Your Design](#)
- [Module Ordering](#) on page 466
- [Performing Page Management Operations](#) on page 476
- [Displaying and Working with Schematic Page Numbers](#) on page 487
- [Importing Designs](#) on page 497
- [Baselining a Design](#) on page 506
- [Creating the Table of Contents for a Design](#) on page 511

## **Finding Nets and Cells in Your Design**

To find nets and cells, do the following:

1. Click the *Search options* icon on the Search toolbar

The *Find* dialog box appears.

## Allegro Design Entry HDL User Guide

### Working with Designs

---

2. In the *Find What* field, do one of the following:

- Type the name of the net or the cell to be located. For example, typing ls04 locates all instances of the part ls04.

**Note:** To find a vectored signal, DATA[3..0] or DATA(3..0), type DATA or DATA<3..0> in the *Name* box.

- Select a previously-entered name from the list.

3. Select the object type to be located, *Net* or *Cell*.

4. To optionally restrict the search by property, do both of the following in the *With Property* section of the dialog box:

- Select property name in the *Filter Options* field.
- Type a property value in the *Find What* field or type an asterisk character (\*) to locate all objects with the specified property name and any property value.

When you enter the property name and the value, they are added to their respective list boxes so that they can be reused during the same design session.

5. Click *Find All* to begin the search.

A message in the *SearchResults* dockable window lists the number of instances that were found. The (unlabeled) status area box displays the instances of the object found.

6. Click on a search result to view it in your design.

When you select a result to view, the page that contains the object appears with the object highlighted. If you chose *Navigate*, the Global Navigation window appears so that you can move across the design to view all the net instances listed in the search results box.

7. Click on another search result to view in the design, search for a different net or cell by entering a new net or cell name in the *Name* box, or click *Close*.

### Objects Not Found by Find

- Objects named:

PIN NAMES DRAWING DEFINE  
MENU

- Bus names:

- having step size

For example, A<31..0:2> is not found by Global Find

- with names ending in \*

For example, A<2..0>\* is not found by Global Find

**Note:** HDL\_POWER symbols and other symbols such as GND, or the SIG\_NAME property are not written to the Xcon file. Therefore, a quick search of only components, nets, properties, or a combination of the three cannot find them.

## Navigating Nets in Your Design

1. Choose *Tools – Global Navigate*.

The *Global Navigation* window appears.

2. Select a net in your design.

The hierarchical name for the net appears in the *Hierarchical Names* box, and the message box indicates how many nets were located. The status area box lists all the net instances located. You can also select how the search results are to be viewed by right-clicking the status area and selecting *Hierarchical Names* or *Library Locations* from the pop-up menu.

3. Select *Zoom to Object* to zoom in on a search result, otherwise, go to step 6.

4. To view a search result, select a net instance in the status area box.

When you select a search result, the page containing the object appears with the object highlighted.

5. To navigate to a different net, select another net in your design.

You can use the **Back** and **Next** buttons to move between different nets that you have selected.

To clear the search results and search history, right-click the *Global Navigation* window and choose *Clear Results* or *Clear History* as required. For more information about the *Global Navigation* window, refer to the [Global Navigation](#) section of Allegro Design Entry HDL Reference Guide.

## Global Modification

The Global Modify feature in Design Entry HDL helps you delete or modify any net, pin, or component property from the whole design. It also helps you replace a component with a new component across a design. You can either choose the new component from the physical part filter or pick a replacement from the design. Component change is also supported in the logical mode. .



The Global Modify feature contains Java 1.3.1 code that uses TrueType fonts. For the Global Modification window to appear correctly, you must install the *SUNWi1of* package.

To open the Global Modification window, do the following:

1. In Design Entry HDL, choose *Tools – Global Update*.
2. In the *Global Update* submenu, select any option.

Each tabbed page in the Global Modification window contains fields that you can use to modify properties.

Property Change      Use this tabbed page to change the properties of components, pins, and nets across a design.

Property Delete      Use this tabbed page to delete the properties of components, pins, and nets across a design.

Component Change      Use this tabbed page to replace a component across a design with a new component.

Global modification can also be carried out in batch mode. To know more, see the [The Batch Mode Operation](#) section below.

## Modifying Component, Pin, and Net Properties

To change a net, pin, or component property, do the following:

1. With the schematic page open in Design Entry HDL, choose *Tools – Global Update*.
2. In the *Global Update* submenu, select *Global Property Change*.

The Global Modification window appears with the *Global Property Change* tabbed

page open.

3. In the *Change (Name)* field, do one of the following:
  - a. Type the property name.
  - b. Select a previously-entered name from the drop-down list. For example, if you want to change the name of the ASSIGN\_TOPOLOGY property, select ASSIGN\_TOPOLOGY from the drop-down list.
4. In the *To (Name)* field, do one of the following:
  - a. Type the new property name. For example, if you want to change the name of the ASSIGN\_TOPOLOGY property to ALLOT\_TOPOLOGY, type ALLOT\_TOPOLOGY in this field.
  - b. Select a previously-entered name from the drop-down list.
5. Select the Design Entry HDL object type. For example, if you want to change the ASSIGN\_TOPOLOGY property on a net, select the *Nets* check box.

Wildcards are supported for the original property name and value. The asterisk character (\*) is always handled as a wildcard in the original property name. The *Enable Wildcard* check box controls whether the asterisk character (\*) in the original property value is handled as a wildcard or as a literal. To know more about the use of wildcards in the *Property Change* tab, refer to *Dialog Box Help*.

**Note:** If you are changing a property on a pin or component, select the appropriate check box.

6. Select the scope of modification:
  - If you want to apply the modifications to all pages in all modules of the design, select the *Design* radio button.
  - If you want to apply the modifications only to the current page, select the *Current Page* radio button.
  - If you want to specify a comma-separated list of pages and page ranges, for example 1, 3, 5, 7–12, select the *Page Range* radio button.
  - If you want to process the current module instead of the current page or hierarchy, select the *Current Module* radio button.

## Allegro Design Entry HDL User Guide

### Working with Designs

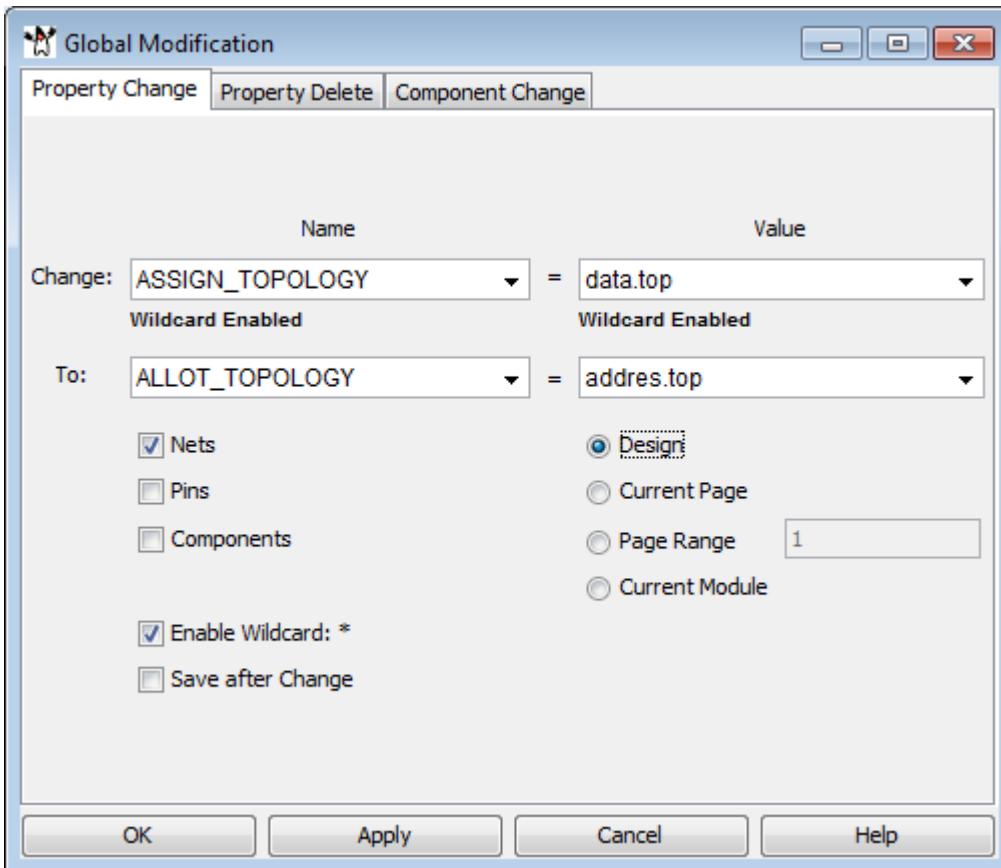
7. Apart from changing the property name, if you want to change the value of the property on the schematic, specify the value in the *Change (Value)* field. For example, if you want to change the value of the property from `data.top` to `address.top`, type `data.top` in this field.

8. In the *To (Value)* field, specify the new property value. In this case, type `address.top`.

The `++Preserve Source Value++` option in the *To (Value)* drop-down list is used to retain the value of the source property. This is because the asterisk character (\*) is always treated as a literal character in the new property value field.

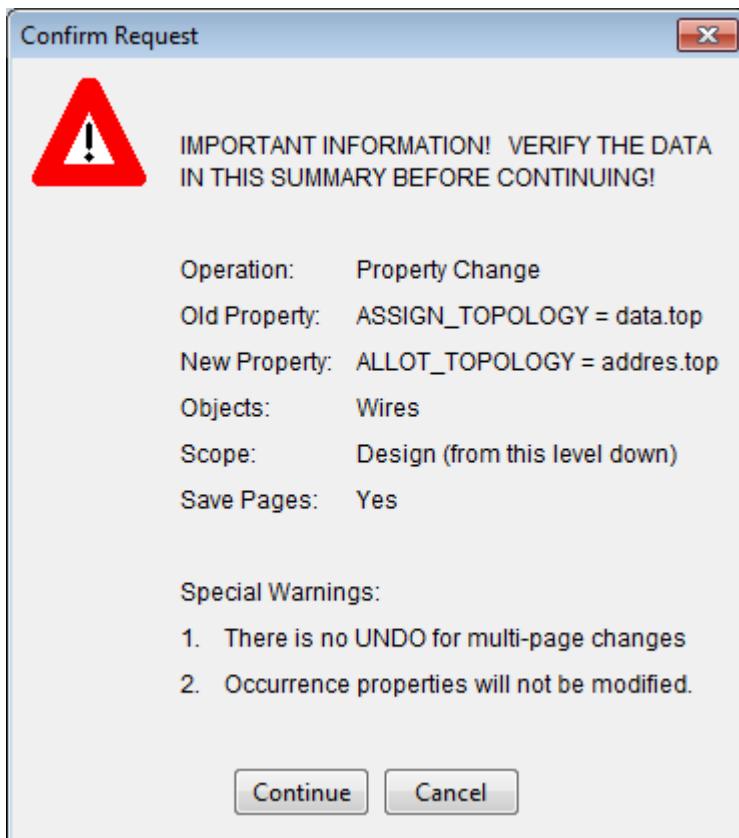
9. To specify that the schematic sheet be saved after it is modified, select the *Save after Change* check box.

After specifying the changes to be made, the Property Change page should appear as shown below:



10. To save all the changes you made without closing the Global Modification window, click the *Apply* button.

The *Confirm Request* message box appears.



#### 11. In the *Confirm Request* message box, click *Continue*.

The *Results* box appears with the current status.

After all the changes are made based on the defined scope, the *Results* box displays a *Successful Completion* status.

The *Results* box also displays other important information about the modifications you made, such as the number of pages processed, the number of read-only pages, the number of properties changed, and the number of default body properties. If you want to view the log file, select the *View Results Log File* check box and click *OK*.

The log file contains the changes made to the schematic and is stored in the `temp` folder of the project.

## Log Files Support

The Global Modify feature maintains three backup log files, `gc.log.1`, `gc.log.2`, and `gc.log.3`, where 1 is the most recent backup and 3 the oldest. The most recent data is stored in a log file called `gc.log`.

## Deleting Component, Pin, and Net Properties

To delete a net, pin, or component property, do the following:

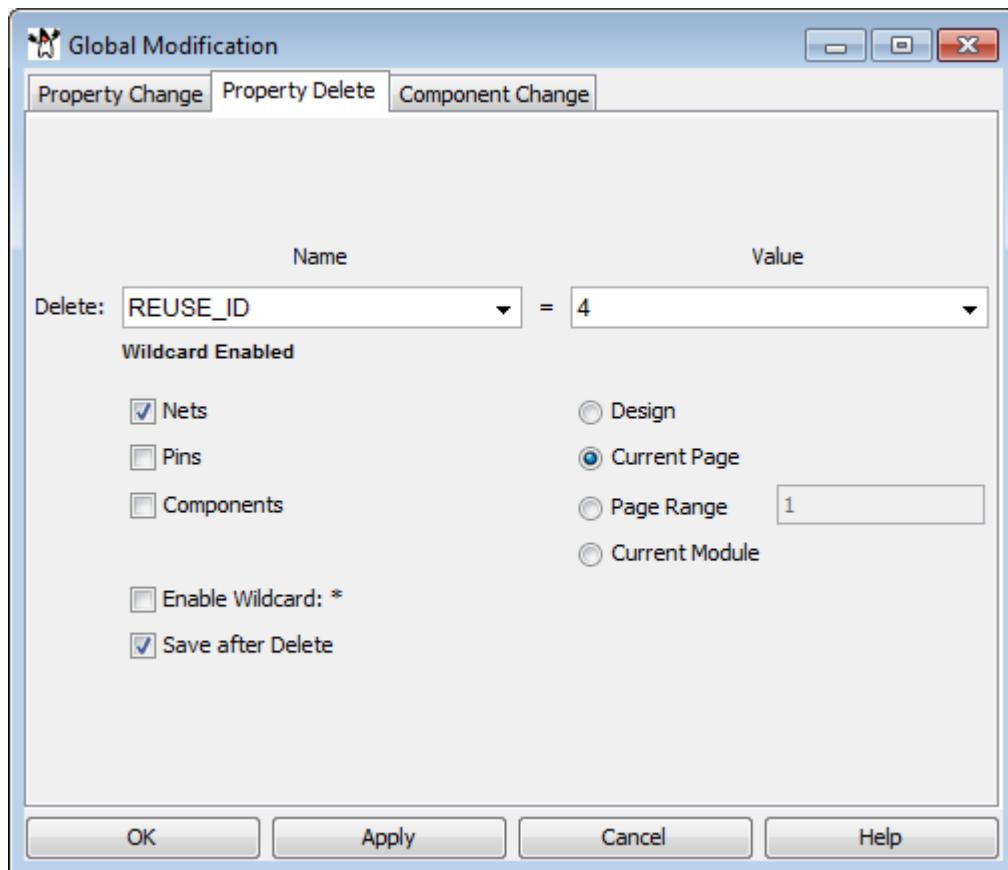
1. Click the *Property Delete* tab.
2. In the *Delete (Name)* field, do either one of the following:
  - Type the property name.
  - Select a previously entered name from the drop-down list. For example, if you want to delete the `REUSE_ID` property, select `REUSE_ID` from the drop-down list.
3. In the *Delete (Value)* field, do one of the following:
  - Specify the value of the property on the schematic to be deleted, for example 4.
  - Select a previously-entered value from the drop-down list.
- Wildcards are supported when deleting property names and values. The asterisk character (\*) is always handled as a wildcard in the original property name. To know more about the usage of wildcards in the *Property Delete* tab, refer to *Dialog Box Help*.
4. Select the Design Entry HDL object type. For example, if you want to delete the `REUSE_ID` property on a net, select the *Nets* check box.  
**Note:** If you are changing a property on a pin or component, select the appropriate check box.
5. Select the scope of modification:
  - If you want to apply the modifications to all pages in all modules of the design, select the *Design* radio button.
  - If you want to apply the modifications only to the current page, select the *Current Page* radio button.
  - If you want to specify a comma-separated list of pages and page ranges, for example 1, 3, 5, 7–12, select the *Page Range* radio button.

## Allegro Design Entry HDL User Guide

### Working with Designs

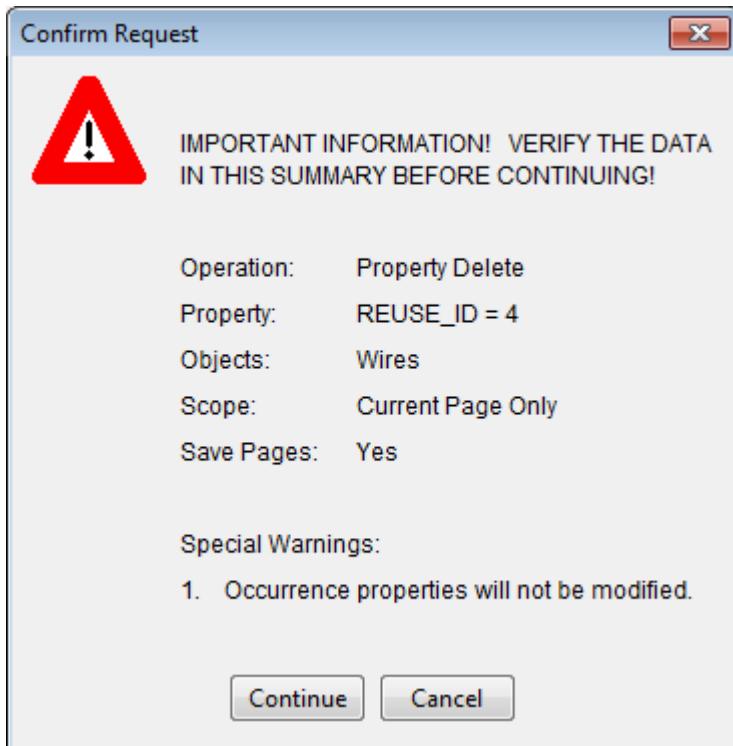
- If you want to process the current module instead of the current page or hierarchy, select the *Current Module* radio button.
6. To specify that the schematic sheet be saved after changes are made, select the *Save after Change* check box.

After specifying the changes to be made, the Property Delete page should resemble the following:



7. To start the processing and close the Global Modification window, click the *OK* button.

The *Confirm Request* message box appears.



8. In the *Confirm Request* message box, click *Continue*.

The *Results* dialog box appears with the current and modified status.

### Selecting the Component to Replace from the Physical Part Filter

The *Component Change* tab can be used to perform functions such as global delete, global modify, global replace, and global refresh. While globally replacing components, you can choose the new component from the physical part filter or pick a replacement from the design.

**Note:** These *Component Change* functions are also supported in the logical mode.

There are two ways in which you can select the component you want to replace and the new component:

- Select the component from the physical part filter
- Select the component from the design (Schematic Pick)

## Allegro Design Entry HDL User Guide

### Working with Designs

---

There is no wildcard support in the *Component Change* tab. An asterisk (\*) is allowed in the original property value and new property value fields, but it is a character literal.

1. Click the *Change* button in the *Original Component* group.

The *Select Component to Change* dialog box appears.

2. To select the component from the physical part filter, click the *Browse* radio button.
3. Select the library the component belongs to from the *Library* drop-down list, for example, `parts_lib`.
4. Select the component from the *Component* drop-down list, for example, `cap`.
5. Select the version of the component from the *Version* drop-down list, for example, `1`.

Select the *Select Physical Component* check box to ensure that the changes are made to the physical component.

Be careful in using wildcards for versions. You might inadvertently make more changes than you want to.

6. Click the *OK* button.

The Physical Part Filter window appears.

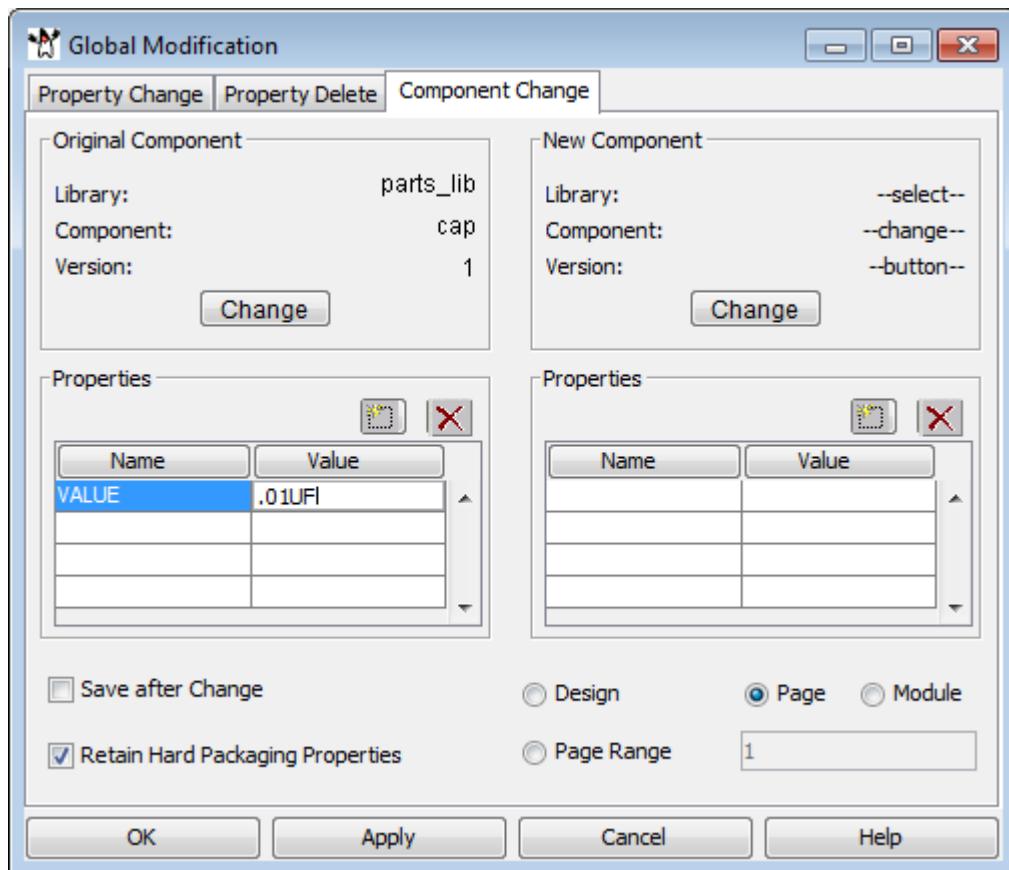
7. In the Physical Part Filter window, select the part and click *OK*.

The Global Modification window appears with the properties of the selected component

## Allegro Design Entry HDL User Guide

### Working with Designs

listed in the *Properties* list below the *Original Component* group.



**Note:** When you select a component, the *Properties* list displays the properties of the component. Each property in the *Properties* list is represented by a Name-Value pair, for example **VALUE** and **.01UF**.

You can also add a new property to the list and delete a selected property from the list using the *New* and *Delete* icons in the *Properties* list. There is no wildcard support in the property list of the *Component Change* tab. An asterisk (\*) is allowed in the Original Property Value and New Property Value fields, but it is considered a character literal.

### Selecting the New Component from the Physical Part Filter

To select the new component from the physical part filter, do the following:

1. Click the *Change* button in the *New Component* group.

The *Select New Component* dialog box appears.

## Allegro Design Entry HDL User Guide

### Working with Designs

2. Select the library that the new component belongs to from the *Library* drop-down list, for example `parts_lib`.

3. Select the new component from the *Component* drop-down list, for example `pres`.

4. Select the version of the new component from the *Version* drop-down list, for example `1`.

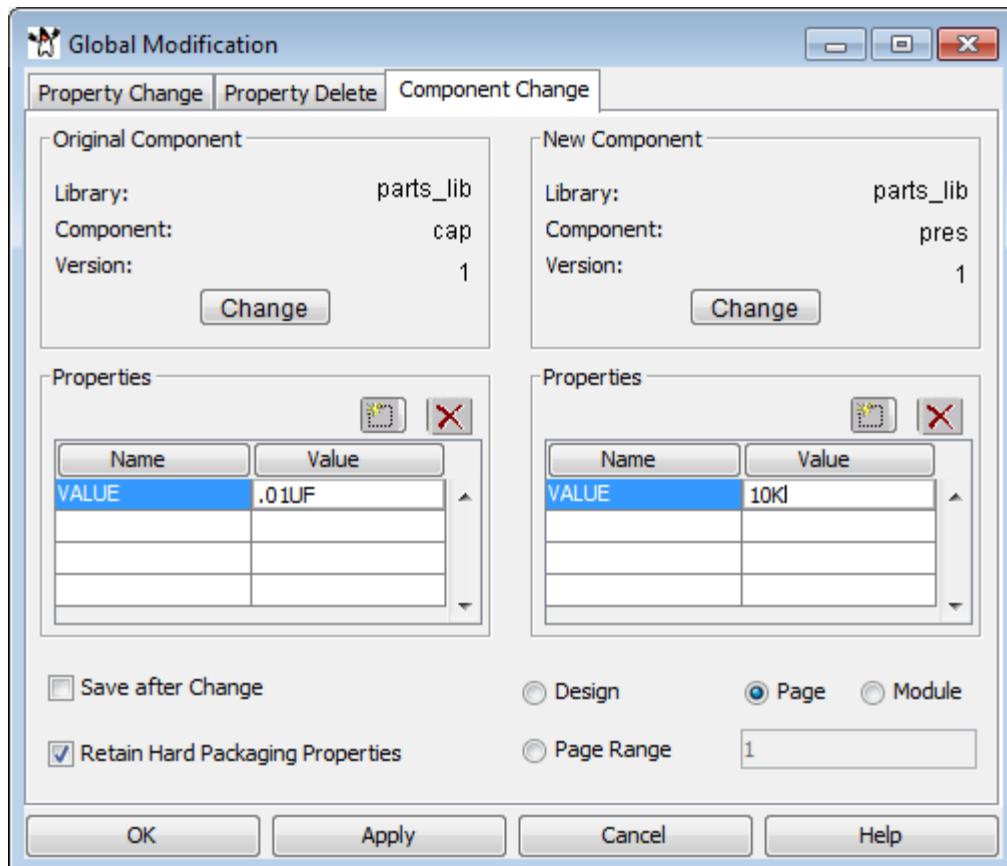
Be careful in using wildcards for versions. You might inadvertently make more changes than you want to. The probability of making changes by mistake is high when wildcards are used both in source and destination.

5. Click the *OK* button.

The Physical Part Filter window appears.

6. In the Physical Part Filter window, select the part and click *OK*.

The Global Modification window appears as shown below with the properties of the selected component listed in the *Properties* list below the *New Component* group.



7. Click the *OK* button.

The *Confirm Request* message box appears.

8. In the *Confirm Request* message box, click *Continue*.

The *Results* dialog box appears, indicating the replacement status.

## Selecting the Component to Replace from the Design (Schematic Pick)

To select the component to replace, do the following:

1. Click the *Change* button in the *Original Component* group.

The *Select Component to Change* dialog box appears.

2. Under *Schematic Pick*, select *Get Annotated Part Table Properties* to retrieve only the annotated part table properties of the original component. The *Get Annotated Part Table Properties* radio button is selected by default.

3. Click the *Advanced* button.

The *Original Component Options* dialog box appears.

4. In the *Original Component Options* dialog box, select *Selected Version* to replace instances of only the selected version of the original component. The *Selected Version* radio button is selected by default.

5. Click the *OK* button.

The *Select Component to Change* dialog box appears.

6. Click the *OK* button.

7. In the *Design Entry-HDL* message box, click *OK*.

8. Click a component in the schematic page to select the component.

The Physical Part Filter window appears.

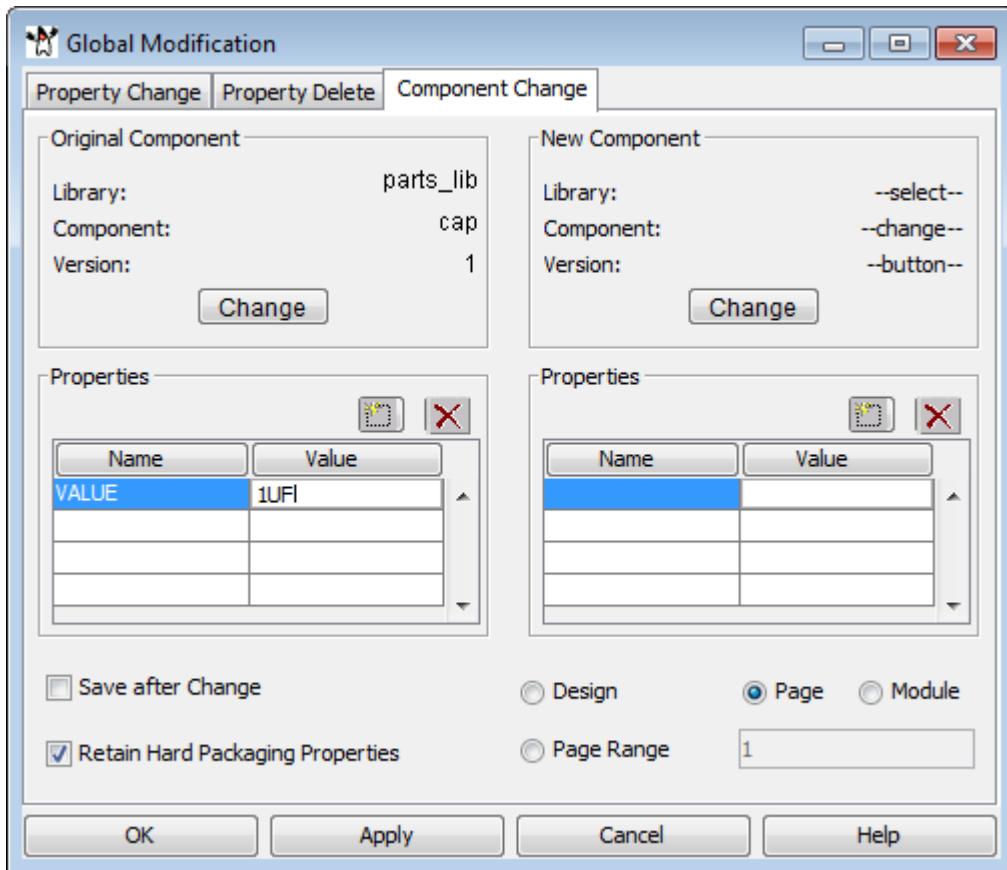
9. In the Physical Part Filter window, select the part and click *OK*.

The Global Modification window appears with the properties of the selected component

## Allegro Design Entry HDL User Guide

### Working with Designs

listed in the *Properties* list below the *Original Component* group.



### Selecting the New Component from the Design (Schematic Pick)

To select a new component, do the following:

1. Click the *Change* button in the *New Component* group.  
The *Select New Component* dialog box appears.
2. Under *Schematic Pick*, select *Get Annotated Part Table Properties* to retrieve only the annotated part table properties of the new component. The *Get Annotated Part Table Properties* radio button is selected by default.
3. Click the *Advanced* button.  
The *New Component Options* dialog box appears.

## Allegro Design Entry HDL User Guide

### Working with Designs

4. In the *New Component Options* dialog box, select *Selected Version* to replace the original component with the selected version of the new component. The *Selected Version* radio button is selected by default.

5. Click the *OK* button.

The *Select Component to Change* dialog box appears.

6. Click the *OK* button.

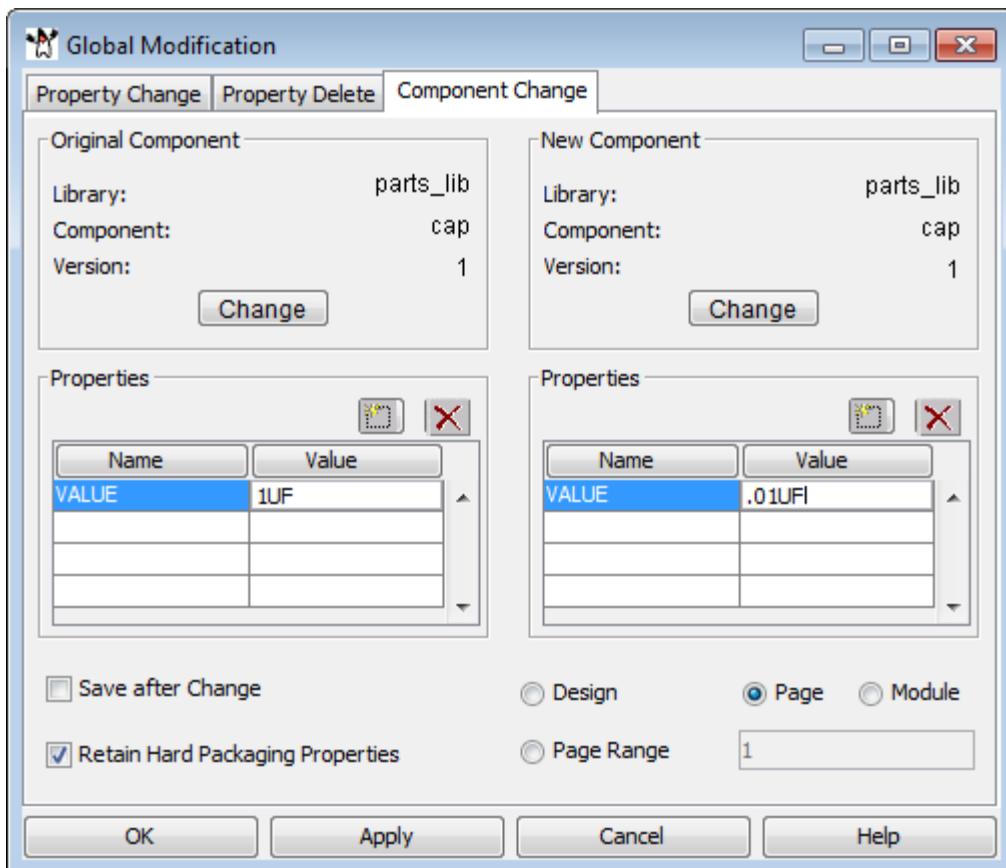
7. In the *Design Entry-HDL* message box, click *OK*.

8. Click a component in the schematic page to select the component.

The Physical Part Filter window appears.

9. In the Physical Part Filter window, select the part and click *OK*.

The Global Modification window appears as shown below with the properties of the selected component listed in the *Properties* list below the *New Component* group.



10. Click the *OK* button in the *Global Modification* window.

11. In the *Confirm Request* message box, click *Continue*.

The *Results* message box appears, indicating the replacement status.

### Deleting a Component

The guidelines for deleting a component are as follows:

- The Original Lib / Name / Version must be provided.
- The New must be left in the default mode. This can be obtained with a RMB click on the New component property table. There are 2 available options, the first blanks the table and the second blanks the table and restores the default to the Lib / Name / Version fields.
- Properties can also be used to qualify the Original component selection.

### Modifying a Component

The features of a component modification are as follows:

- The Lib / Name / Version fields must be the same in the Original and the New fields.
- If there are properties in the Original list that are not in the New list, they are deleted. In the case of default body properties, they are not deleted. However, this is noted in the summary GUI and the log file. The log file contains the changes made to the schematic and is stored in the `temp` folder of the project. The Global Modify solution supports three backup log files, `gc.log, 1`, `gc.log, 2`, and `gc.log, 3`, where 1 is the most recent backup and 3 the oldest. The most recent data is stored in a log file called `gc.log`.
- If there are properties in the New list that are not in the Original list, they are added. This is noted in the log file, but not in the Summary dialog.
- All packaging data is retained.
- A user property is one that is on the source component and not listed in the original component table of properties. These properties are retained during the modify operation.

### Replacing a Component

The features of a component replace are as follows:

## Allegro Design Entry HDL User Guide

### Working with Designs

---

- There must be at least one difference between the Original and the New Lib / Name / Version fields. The property fields do not determine the case, although they are taken into account when performing the requested change.
- The Reference Designator is always retained.
- When Retain Hard Packaging Info is enabled and the number of pins and location of pins on the Original and New component match, the Packaging Data (SEC and PN) is retained.
- When Retain Hard Packaging Info is disabled, the Packaging data is changed to SOFT.
- Anytime the number of pins or location of the pins relative to the origin changes, the packaging data (PN, SEC) is removed, regardless of the Retain Hard Packaging Info option.
- A user property is one that is on the source component and not listed in the original component table of properties. These properties are retained during the replace operation.
- The characteristics of properties (color, xy, justification, visibility, and size) are taken from the new component's definition. If a definition does not exist, the characteristics from the original component instance properties are used.
- If there are properties in the Original list that are not in the New list, they are deleted. In the case of default body properties, they are not deleted. However, this is noted in the summary GUI and the log file.
- If there are properties in the New list that are not in the Original list, they are added. This is noted in the log file, but not in the Summary dialog.

### Refreshing a Component

The features of a component refresh are as follows:

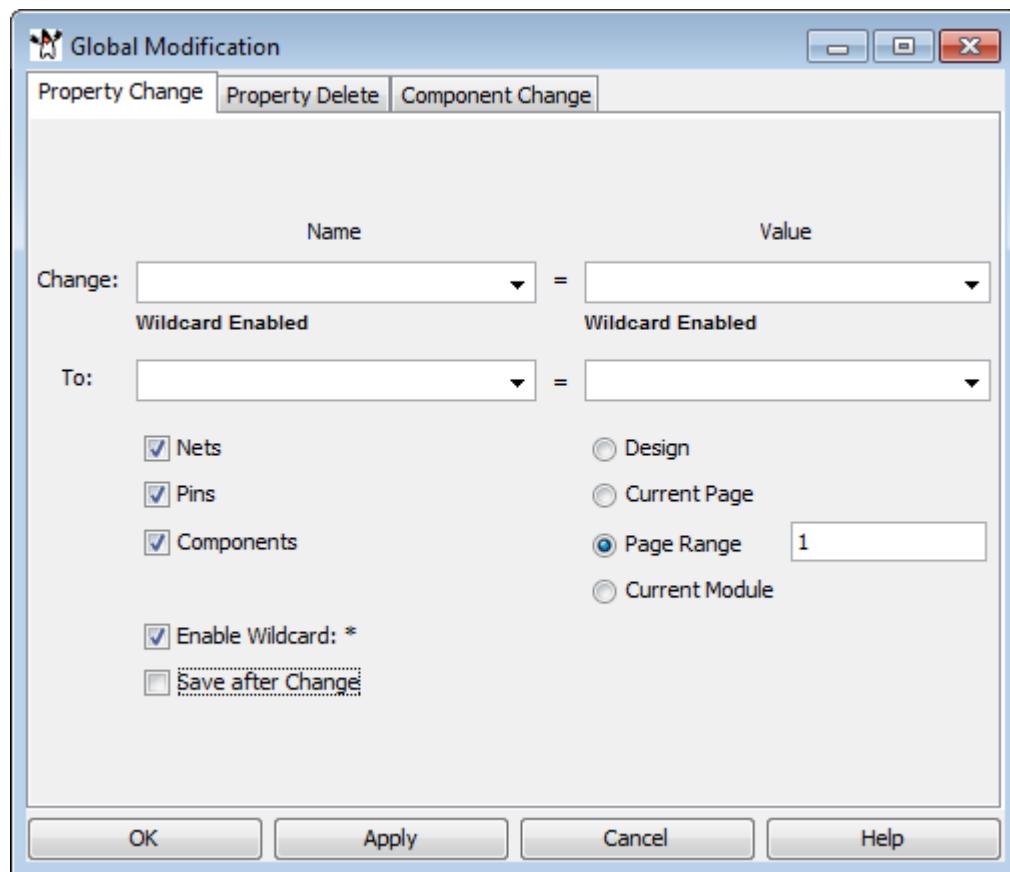
- The Lib / Name / Version and the qualifying properties are identical in the Original and New fields.
- This forces a Design Entry HDL Replace with the same component. The only changes that are seen are those that come from a library change of the component.
- The packaging data is retained.

## Erroneous Conditions

The following conditions are checked for before displaying the final confirmation dialog box. If any of these conditions are flagged, the confirmation dialog box is not displayed. You must fix these errors before executing the requested schematic modification. Each of these checks is also performed during batch command executions and any errors found are reported at the top of the log file.

### Property Change Tab

The conditions that are checked during a property change are as follows:



1. Wildcards exist in both the original property name and value fields.
2. The Preserve options are selected for the name and value in the same run.
3. Blank property names in the original and new property name fields.
4. Object type not selected.

## **Allegro Design Entry HDL User Guide**

### Working with Designs

---

#### **5. Invalid original property name syntax.**

Names must begin with a letter and contain letters, numbers, and the underscore character. The \$ character is allowed only at the beginning to designate soft properties. Wildcards are allowed.

#### **6. Invalid new property name syntax.**

Names must begin with a letter and contain letters, numbers, and the underscore character. The \$ character is allowed only at the beginning to designate soft properties. Wildcards are not allowed.

#### **7. Internal Design Entry HDL property names used in the original or new property name fields.**

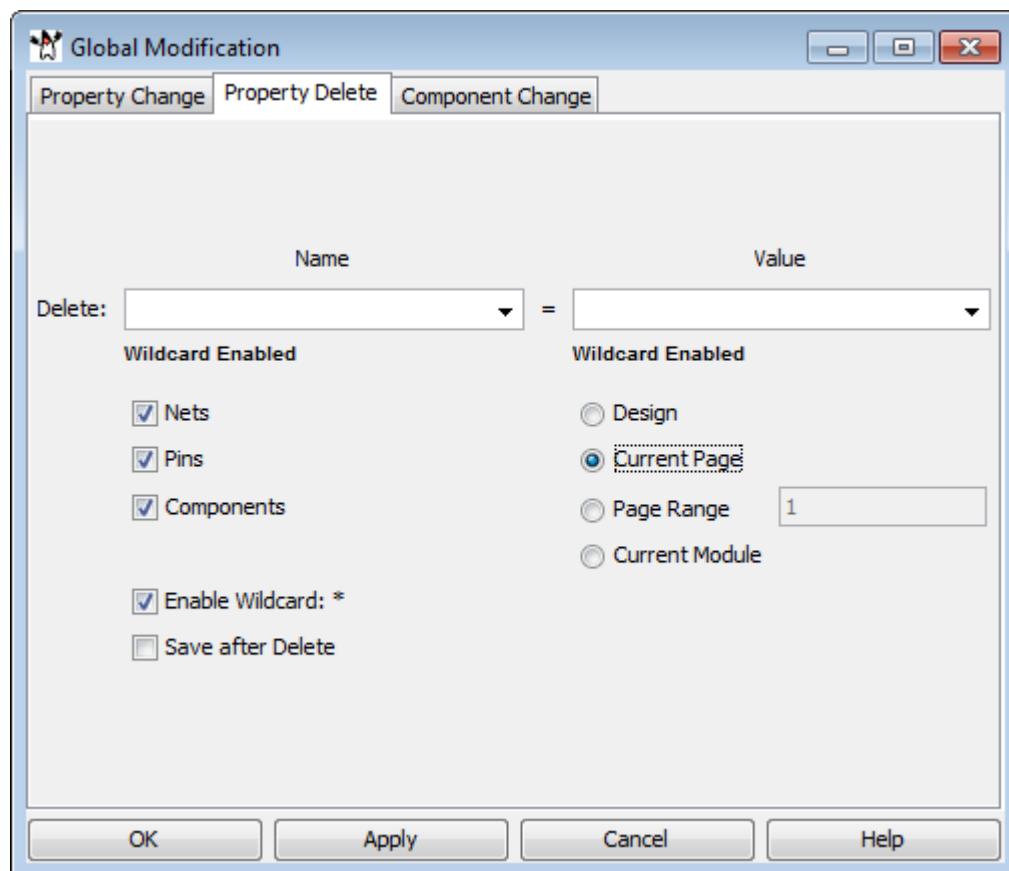
Examples of Design Entry HDL property names include CDS\_\*, CR\*, \$CR\*, PN, \$PN, and SEC\_TYPE.

#### **8. Identical source and destination property Name -Value pairs.**

#### **9. Invalid page range syntax.**

## Property Delete Tab

The conditions that are checked during a property delete are as follows:



1. Wildcards exist in both the delete property name and value fields.
2. Blank property name in the delete property name field.
3. Object type not selected.
4. Invalid delete property name syntax.

Names must begin with a letter and contain letters, numbers, and the underscore. The \$ is allowed only at the beginning to designate soft properties. Wildcards are allowed.

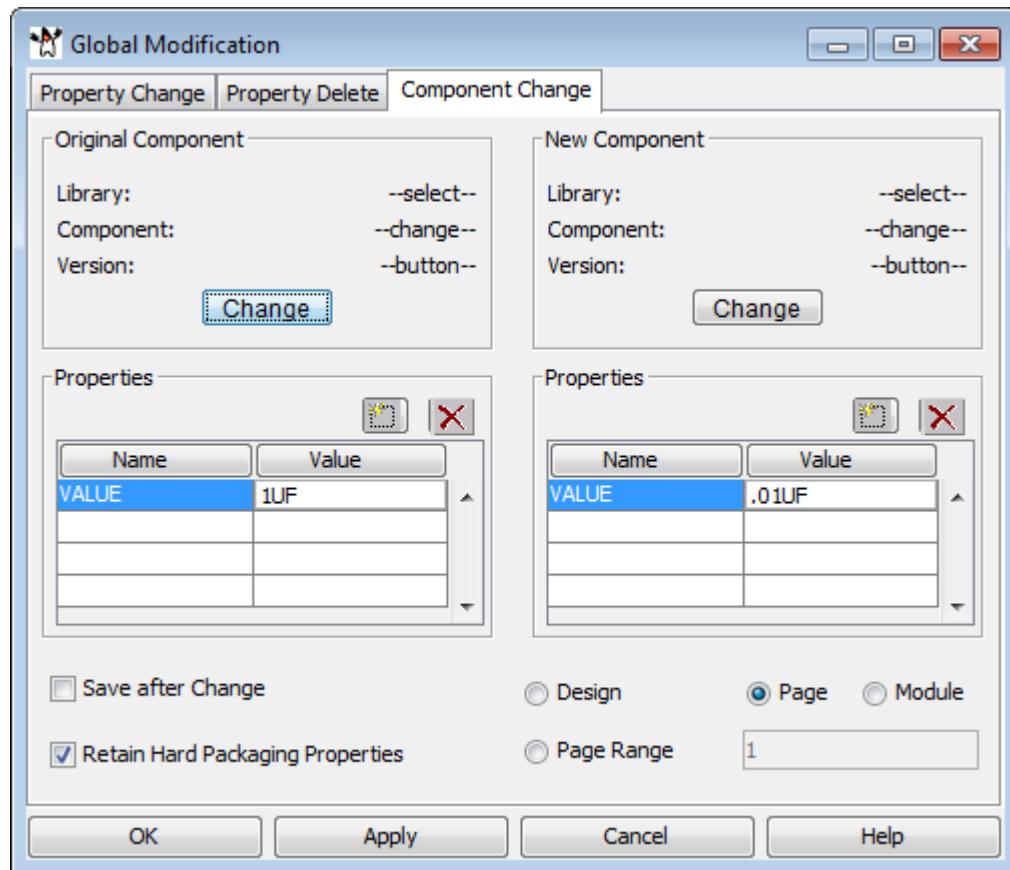
5. Internal Design Entry HDL property name used in the delete property name field.

Examples of Design Entry HDL property names include CDS\_\*, CR\*, \$CR\*, PN, \$PN, and SEC\_TYPE.

6. Invalid page range syntax.

## Component Change Tab

The conditions that are checked during a component change are as follows:



- Blank or default values in the Library/Component/Version fields.

This check also takes into account the component delete case, where the new component is left blank or with the defaults and the new component is valid.

- Blank property names in the original and new property fields.

- Invalid original and new property name syntax.

Names must begin with a letter and contain letters, numbers, and the underscore. The \$ is allowed only at the beginning to designate soft properties. All \* are treated literally. Wildcards are not allowed.

- Internal Design Entry HDL property names used in the original or new property name fields.

Examples of Design Entry HDL property names include CDS\_\*, CR\*, \$CR\*, PN, \$PN,

and SEC\_TYPE.

#### 5. Invalid page range syntax.

## Design Entry HDL Issues Affecting the Global Modify Solution

- If you select a component that does not have physical part data, Part Information Manager shows the data from the last component that had physical part information.
- When using the batch command, file option changes cannot be ignored because Design Entry HDL does not provide an automated way to exit and ignore modified drawings.

## The Batch Mode Operation

Access to the batch mode operation is provided through a Design Entry HDL console command called \_globalBatch. The \_globalBatch command is used only for flat designs. This command takes a single argument, that is, the name of a command file. Relative paths are resolved according to the location of the CPM file.

A command file can contain a single command or as many commands as you want. All commands contained from within a command file are dumped to a single log file. If multiple log files are desired, you must use multiple command files. Command files can handle comments.

## Command File Sample

```
; ; Sample Global Change/Delete/Modify/Replace Command File
; ; A Semicolon NOT FOUND inside double quotes designates a comment
; ; This file must contain 1 master structure but the structure can
; ; contain as many commands as desired.
; ; White space is ignored as long as it is NOT within double quotes
; ;
; ; The following are case insensitive keywords and do not need to be quoted:
; ; True, False, Design, Page, Module
; ;
; ; All property names, values, component names, library names, component
; ; versions and page ranges must be quoted.
; ;
; ; The -SCOPE option supports keywords or a range of pages. Even though
; ; the keywords do not need quotes the range range does. The page range
; ; accepts comma separated list of pages and page ranges designated by a '-'
; ; Example: "1,3,5,7-12"
; ;
; ; A special keyword string "<<PRESERVE>>" is allowed in the _globalchange
```

## Allegro Design Entry HDL User Guide

### Working with Designs

---

```
;; -ToProp fields. This indicates to retain the source property name or
;; source property value. <<PRESERVE>> cannot be used for both the
;; name and value in the same run, otherwise there would be nothing to change!

( ;; The parenthesis starts the definition of the master structure
( _globalDelete
( -Nets      true          ) ;; True / False
( -Pins      true          ) ;; True / False
( -Comps     true          ) ;; True / False
( -Scope     design        ) ;; Design / Page / Module / "1,2,5-7"
( -Save      true          ) ;; True / False
( -Wild      true          ) ;; True / False
( -Prop      "name" "value" ) ;; Double-Quoted Strings

) ;; Each command must also have starting and ending Parenthesis
;; This parenthesis ends the _globalDelete Command

( _globalChange
( -Nets      false         ) ;; True / False
( -Pins      false         ) ;; True / False
( -Comps     false         ) ;; True / False
( -Scope     page          ) ;; Design / Page / Module / "1,2,5-7"
( -Save      true          ) ;; True / False
( -Wild      true          ) ;; True / False
( -FromProp  "name" "value" ) ;; Double-Quoted Strings
( -ToProp    "name" "value" ) ;; Double-Quoted Strings or "<<PRESERVE>>"

) ;; This ends the _globalChange Command

( _globalModify
( -Scope     page          ) ;; Design / Page / Module / "1,2,5-7"
( -Save      true          ) ;; True / False
( -HardProp   true         ) ;; True / False
( -FromLib   "lib"         ) ;; Double-Quoted String
( -FromCell   "cell"       ) ;; Double-Quoted String
( -FromVer    "ver"         ) ;; Double-Quoted String
( -FromProp   "name" "value" ) ;; Double-Quoted Strings
( -FromProp   "name" "value" ) ;; Double-Quoted Strings
( -FromProp   "name" "value" ) ;; Double-Quoted Strings
( -ToLib     "lib"         ) ;; Double-Quoted String
( -ToCell    "cell"       ) ;; Double-Quoted String
( -ToVer     "ver"         ) ;; Double-Quoted String
( -ToProp    "name" "value" ) ;; Double-Quoted Strings
( -ToProp    "name" "value" ) ;; Double-Quoted Strings
( -ToProp    "name" "value" ) ;; Double-Quoted Strings

) ;; This ends the _globalModify Command

( Exit )
) ;; This parenthesis ends the definition of the master structure
```

## Part Manager

The Part Manager utility in Design Entry HDL provides you a convenient way of viewing information about the part table file (PTF) rows associated with part instances on a schematic. Part Manager is a GUI-based utility that helps you check the status of physical properties of part instances in a design and prevent errors, which would otherwise occur when you package the design.

Part Manager provides you a spreadsheet-like interface to update part instances on a schematic with appropriate PTF rows with a simple click of a mouse. Part Manager also comes handy in case of design reuse, where updating each reused part instance on a schematic with a PTF row can be a time-consuming task.

This section covers:

- [How Part Manager Works](#) on page 447
- [The Part Manager User Interface](#) on page 448
- [Working with Part Manager](#) on page 453

### How Part Manager Works

Part Manager shows the complete summary of a design, including names of parts, key and injected properties, and part status of all the parts used in the design.

You use Part Manager to accomplish the following tasks:

- Check the status of all the parts used in the design with respect to the corresponding ptf. The status depicts whether the physical properties of a part match any row in the ptf.
- Highlight part instances on the schematic from within the Part Manager user interface. This is particularly useful when you want to check for conflicts between ptf rows and part instances on the schematic.
- Update one or more part instances on the schematic with the associated ptf row.

### Opening Part Manager

You can open Part Manager in one of the following ways:

- From the *Tools* menu of Project Manager, click *Part Manager*.
- From the *Tools* menu of Design Entry HDL, click *Part Manager*.

## Allegro Design Entry HDL User Guide

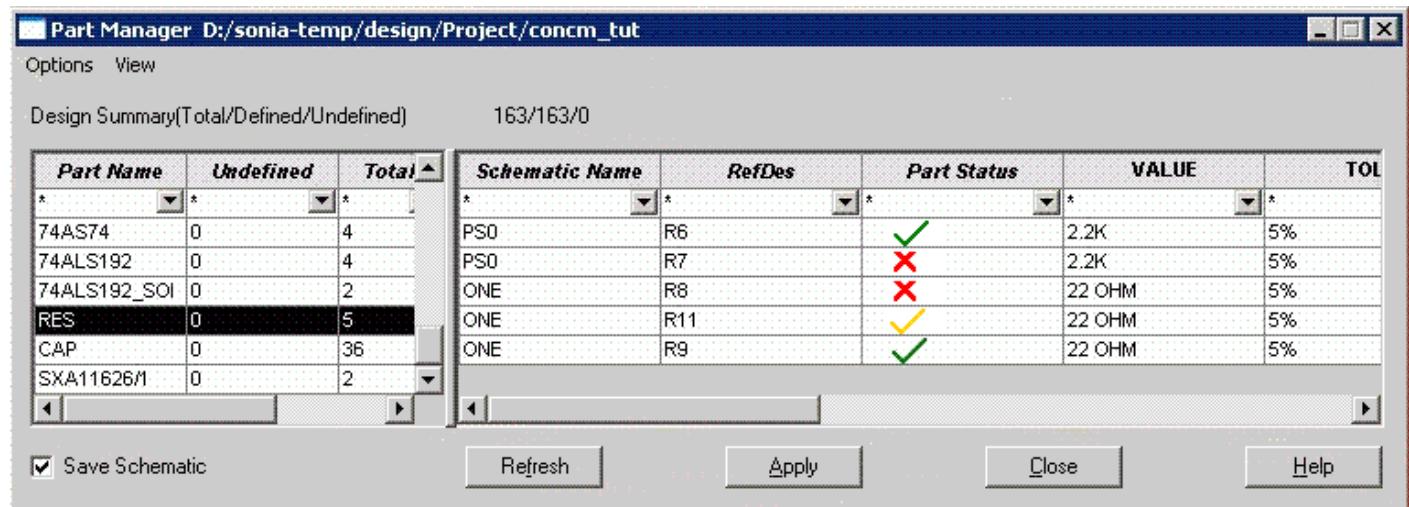
### Working with Designs

- Type partmgr and press Enter in the console window of Design Entry HDL.
- Click  in the Design Entry HDL main window.

Part Manager loads the design and displays the complete information about the parts used in the design.

### The Part Manager User Interface

The Part Manager user interface is a two pane window. The left pane lists the names of the parts used in the design, total number of instances of the part, and the instances that are out of sync with the corresponding ptf. The left pane also lists the summary of parts instances indicating the number of defined and undefined part instances. The right pane displays a detailed grid of part information including the key and injected properties, and the part status.



### Left Pane

The Design Part Names list on the left pane lists the physical part names of all the electrical parts used in the design. By default, the physical name of the part which has the maximum number of out-of-sync instances, is selected and a detailed grid on the right pane shows complete information about the part. Multiple selections are not allowed in the Part Names list. The left pane also shows the summary of the part instances.

In case of logical components, the part names are picked up from the `chips.prt` file. If the `chips.prt` file is not found, cell name is used.

#### Right Pane

When you select a part name in the left pane, a detailed list of information about all the instances of the selected physical part, is displayed on the right pane of Part Manager in a grid format. Both key and injected properties appear on the grid.

**Example:** If you want to get a snapshot view of all the instances of resistors used in a design, select “RES” in the Design Part Name list and the relevant physical part information will be listed in the grid on the right pane.

#### Columns

The first three column headers of the grid are hard-coded and are available for all physical parts, by default. The column headers of the first three columns are distinguished by bold and italicized text.

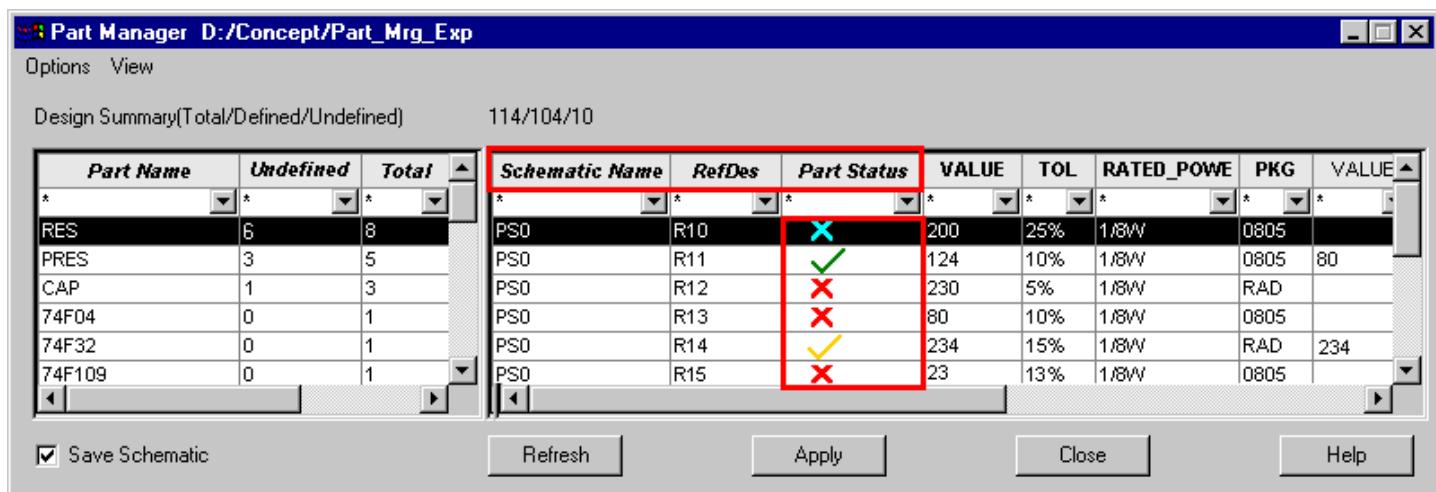
## Allegro Design Entry HDL User Guide

### Working with Designs

---

**Table 11-1**

<b>Column Header</b>	<b>Description</b>
Schematic Name	This column shows the block name of the part instance
RefDes	This column shows the reference designator of the component. If the location property is not available, a question mark ("?") is displayed in this column
Part Status	This column shows the status of parts represented by icons. The values that this column can show are listed in <a href="#">Table 11-2</a> on page 450



**Table 11-2**

<b>Icon</b>	<b>Part Status</b>	<b>Description</b>
	MATCHED (Green)	This part instance was added in logical mode (no ptf rows used) and the PXL directive FORCE_PTF_ENTRY is not set. Part Manager recognizes a part as logical if the part does not have a ptf associated with it.
	MATCHED (Green)	This part instance matches a row in any of the part table files. The part was added in physical mode and no part table property has been updated manually.

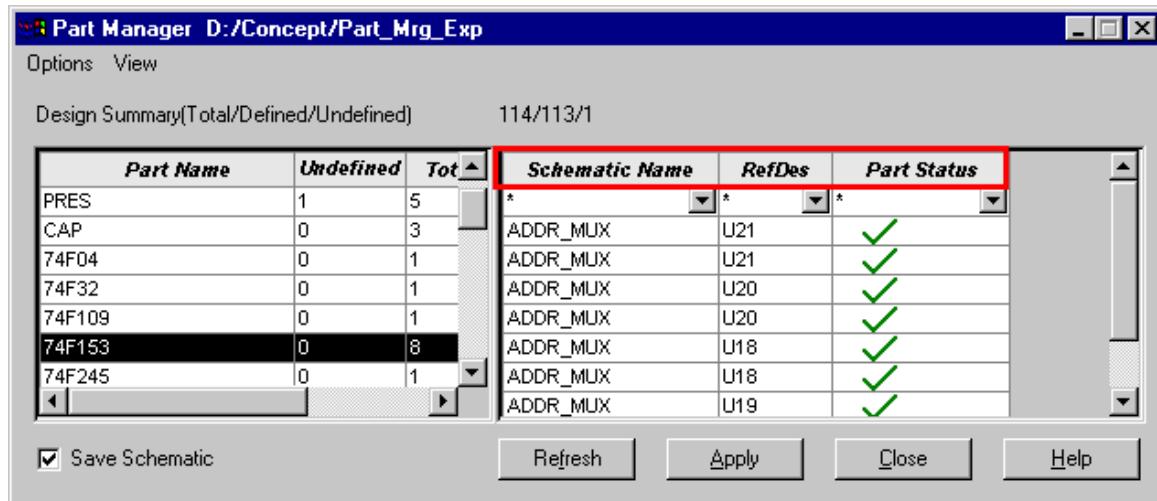
## Allegro Design Entry HDL User Guide

### Working with Designs

Icon	Part Status	Description
(Yellow)	INJECTED MISMATCH	This part instance was added in logical mode, but the PXL directive FORCE_PTF_ENTRY is set. PXL packages such a part with a warning.
(Yellow)	INJECTED MISMATCH	This part instance matches a row in one of the part table files, such that all key properties match, but one or more injected properties do not match. PXL packages such a part without any warning.
(Red)	NOT MATCHED	This part instance does not match any row in any part table file.
IGNORED	IGNORED	This status is shown for the parts which have a PACK_IGNORE property.

#### ***Status of logical parts***

Part Manager displays only three default columns for logical parts. Depending on the PXL directive FORCE\_PTF\_ENTRY, the status will be Green or Yellow.

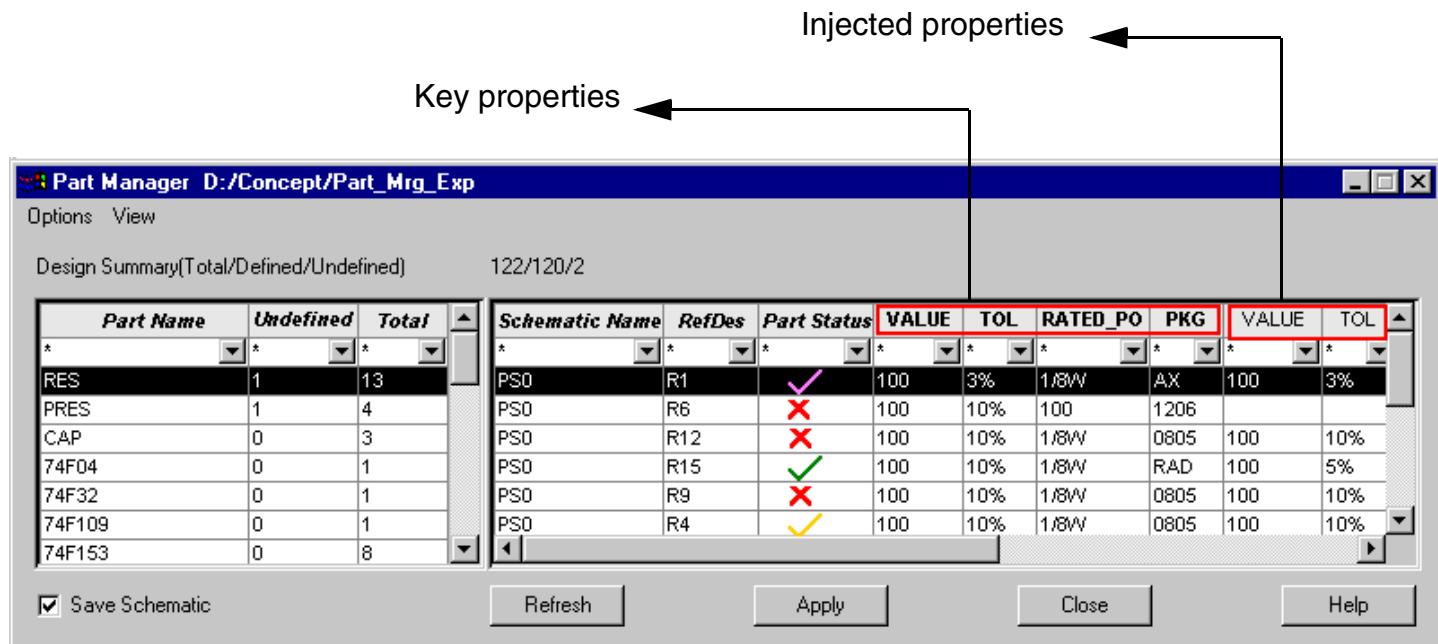


In addition to the default Part Manager columns, key and injected properties of a part instance are displayed. For example, in the figure shown below, the part "RES" has four key properties,

## Allegro Design Entry HDL User Guide

### Working with Designs

RATED\_POWER and PKG and two injected properties, VALUE and TOL. Note that a key property column header appears in bold.



### Filters

You can choose to display specific part instances by applying filters. Each of the columns of the Part Manager window has filters that support wildcard characters. These filters have a drop-down combo box that shows all the valid values which you can use to filter out the required values.

### Summary (Total/Defined/Undefined)

The Part Manager user interface also displays a summary of all the parts of the design and their status. The summary lists:

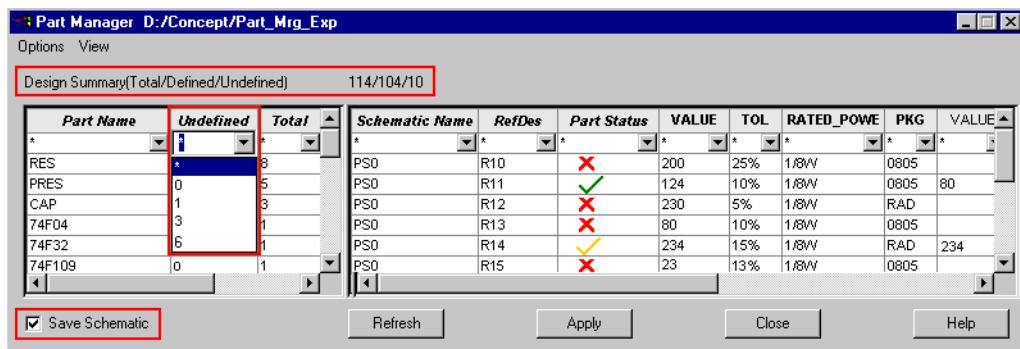
- Total number of parts
- Defined parts (all parts with a “MATCHED” or “INJECTED MISMATCH” status)
- Undefined parts (all parts with a “NOT MATCHED” status)

The figure displayed below displays Part Manager information for a design, which contains 114 part instances, out of which 104 are defined and 10 are undefined. When you update an

# Allegro Design Entry HDL User Guide

## Working with Designs

undefined part instance with a valid ptf entry and apply the changes, the summary is adjusted to reflect the current status of the design.



### Save Schematic

The *Save Schematic* check box on Part Manager facilitates the updating of the schematic with the changes that you make in Part Manager. If this check box is not selected, changes are passed to schematic, but not saved. If you save the schematic, you will see the updated values.

## Working with Part Manager

Part Manager provides functions for updating the part instances on a schematic with their corresponding values in a part table file. You can also update multiple part instances simultaneously with a single ptf row. In addition, you can highlight a specific part instance on the schematic by selecting it in the Part Manager grid. Part Manager also provides you with a toggle to show or hide the canonical path of all the part instances. This section covers:

- [Updating Part Instances](#)
- [Updating All Part Instances with Mismatched Injected Properties](#)
- [Applying Changes to Part Instances](#)
- [Resetting Changes Made to Part Instances](#)
- [Updating Multiple Parts with a Single PTF Row](#)
- [Refreshing Contents of Part Manager](#)
- [Highlighting Part Instances on the Schematic](#)
- [Displaying the Canonical Path](#)

## Allegro Design Entry HDL User Guide

### Working with Designs

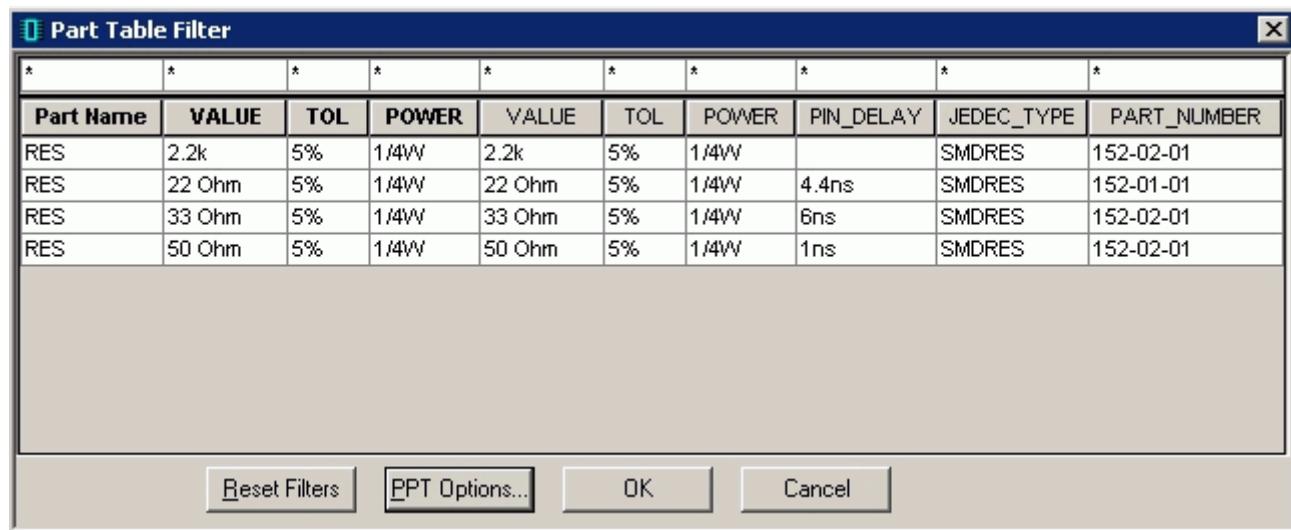
## Updating Part Instances

To resolve undefined parts in the design, you update part instances displayed on the Part Manager grid with appropriate ptf rows.

To update an undefined part instance:

1. On the Part Manager grid, right-click a row with a NOT MATCHED (red) or INJECTED MISMATCH (yellow) part status.
2. On the pop-up menu, click *Update Instance*.

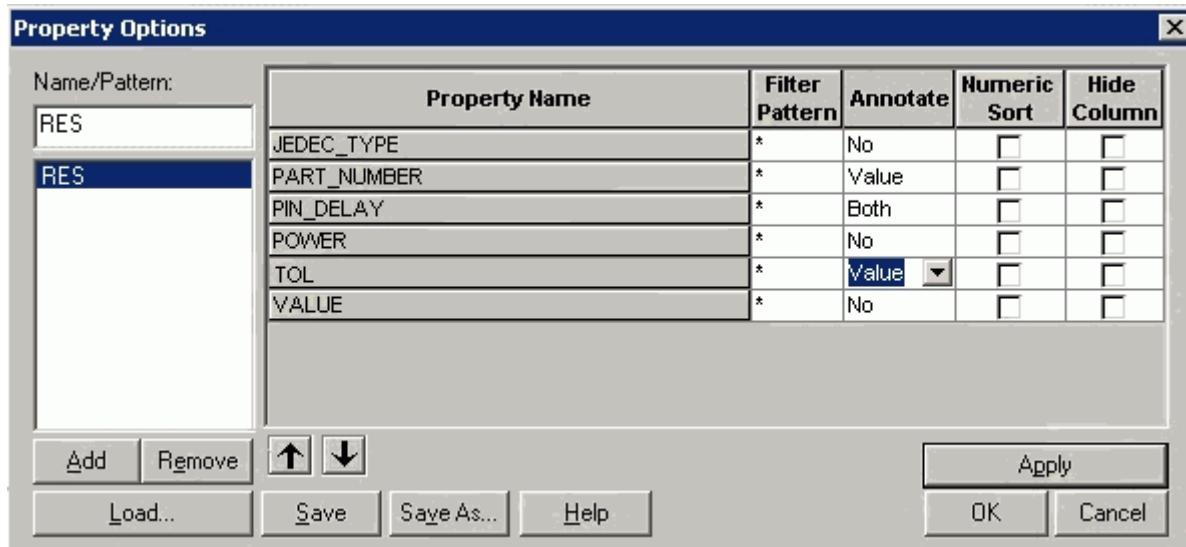
Alternatively, you can select *Update Instance* from the *Options* menu. This brings up the Part Table Filter window.



## Allegro Design Entry HDL User Guide

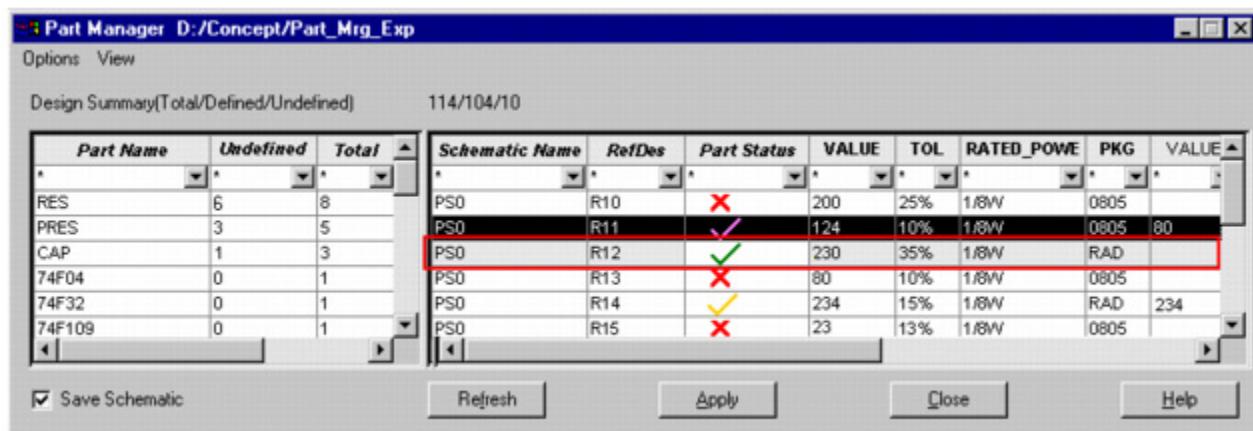
### Working with Designs

The value that you select for the *Annotate* option for an injected property in the Part Table Filter window will be preserved and displayed in the schematic.



- From the Part Table Filter window, choose an appropriate row with which you want to update the selected part instance.

The status of the selected row in the grid changes from NOT MATCHED (red) or INJECTED MISMATCH (yellow) to MATCHED (green) and the background is grayed out. The grayed out background indicates that the part instance has been modified in the memory. However, the change is yet to be reflected on the schematic.



**Note:** Make sure that the *Save Schematic* check box is selected when you update a part instance, else changes will not be saved to the schematic.

**Note:** When you launch Part Manager from Project Manager, the component instance is not updated in the same session if DE-HDL is already launched. To view the updated component instance in the schematic, you need to relaunch DE-HDL.

## Updating All Part Instances with Mismatched Injected Properties

In previous releases of Part Manager, you had to update mismatched injected properties for part instances one-by-one. Now, you can update all the part instances in the design, that have mismatched injected properties in one go.

To update all the instances of the parts with mismatched injected properties, do one of the following:

- Choose *Options - Update All Injected Mismatch* in Part Manager.
- Run `partmgr -updateallinj` from the DE-HDL console window.

All the part instances that have mismatched injected properties are updated and a log file *update\_inj.log* is created in the temp directory of the project which contains update details of design. If a design or a page is locked or read-only, a message is logged in this file.

## Applying Changes to Part Instances

Merely updating the part instance does not update the schematic with the changes. For the changes to take effect, you also need to apply the changes to schematic.

You can apply change to the schematic with the new values in one of the following ways:

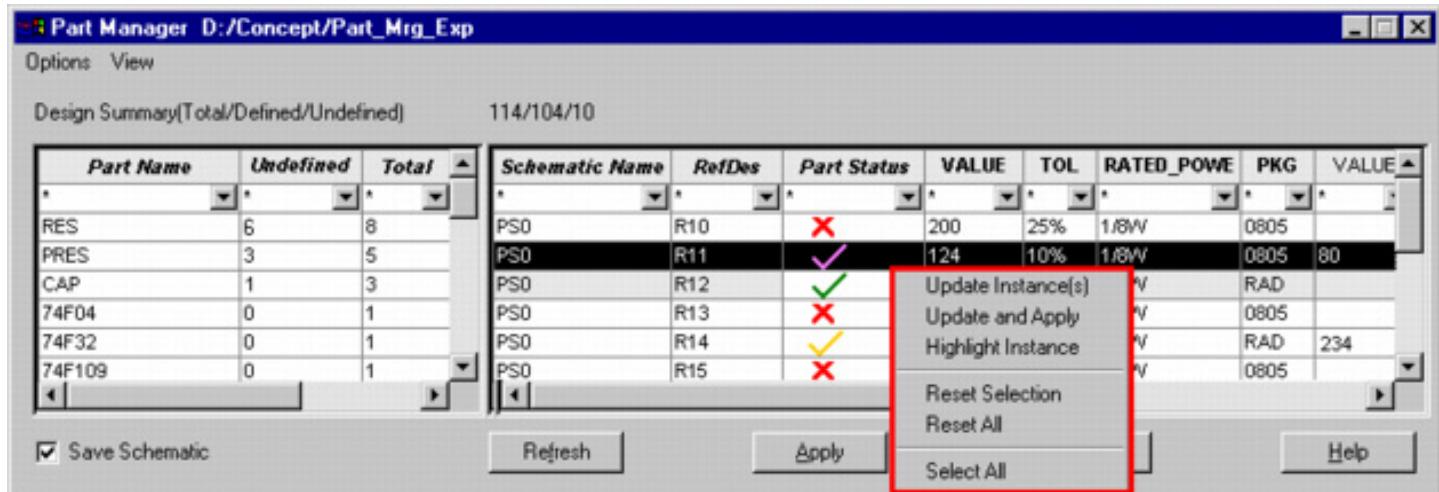
- Select the updated row and click the *Apply* button.
- From the *Options* menu on the Part Manager window, select *Apply Changes*.

Alternatively, you can use the *Update and Apply* command from the *Options* menu, which works exactly like the *Update Instance* and *Apply Changes* commands put together.

**Note:** In case the selected row is part of a reuse block, Part manager will show a warning: "This component will be replaced in all the instances of the reuse block. Do you want to continue?". Selecting "Yes" replaces that part in all instances of the reuse block. Selecting "No" aborts the update process.

## Resetting Changes Made to Part Instances

If you have updated a rows with new values, but not yet applied the changes, you can revert back to the original schematic values. The *Reset Selection* and *Reset All* commands help you achieve this. This option is particularly useful when you update a row incorrectly and want to undo the update operation.



To reset original schematic values:

1. Right-click the affected row.
2. From the pop-up menu, choose *Reset Selection*. In case you want to undo changes on multiple rows, select *Reset All*.

Alternatively, you can choose *Reset All* or *Reset Selection* from the *Options* menu.

The original values are restored.

## Updating Multiple Parts with a Single PTF Row

You can update multiple or all instances of a part, simultaneously. This action updates all the selected part instances with a single ptf row irrespective of the individual statuses of the part instances.

To update multiple part instances with a single ptf row:

1. On the Part Manager grid, right-click any row.
2. From the pop-up menu, choose *Select All*.

## Allegro Design Entry HDL User Guide

### Working with Designs

Alternatively, you can choose *Select All* from the *Options* menu. To select specific part instances on the grid, use the Ctrl + click or Shift + Ctrl + click combinations.

3. Right-click again and select *Update and Apply*.

All part instances are updated in the Part Manager grid and on the schematic with the single row that you selected in the Physical Part Filter window.

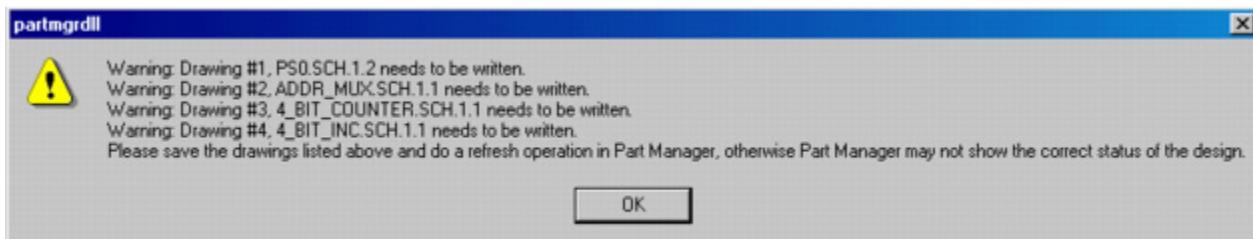
## Refreshing Contents of Part Manager

You can reload Part Manager with the updated details of the parts used in the design. This helps you verify if the design has actually been updated with the modifications that you made using the *Update Instance* and *Apply Changes* commands. Refreshing also helps in synchronizing Part Manager with Design Entry HDL, in case you make any changes in Design Entry HDL.

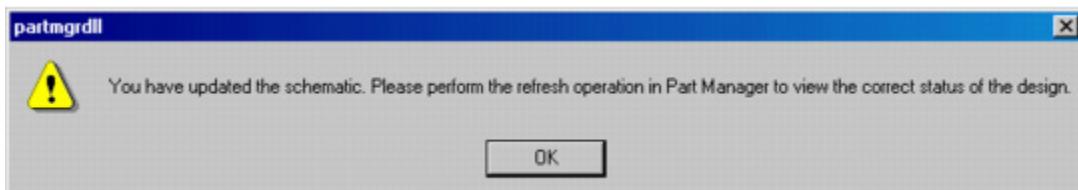
- To refresh the contents of Part Manager, click the *Refresh* button on Part Manager.

Part Manager reads the schematic and reloads the updated part instance details from the schematic.

**Note:** If you make any changes to the schematic in Design Entry HDL and move the focus back to Part Manager without saving the schematic, Part Manager gives a warning and prompts you to save the schematic pages where you made changes and refresh Part Manager.



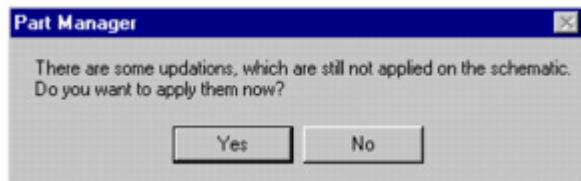
**Note:** If you save the schematic and then move the focus to Part Manager, it prompts you to refresh Part Manager.



## Allegro Design Entry HDL User Guide

### Working with Designs

**Note:** If you try to close Part Manager without applying changes on the schematic, Part Manager displays a message prompting you to apply the changes to the schematic.



## Highlighting Part Instances on the Schematic

With Part Manager you can easily locate a specific part instance on a cluttered schematic. You can highlight a specific part instance on a schematic from within the Part Manager grid.

To highlight a part instance on a schematic:

1. Select the appropriate row in the Part Manager grid and right-click.
2. Select *Highlight Instance* from the pop-up menu.

Alternatively, you can select *Highlight Instance* from the *Options* menu. The part instance is highlighted on the schematic. This option is available for single rows only.

## Displaying the Canonical Path

Another way of locating a part instance on a schematic is using its canonical path. In the Part Manager window, you can display the complete hierarchical path of all the part instances in the Part Manager grid.

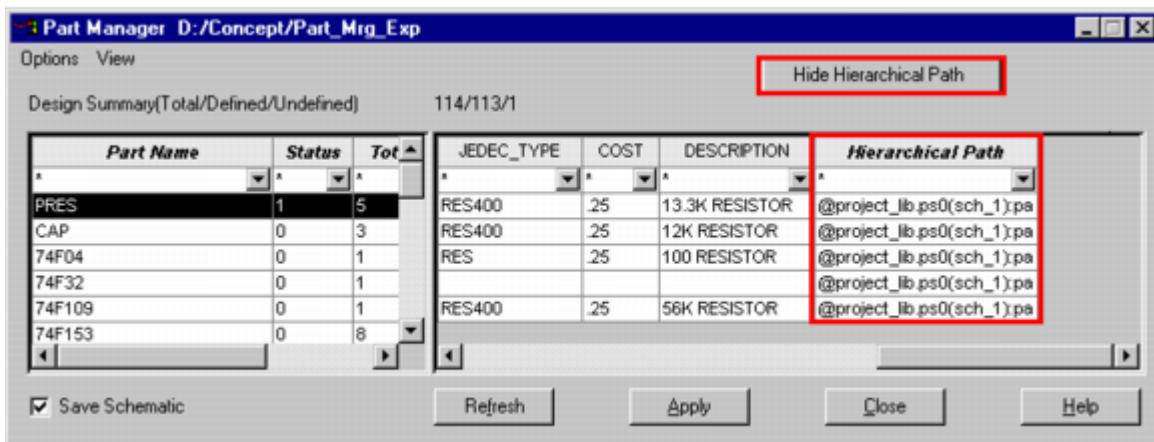
To show the hierarchical path of all part instances:

1. Right-click anywhere in the Part Manager window outside the grid and the Design Part Names list.
2. Select *Show Hierarchical Path*.

## Allegro Design Entry HDL User Guide

### Working with Designs

Alternatively, you can select *Show Hierarchical Path* from the *View* menu. A new column, titled *Hierarchical Path*, is added at the end of the Part Manager grid. This column displays the canonical path of each part instance.



Note that the menu option in the pop-up menu changes to "Hide Hierarchical Path". You can hide the column by selecting this menu option.

## Running Scripts

You can create a text file containing a list of Design Entry HDL commands (a script) to run in batch mode. Scripts can call other scripts and can be interactive.

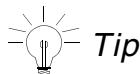
Within a script:

- The `pause` command temporarily interrupts the Design Entry HDL editor until you press a key.
- The `echo` command displays messages from the script file in the Design Entry HDL console window. This lets you track the progress of a script and is useful for debugging.
- You can specify X-Y coordinates in a script.
- You can use environment variables in a script using the `($ENV_VARIABLE)` syntax.

For example, if you are maintaining all your scripts at `/net/foo/script_home` and want to call a script named `check.scr` located at `/net/foo/script_home` from within your script, you can set the environment variable `$SCRIPT_HOME=/net/foo/script_home` and use the following command in your script file:

```
script ($SCRIPT_HOME)/check.scr
```

- You can include user input tokens to allow a script to request user inputs during an operation. For more information, see [User Input Tokens](#) on page 461.
- You can redirect all messages, warnings, errors to the command line irrespective of the *project.cpm* settings. This is useful when a Design Entry HDL script might suspend if the message output mode for errors or warnings is set to *Dialog* from the *Design Entry HDL Options* dialog box (Choose *Tools – Options*). For more information, see [Redirecting Messages to the Command line](#) on page 461.



#### Tip

You also can include comments in a script enclosed in curly braces '{}'.

### User Input Tokens

User input tokens must be placed at the beginning of a new line. There are two input tokens:

\$< When Design Entry HDL encounters this token in a script, it prints from the token to the end of the text line as a prompt, then waits for one item of input. The input can be a typed line, a pressed function key, a mouse point, or a *Ctrl + C* operation. You cannot press *Enter* in response to a user input request.

\$; This token also prints from the token to the end of the text line as a prompt and waits for input, but this token accepts and interprets input until you enter a semicolon. If this token is included, Design Entry HDL follows the prompt with the message:

Type ; when done with user input.

### Redirecting Messages to the Command line

In the script file, you can add the following `set` commands:

#### To

Redirect all the messages to the command line

#### Use

`set dialogs off`

Restore the .cpm setting

`set dialogs on`

So the script would look similar to:

```
set dialogs off
```

```
...
```

```
...
...
set dialogs on
exit
```

## Running a Script

You can run a script in the following four ways:

- [Running a script from the Tools menu](#) on page 462
- [Running a script from the Design Entry HDL console window](#) on page 463
- [Running a script from the Design Entry HDL console window](#) on page 463
- [Running a script from the Windows command prompt](#) on page 463

### ***Running a script from the Tools menu***

1. Choose *Tools – Script – Run Script*.  
The *Open* dialog box appears.
2. Navigate to the script file you want to run and highlight the filename.
3. Click *Open*.

### ***Running a script every time you open a project in Design Entry HDL***

If you want to automatically run a script every time you open a project in Design Entry HDL, do the following:

1. Choose *Tools – Options*.  
The *Design Entry HDL Options* dialog box appears.
2. Select the *Paths* tab.
3. Enter the name of the script file in the *Input Script* field or click *Browse* to select the script file.

If you want to run more than one script every time you open a project in Design Entry HDL, create a master script file and list the sequence in which you want to run the scripts in the master script file. Specify the name of the master script file in the *Input Script* field. For example, if you want to run a script named `check.scr` and then run a script named `zoom.scr`, do the following:

## Allegro Design Entry HDL User Guide

### Working with Designs

---

1. Create a script file named `master.scr` with the following entries:

```
script check.scr  
script zoom.scr
```

2. Enter `master.scr` in the *Input Script* field in the *Paths* tab of the *Design Entry HDL Options* dialog box.

#### ***Running a script from the Design Entry HDL console window***

- Use the `script <file_name>` command

You can specify the path to the script file or use an environment variable to specify the path to the script file. For example, if you are maintaining all your scripts at `/net/foo/script_home` and want to run a script named `check.scr` located at `/net/foo/script_home` from the console window, you can set the environment variable `$SCRIPT_HOME=/net/foo/script_home` and use the following command in your script file:

```
script ($SCRIPT_HOME)/check.scr
```

For more information on the `script` command, see [Script](#) in *Allegro Design Entry HDL Reference Guide*.

#### ***Running a script from the Windows command prompt***

- Use the following command:

```
nconcepthdl -proj <project_name>.cpm -scr <script_file_name>
```

For more information on the `nconcepthdl` command, see [Nongraphical Design Entry HDL](#) in *Allegro Design Entry HDL Reference Guide*.

## **Stopping a Script**

- Use `Ctrl+C` to stop a script.

## **Sample Scripts**

Some simple examples of scripts are given below.

#### ***Script to add a LS04 component to a drawing and use the mouse to position the part***

```
add ls04
```

\$<Place the LS04

**Script to run multiple script files in a specific sequence**

```
script /net/foo/scripts/set_options.scr  
script /net/foo/scripts/check.scr  
script /net/foo/scripts/zoom.scr
```

**Script to add a SIZE property to a part with a size specified at the time of entry**

```
property  
$<Choose the part to add a size to  
size =  
$<Type in the size you want and press Enter  
$<Place the property on the drawing
```

**Script to rotate an object until the user enters a semicolon**

```
rotate  
$;Rotate the object until properly oriented
```

A more complicated script might contain a large number of `signame` commands and prompt the user for a point to place each `SIG_NAME` property.

## Highlighting (Cross-Probing) Objects

You can highlight selected objects

- In expanded drawings - to trace a signal on multiple pages of a drawing and across multiple levels.
- Between Design Entry HDL and other system tools - to correlate the circuit logic to changes you made in the schematic or to navigate nets between a physical layout and the corresponding schematic.

To cross probe between Design Entry HDL and PCB Editor:

- a. Launch Project Manager and open a project.
- b. Launch Design Entry HDL from Project Manager.
- c. Launch PCB Editor from Project Manager.
- d. In PCB Editor, select the *Highlight* button to put PCB Editor in highlight mode or choose *Display – Highlight*.

- e. Right-click a net on the canvas and choose *Highlight* from the pop-up menu.

The corresponding signal in PCB Editor is highlighted.

- f. Similarly, select any signal in PCB Editor.

The corresponding net is highlighted in Design Entry HDL.

For more information on highlighting and dehighlighting objects, see [Highlighting Objects](#) on page 116 and [Turning Off Highlighting](#) on page 117.

## Distributing Design Changes between Physical and Logical Designs

1. Choose *Tools – Design Differences*.

The *Design Differences* dialog box appears.

2. Select one or both options to:

- update the board view to specify the Allegro PCB Editor board name in the PCB Editor Board box.
- update the package view.

3. Click *OK*.

See the [Design Synchronization and Packaging User Guide](#) for more information on handling design differences.

## Applying Connectivity Changes from the Physical Design to Your Schematic

1. Choose *Tools – Design Association*.

The *Design Association* dialog box appears with markers information.

2. Use the menu commands in the *Design Association* dialog box to apply the connectivity changes.

See the *Using Design Association* chapter of the [Design Synchronization and Packaging User Guide](#) for more information on applying connectivity changes from the board layout to your schematic.

## Back Annotating Your Design

- Choose *Tools – Back Annotate*.

Design Entry HDL reads the `psback.dat` file containing the physical part and adds the information to the design.

**Note:** To know more about the `psback.dat` file, refer to the *File Formats* chapter of the *Packager-XL Reference* guide.

## Module Ordering

In hierarchical designs, you can change the order in which child blocks are plotted and cross-referenced. You can also exclude certain modules from being cross-referenced or plotted. Design Entry HDL lets you reorder modules using drag and drop operations. You can exclude or include modules by simply right clicking on them and selecting a menu item.

**Note:** In the context of module ordering, a module refers to a hierarchical block that has a schematic associated with it.

The following conditions apply to module ordering:

- The modules that are being re-ordered must be at the same level of hierarchy.
- The modules that are being re-ordered must have the same parent module.

Module ordering saves all ordering and the exclusion or inclusion information in a module order file named `module_order.dat` in the `<root design>/sch_1` directory. The `module_order.dat` file is read during cross referencing and hierarchical plotting.

If you cannot view a cell in the hierarchy viewer window, save it in Design Entry HDL. Saving a cell stores its information in the `module_order.dat` file and as a result, the cell appears in the hierarchy viewer window when the file is read the next time. Note that a cell will appear in the hierarchy viewer window only when it is saved.

The hierarchy of modules is displayed in the form of a tree. You can choose to view the tree with or without the modules that have been excluded.

## Module Ordering in Design Entry HDL

The hierarchy viewer window displays the complete hierarchy of a design. The primary functions of this window are to provide a designer the ability to navigate through a design and reorder modules in the hierarchy.

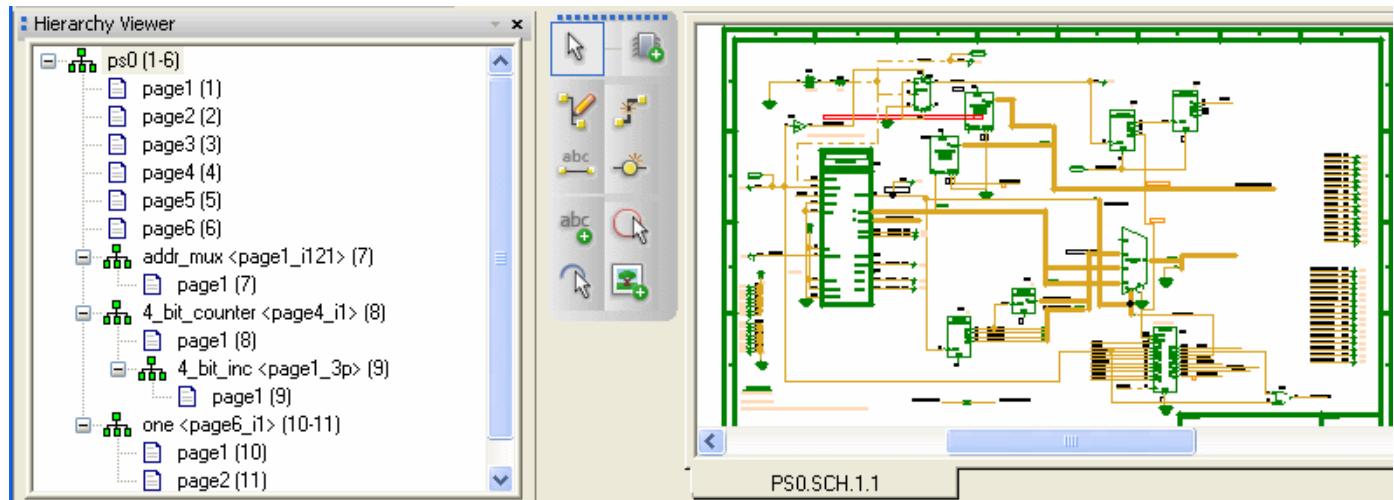
Moreover, the hierarchy viewer window is movable and dockable, which means that you can drag the window to any part of the screen and place it on an area that suits your requirement.

The hierarchy viewer window comprises of a tree structure, with the top-level design as the root node. In addition, all the hierarchical modules in the design are displayed in the window.

The tree structure displays sheet numbers for all non-excluded modules in the design. However, you can switch off the display of sheet numbers by selecting the *Hide Sheet Numbers* check box in the *Design Entry HDL Options* dialog box.

## The Hierarchy Viewer Tree

For a hierarchical design, the tree structure that appears in the hierarchy viewer window resembles the following figure:



For flat designs, only the top-level cell is displayed.

The hierarchy viewer tree supports all the module-ordering commands provided by the module-ordering dialog in previous releases. The tree also displays the existing module ordering of the design, as well as the excluded modules that are greyed out.

The hierarchy viewer tree always shows the sch\_1 hierarchy even if any other view is opened.

## Using the Hierarchy Viewer

You can display the hierarchy viewer window by selecting the *View Hierarchy* option in the View menu.

You can perform the following navigation and module-ordering functions in the hierarchy viewer window:

#### **Navigation functions**

##### ***Open a module in the Design Entry HDL window***

You can either click or double-click a module in the hierarchy viewer window to open it in the Design Entry HDL window for editing. Performing a single or a double-click is equivalent to right-clicking a module and selecting *Open* in the shortcut menu.

##### ***Open a module in a new Design Entry HDL window***

To open a module in a new Design Entry HDL window, right-click the module in the hierarchy viewer window and select *Open in New Window*.

##### ***Select an instance of a module***

The *Select Instance* menu option highlights the instance of the selected module on the parent schematic with a blinking box. The *Select Instance* option can be accessed by right-clicking a module in the hierarchy viewer window.

**Note:** The *Select Instance* option is disabled for the top-level module.

##### ***Jump to a page***

The *Go To Page* option allows you to jump to a page/symbol in the design. The *Go To Page* option can be accessed by right-clicking a module in the hierarchy viewer window.

##### ***Hide sheet numbers***

Select the *Hide Sheet Numbers* option to hide sheet numbers from appearing in the hierarchy viewer window. The *Hide Sheet Numbers* option can be accessed by right-clicking a module in the hierarchy viewer window.

**Note:** When displaying the number of pages, the hierarchy viewer gives a count of only those pages for which the corresponding `.csb` files exist. Pages without `.csb` files are ignored.

#### ***Hide instance names***

Select the *Hide Instance Names* option to hide instance names from appearing in the hierarchy viewer window. The *Hide Instance Names* option can be accessed by right-clicking a module in the hierarchy viewer window.

#### ***Refresh the Hierarchy Viewer***

The *Refresh Hierarchy* option updates the tree structure in the hierarchy viewer window with any changes made to the design, such as deleting or adding a new module to the design. The *Refresh Hierarchy* option can be accessed by right-clicking either a module or inside the hierarchy viewer window.

#### **Module-Ordering functions**

The module ordering functions are available as context-sensitive options for modules. These options can be accessed by right clicking a module and then selecting the *Module Order* option.

When you do module ordering using the hierarchy viewer window, the cross-references on the schematic are not updated automatically. You must run CRefer again to synchronize the cross-references on the schematic with the module ordering operation.

The following are the module-ordering functions available under *Module Order*.

##### ***Exclude Occurrence***

Excludes only the current occurrence of the module.

**Note:** If a cell has been excluded using module ordering or xmodules.dat, the sheet number for the cell is not shown.

##### ***Exclude All***

Excludes all occurrences of the module.

##### ***Include Occurrence***

Includes only the current occurrence of the module.

#### ***Include All***

Includes all occurrences of the module.

#### ***Hide Excluded Modules***

Hides excluded modules.

#### ***Excluded Modules***

Displays a list of all excluded modules.

#### ***Reset Module Order***

Clears all exclusions and inclusions.

### **Excluded Modules File**

Apart from excluding modules through the hierarchy viewer window, modules can also be excluded by mentioning the module name in a file named `xmodules.dat`. This file can reside in the following places:

- Hierarchy - `<your_inst_dir>/share/cdssetup`
- Home - `$HOME/cdssetup`
- Project - `<proj_dir>/cdssetup`

**Note:** For a given project, the file at the project level is given precedence over other projects. If the file is not present in the `<proj_dir>/cdssetup` then the one in the Home directory takes precedence over the project in hierarchy and applies to all of the projects. If there is no such file in the Home directory, the information is read from the one in the hierarchy and it applies to all projects at the site.

The format of the `xmodules.dat` file is

```
(“<module-name-1>” “<module-name-2>” “<module-name-3>”)
```

For example, if a user has two hierarchical blocks, `capacitor` and `gnd`, and wants to exclude them from plotting and cross referencing, the `xmodules.dat` file should appear as

```
(“capacitor” “gnd”)
```



Ensure that there are no spaces before or after the module name. For example, to exclude a module named `clock`, if you specify ("clock") in the `xmodules.dat` file, the module will not be excluded. Note that you can have spaces within the module name. For example, to exclude a module named `power supply`, you can specify ("power supply") in the `xmodules.dat` file.

**Note:** If a module is manually removed by writing it in the `xmodules.dat` file, the `module_order.dat` file is not updated. The module still appears excluded in the hierarchy viewer window and the Plot dialog box. If you want to include this module again, include the module in the hierarchy viewer window.

## Retaining Module Ordering While Importing Blocks

The module order defined in an imported block can be inherited and updated in the main project. This can be achieved by setting the following directives to 'ON' in the project or site cpm files. By default, they are set to 'OFF' in the installation `cds.cpm` file.

```
START_CONCEPTHDL  
READ_LOWER_LEVEL_MODULEORDER 'OFF'  
HONOR_SOURCE_MODORDER_ON_REIMPORT 'OFF'  
END_CONCEPTHDL
```

### **READ\_LOWER\_LEVEL\_MODULEORDER**

When you import a new block into a hierarchical block, the module ordering in the lower-level (imported) block is read and the same module order is used in the main design. This directive should be set whether you import or re-import a block.

### **HONOR\_SOURCE\_MODORDER\_ON\_REIMPORT**

When you re-import a block, the existing module ordering at the destination is ignored if a `module_order.dat` file is present in the re-imported block. The module order from the re-imported block is used and updated at the top-level design. If the directive is set to off, which is also the default value, the module order set in the top-level design is retained.

**Note:** The `module_order` of only the root block is honored. For any lower-level block to be imported or re-imported, module ordering needs to be performed with the block launched as the root design.

## Sheet Names in Hierarchy Viewer

The hierarchy viewer shows sheet names along with the block names and page numbers under each block in the design. The pages appear just below the block, before the child blocks.

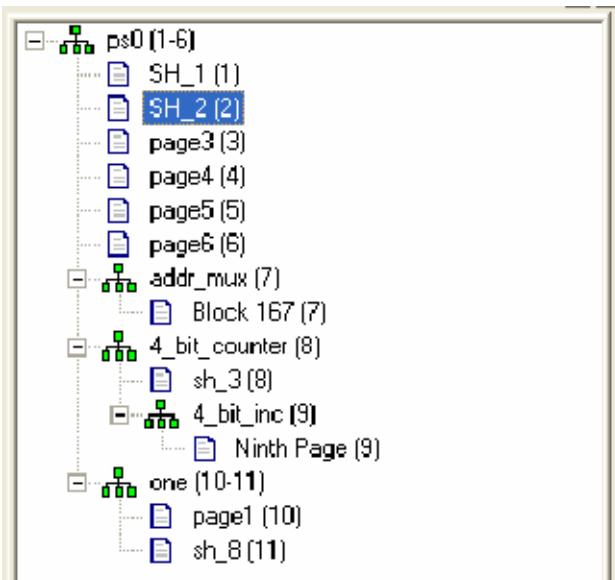
**Note:** Sheet names also appear in the *Go To Sheet* dialog box as well as the title bar of the main window.

**Note:** This feature is only available with the following licenses of Allegro Design Entry HDL:

- Allegro Design Authoring
- Allegro Design Entry HDL SI XL
- Allegro PCB Design HDL - GXL
- Allegro PCB Design HDL XL

Some of the salient features of the hierarchy viewer include the following:

- Page numbers/sheet names in the hierarchy viewer along with block names



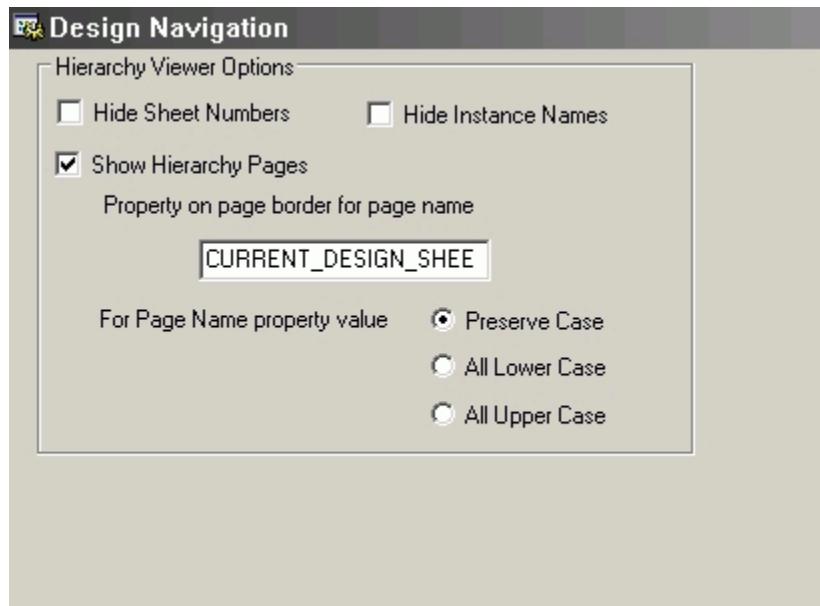
- Sheet name in title window



## Allegro Design Entry HDL User Guide

### Working with Designs

- Design Navigation page in the Design Entry HDL Options dialog box (*Tool – Options*)



## Setting Up Page Numbers

To show page names, you need to perform the following tasks:

1. Set up the `PAGE_NAME_PROP` directive to a *property name*.

The property name that you set will hold the Sheet Name text. The Sheet Name will be suffixed with *Sheet Number*, which will be unique across the hierarchy. Sheet number refers to numerals, such as 1, 2, 3 and so on.

By default, `PAGE_NAME_PROP` is set to "(Empty)", this implies that the page numbers will be visible in the hierarchy viewer.

Follow these steps to set the `PAGE_NAME_DIRECTIVE` from the Design Entry HDL dialog box:

- a. Choose *Tools – Options*.
- b. Select the Design Navigation tab.
- c. Type `<property name>` in the *Property on page border for page name* field.
- d. Select the casing for the page name. For example, *Preserve Case*.
- e. Click *Apply*.
- f. Click *Save All*.

Follow these steps to set the `PAGE_NAME_DIRECTIVE` from the command console:

- a. Type `set PAGE_NAME_DIRECTIVE <property name>`.
- b. Type `hier_write`.

The hierarchy is reloaded and will start showing the sheet names.

2. Specify the sheet name.

To set the page name by using the page border follow these steps:

- a. Right-click the page border.
- b. Select *Attribute*.
- c. Type `<value>` for the `<property name>`.
- d. Click *OK*.

To set the page name by using the hierarchy viewer follow these steps:

- a. Right-click the page.
  - b. Select Edit Page Name.  
The Page Name dialog box appears.
  - c. Type <page name>.
  - d. Click OK.
3. Save the design.

The page names are displayed in the hierarchy viewer.

## Sheet Name Input in New Designs

On saving the design, you are prompted to specify a sheet name for the pages that contain the Page Border symbols. In case the Property placeholder is not found the Property is created and attached to the cursor to be placed on the schematic. This behavior will be disabled when the Save command is called from scripts.

When a new page is created or added to the design, and you save the design, you are prompted for a Sheet Name for the pages that contain the Page Border symbols. In case the property placeholder is not found, the Property is created and placed on the origin of page border.

Additionally, the visibility is set to *None*, by default. If the placeholder is there on page border, its value is updated when you specify the page name.

## Using the Design Navigation tab

From the Design Navigation tab, you can perform the following tasks:

- Show/Hide the sheet names
- Show/Hide the instance names
- Show/Hide Hierarchy Pages
- Set the PAGE\_NAME\_PROP
- Specify the casing for the page name (lower, upper, or preserve)

## Performing Page Management Operations

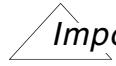
Using the page management options or the \_PAGE commands on a page or pages, you can insert, delete, or move pages in a schematic with ease. When you work on pages using the page management user interface or the \_PAGE commands, all page numbers are automatically adjusted.

For a detailed description of the corresponding console commands, refer to [PAGE commands](#) in *Allegro Design Entry HDL Reference Guide*.



### *Important*

Page management operations are only applicable to the pages of the currently open block.



### *Important*

Before you perform a page management operation, review [Points to Remember](#) on page 485.

## Inserting a Page

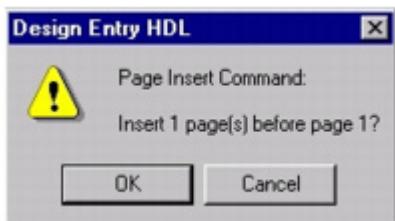
You either insert a single page or a set of pages in a schematic using the page management user interface. You can insert new pages before or between existing or [blank pages](#). All subsequent pages are renumbered automatically and you need not worry about renumbering them manually.

### Inserting a Single Page at the Current Location

You insert a single page at the current location in a schematic by using the *File – Edit Page/Symbol – Insert Page* menu command. To insert a single page at the current location:

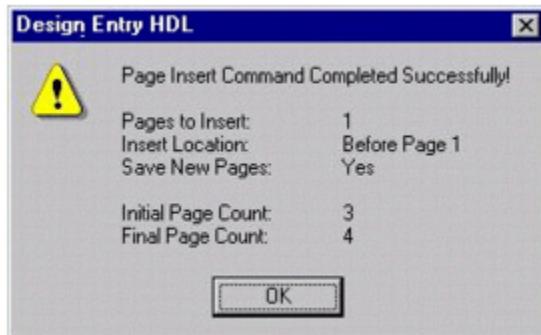
1. Choose *File – Edit Page/Symbol – Insert Page*.

A confirmation box is displayed.



2. Click *OK* to confirm.

A blank page is inserted and a success notification is displayed.



## Inserting a Set of Pages

You can also insert a page or a set of pages anywhere in a schematic. The maximum number of pages that you can insert in a single command is 250.

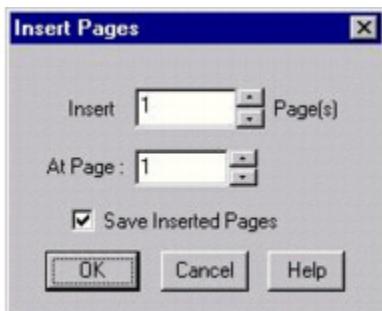
To insert a single page at the current location:

1. Choose *File – Edit Page/Symbol – Insert (n) Pages*.

## Allegro Design Entry HDL User Guide

### Working with Designs

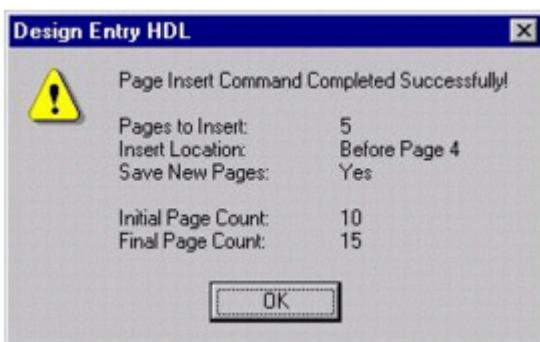
The Insert Pages dialog box is displayed



In this dialog box, you can specify the number of pages to insert and the location from where you want to insert the pages. Pages are always inserted before the current page at the target location.

2. To insert 5 pages, select 5 in the Insert spin box.
3. To insert pages before page 4, select 4 in the At Page spin box.
4. Click the *Save Inserted Pages* check box to create Page\* files for the pages to be inserted. If this option is unchecked, a page gap will be created in the schematic. Page gaps do not have corresponding Page\* files.
5. Click *OK*.

A set of 5 pages will be inserted before page 4. The current page 4 will become page 9. All the pages will be saved and the corresponding Page\* files will be created under the sch\_1 directory. Finally, a message will display a summary of the newly inserted pages.



**Note:** If you insert a new page at a location where there is already a page gap, the command adds an additional page and the size of the page gap is unchanged.

#### Inserting Pages beyond the End of the Schematic

Apart from inserting pages between existing pages, you can also insert a page beyond the end of the existing schematic. For example, to insert 2 pages before page 25 in a schematic with the page sequence, 1-3, 6-10, 12-15, perform the following steps:

1. Choose *File – Edit Page/Symbol – Insert (n) Pages.*
2. In the Insert Pages dialog box, select 2 in the Insert spin box.
3. Select 25 in the At Page spin box.

**Note:** The *Save Inserted Pages* option is of no value for this case as everything beyond the end of the schematic module is already blank.

Two pages will be inserted starting from page 25. There will be a 9 page gap between pages 15 and 25. The new page sequence will be 1-3, 6-10, 12-15, 25-26.

#### Inserting Page Gaps between two Pages

You can also add page gaps between two pages using the Page Insert dialog box. For example, to create a 3 page gap at page 8 in a schematic with the page sequence, 1-3, 6-10 12-15, perform the following steps:

1. Choose *File – Edit Page/Symbol – Insert (n) Pages.*
2. In the Insert Pages dialog box, select 3 in the Insert spin box.
3. Select 8 in the At Page spin box.
4. Deselect the *Save Inserted Pages* check box.

All pages, page 8 onwards, will be moved by three places to accommodate the new pages. As a result, the existing page 8 will become page 11. Three blank pages, 8,9,10 will be inserted:

- If you click the Next Page or Previous Page buttons to move to other pages, you will be prompted to save page 8. If you choose to save this page, the corresponding *Page8.\** files will be created and the page number count will increase by one. The new page sequence will be 1-3, 6-8, 11-13, 15-18.

## Allegro Design Entry HDL User Guide

### Working with Designs

- However, if you choose not to save page 8, the result would be a 3 page gap in the schematic at the point of insert. The new page sequence will be 1-3, 6-7, 11-13, 15-18



## Deleting a Page

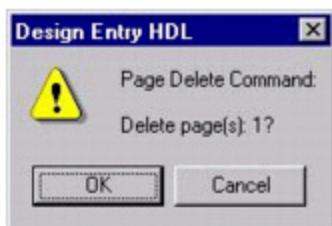
You can delete existent or blank (non existent) pages from a schematic using the Delete Pages dialog box. When you delete a page, the pages following the page to be deleted are moved in without leaving a page gap.

### Deleting the Current Page

You delete the current page in a schematic by using the File – *Edit Page/Symbol* – Delete Page menu command. To delete the current page, perform the following steps:

1. Choose *File – Edit Page/Symbol – Delete Page*.

A confirmation box is displayed.

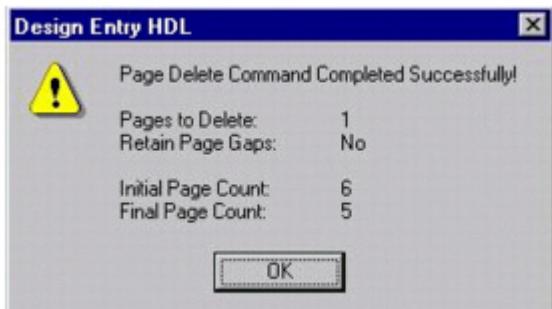


2. Click OK to confirm.

## Allegro Design Entry HDL User Guide

### Working with Designs

The current page is deleted and a success notification is displayed



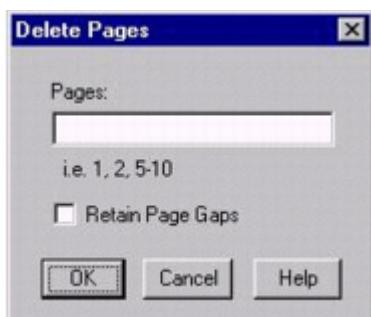
### Deleting a Set of Pages

You can also delete a set of pages. You can specify an explicit number or a range of numbers to be deleted. For example, 1,2,3,5-7, is a valid range. Spaces are not allowed between page numbers.

To delete a set of pages, perform the following steps:

1. Choose *File – Edit Page/Symbol – Delete (n) Pages.*

The Delete Pages dialog box is displayed. In this dialog box, you specify a range of pages to delete.



2. To delete pages in the range 6-8, type *6-8* in the Pages text box.
3. Select the *Retain Page Gaps* check box if you want to retain the physical page numbers of the pages following the pages being deleted. This will create page gaps for the pages you delete. For this example, leave this option unchecked.
4. Click *OK*.

## Allegro Design Entry HDL User Guide

### Working with Designs

Pages 6, 7, and 8 will be deleted. The physical page numbers of all the pages following page 8 will be moved in by 3

### Removing Page Gaps

You can also remove page gaps or reduce the size of the page gap in a schematic using the Delete Pages dialog box. For example, to delete page 4 from a schematic with page sequence, 1-3, 6-10, 12-15, perform the following steps:

1. Choose *File – Edit Page/Symbol – Delete (n) Pages*.
2. Type 4 in the Pages text box.
3. Click *OK*.

The size of the 2-page gap between pages 3 and 6 will be reduced by 1. Consequently, the pages after page 4 will move in by 1. The new page sequence will be 1-3, 5-9, 11-14.

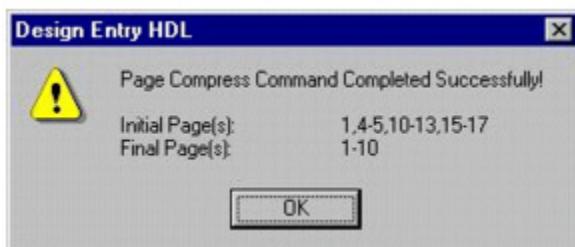
**Note:** If you try to delete a page gap, which is out of the page range of the schematic, it will result in an error:

### Compressing Pages

You remove all the page gaps in a schematic by using the *File – Edit Page/Symbol – Compress Pages* menu command.

- To remove all the page gaps in a schematic, choose *File – Edit Page/Symbol – Compress Pages*.

All the page gaps will be removed and a success notification will be displayed.



### Moving a Page

You can move a page or a set of pages to existent or non-existent locations using the *File – Edit Page/Symbol – Move Pages* menu command. Using the new Move Pages dialog box

you can move pages between existing pages of a schematic. You can also move non-contiguous pages to contiguous locations

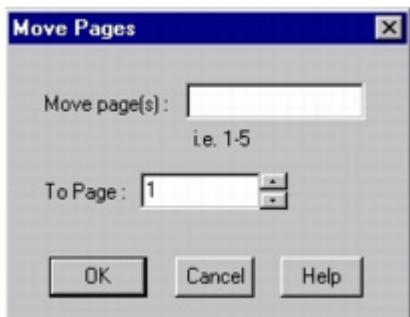
The move pages option works as a drag-and-drop functionality in a GUI, and does not create any page gaps for the moved pages. As a result, the total page count remains the same. However, gaps existing in page numbers are retained.

#### Moving a Page before an Existing Page

Using the Move Pages dialog box you can move a set of pages in a schematic. For example, to move pages 4-8 between pages 15 and 20 in a schematic with page sequence 1-30, perform the following steps:

1. Choose *File – Edit Page/Symbol – Move Pages*.

The Move Pages dialog box is displayed.



2. Specify the page or the set of pages to be moved in the *Move Pages* text box. For example, type *4-8*.

In the *To Page* spin box, you specify the location in the schematic where the pages will be moved. For example, if you specify 5 the moved pages will precede the current page 5. To move pages to the end, use last physical page + 1.

3. Type 15 in the To Page spin box as the page location where you want to move the pages.

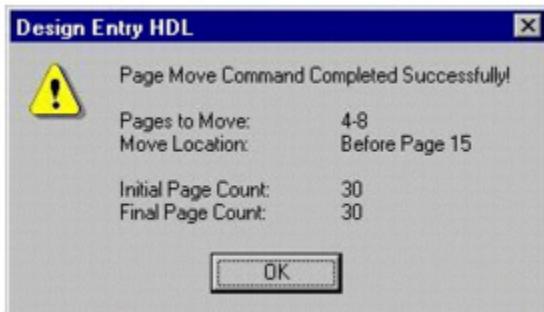
4. Click *OK*.

- Pages 15-30 will be moved out 5 pages to make space for the 5 pages being moved.
- Pages 4-8 will be moved to the blank slots created in step 1.
- The 5 page gap (4-8) created in step 2 by moving all pages in by 5 will be removed.

## Allegro Design Entry HDL User Guide

### Working with Designs

- All existing page gaps are maintained, but the page gaps created in the process of the move command are closed.



### Moving a Set of Pages outside the Current Range of Pages

Consider a scenario where you have the following pages in a schematic: 1-3, 6-10, 12-15. Perform the following steps:

1. Choose *File – Edit Page/Symbol – Move Pages*.
2. Type 6-9 in the Move Pages box.
3. Select 17 from the To Page spin box.
4. Click *OK*.

Pages 6-9 will be moved before the blank page 17 inserting pages backwards from page 16. Therefore, initial page 9 will be moved to page 16, the initial page 8 will be moved to page 15, and so on and so forth. The new page sequence will be 1-3, 6, 8-11, 13-16.

### Moving Non-Contiguous Pages to Contiguous Locations

Consider a scenario where you have the following pages in a schematic: 1-3, 6-10, 12-15. If you move pages 3,7,9 to page 15, pages 3, 7, and 9 will move to pages 12, 13, and 14, respectively, and the other pages will be adjusted accordingly. The confirmation message will be suppressed and the pages will be moved without prompting you for confirmation. The new page sequence will be 1-2, 5-7, 9-15.

## Definitions

### blank page

A blank page indicates that the page does not exist in the schematic; the corresponding Page\* files do not exist for the page. For example, in a schematic with pages 1,2,3,6, pages 4 and 5 are blank pages which form a page gap. In the sch\_1 view, Page4.\* and Page5.\* files will not exist for the two blank pages.

The *Previous Page* and *Next Page* buttons in Design Entry HDL skip over blank pages.

### page gap

A sequence of blank pages. For example, a schematic with pages 1,2,3,4,5 contains no page gaps, whereas a schematic with pages 1,2,3,6 contains a single page gap with size equal to two - blank pages 4-5.

## Points to Remember

This section describes a few important points you should remember when using the page management commands on a schematic:

### Performing Page Operations on a Schematic with Missing .csa or .csb Files

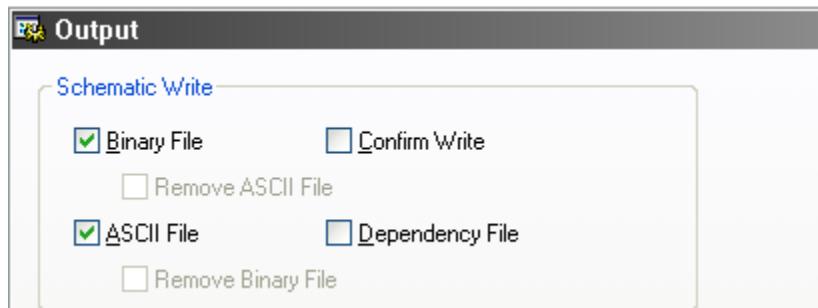
Page management operations will not work unless both the .csa (ASCII) and the .csb (binary) files exist for all the pages in the schematic. In case, you have either of the two files missing for any page, you will get an error message when you perform any page management operation.



## Allegro Design Entry HDL User Guide

### Working with Designs

To avoid this error, select the *Binary File* and *ASCII File* check boxes in the Design Entry HDL Options dialog box and save the schematic before performing any page management operation.



### Running the Hier\_Write Command

When you copy or import pages into a design from another design, you must run the *hier\_write* command on the design performing page management operations. When you run the *hier\_write* command, you must ensure that the *Binary File* and *ASCII File* check boxes in the Design Entry HDL Options dialog box are selected. You must also run the *hier\_write* command before performing page management operations on a schematic created in an earlier version of Design Entry HDL.

### Saving Affected Pages before Running Page Management Operations

If you perform a page management operation which impacts a page that has unsaved changes in it, an error is generated and the page management operation is terminated. As a result, before you perform a page management operation, save all the affected pages with unsaved changes.

For example, if you delete pages 3-4 in a schematic that has pages 3, 4 and 5 with unsaved changes, an error message will be displayed. You will not be able to perform any page management operation affecting these pages unless you save them.

Page 5 is also flagged as it would have to be moved to close the gap created if 3 and 4 were deleted. If the *Retain Page Gaps* option is checked, only pages 3 and 4 would be flagged since 5 would not be moved to close the gaps left by the delete operation.

However, if the page management operation does not affect the pages with unsaved changes, no errors are generated and the command is executed successfully.

For example, if page 3 of a schematic is modified and you insert 2 pages before page 7 without saving page 3, no errors are generated as this operation does not affect page 3.

## Performing Page Management Operations in a Read-only Design

If you try to perform a page management operation on a read-only design, an error message will be displayed and the operation will terminate.

## Performing Page Operations on a Locked Page

If a `<page>.lck` file exists for a page, you cannot run any page management operation on that page or any other page affecting the page. For example, if there is a lock on page 5, and you try to delete page 4, an error will be generated as page 5 would have to be moved to close the gap created by deleting page 4.

# Displaying and Working with Schematic Page Numbers

Design Entry HDL allows you to display page numbers on a schematic page using custom text variables for page numbers. You can modify the custom text to change the way in which page numbers are displayed. You can also renumber the pages in a design. For more information on displaying and working with schematic page numbers, see the following sections:

- [Displaying Page Numbers in a Schematic](#) on page 487
- [Modifying Custom Text for Page Numbers](#) on page 491
- [Updating Custom Text Variables for Page Numbers](#) on page 492
- [Managing Schematic Pages](#) on page 492

## Displaying Page Numbers in a Schematic

Design Entry HDL allows you to display page numbers on a schematic page using custom text variables for page numbers. When you plot a schematic page, the page number of the schematic page is plotted only if you have added the custom text variables on the schematic page.

Cadence recommends that you add custom text variables for page numbers on the schematic pages. This allows you to easily refer to a page in Design Entry HDL against a plotted page or a cross-reference report using the *File – Edit Page/Symbol – Go To* command. For more information, see [How do I go to a specific page in a design?](#) on page 47.

- If you plot a schematic page on which custom text variables for page numbers are not added, the plotted page will not contain the page number. If you later want to refer to a

## Allegro Design Entry HDL User Guide

### Working with Designs

---

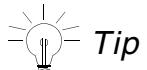
page in Design Entry HDL against its plotted page, you will not know which page to open in Design Entry HDL.

- When you perform cross-referencing on a design, the cross-reference reports display the schematic page numbers. It will be easier to refer to a page in Design Entry HDL with the cross-reference report if the page number is displayed on the schematic page.

**Note:** The cross-reference reports will contain the page number information even if you have not added the custom text variables on the schematic page.

The following custom text variables allow you to display page numbers on a schematic page. For more information on custom text variables, see [Working with Custom Text](#) on page 377.

- CON\_PAGE\_NUM
- CON\_TOTAL\_PAGES
- CURRENT DESIGN SHEET
- TOTAL DESIGN SHEETS



*Tip*  
To use the Quickplace feature of PCB Editor for placing selected unplaced components on the board, you can add the CON\_PAGE\_NUM and CON\_PARENT\_CNAME variables to your design. For details, refer to the quickplace command in *PCB Systems Physical Command Reference*.

#### ***The CON\_PAGE\_NUM and CON\_TOTAL\_PAGES Custom Text Variables***

The CON\_PAGE\_NUM and CON\_TOTAL\_PAGES variables allow you to display the page numbering information for pages in a cell. For example, if the root design A in a hierarchical design has three pages, and sub design B has four pages, the value of:

- CON\_PAGE\_NUM variable will be 1 for the first page of the root design A and 3 for the last page in the root design A. The value of the CON\_PAGE\_NUM variable for the first page of sub design B will also be 1 and the value of the variable for the last page of sub design B will be 4. If you are in the third page of the sub design B, the value of the CON\_PAGE\_NUM will be 3 and not 6. This means that the CON\_PAGE\_NUM variable does not take into account the pages in all the cells in a hierarchical design.
- CON\_TOTAL\_PAGES variable will be 3 for the root design A and 4 for the sub design B. This means that the CON\_TOTAL\_PAGES variable does not take into account the pages in all the cells in a hierarchical design.

## Allegro Design Entry HDL User Guide

### Working with Designs

The value of CON\_TOTAL\_PAGES will be the value of the highest page number you have assigned for a page in the cell. For example, if you have four pages in a cell with numbers 1, 2, 3, and 6, the value of CON\_TOTAL\_PAGES is 6.

#### ***The CURRENT DESIGN SHEET and TOTAL DESIGN SHEETS Custom Text Variables***

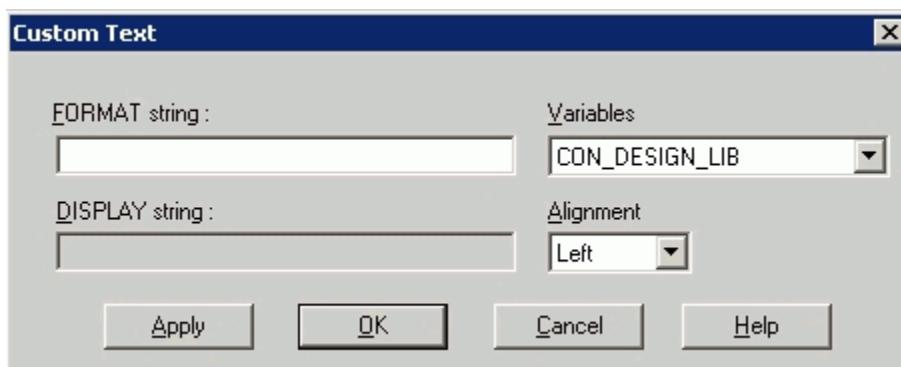
The CURRENT DESIGN SHEET and TOTAL DESIGN SHEETS variables allow you to display the page numbering information for pages in a hierarchical design. For example, if the root design A in a hierarchical design has three pages, and sub design B has four pages, the value of:

- CURRENT DESIGN SHEET variable will be 1 for the first page of the root design A and 3 for the last page in the root design A. The value of the CURRENT DESIGN SHEET variable for the first page of sub design B will be 4 and the value of the variable for the last page of sub design B will be 7.
- The TOTAL DESIGN SHEETS variable will be 7. The TOTAL DESIGN SHEETS variable takes into account the actual number of pages in a hierarchical design. Even if the highest page number assigned to a page in a hierarchical design is greater than the actual number of pages in the design, the TOTAL DESIGN SHEETS variable will display only the actual number of pages in the design.

#### ***To add the custom text variables for page numbers***

1. Choose Text – Custom Text.

The Custom Text dialog box appears.



2. Select the following variables from the *Variables* drop-down list as required:

- CON\_PAGE\_NUM
- CON\_TOTAL\_PAGES

## Allegro Design Entry HDL User Guide

### Working with Designs

---

- CURRENT\_DESIGN\_SHEET
- TOTAL\_DESIGN\_SHEETS

The variable is displayed in the *FORMAT string* field. The *Display string* field displays the current value of the variable. For example, if you are adding the CURRENT\_DESIGN\_SHEET variable on the 15th page of a hierarchical design, the *Display string* field displays the value 15.

#### 3. Edit the format string for the custom text as required.

For example, if you have selected the CURRENT\_DESIGN\_SHEET, the *FORMAT string* field displays <CURRENT\_DESIGN\_SHEET> and the *Display string* field displays, say 15.

If you want to display the page number as PAGE 15 on the schematic page, change the text in the *FORMAT string* field to Page <CURRENT\_DESIGN\_SHEET>. The *Display string* field now displays PAGE 15.

Similarly, if you have selected both the CURRENT\_DESIGN\_SHEET and TOTAL\_DESIGN\_SHEETS variables, and want to display running total page numbers, such as Page 15 of 20, change the text in the *FORMAT string* field to:

Page <CURRENT\_DESIGN\_SHEET> of <TOTAL\_DESIGN\_SHEETS>

The *Display string* field displays:

PAGE 15 of 20

**Note:** If you want to add an environment variable to the format string, precede it with a \$ sign. The current value of the environment variable is displayed in the *Display string* field.

#### 4. Click *OK*.

The custom text for the page number is attached to the cursor.

#### 5. Click on an object to attach the custom text to it.

**Note:** You can add multiple custom text to the same object on the schematic.

#### 6. Click again to place the custom text on the schematic.

The page number is displayed in the schematic.

#### 7. Right click and choose *Done*.

**Note:** Lie with properties, custom text can be deleted, moved, copied, and rotated.

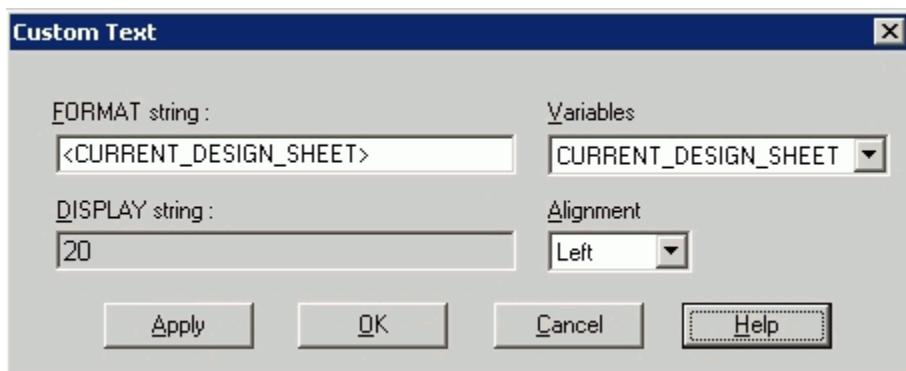
## Modifying Custom Text for Page Numbers

Design Entry HDL allows you to modify the custom text variables for page numbers to change the way in which the page numbers are displayed in the page number border.

For example, if you have added only the CURRENT\_DESIGN\_SHEET custom text variable in the 20th page of a hierarchical design that has 25 pages, the schematic page will display the page number as 20. If you want to display running total page numbers such as Page 20 of 25, do the following:

1. Choose *Text – Change*.
2. Click on the page number custom text 20 that is displayed on the schematic page.

The *Custom Text* dialog box appears.



3. Select the TOTAL\_DESIGN\_SHEETS variable from the *Variables* drop-down list.
4. Modify the text in the *FORMAT string* field to:

Page <CURRENT\_DESIGN\_SHEET> of <TOTAL\_DESIGN\_SHEETS>

The *Display string* field displays:

PAGE 15 of 20

5. Click *OK*.

The page numbering information on the schematic changes to PAGE 15 of 20.

**Note:** The page number format is always an integer. Design Entry HDL does not allow you to change the page number format to i, I, a, and so on.

## Updating Custom Text Variables for Page Numbers

The CON\_PAGE\_NUM and CON\_TOTAL\_PAGES custom text variables for page numbers are automatically updated whenever any changes are made in the design. However, you may need to update the CURRENT\_DESIGN\_SHEET and TOTAL\_DESIGN\_SHEETS custom text variables for page numbers to ensure that the schematic page displays the correct page number. This update needs to be done in the following cases:

- If you have modified the design by adding or deleting pages or blocks.
- When you add custom text variables for page numbers on a schematic page, the name of the variable is substituted by the page number. If the schematic page continues to display the variable name instead of the page number, you need to update the variable. For example, if you add the CURRENT\_DESIGN\_SHEET custom text variable in the 20th page in a hierarchical design, the schematic page may display <CURRENT\_DESIGN\_SHEET> instead of 20.
- When you renumber schematic pages using the page renumbering commands, the schematic pages may not display the correct page number. For more information on renumbering pages, see [Managing Schematic Pages](#) on page 492.

**Note:** When you perform module ordering, cross-referencing, or plotting of the design, the custom text variables are updated automatically.

### **To update the custom text variables for page numbers**

- Choose *Text – Update Sheet Variables*.

The custom text variables for page numbers on all pages in the design are updated to display the correct page number.

You can also use the `updatesheetvars` console command to update the custom text variables for page numbers. For more information, see [updatesheetvars](#) in *Allegro Design Entry HDL Reference Guide*.

## Managing Schematic Pages

In multiple page designs, you may have non-contiguous pages in the design. You may want to remove those pages so as to make your design contiguous. There can also be situations when you would want to add a new page in the middle of a design. You can use the page management commands of Design Entry HDL to perform these tasks. The Page management commands let you change the order in which pages are arranged in the design. You can interchange two pages, move a page from one place to another, or delete a page.

For more information on page renumbering commands, see [Page Management Commands](#) on page 494.

In the context of page renumbering, there are two types of page numbers, physical and logical.

#### **Physical Page Numbers**

Physical page numbers are the ones that you see on the Design Entry HDL title bar. For example, if `CLOCK.SCH.1.4` is displayed on the Design Entry HDL title bar, `CLOCK` indicates the name of the cell, `SCH` indicates that the view type is schematic, 1 indicates the version number of the view and 4 indicates the physical page number of the schematic page.

The physical page numbers change when you renumber schematic pages.

#### **Logical Page Numbers**

When you create a new schematic page, the logical and physical page numbers are the same. Design Entry HDL assigns the logical page number for a schematic page using the directive `set page_number Pn` to the `page*` files. The logical page number is the original page number for the schematic page. When you renumber a schematic page, the physical page number changes but the logical page number does not change. Design Entry HDL keeps track of the logical page numbers of pages that have been renumbered by reading the `set page_number Pn` in the `page*` files.

For example, when you create a three page design, the logical and physical page numbers of the schematic pages will be 1, 2 and 3. While cross-probing or globally locating objects, canonical names are shown with the logical page numbers. For example, the canonical name `@top_lib.top.(sch_1):page2_i5` displayed in the *Global Find* dialog box means that the component that has the `PATH=i5` property is located in the logical page 2 in the `sch_1` view of the cell `top` in the library named `top_lib`.

In the above design, if you swap page 2 with page 3, and perform a global find for the component that originally existed on page 2 of the design, but now exists on page 3 of the design, the *Global Find* dialog box continues to display the canonical name for the component as `@top_lib.top.(sch_1):page2_i5`. This means that the logical page number has not changed even after you swapped the pages in the design. If you click on the canonical name in the *Global Find* dialog box, Design Entry HDL zooms in to display the object on page 3 (the physical page) of the design.

#### Renumbering Pages

If you are renumbering the pages of a design that is not of the current release, you need to first save the design using the `hier_write` console command. Writing the design once adds the directive `set page_number Pn` to the `page*` files. The number *Pn* is called the logical page number. This directive is required for Design Entry HDL to keep track of the original page numbers of pages that have been renumbered.

If you have renumbered the pages of a design but have not changed the schematic, you do not need to save the design. The change is visible in all the tools.

An example of the need to use the page management feature would be a design that has 12 pages of which page numbers 5 and 6 are blank. You may want to remove the blank pages and collapse the design.

You can delete the blank pages using the PAGEDelete command and the other pages will be renumbered automatically.

You may need to add a new page somewhere in the middle of a design. You can use the page management commands to insert new pages and the other pages will be renumbered accordingly.

For example, if you have pages 1 to 10 and you want to add another page in between and number it 8, you can use the PAGEInsert command (command.fm) command. The new page will be inserted and the other pages will be moved and renumbered automatically.

#### Page Management Commands

You can use the following console commands to manage pages in a design:

- PAGEInsert
- PAGEDelete
- PAGECompress
- PAGEMove
- page move
- page swap
- page delete
- page reset
- page forcereset

You cannot run these commands in a design if a page in the same design is opened by another user who has write permissions. Design Entry HDL displays the following error message in the console window if you run these commands when the design is opened by another user:

```
This design is being simultaneously edited by multiple users. Ignoring page command.
```

**Note:** If you are working on a block used in the design and another user is working on another block, you can run the page management commands on the pages in the block.



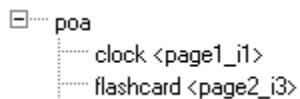
Page management commands, like all other commands implemented using Design Entry HDL SKILL, cannot be launched from scripts.

### Page Renumbering, Module Ordering and Hierarchical Plotting

When you run the page management commands, the order of the pages in the design hierarchy tree in the hierarchy viewer and the Plot dialog box does not change automatically. You must manually change the order of the pages in the hierarchy viewer window. For example, suppose that you have a block `POA` with:

- Block `clock` instantiated on the page 1 and
- Block `flashcard` instantiated on page 2.

The hierarchy viewer window displays the following design hierarchy tree:

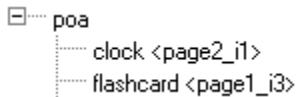


The *Plot* dialog box displays the following design hierarchy tree:



Now swap page 1 with page 2 using the `page_swap 1 2` Design Entry HDL console command.

### Hierarchy Tree in the Hierarchy Viewer after Page Swap



Note that though the page numbers in the hierarchy tree in the hierarchy viewer have changed, the order of the pages has not changed. You must manually change the order of the pages in the hierarchy viewer window, as shown in [Figure](#) on page 496.

### Hierarchy Tree in Plot Dialog Box after Page Swap



Note that the page number for the block `clock` has not changed from 3 to 4 and the page number for the block `flashcard` has not changed from 4 to 3 in the *Plot* dialog box. This results in the pages for the blocks not being plotted in the expected order. In this example, you would expect the pages in the design to be plotted in the following order:

1. Page 1 of block POA that has the block `clock` instantiated on it
2. Page 1 of block POA that has the block `flashcard` instantiated on it
3. Pages of block `clock`
4. Pages of block `flashcard`

However, the pages are plotted in the following order:

1. Page 1 of block POA that has the block `clock` instantiated on it
2. Page 1 of block POA that has the block `flashcard` instantiated on it
3. Pages of block `clock`
4. Pages of block `flashcard`

If you want the pages for the block `flashcard` to be plotted before the pages for the block `clock`, you must change the order of the pages in the hierarchy viewer as shown in the figure below:



# Importing Designs

## Overview

Design Entry HDL includes support for reusing designs by importing schematic sheets and blocks across projects in order to enhance the productivity of designers. This feature saves you the trouble of copying physical page files through system commands, which involves additional work like resolving page conflicts, and so on.

Using the Import Design feature, you can do the following:

- Browse and select a new project .cpm file from within Design Entry HDL.
- View the sheets to be imported in a new Design Entry HDL window in read-only mode before importing.
- Import schematic sheets including sheets with hierarchical blocks.
- Import blocks.
- Add lower-level blocks to libraries used in an existing project.
- Re-import read-only blocks into the destination block.

## Importing a Design Sheet

You can import one or more schematic design sheets to the currently open project from another project.

**Note:** Before you start importing sheets and blocks, make sure that the source and destination design cds.lib point to the same libraries. All the libraries being used in the source project should also be accessible in the destination design through cds.lib. Otherwise, there will be placeholders on the schematic for the cells which are not found in the destination design.

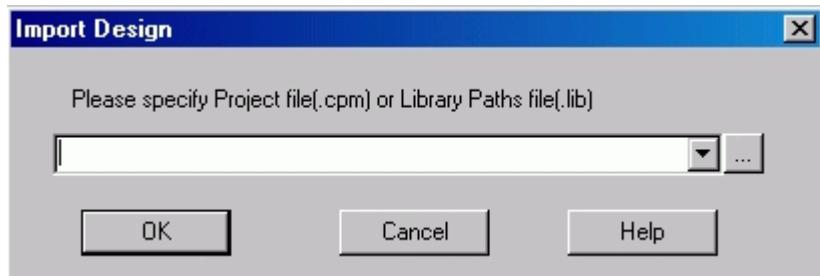
To import a design sheet, do the following:

1. Choose *File – Import Design*.

## Allegro Design Entry HDL User Guide

### Working with Designs

The Import Design dialog box is displayed. You are prompted to specify the project.cpm file of the project from which you want to import the design sheets.



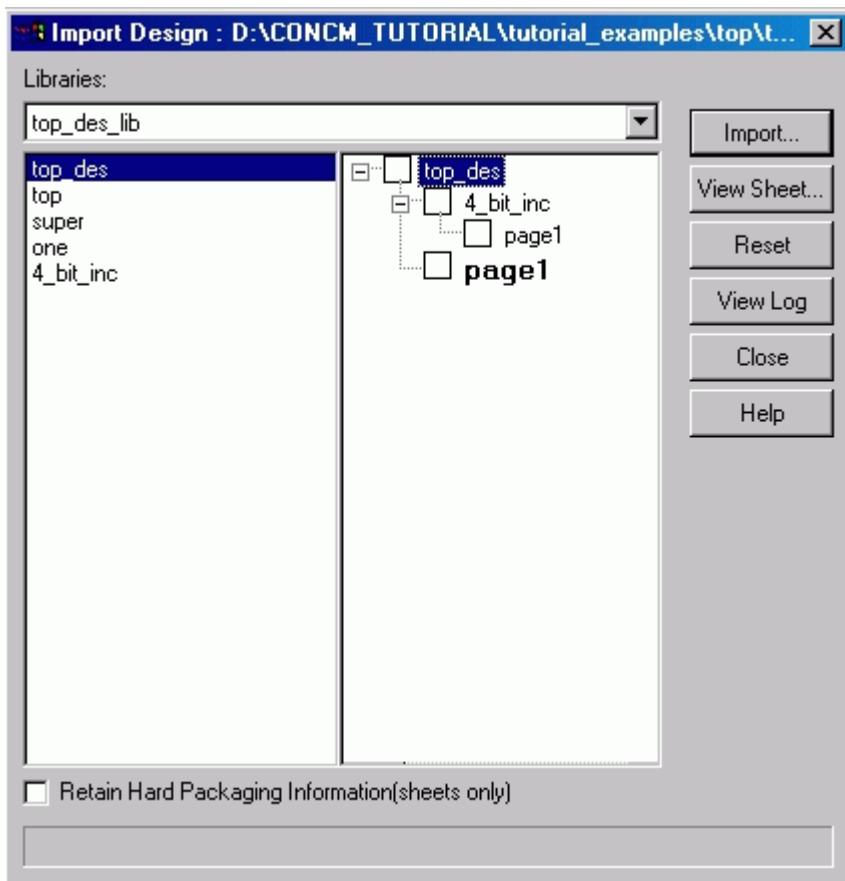
2. Specify the path to the appropriate project file (.cpm) or the cds.lib file in the *Project File (.cpm)/Library Paths file (.lib)* field. You can also browse to the required file.

**Note:** The projects you select are displayed in the *Projects* list box. Only absolute paths are displayed in the list box. If you have assigned a relative path for a file, its absolute path appears in the list box. You can place the preferred\_projects.txt file in the home or <home>/cdssetup or at the project level.

## Allegro Design Entry HDL User Guide

### Working with Designs

3. Click **OK**.



The Import Design dialog box displays the hierarchy of the project. The root library is selected by default. All the sheets and blocks in the selected design are displayed.

You can select the sheet or sheets that you want to import into the current project. You also have the option to view a sheet before importing it. When you view a sheet, Allegro Design Entry HDL opens in read-only mode.

4. Select the library from which you would like to import a block in the *Libraries* field.

All blocks and pages in the selected library appear. The blocks and top design sheets are listed separately. The pages that appear in the bold letters indicate that they have blocks. When you move the mouse over a bold page, a tool tip appears listing all the blocks in the selected page.

**Note:** If you select a library that does not contain any blocks, then the Import Design dialog box displays blank columns. A message prompting non-availability of blocks in the library appears.

5. Select the required option:

## Allegro Design Entry HDL User Guide

### Working with Designs

---

- A sheet or multiple sheets—You can select multiple sheets by clicking in multiple check boxes.

**Note:** You can select multiple pages of the same block but cannot select pages across blocks.

- A block—if you select a block all pages within it are selected.
- A bold page—if you select a bold page, blocks within that page will automatically be imported.

#### 6. To view the selected sheet before import, click *View Sheet*.

**Note:** You can also select *View Sheet*, *Import* or *Reset* options for the selected sheets from the context-sensitive menu in Windows. You can use the View Sheet, Import and Reset buttons in the Design Import dialog box to perform operations on different sheets.

The hierarchy of the project is displayed in a read-only instance of Design Entry HDL and first sheet of the project appears in the Design Entry HDL canvas as a viewer. Note that there are no toolbars or the console command window. Most of the menu items are also grayed out. You can perform the following activities in the read-only viewer:

- View the hierarchy of the project to select a specific sheet
- Descend to a particular block/sheet.

#### 7. Select the *Retain Hard Packaging Information (sheets only)* check box to import hard package properties.

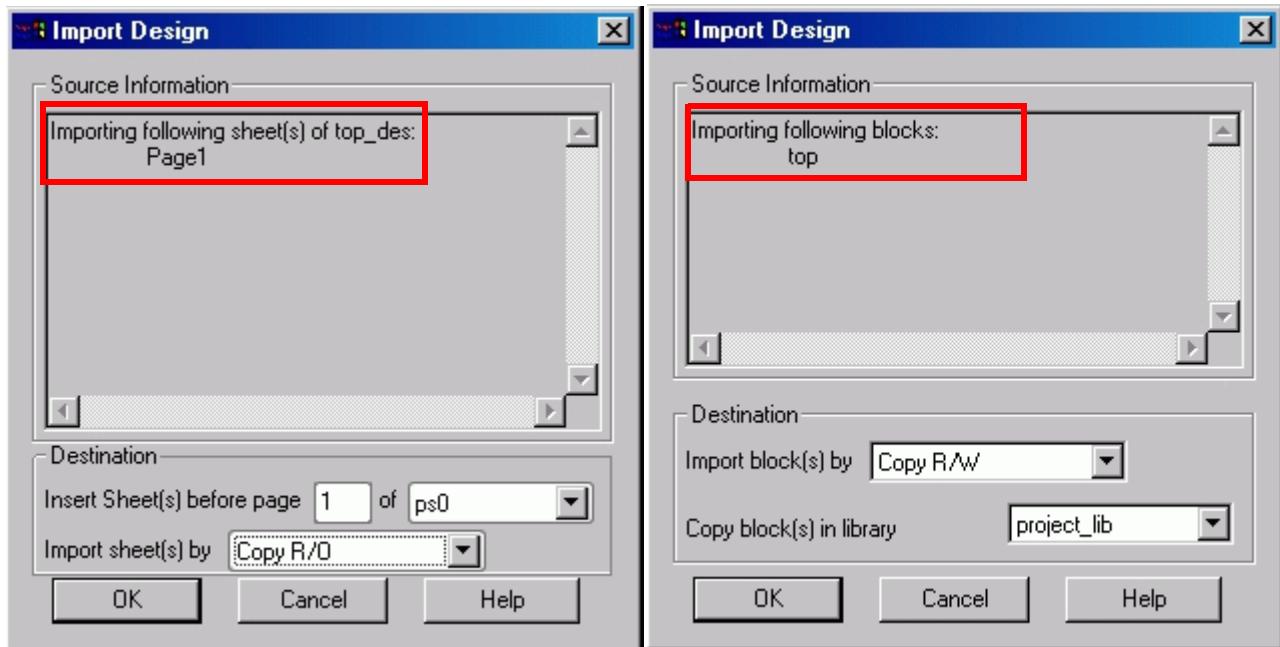
**Note:** This option is used to retain information in the pages. Context packaging data or packaging data not backannotated is not preserved while importing sheets.

#### 8. To import the selected sheets, click *Import*.

## Allegro Design Entry HDL User Guide

### Working with Designs

The *Import Design: Source Information* dialog box appears. The user interface of this dialog box changes based on whether you select a sheet or a block for importing.



To import, do the following:

- Sheets—Specify the page location and design name where you want to insert the imported sheet in the *Insert Sheet(s) at page* text box. Specify whether the sheet should be imported in read-only (R/O) mode or read-write (R/W) mode in the *Import sheet(s) by* field.

**Note:** If you want to reimport a read-only sheet, first delete the sheet in your target design then reimport the sheet from your source design.

- Blocks—Specify whether the blocks should be imported in read-only (R/O) mode or read-write (R/W) mode in the *Import Blocks by* field and specify the library where you want to copy the imported block in the *Copy block(s) in library* field.
- Sheets containing blocks (bold pages)—For sheets, specify the page location and the design name where you want to insert the sheet in the *Insert Sheet(s) at page* text box. For blocks, specify whether the blocks should be imported in Read/Only (R/O) mode or Read-Write (R/W) mode in the *Import Blocks by* field and specify the library where you want to copy the imported block in the *Copy block(s) in library* field.

**Note:** You cannot import blocks in worklib with read-only rights. Use a separate folder for importing read-only blocks.

#### 9. Click *OK*.

**Note:** You might see a warning for missing primitives if the lib-cells of the sheet being imported are not found in the library accessible to the current design. If the lib-cell is found in the reflib referenced in the current design, it will add that library in the .cpm file and the component is placed on the schematic. This warning also appears if the cell is available in the reflib but is found under some other library name. However, Design Entry HDL places the cell from the library accessible to the current design.

If there are signal name clashes in source and target design, the Import Design: Signal Name Clash dialog box appears.



You can change signal names in one of the following ways:

- Edit a signal name in the *New Signal Name* field.
- Select the *New Signal Name* field for a few signals and add a suffix or prefix to them. For this, add text in the *Process* field and click the *Add Prefix* or the *Add Suffix* button.

#### 10. Click *OK* to process signal name changes.

As soon as the sheets are imported, the read-only Design Entry HDL viewer closes and the Import Design dialog box appears. You can now import more blocks for which you can reset the selection by clicking the *Reset* button or click *Close* to complete the operation.

**Note:** When you import a multi-level block, its low level blocks are also imported and

stored in the same folder with the same rights (R/O or R/W).

11. To view the log file report for the design import operation, click *View Log*.

12. Click *Close* to close the Import design dialog box.

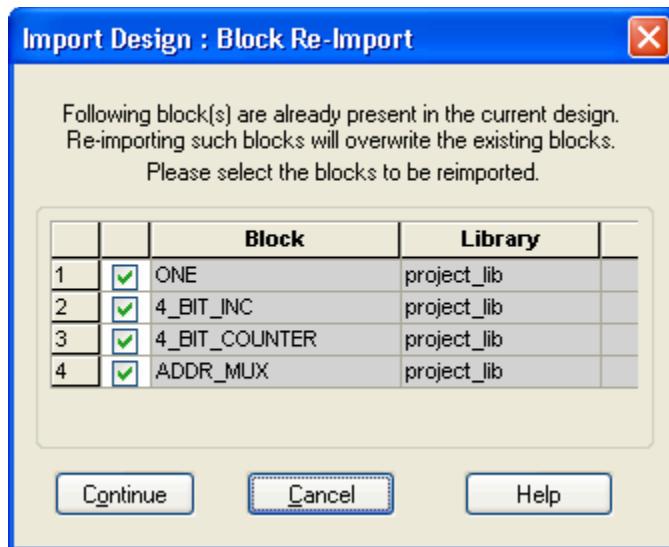
**Note:** If you have cross-referenced pages at the end of a schematic, you must save such pages in Design Entry HDL before you import a sheet. Otherwise, you will not be able to import sheets into the schematic. This happens because .csa files do not exist for the pages created by Crefer.

**Note:** Context data is not imported when importing a sheet. Only the schematic pages (.csb) are copied on import sheet. Packaging data is not imported until it is backannotated in the source design.

For information on how to retain module ordering while importing blocks, see [Retaining Module Ordering While Importing Blocks](#) on page 471.

## Re-importing a Block

If the sheets you select for re-importing have blocks that were already imported into the design, a dialog box appears indicating the presence of those blocks in the selected design. You can click the *Continue* button to re-import the blocks or *Cancel* to discontinue the block re-import.



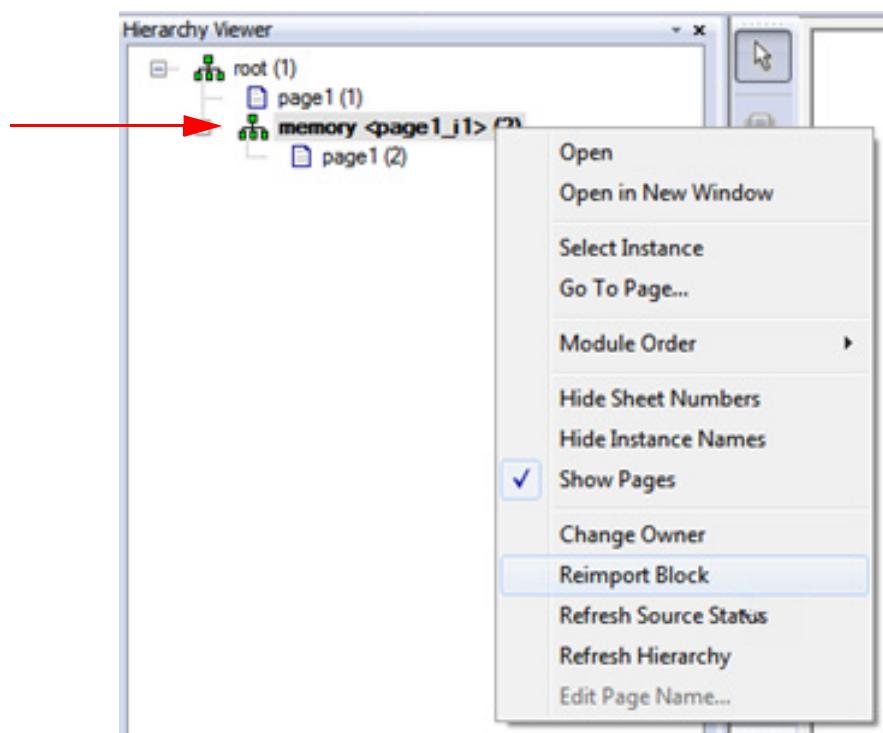
**Note:** While re-importing blocks, pages that contain root blocks which are being re-imported are saved. If any page being written is read-only or locked, the page is not saved. A summary of such pages is written to the design import log and the *Block Import* status is set to *Block Import Completed with Warnings*.

**Note:** A cell with a schematic view (sch\_n) that contains the pc.db file is considered a block and not a primitive. Such cells appear as blocks in the drop-down list.

## Re-importing Read-only Blocks

DE-HDL provides an option to reimport read-only blocks. This allows you to re-import a block whose source block has changed in another design.

DE-HDL stores the location from where the source block was imported, and periodically checks the source block for changes. If the source block, for example, MID, has changed, the same MID block in the destination design is displayed in bold.



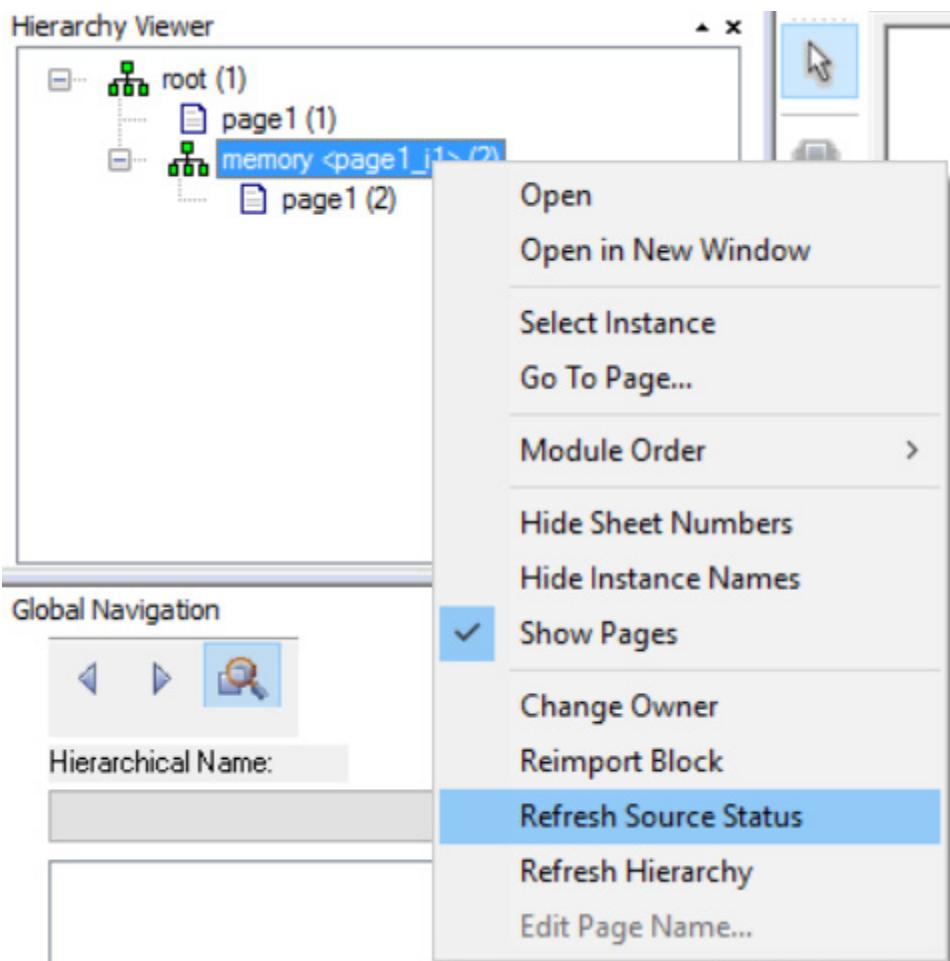
To enable the reimport of blocks as read-only blocks, set the ALLOW\_TRIGGER\_REIMPORT directive to 'ON' in the START\_CONCEPTHDL section of the .cpm file.

You can also specify a reimport timer using the TRIGGER\_REIMPORT\_TIMER directive. The default unit of measure for the timer is minutes. This cannot be changed.

## Allegro Design Entry HDL User Guide

### Working with Designs

Depending on the number of minutes you have defined in this directive, DE-HDL checks the source block for changes. You could also switch this directive off, and when required, use the *Refresh Source Status* option in the context menu.



If you want specific views in the source block to be ignored, for example, physical, use the REIMPORT\_IGNORE\_VIEWS directive as illustrated:

```
REIMPORT_IGNORE_VIEWS '<view>'
```

## Viewing a Design

Design Entry HDL provides you with a read-only Design Entry HDL Viewer, which enables you to browse a design even when you are working on another design. You can do the following in the read-only viewer:

- View the hierarchy of the project.

- Descend to a particular block/sheet.

To reference a block in the currently open project from another project,

1. Choose *File – View Design*.

You are prompted to specify the name of the design that you want to browse.

2. Specify the name of the design you want to browse.

3. Click *OK*.

An instance of the Design Entry HDL viewer is spawned. As you can notice in the figure below, there is no hierarchy viewer or console command window.

As you move through the various menus and the toolbar, you will notice that most of the tool bar icons and menu options are also disabled. The reason is that the design has been opened in a read-only mode, which is meant only for the purpose of browsing.

## Baselining a Design

Schematic design can be a collaborative process where many designers work on different parts of a schematic. Once they complete their part of design, an integrator can merge the various parts to create the complete schematic.

In order to be able to do this, Allegro Design Entry HDL now empowers you to create and save details of changes made to a schematic at any stage of design process. This is known as baselining. Whenever you baseline a schematic, the details of changes made are saved and mapped to a version number. This version number differentiates between multiple versions of schematic.

With this feature, you can:

- Save details of changes made to a schematic.
- View changes made to the schematic from the first version.
- View user-entered comments associated with a version.

## How Baselining Works

Baselining a design involves creation of the `metadata` folder in the design project folder. This folder saves the up-to-date design information for the project. Each time you make changes to the design and baseline it, the metadata information is updated.

## Allegro Design Entry HDL User Guide

### Working with Designs

To ensure that baseline feature works correctly, you must enable the creation of metadata in Allegro Design Entry HDL. For information on how to configure schematic metadata creation preferences, see [Setting Preferences for Metadata Creation](#).

**Note:** The location of reference or local libraries is defined in the `cds.lib` file of your design project.

**Note:** If a schematic cell does not contain any metadata, Component Revision Manager does not check it for differences with the library cell.

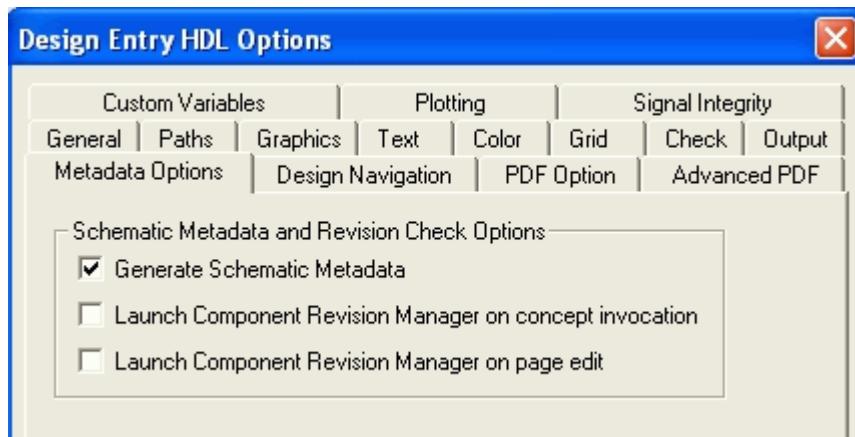


Make sure that metadata is generated in the current release. Cadence strongly recommends you not to use metadata created using earlier versions.

### Setting Preferences for Metadata Creation

Before you start using baseline feature, make sure that Allegro Design Entry HDL has been configured to create metadata for your design projects. To ensure that:

1. Choose *Tools – Options*. The *Design Entry HDL Options* dialog box appears.
2. Select the *Metadata Options* tab.



3. To generate schematic-related metadata in Allegro Design Entry HDL, select the *Generate Schematic Metadata* check box in the *Schematic Metadata and Revision Check Options* section.

**Note:** The metadata creation, by default, is set to off. Use the `GENERATE_SCH_METADATA 'ON'` directive in the `cpm` file to enable metadata creation, by default.

## Allegro Design Entry HDL User Guide

### Working with Designs

---

4. To ensure that your schematic is automatically checked for differences between the library cells and schematic when:
  - a. You launch Allegro Design Entry HDL, select the *On Design Entry HDL Invocation* check box in the *Schematic Metadata and Revision Check Options* section.
  - b. You move from one page to another page of design, select the *On page Edit* check box in the *Schematic Metadata and Revision Check Options* section.
5. Click *OK*.

**Note:** By default, the ability to check for differences between the schematic and library cells is set to off. Alternatively, you can use the following directives (to be specified in cpm file) to control the default revision check behavior.

Directive	Description
SYNC_ON_STARTUP 'ON'	Checks for differences between the schematic and library cells at Design Entry HDL startup.
SYNC_ON_PAGE_EDIT 'ON'	Checks for differences between the schematic and library cells every time you move from one page to another in design.

**Note:** After enabling metadata creation, make sure that you run the `hier_write` command (by entering it in the console window of Design Entry HDL) on the design. This will update the metadata for the existing components, and will create metadata for the newly added components, if any. This ensures that the Component Revision Manager detects the differences between the schematic and library cells correctly.

## Baselining a Design

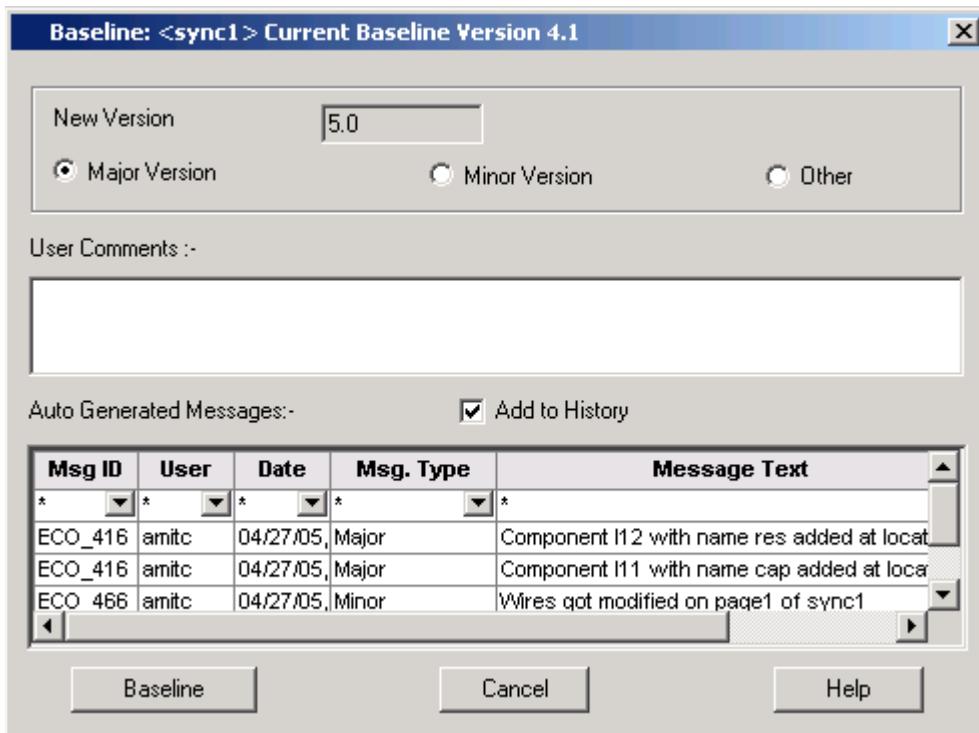
To baseline a design, do the following:

1. In Allegro Design Entry HDL, choose *File – Save and Baseline*.

## Allegro Design Entry HDL User Guide

### Working with Designs

The *Baseline: <design name> Current Baseline Version* dialog box appears.



2. In the *New Version* section, specify a version type for the schematic.

- a. If the design has undergone major changes, select the *Major Version* radio button.

Selecting a major version, by default, automatically increments the current version number by 1 (a whole number). For example, if the current version of the design is 2.0, selecting the *Major Version* radio button increments the new version number to 3.0.



- b. If the design has undergone minor changes, select the *Minor Version* radio button to automatically increase the current version number by the nearest decimal point.

For example, if the previous version is 2.0, choosing the *Minor Version* radio button changes the version number to 2.1.



## Allegro Design Entry HDL User Guide

### Working with Designs

- c. Select the *Other* radio button to enter a desired version number of the schematic in the text box.

The version number should have zero as the last decimal point (for example: 1.3.0 and 1.2.1.0). You cannot have version numbers such as 1.4.5 and 2.3.7.



**Note:** By default, the *Major Version* radio button is chosen.

3. Optionally, enter comments specific to the new version in the *User Comments* section.
4. Select the *Add to History* check box in the *Auto-Generated Messages* section to include system-generated messages in the log files, along with user comments. These messages describe the differences you have made since the last baselined version.

Selecting the *Add to History* check box also ensures that these messages appear in the Component Revision Manager dialog box.

Msg ID	User	Date	Msg. Type	Message Text
*	*	*	*	*
ECO_416	amitc	04/27/05	Major	Component I12 with name res added at locat
ECO_416	amitc	04/27/05	Major	Component I11 with name cap added at loca
ECO_466	amitc	04/27/05	Minor	Wires got modified on page1 of sync1

If you do not select this check box, then only the user comments will be added in the log files, and they will appear in the Component Revision Manager dialog box for the respective versions. By default, the *Add to History* check box is selected.

The *Add to History* check box also gives the flexibility to add or reject the system-generated messages, which can be too large in number or be irrelevant at times. In such a situation, you can copy the desired system-generated messages from the grid, and paste them into the *User Comments* section.

5. Click *Baseline* to save and baseline the schematic.

**Note:** If you try to baseline a schematic which has no changes as compared to the previous version, a message appears that there are no changes in the schematic, and the baseline operation is canceled.

**Note:** Whenever you synchronize the differences between the schematic and library cells in a design, only the baselined schematic versions are used. Hence, it is necessary to baseline a schematic.



Saving or baselining any of the lower-level blocks of a hierarchical design does not automatically update the metadata linkages with the version number at the top level. To update the linkages at once, it is necessary to save the schematic at the top level. Alternatively, browse all the pages containing such blocks (moving towards the top level); and update them using the Component Revision Manager window that appears when you move between pages of the schematic.

## Creating the Table of Contents for a Design

In many of the designs you create, the first sheet of the schematic contains the table of contents (TOC). A TOC provides a quick view of the design contents and also helps to navigate the design when a schematic design is plotted or published. The information displayed in the TOC can range from sheet-specific data, such as sheet numbers, sheet names, module or block names to design-specific data, such as the name of the project, the name of the engineer, date, the total number of pages, and so on. The data on the page borders is referred to as master data.

**Note:** In-context data such as a subdesign suffix which is present on different block instantiations, and is different for different instantiations, cannot be included in the TOC.

Design Entry HDL provides functionality for creating and automatically updating the TOC with design information. This feature replaces any indigenous solutions that you might have been using. The automatic TOC generator feature saves you design time as it does not require any

## Allegro Design Entry HDL User Guide

### Working with Designs

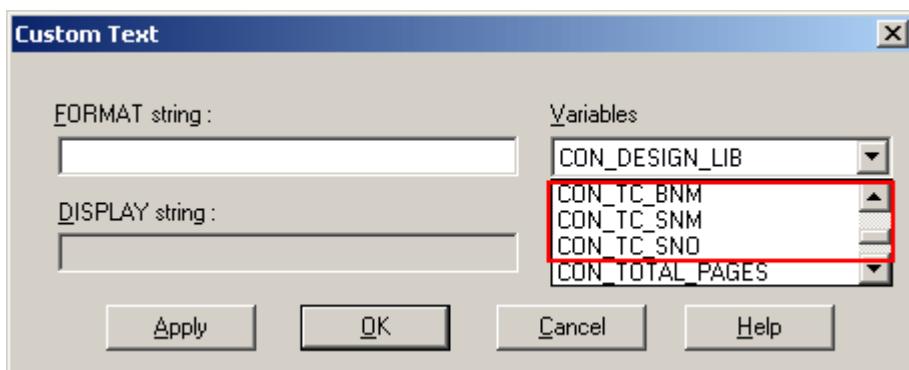
manual validation. A TOC is automatically updated whenever the title of a page changes or a sheet or a block is added, deleted, or moved.

TABLE OF CONTENTS	
1	COVER
5	NOTES
6	BLANK
7	BLANK
8	BLANK
9	BLANK
10	BLANK
11	SYSTEM BLOCK DIAGRAM
12	MARVELL BLOCK DIAGRAM
13	DETAILED MEMORY BLOCK DIAGRAM
14	BLANK
15	BLANK
16	MARVELL - ALKINDI - PHY - BLOCK
17	LAYOUT, TOP SIDE
18	BLANK
19	BLANK
20	MIDPLANE, SPINE 1
21	MIDPLANE, SPINE 2
22	MIDPLANE, SPINE 3

TABLE OF CONTENTS	
56	SANTA CRUZ, 4/6
57	SANTA CRUZ, 5/6
58	SANTA CRUZ, 6/6
59	FLTR
60	FLTR, VREF
61	BYPASS CAPS
62	BYPASS CAPS
63	BYPASS CAPS
64	BLANK
65	AC COUPLING CAPS
66	SCR1 TO OCT1 CAPS
67	SCR1 TO OCT1 CAPS
68	SCR1 TO OCT1 CAPS
69	SCR1 TO OCT1 CAPS
70	OCT1 - 1 OF 11 CP
71	OCT1 - 2 OF 11 LT
72	OCT1 - 3 OF 11 MD
73	OCT1 - 4 OF 11 FA
74	OCT1 - 5 OF 11 ME

### Adding the Table of Contents to a Design

In order to include a TOC in your design, you need to instantiate the TOC symbol on your schematic. Typically, a TOC symbol includes predefined custom text variables for sheet-specific data, such as the sheet number (CON\_TC\_SNO), the sheet name (CON\_TC\_SNM), and the block name (CON\_TC\_BNM). Allegro Design Entry HDL evaluates these variables at run time. You can add custom text variables for design-specific data, such as the project name, the name of the engineer, the date, and the total number of pages to the default symbols.



## Allegro Design Entry HDL User Guide

### Working with Designs

#### Instantiating a TOC Symbol on a Schematic

You instantiate a TOC symbol on a schematic just as you instantiate any other symbol. Consider the example of a sample design consisting of 86 sheets. Let us add a TOC symbol to this design:

1. Open the design.

Notice that the names of all the sheets appear in the hierarchy viewer.

2. Click a page in the hierarchy viewer.

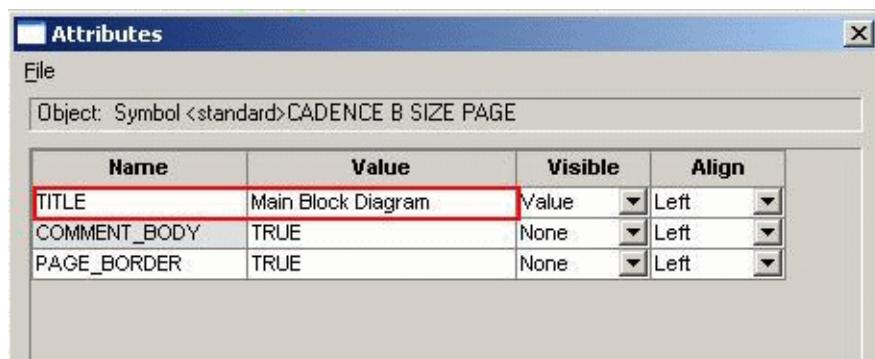
Page titles are determined from a property in the page border and displayed in the hierarchy viewer. The property from which the page title is determined can be specified in the setup. Also, you can modify this property in the Attributes form. You can change the border by:

- a. Add the property on page border for page name from *DE-HDL – Tools – Options – Design Navigation*.
- b. Change the page name from Hierarchy Viewer, select the page, right-click and select *Edit Page Name*.

The page name gets changed in Hierarchy Viewer and also in the Attribute form of the Page Border.

3. Now, right-click the page border and choose *Attribute*.

Notice that the page title (sheet name) is defined in the TITLE property on the page border. The property from which the page title is determined can be configured.



4. Insert one sheet before page 1

A new page is inserted before page 1.

5. Place the TOC symbol on this sheet:

- a. Select page 1 in the hierarchy viewer.
- b. Choose *Component – Add*.
- c. Select *standard* from the list of libraries.
- d. Select the name of the TOC symbol. For example, *toc*.
- e. Click the *Add* button.
- f. To put the TOC on the first page of the design, click page 1 in the hierarchy viewer.
- g. Right-click and choose *Done* from the pop-up menu.



The TOC symbol is added to page1.

6. Choose *File – Save*.

## Allegro Design Entry HDL User Guide

### Working with Designs

7. Click a page in the hierarchy viewer other than page 1 and then click page 1 again.

SHEET NUMBERS	SHEET DETAILS
1	page1
2	Index
3	Main Block Diagram
4	Processors

The TOC is generated.

## Allegro Design Entry HDL User Guide

### Working with Designs

#### ***Handling Movement of Pages in the TOC***

If you move pages around in a design, the TOC is updated with the new sequence of pages. For example, if you move page 1 to 2, the change in the sequence of pages is also reflected in the TOC.

The diagram shows a top-level schematic with three components labeled TAB, A, and B. Below this is a Table of Contents grid. The grid has two columns: 'SHEET NUMBERS' and 'SHEET DETAILS'. It contains three rows. The first row has '1' in the 'SHEET NUMBERS' column and 'Index' in the 'SHEET DETAILS' column. The second row has '2' in the 'SHEET NUMBERS' column and 'page1' in the 'SHEET DETAILS' column. Both the '1' row and the '2' row are highlighted with red boxes.

SHEET NUMBERS	SHEET DETAILS
1	Index
2	page1

#### ***Handling Changes in Page Titles in the TOC***

Any change made to the page titles is also reflected in the TOC. For example, if you change the title of the TOC page to Table of Contents, the TOC is updated to reflect the change.

The diagram shows a top-level schematic with three components labeled TAB, A, and B. Below this is a Table of Contents grid. The grid has two columns: 'SHEET NUMBERS' and 'SHEET DETAILS'. It contains three rows. The first row has '1' in the 'SHEET NUMBERS' column and 'Index' in the 'SHEET DETAILS' column. The second row has '2' in the 'SHEET NUMBERS' column and 'Table of Contents' in the 'SHEET DETAILS' column. The third row has '3' in the 'SHEET NUMBERS' column and 'Main Block Diagram' in the 'SHEET DETAILS' column. All three rows are highlighted with red boxes.

SHEET NUMBERS	SHEET DETAILS
1	Index
2	Table of Contents
3	Main Block Diagram

#### ***Handling Multiple TOC Pages***

If the design is large, you can insert multiple sheets with the TOC symbol to accommodate page information in all the sheets in the design.

A design might also have many rows of sheets with the same page title. Instead of having multiple entries for such rows in the TOC, you can show such entries in a single row with the sheet numbers as a range in the Sheet Number column.

36	EGRESS QDRII SRAM CHANNEL 0
37	EGRESS QDRII SRAM CHANNEL 0
38	EGRESS QDRII SRAM CHANNEL 1
39	EGRESS QDRII SRAM CHANNEL 1
40	EGRESS QDRII SRAM CHANNEL 2
41	EGRESS QDRII SRAM CHANNEL 2

To display a page range in the TOC, you need to specify the following CPM directive in the START\_CONCEPTHDL section of the project's .cpm file:

```
START_CONCEPTHDL  
TOC_DISPLAY_SHEET_RANGE 'ON'  
...  
END_CONCEPTHDL
```

## Allegro Design Entry HDL User Guide

### Working with Designs

The TOC shows page ranges in the Sheet Number column wherever applicable

35	Decoupling Caps
36–37	EGRESS QDRII SRAM CHANNEL 0
38–39	EGRESS QDRII SRAM CHANNEL 1
40–41	EGRESS QDRII SRAM CHANNEL 2
42–43	EGRESS QDRII SRAM CHANNEL 3

**Note:** You can also control the line spacing between TOC rows by using the following directive:

```
START_CONCEPTHDL  
TOC_ROW_SPACING_MULTIPLIER '1'  
...  
END_CONCEPTHDL
```

## Creating a Symbol for TOC

You can also create custom TOC symbols for your design. For example, you can enhance the Page Border symbol and use it as a TOC symbol. You can use lines (wires) to draw tables and add logo bitmaps to the symbols. The TOC symbol is identified by the property TOC\_SYMBOL set to TRUE.

1. Select a sheet with a page border in the hierarchy viewer.
2. Double-click on the page border and open the symbol.
3. Save the symbol as a TOC symbol.

## Allegro Design Entry HDL User Guide

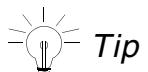
### Working with Designs

4. Choose *Text – Custom Text*.
5. In the Custom Text dialog box, select CON\_TC\_SNO as the custom text and specify the repeat count in the *Repeat* field. The repeat count determines the number of entries for the custom text in the TOC symbol.
6. Click on the symbol origin to attach the custom text.

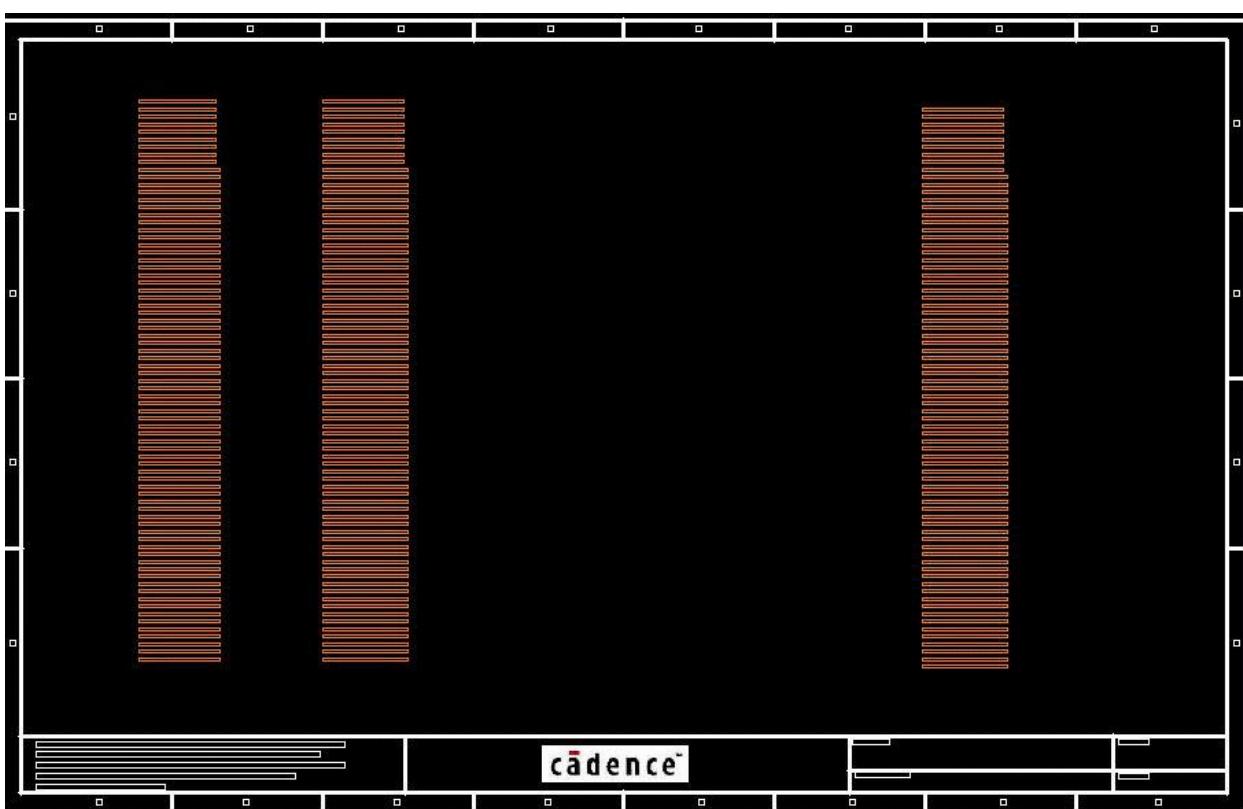
Now the custom text variable shows a linkage with the symbol origin.

7. Place the custom text variable on the top left part of the sheet. Based on the repeat count specified in the Custom Text dialog box, the custom text entries are added to the symbol one below the other.
8. Similarly, you can add the CON\_TC\_SNM and CON\_TC\_BNM custom variables to the TOC symbol.

The three variables you have selected will appear as three columns in the TOC symbol.



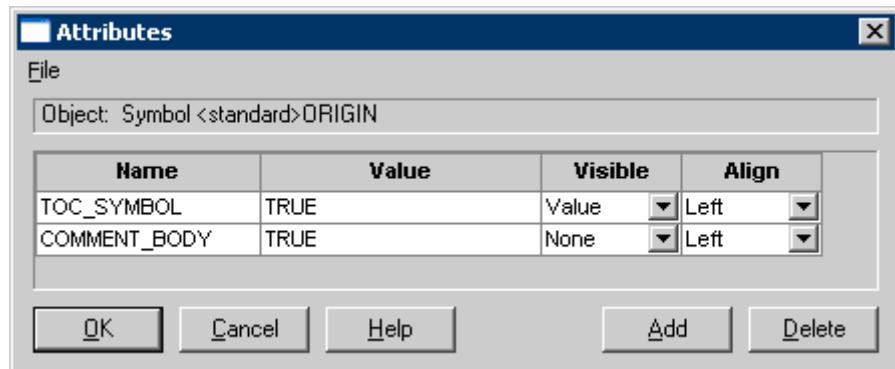
**Tip**  
Ensure that you have all the three columns horizontally aligned. Keep enough space for the sheet names because they are usually long.



## Allegro Design Entry HDL User Guide

### Working with Designs

9. Add graphics, such as rectangles or boxes created using wires, or a bitmap if required.
10. Set the TOC\_SYMBOL property to TRUE and attach it to the origin of the symbol:
  - a. Choose *Text – Attributes*.
  - b. Trace and click the origin of the symbol.
  - c. Click *Add* on the Attributes form.
  - d. Type TOC\_SYMBOL in the Name column.
  - e. Specify *TRUE* as the value.



11. Choose *File – Save* to save the symbol.

You can now instantiate this symbol in a schematic:

1. In the schematic, insert a new sheet before sheet 2. The sheet opens with the default page border.
2. Replace the page border with the TOC symbol.
3. Choose *File – Save* to save the design.

## Allegro Design Entry HDL User Guide

### Working with Designs

4. Refresh the new TOC page by selecting any other page from the hierarchy viewer and then moving back to the newly added TOC page

SHEET NUMBER	SHEET NAME
1	Index
2	page2
3	Main Block Diagram
4	Processors
5	SPI4/SLOW PORT MEDIA CONNECTOR
6	Processor MEDIA <SPI4>POWER
7	NEW SINK FABRIC INTERFACE CONNECTOR
8	FABRIC INTERFACE CONNECTOR
9	FABRIC SIDE - UTILITY PCI BUS
10	EGRESS PIN DECOUPLING
11	INGRESS PIN DECOUPLING
12	PCI SECTION BLOCK DIAGRAM
12	HOST PCI INTERFACE <CPCI T1 AND T2>
14	HOST_PCI_INTERFACE
15	INGRS AND EGRESS IXFB800
15	21155 AND 21154 PCI INTERFACE <NPU DOMAIN>
15	21154 PCI INTERFACE <UTILITY DOMAIN>
18	B2559 NIC Interface <Utility Domain>
19	ETHERNET AND SERIAL CONNECTORS
20	PMC EXTENSION CARD CONNECTOR <UTILITY INTERFACE>
21	EGRESS NPU SLOW PORT AND PLD

The TOC is updated, and the page numbers appear as sheet ranges where appropriate

# **Allegro Design Entry HDL User Guide**

## Working with Designs

---

---

## **Netlisting Your Design**

---

A netlist contains the list of signals, parts, and pins in a design and information on how they are connected with other devices.

Design Entry HDL allows you to generate netlists that can be used for:

- Digital simulation
  - For more information, see [Netlisting for Digital Simulation](#) on page 524.
- Programmable IC simulation
  - For more information, see [To specify the options for generating the VHDL netlist for synthesizing a design](#) on page 526.
- Analog and Mixed Signal simulation
  - For more information, see [Netlisting for Analog and Mixed Signal Simulation](#) on page 529.

You can specify the options for generating the netlist.

When Design Entry HDL generates the netlist, it does the following:

- Checks the drawings for Verilog and VHDL compatibility.
- Performs cross-view checking of ports, port modes, and port types between schematic views and symbol views.

For more information, see [Cross-View Checking](#) in *Allegro Design Entry HDL Reference Guide*.

- Performs entity declaration checking for instantiated components.

For more information, see [Entity Declaration Checking for Instantiated Components](#) in *Allegro Design Entry HDL Reference Guide*.

See [Netlisting Errors](#) for solutions or workarounds for errors that occur during the netlisting process.

## Netlisting for Simulation

This section describes the following:

- [Netlisting for Digital Simulation](#) on page 524
- [To specify the options for generating the VHDL netlist for synthesizing a design](#) on page 526
- [Netlisting for Analog and Mixed Signal Simulation](#) on page 529

### Netlisting for Digital Simulation

The Design Entry HDL Digital Simulation Interface allows you to generate the Verilog and VHDL netlist for digital simulation. You can generate the netlist for use with the following simulators:

- Verilog-XL simulator
- Affirma NC Verilog simulator
- Third-party Verilog simulators
- Leapfrog VHDL simulator
- Affirma NC VHDL simulator
- Third-party VHDL simulators

You can specify the options for generating the netlist in the Design Entry HDL digital simulation interface.

#### ***To specify the options for generating the netlist for digital simulation***

1. Start *Project Manager*.
2. Choose *File – Open*.
3. Select the project file (.cpm) and click *OK*.
4. Choose *Tools – Setup*.  
The *Project Setup* window appears.
5. Select the *Tools* tab.
6. Click *Simulation Setup*.

The *Choose Simulator* dialog box appears.

7. Select the simulator you want to use for performing digital simulation.
8. Click *Setup*.

The setup dialog box for the simulator appears.

9. Select the *Netlist* tab and specify the options for generating the netlist.

**Note:** If you have a hierarchical design, you can select the *Single File Netlist* check box in the *Netlist* tab to generate a single file netlist for the entire design.

For more information on how to specify the options for generating the netlist for each of the simulators, see *Allegro Design Entry HDL Digital Simulation User Guide*.

If you have already selected the simulator (steps 1 to 6 above) you want to use for performing digital simulation, you can also specify the netlisting options by doing the following:

1. In Design Entry HDL or Project Manager, choose *Tools – Simulate*.

The simulation interface dialog box appears.

2. Click *Setup*.

The setup dialog box for the simulator appears.

3. Select the *Netlist* tab and specify the options for generating the netlist.

**Note:** If you want to generate a single file netlist for a hierarchical design, select the *Single File Netlist* check box in the *Netlist* tab.

For more information on specifying the netlisting options for each of the simulators, see *Allegro Design Entry HDL Digital Simulation User Guide*.

#### ***To generate the netlist for digital simulation***

Once you have specified the options for generating the netlist for a specific simulator, you can generate the netlist for use with the simulator.

To generate the netlist:

1. In Design Entry HDL or Project Manager, choose *Tools – Simulate*.

The simulation interface dialog box appears.

2. Click *Run*.

## Allegro Design Entry HDL User Guide

### Netlisting Your Design

---

The *Simulation Progress Status* window appears displaying the progress of the netlisting and simulation processes. Click *Details* to view the details of these processes.

For the Verilog-XL, NC Verilog, and third-party Verilog simulators, the Verilog netlist file `verilog.v` is generated. For the Leapfrog, Affirma NC VHDL, and third-party VHDL simulators, the VHDL netlist file `vhdl.vhd` is generated. The netlist files are located in the following directory:

```
<project_directory>/worklib/<design_name>/sim_sch_1/
```

**Note:** If you have selected the *Single File Netlist* check box in the *Netlist* tab, a single file Verilog netlist `<design_name>.v` or a single file VHDL netlist `<design_name>.vhd` are generated (for hierarchical designs). These files are located in the *run* directory you specify in the simulation interface dialog box.

The schematic errors that are identified during the netlisting process are displayed in the *Markers* dialog box in Design Entry HDL. Click on an error to highlight the area on the schematic where the error has occurred. Correct the schematic errors and generate the netlist again.

The schematic errors that are identified during the process of generating the netlist for digital simulation are stored in the `simNetlist.mkr` file that is located in the *run* directory you specify in the simulation interface dialog box. You can view the errors by loading the file in the *Markers* dialog box. The `netassembler.log` file located in the *run* directory contains the details of the netlisting process.

#### To view netlisting errors in the digital simulation netlist

1. In Design Entry HDL, choose *Tools – Markers*.

The *Markers* dialog box appears.

2. Choose *File – Load* and open the `simNetlist.mkr` file located in the *run* directory you specified in the simulation interface dialog box.

The schematic errors that are identified during the netlisting process are displayed in the *Markers* dialog box.

#### To specify the options for generating the VHDL netlist for synthesizing a design

1. In Design Entry HDL, choose *Tools – Options*.

The *Design Entry HDL Options* dialog box appears.

2. Select the *Output* tab.

Ensure that the *Create Netlist* check box is selected.

3. Select the *VHDL* check box.

4. Click the *Options* button next to the *VHDL* check box.

The *VHDL Netlist* dialog box appears.

5. Select the *Verbose Output* check box to log the debug messages of the netlisting process in the `hdlmdir.log` file located in the `<project_directory>/temp/` directory.
6. Select the *Strict Entity Check* check box to enable checking by comparison of instance properties and symbol properties like the `VHDL_MODE` property.
7. Select the *Check Instance Vs Signal* check box if you want Design Entry HDL to check if the name of any signal on the schematic is the same as `page<page_number>_<value of PATH property on any instance>`. If this check box is selected, Design Entry HDL displays the following error message for every signal that has the same name as `page<page_number>_<value of PATH property on any instance>`:  

```
126 ERROR "Identifier is used as both a PATH value and a signal name."
```
8. Specify the maximum number of netlisting errors that you want to allow in the *Max Errors* field. If the number of netlisting errors in the design exceeds the number specified here, Design Entry HDL will not generate the netlist.
9. Specify the VHDL logic type for all the vectored ports and signals in the design. You can specify any legal VHDL vector type, such as `STD_LOGIC_VECTOR` and `BIT_VECTOR`. The default value is `STD_LOGIC_VECTOR`.  
**Note:** The specified vector type applies to all drawings in the design. You can override the vector type for individual drawings by using the `VHDL_VECTOR_TYPE` property on a `VHDL_DECS` symbol.
10. Specify the VHDL logic type for all scalar ports and signals in the design. You can specify any legal VHDL scalar type, such as `STD_LOGIC` and `BIT`. The default value is `STD_LOGIC`.  
**Note:** The specified scalar type applies to all drawings in the design. You can override the scalar type for individual drawings by using the `VHDL_SCALAR_TYPE` property on a `VHDL_DECS` symbol. For more information on specifying the VHDL logic type of ports and signals, see [Setting the VHDL Logic Type for Ports and Signals](#) on page 234.
11. Specify the names of the libraries that are to be used in VHDL library clauses in the VHDL entity and architecture text generated from the schematic. If you do not specify a library, IEEE will be used as the default library.

## Allegro Design Entry HDL User Guide

### Netlisting Your Design

---

**Note:** The default library (`IEEE`) is not used if you specify any other library. If you want to use the `IEEE` library along with other libraries, you must explicitly add the `IEEE` library.

You can also add libraries for a drawing by using the `LIBRARY` property on a `VHDL_DECS` symbol. In the VHDL entity and architecture text generated from the schematic, the libraries on the symbol will be appended to the list of libraries you specify here.

12. Specify the names that are to be used in VHDL use clauses in the VHDL entity and architecture text generated from the schematic. There is no limit on the number of use clauses you can add. If you do not specify any use clauses, `IEEE.STD_LOGIC_1164.ALL` will be used as the default.

**Note:** The `IEEE.STD_LOGIC_1164.ALL` use clause will not be used if you add any other use clauses. If you want to use `IEEE.STD_LOGIC_1164.ALL` along with any other use clause, you must explicitly add `IEEE.STD_LOGIC_1164.ALL`.

You can also add use clauses for a drawing by using the `USE` property on a `VHDL_DECS` symbol. In the VHDL entity and architecture text generated from the schematic, the use clauses on the symbol will be appended to the list of use clauses you specify here.

13. Click *OK* to save the settings and close the *VHDL Netlist* dialog box.
14. Select the *Annotate Synthesis Constraints in Netlist* check box if you want the synthesis constraints specified in Design Entry HDL to be written in the netlist.  
Synthesis constraints are specified in Design Entry HDL using vendor-specific (Xilinx, Altera, or Actel) properties. For more information, see the [Programmable IC Tutorial](#).
15. Click *OK* to close the *Design Entry HDL Options* dialog box.

#### **To generate the netlist for synthesizing a design**

- In Design Entry HDL, choose *File – Save*.

The Verilog netlist file `verilog.v` and VHDL netlist file `vhdl.vhd` are generated in the following directory:

`<project_directory>/worklib/<design_name>/sch_1/`

The schematic errors that are identified during the netlisting process are displayed in the *Markers* dialog box in Design Entry HDL. Click on an error to highlight the area on the schematic where the error has occurred. Correct the schematic errors and generate the netlist again.

The schematic errors that are identified during the process of generating the netlist for packaging are stored in the `netlister.mkr` file that is located in the `<project_directory>/temp/xxnedtmp/` directory. You can view the errors by loading

## Allegro Design Entry HDL User Guide

### Netlisting Your Design

---

the file in the *Markers* dialog box. The `hdlmdir.log` file located in the `<project_directory>/temp/` directory contains the details of the netlisting process.

#### **To view netlisting errors**

1. In Design Entry HDL, choose *Tools – Markers*.

The *Markers* dialog box appears.

2. Choose *File – Load* and open the `netlister.mkr` file located in the `<project_directory>/temp/xxnedtmp/` directory.

The schematic errors that are identified during the netlisting process are displayed in the *Markers* dialog box.

## Netlisting for Analog and Mixed Signal Simulation

Design Entry HDL supports analog and mixed signal simulation using the PSpice A/D simulator. PSpice A/D is a simulation program that models the behavior of a design containing any mix of analog and digital devices.

#### **To generate the netlist for PSpice simulation**

- In Design Entry HDL, choose *PSpice – Create Netlist*.

A netlist file `<design_name>.net` will be generated in the following directory:

`<project_directory>\worklib\<design_name>\cfg_analog\`

The schematic errors that are identified during the netlisting process are displayed in the *Markers* dialog box in Design Entry HDL. Click on an error to highlight the area on the schematic where the error has occurred. Correct the schematic errors and generate the netlist again.

The schematic errors that are identified during the process of generating the netlist for packaging are stored in the `<design_name>.mkr` file that is located in the `<project_directory>\worklib\<design_name>\cfg_analog\` directory. You can view the errors by loading the file in the *Markers* dialog box. The `hdlmdir.log` file located in the `<project_directory>/temp/` directory contains the details of the netlisting process.

#### **To view netlisting errors in the PSpice netlist**

1. In Design Entry HDL, choose *Tools – Markers*.

The *Markers* dialog box appears.

2. Choose *File – Load* and open the <design\_name>.mkr file located in the <project\_directory>\worklib\<design\_name>\cfg\_analog\ directory.

The schematic errors that are identified during the netlisting process are displayed in the *Markers* dialog box.

## Generating Netlist in ISCF format

When creating a schematic design for Intel, you can get it reviewed early in the design cycle by Intel to ensure that all Intel design guidelines are followed. For the review, Intel Schematic Checking teams need schematic data in a specific format.

Since every Original Equipment Manufacturer (OEM) has its own set of design tools from different vendors, each vendor provides schematic data in its own specific format. To stay format neutral, Intel has developed a format called Intel Schematic Connectivity Format (ISCF).

Design Entry HDL allows you to export the details of a schematic to ISCF using the geniscfnetlist console command.

### geniscfnetlist

Creates a design report with .iscf extension. You can execute this command directly from the Windows command line without launching Design Entry HDL. The report lists details of components, pins, connections, properties, power nets, and ground that are available in the design and packaged.

### Syntax

geniscfnetlist

```
-proj <cpmfile>
-o <path_for_iscf_file>
[-temp <location_for_log_file>]
[-compprops <comma_separated_list_of_propnames>]
[-pinprops <comma_separated_list_of_propnames>]
[-groundnets <comma_separated_list_of_groundnetnames>]
```

## Allegro Design Entry HDL User Guide

### Netlisting Your Design

---

#### Parameter Description

Parameter	Description
proj	Path to the project file.
o	Destination path with the filename where the .iscf file will be created.
temp	<Optional> Destination path where the log file with the name geniscfnetlist.log will be created. This log file stores all the program execution information.
compprops	<Optional> Use this parameter if you want to include specific component properties in the report. You can specify component property names in a comma-separated list.  If this parameter is not specified, the default component properties PART_NAME, PART_NUMBER, and JEDEC_TYPE are included in the .iscf file.
pinprops	<Optional> Use this parameter if you want to include specific pin properties in the report. You can specify pin property names in a comma-separated list.  If this parameter is not specified, the default pin properties PIN_NUMBER, PIN_NAME, and TYPE are included in the .iscf file.
groundnets	<Optional> geniscfnetlist considers all signals with the VOLTAGE property as ground and power signals, and hence includes them in a common section in the report.  Use this parameter if you want to include specific ground signals in a separate section in the report. You can specify ground signal names in a comma-separated list.  If this parameter is not specified, the default signal names GND, GND_EARTH, 0, and 0V are included in the .iscf file.  Signals that have the VOLTAGE property added and are not specified in this parameter are included in the power signals section in the report.

# Allegro Design Entry HDL User Guide

## Netlisting Your Design



Specify property names in capital letters and without any spaces.

### Example

```
geniscfnetlist -proj C:/Projects/ISCF_Netlist_generation/
reference.cpm -o C:/Netlist/netlist_default.iscf
```

This command generates a report with the name `netlist_default.iscf`, which includes the default property names.

```
geniscfnetlist -proj C:/Projects/ISCF_Netlist_generation/
reference.cpm -o C:/Netlist/netlist.iscf -temp C:/Netlist/logs -
compprops PART_NUMBER,PART_NAME,JEDEC_TYPE -pinprops
PIN_NUMBER,PIN_NAME -groundnets GND,GND_EARTH,V12N
```

This command generates a report with the name `netlist.iscf`, which includes the property names specified in the `compprops`, `pinprops`, and `groundnets` parameters.

The generated report contains the sections COMPROPS, COMPPINS, E-NETS, BUSES, NETS, GROUND, and POWER as shown in the following figure:

```
1 BEGIN COMPROPS
2 ##Refdes:PART_NUMBER,PART_NAME,JEDEC_TYPE
3 C1:cap123,CAP_NP,SM_1206
4 C2:cap123,CAP_NP,SM_1206
5 C3:cap123,CAP_NP,SM_1206
6 END_COMPROPS
7
8 BEGIN COMPPINS
9 ##Refdes:(PIN_NUMBER:PIN_NAME:TYPE),(...),...,(...)
10 C1((1:A:Analog),(2:B:Analog))
11 U1((1:IO<1>:Output_BIDIR),(2:A<3>:Input),(3:A<2>:Input),(4:A<1>:Input),(5:A<0>:Input),(6:CE*:Input),(7:DQ<1>:TS_BIDIR),(8:VCC:Power),(9:GND:Power),(10:DQ<2>:TS_BIDIR),(11:WE*:Input),()
12 END_COMPPINS
13
14 BEGIN_E_NETS
15 END_E_NETS
16
17 BEGIN_BUSES
18 ##NetName=1,NetName=2,,,,,,NetName=N
19 A<0>,A<1>,A<2>,A<3>,A<4>,A<5>,A<6>,A<7>,A<8>,A<9>,A<10>,A<11>,A<12>,A<13>,A<14>,A<15>,A<16>,A<17>,A<18>,A<19>,A<20>,A<21>,A<22>,A<23>
20 BA<0>,BA<1>,BA<2>,BA<3>,BA<4>,BA<5>,BA<6>,BA<7>
21 END_BUSES
22
23 BEGIN_NETS
24 ##NetName:RefDes(pinNumber:pinLabel),RefDes(pinNumber:pinLabel),...,RefDes(pinNumber:pinLabel);
25 BA<3>;U9(6:B<3>),U13(17:B<3>),U13(6:B<3>),U5(9:DATA2)
26 NC_J1(12:A<12>),J1(48:A<48>),J1(49:A<49>),U12(7:I/O),U5(27:TDO),U5(35:VCLKB),U5(36:ADD16),U5(52:NTRST),U5(72:TCK),U14(13:Q6),U14(15:Q4),U14(17:Q2),U14(19:Q0),U1(1:NC)
27 END_NETS
28
29 BEGIN_GROUND
30 ##NetName:RefDes(pinNumber:pinLabel),RefDes(pinNumber:pinLabel),...,RefDes(pinNumber:pinLabel);
31 AGND_J1(34:A<34>),J1(35:A<35>),J1(36:A<36>),U201_1(3:COMP),U201_1(7:NC2),U201_1(10:AGND1),U202_1(3:GND),C203_1(2:B),C203_1(2:B),U201_2(3:COMP),U201_2(7:NC2),U201_2(10:AGND1),U202_2(3:GND)
32 GND_J1(1:A<1>),J1(2:A<2>),J1(3:A<3>),J1(33:A<33>),C1(2:B),C3(2:B),C2(2:B),C5(2:B),U12(11:I11),U12(12:GND),U12(13:I12),U16(7:GND),U16(15:GND),U16(31:GND),U16(42:GND),U8(1:DIR),U8(7:GND)
33 END_GROUND
34
35 BEGIN_POWER
36 ##NetName:RefDes(pinNumber:pinLabel),RefDes(pinNumber:pinLabel),...,RefDes(pinNumber:pinLabel);
37 GND_J1(29:A<29>),J1(30:A<30>),J1(62:A<62>),C1(1:A),C3(1:A),C2(1:A),C5(1:A),U12(24:VCC),U16(4:VCC),U16(10:VCC),U16(12:VCC),U16(21:VCC),U16(28:VCC),U16(34:VCC),U16(45:VCC),U16(49:VCC)
38 V12N_J1(63:A<63>),J1(64:A<64>),U202_1(4:V-),U203_1(7:V+),C203_1(11:V+),U202_2(4:V-),U203_2(7:V+),C203_2(1:V+)
39 V12N_J1(31:A<31>),J1(32:A<32>),U202_1(1:V-),U203_1(4:V+),U202_2(7:V-),U203_2(4:V+)
40 VCC_J1(29:A<29>),J1(30:A<30>),J1(62:A<62>),C1(1:A),C3(1:A),C2(1:A),C5(1:A),U12(24:VCC),U16(4:VCC),U16(10:VCC),U16(12:VCC),U16(21:VCC),U16(28:VCC),U16(34:VCC),U16(45:VCC),U16(49:VCC)
41 END_POWER
42
43
```

---

## **Managing Variants in Design Entry HDL**

---

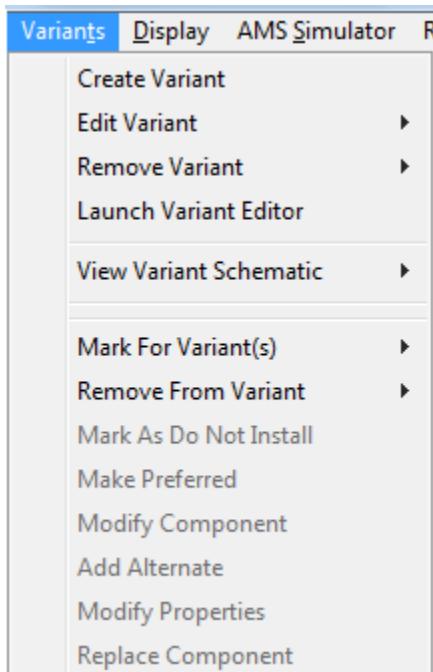
Variant information can now be captured directly on the schematic canvas using new menu options for variant management in Design Entry HDL (DE HDL). Variant-related changes in the schematic canvas are automatically synchronized with Variant Editor and are saved in the variant database. You do not need to manually save your variant-related changes in the schematic.

In the schematic canvas, the following can be done with variants:

- [New Variant Menu Options in Schematic Canvas](#)
- [Creating Variants](#)
- [Editing Variants](#)
- [Removing Variants](#)
- [Launching Variant Editor](#)
- [Opening Variant Schematic View](#)
- [Working with Hierarchical Variants](#)
- [Marking Components for Variants](#)
- [Removing Components from Variants](#)
- [Marking Components as 'Do Not Install'](#)
- [Marking Components as Preferred Components](#)
- [Modifying Components](#)
- [Adding Alternates for a Component](#)
- [Modifying Properties](#)
- [Replacing Components in Functions or Variants](#)

## New Variant Menu Options in Schematic Canvas

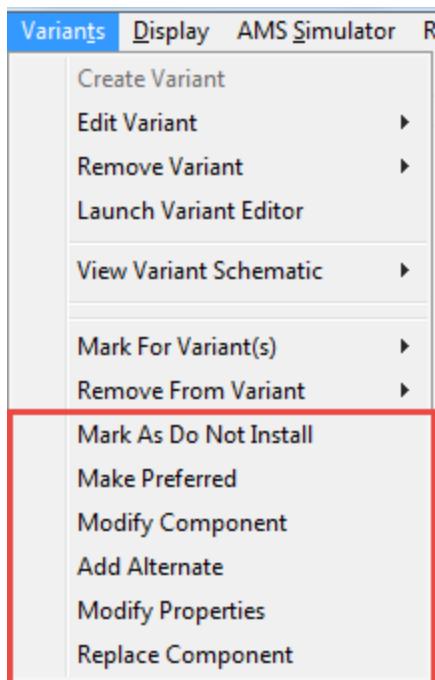
When you open a design in Design Entry HDL, the design is displayed in the base schematic view. The variant menus provide commands that enable you to manage variants in your design. These commands let you create, edit, and remove variants. You can also select multiple objects on the schematic and mark them for modification in variants.



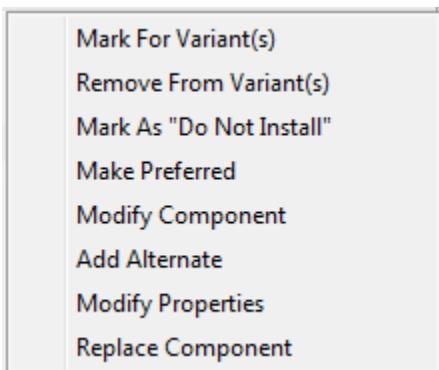
## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL

The menus also provide you the option to view variants on the schematic. When you choose a variant to view in the schematic, variant-specific data is displayed in the schematic and additional menu commands are enabled that allow you to perform variant-specific operations.

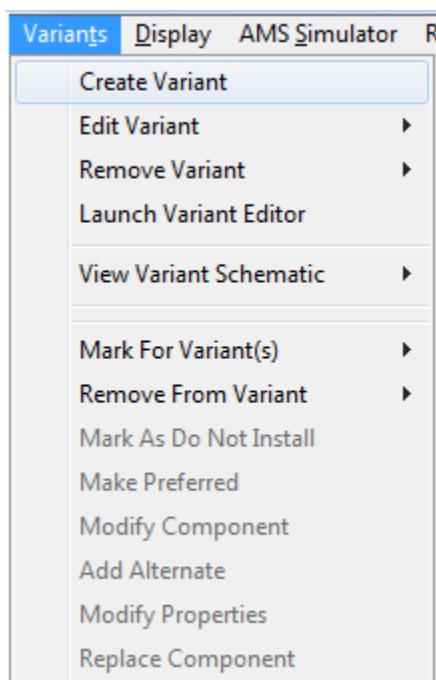


You can also perform variant-specific operations using the pop-up menu. In the variant schematic view, select a component in the schematic and right-click to access the pop-up menu.

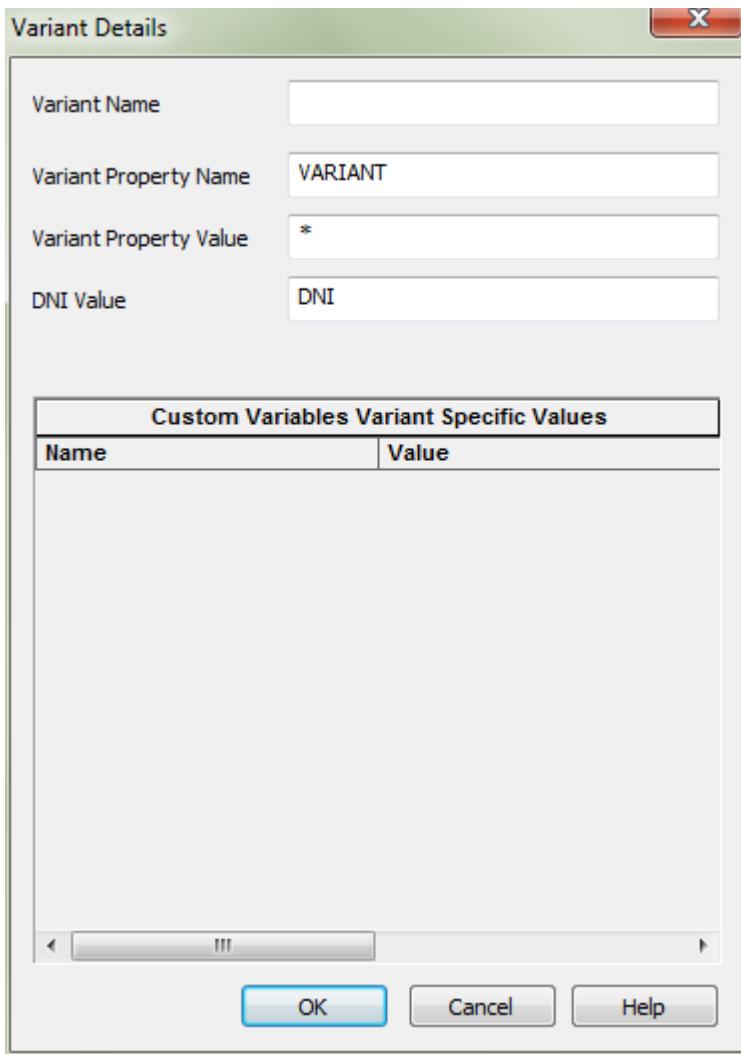


## Creating Variants

1. In the schematic canvas, choose *Variants* — *Create Variant* to create a new variant.



The Variant Details dialog box is displayed.



**2. Enter details as required.**

For more information on working with this dialog box, see the Managing Variants section of the Design Variance User Guide.

## Editing Variants

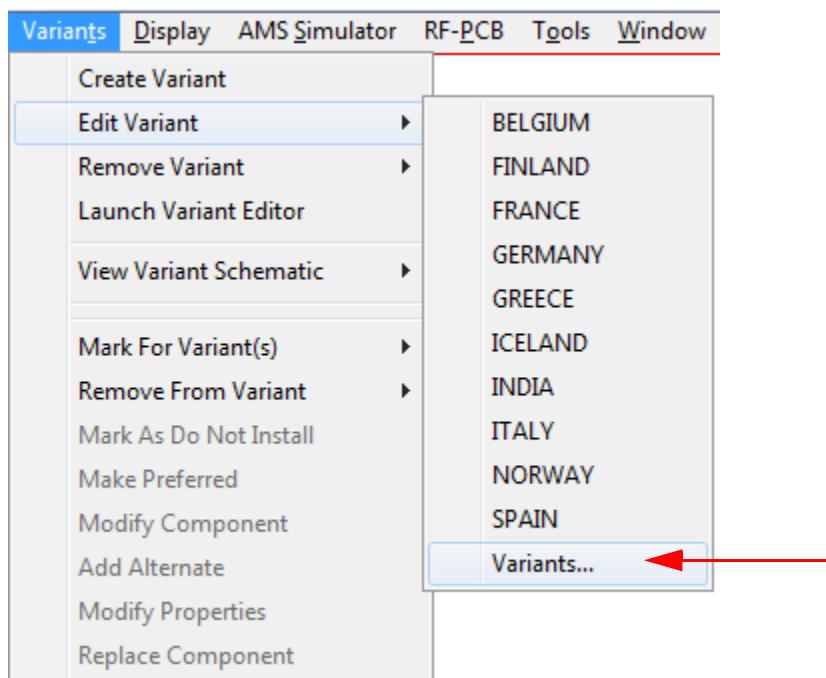
1. To edit an existing variant, choose *Variants — Edit Variant — <variant name>*.

**Note:** A new CPM directive, VAR\_MENU\_COUNT, in the START CONCEPTHDL section allows you to control the number of variants that will be displayed in the Variants menu. The maximum number of variants that can be displayed is 20 and the minimum is 1. By

## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL

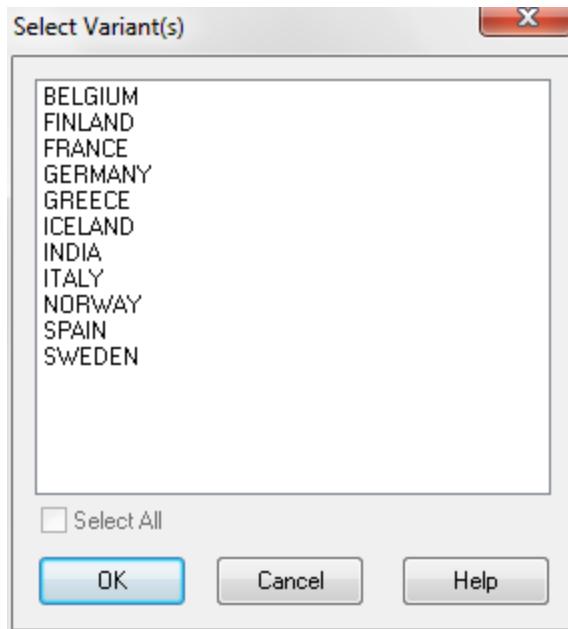
default, if the number of variants in your design exceeds ten, the Edit Variants sub-menu lists the Variants option (indicated by an arrow in the image),



## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL

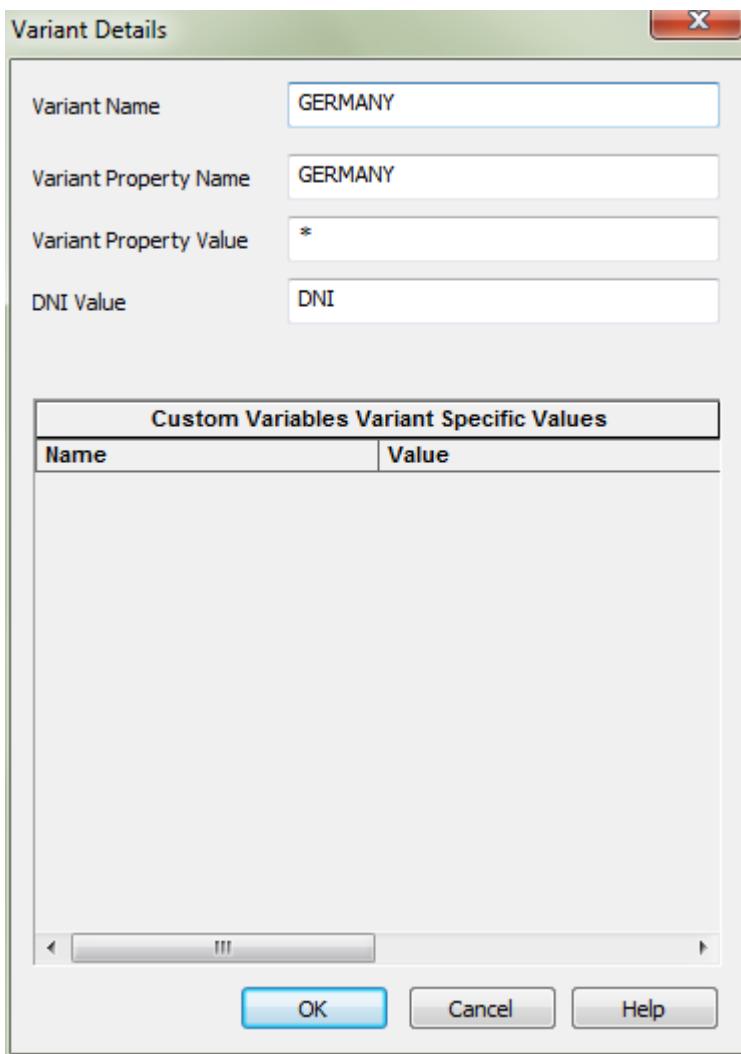
If you want to view the full list of variants in your design, click *Variants*. The Select Variant(s) dialog box displays. Select the variant you want to edit. Note that you cannot check the Select All box when editing a variant.



## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL

The Variant Details dialog box is displayed with the details of the selected variant.

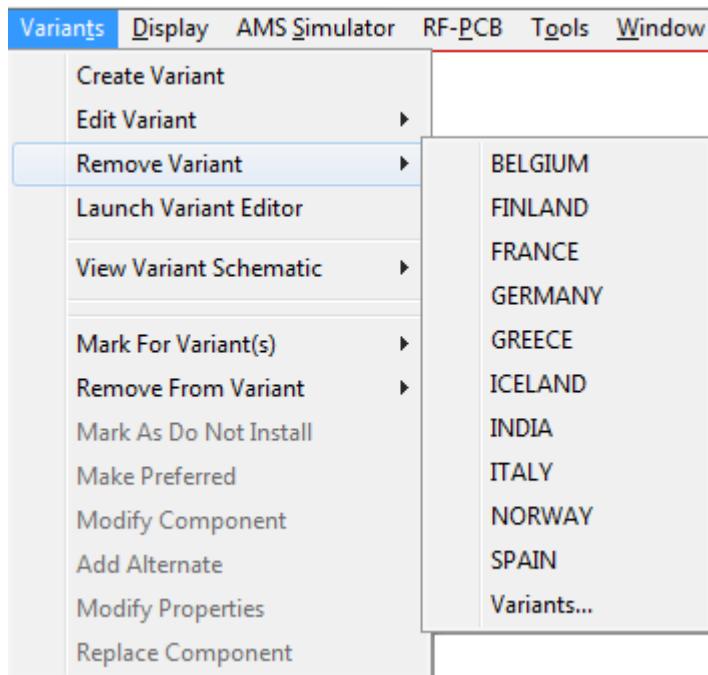


2. Edit the details as required.

For more information on working with the Variant Details dialog box, see the Managing Variants section of the Design Variance User Guide.

### Removing Variants

You can remove a variant from the variant database using *Variants — Remove Variants — <variant name>*.

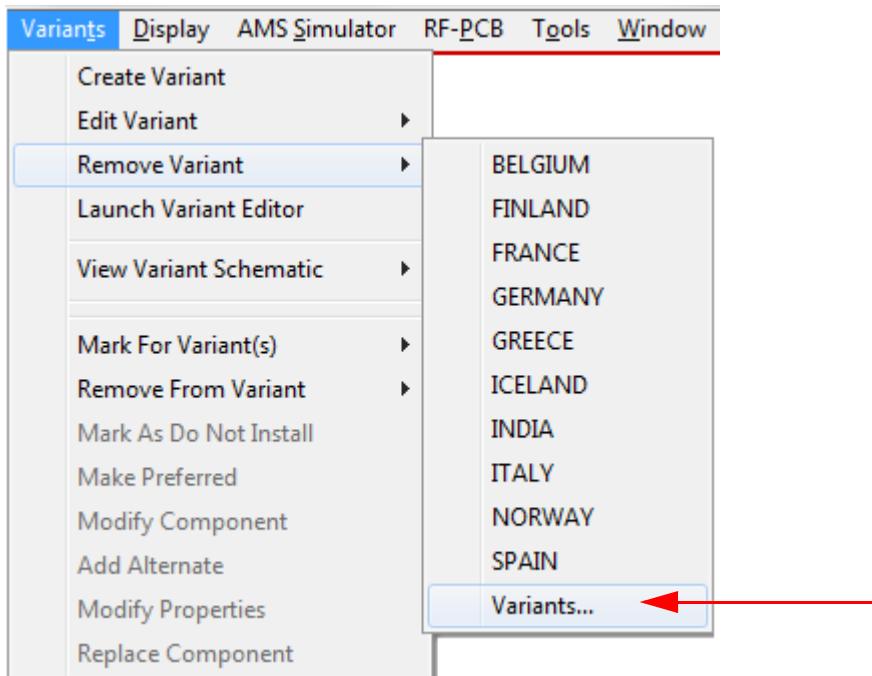


The selected variant is removed from the variant list in `variant.dat`.

# Allegro Design Entry HDL User Guide

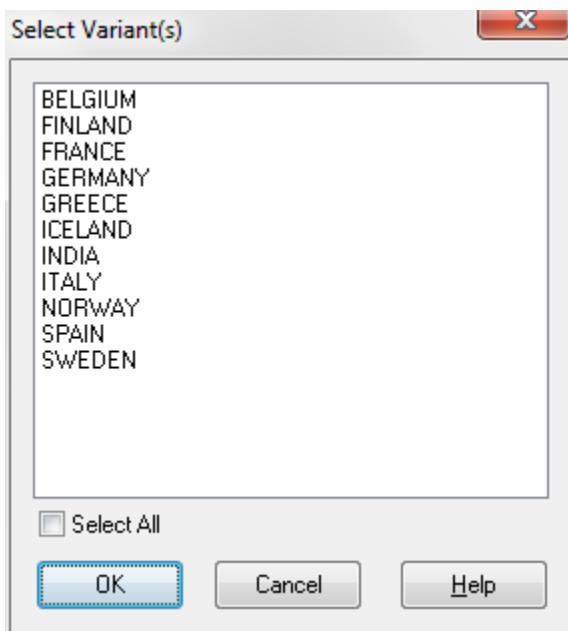
## Managing Variants in Design Entry HDL

1. To remove multiple variants at a time, choose *Variants — Remove Variant — Variants*.



The Remove Variant menu option displays a limited number of variants. To view the full list of variants in your design, click Variants. You can select multiple variants in this list and remove

The Select Variant(s) dialog box appears.



To select non-consecutive items, click on a variant name, then, keeping the Ctrl key pressed, choose the other variants that you want to delete. To choose items serially, click on the first variant name. Keeping the Shift key pressed, choose the last variant. This selects all the variants from the first variant to the last clicked variant. Click *OK*.

You can also check the Select All box to select all the variants.

2. In the confirmation box that appears, click *Yes* or *No* to delete the variants. If you click *Yes*, the variants are deleted from the variant database.

For more information on removing variants, see the Design Variance User Guide.

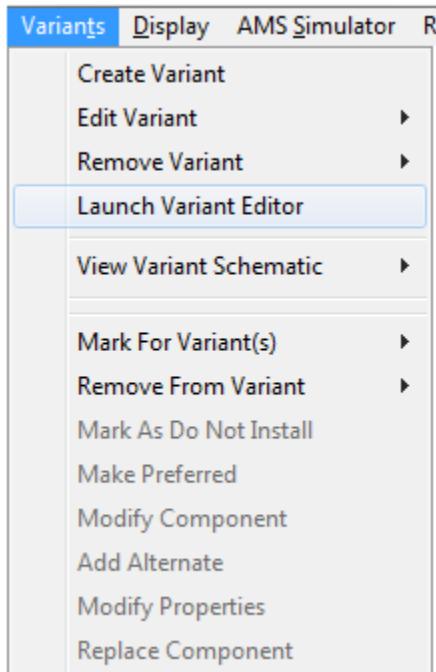
## Launching Variant Editor

Certain advanced operations such as creating functions and groups, defining components as alternates, marking an alternate as a preferred component, generating BOM reports, or replacing components can only be done in Variant Editor.

## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL

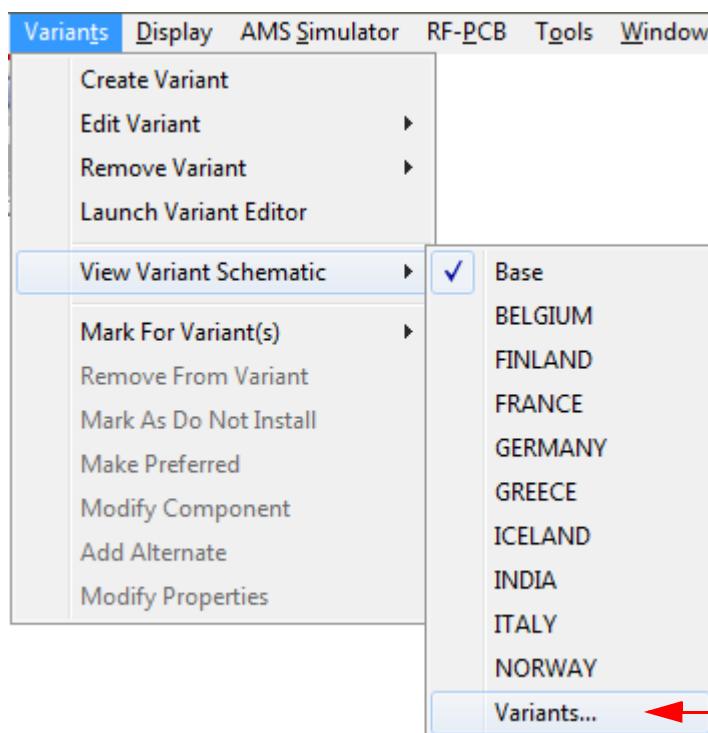
To launch Variant Editor from the schematic, choose *Variants — Launch Variant Editor*.



Variant Editor appears displaying information for all variants in your design. You can perform all variant-related tasks in Variant Editor. On saving changes in Variant Editor, the schematic is automatically synchronized and variant-specific changes can be viewed in the schematic sheet.

## Opening Variant Schematic View

To view changes made for a particular variant, you can open a variant schematic view. You can switch from the base schematic view to any variant view using *Variants* — *View Variant Schematic* — <variant name>.



The View Variant Schematic menu option displays a limited number of variants. To view the full list of variants in your design, click Variants.

When you click the *Variants* sub-menu option, the Select Variant(s) dialog box opens and displays a list of the available variants in your design. The base schematic is selected by default. When you select a variant from the list, the schematic view changes to display the variant-specific view with variant data annotated on the schematic.

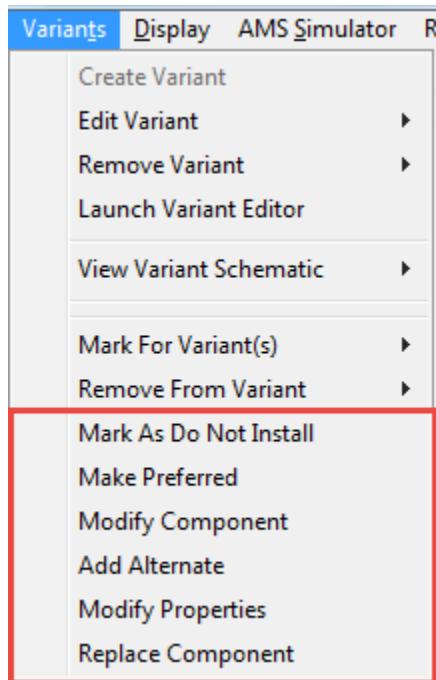
When you switch to the variant view, the title bar of the window indicates that you are now in the viewer mode. This is a viewer mode only and schematic changes made in this mode cannot be saved. You can however make changes, plot, or publish the modified schematic.



## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL

In the variant schematic view mode, you will notice that additional menu options are now available under *Variants*.



For more information about the variant view, refer to the Design Variance User Guide.

## Working with Hierarchical Variants

Design Entry HDL supports hierarchical variant tasks in the schematic canvas. Variants defined in lower-level hierarchical blocks can be applied on block instances in higher-level blocks. These variants are referred to as hierarchical variants in DE-HDL documentation.

All reusable blocks that contain variant definitions can be used for specifying block-level variants. For more information about design reuse, its advantages, and how to implement it, refer to *Design Reuse Tutorial* and the Design Reuse chapter in *PCB Flows*.

## Enabling Hierarchical Variants

To work with lower-level hierarchical variants, variant data from lower-level blocks must first be imported into the top level design. After importing the data, variants from lower-level blocks are available for application to block instances.

Variants can only be applied to block instances; they cannot be applied on individual components. Also note that only those variants that are defined in lower-level blocks can be applied to block instances. This allows you to include a variant from a lower-level block in the top-level variant.

To enable hierarchical variants for a block, do the following:

1. Create a reuse block then place the block where required. For example, if you have a MID reuse block with variants, place the block in the TOP block.
2. Select the block for which you want to enable hierarchical variants.
3. Choose *Enable Hierarchical Variants* from the Variants menu or right-click on the block and choose *Variants — Enable Hierarchical Variants*.

**Note:** If the Enable Hierarchical Variants option is disabled, it could be because of the following reasons:

- There is no variant information in the lower-level block (reusable block).
- Variants are not defined for the top-level block.
- The block has not been identified for hierarchical variants.

For details about HIER\_BOM\_PROP, refer to the Creating Hierarchical Variant BOM Reports section in *Allegro Design Entry HDL Utilities User Guide*.

## Applying Hierarchical Variants

When you enable the hierarchical variants option for a block, this means that the lower-level variant file has been loaded into the top block. You can now apply the hierarchical variant, which means that you apply a selected variant to the root level.

The Apply Block Variant option is activated after you enable the hierarchical variants option for a block. To apply a variant, first select the block to which you want to apply a variant in the schematic canvas.

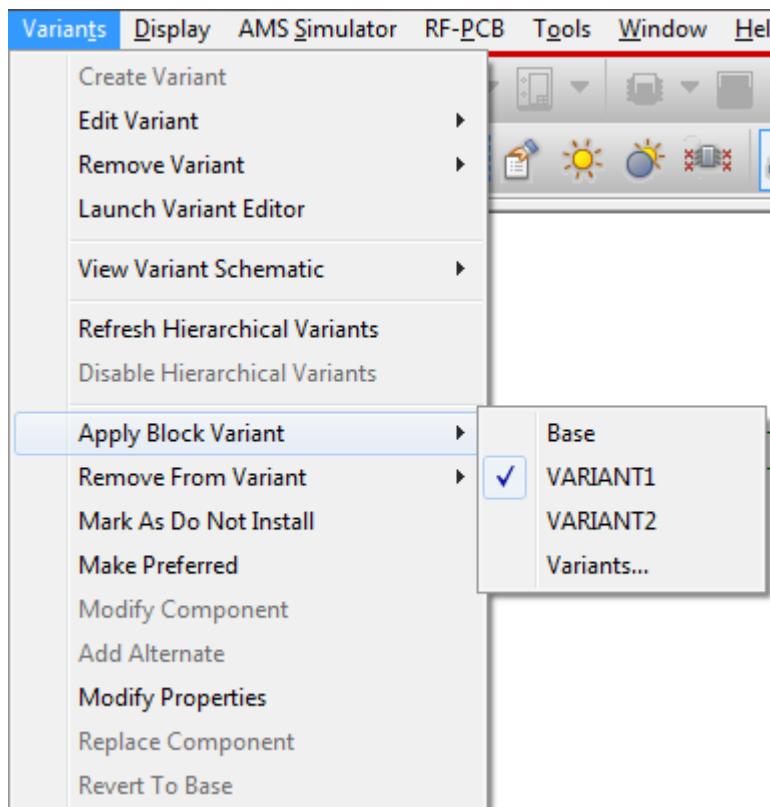
Switch to the variant view mode, then do one of the following:

- Choose *Variants — Apply Block Variant — <variant name>*.

## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL

- Right-click and choose *Variants — Apply Block Variant — <variant name>* from the pop-up menu.



If you have a number of variants, click the *Variants* option and select a variant from the Select Variant(s) list box. When you apply a variant to a block, the variant name is displayed in the canvas.



## Disabling Hierarchical Variants

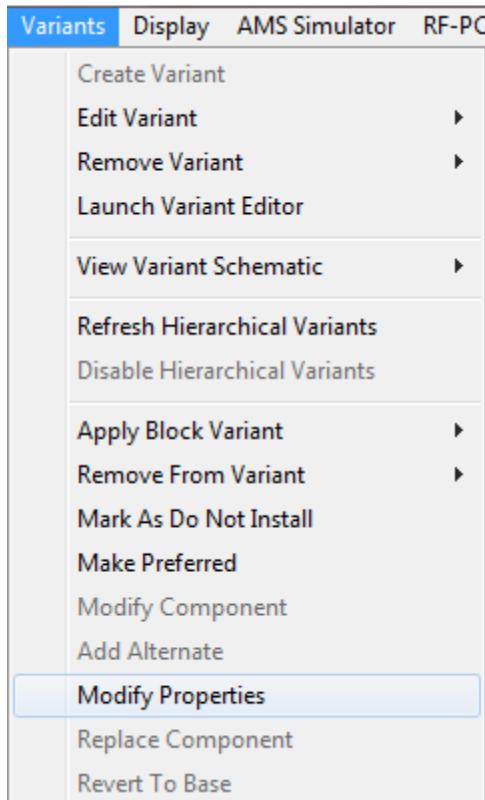
If you have enabled hierarchical variants for a particular block, you can choose to disable it for that block. When you disable hierarchical variants for a block, hierarchical variants for the lower-level reusable block are unloaded and the block reverts to the base state.

To disable the hierarchical variant option for a block, select a block on the schematic then choose *Variants — Disable Hierarchical Variants*, or right-click on a hierarchical variant-enabled block and choose *Variants — Disable Hierarchical Variants* from the pop-up menu.

## Modifying Properties for Hierarchical Variants

You can add customized properties for hierarchical BOM generation or modify the properties that you may have previously added for a block.

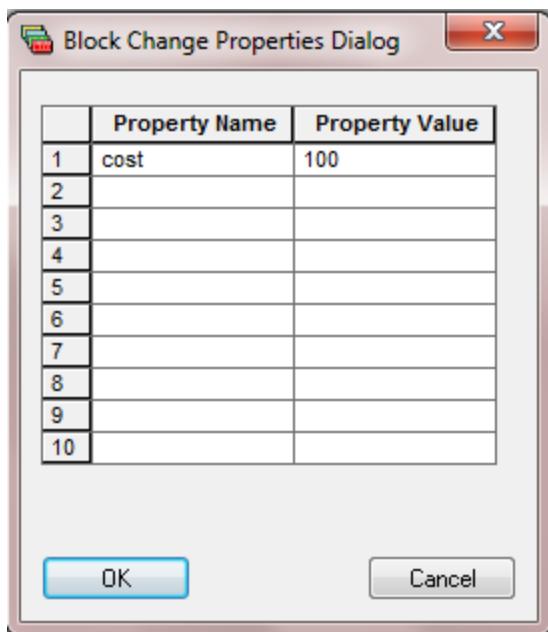
1. Ensure that you are in the required variant view, then choose *Variants — Modify Properties*.



2. Select the block whose properties you want to modify or for which you want to add customized properties for hierarchical BOM generation.

You can also first select a block in the schematic canvas, right-click, and choose *Variants — Modify Properties* from the pop-up menu.

The Block Change Properties Dialog box appears displaying customized, hierarchical BOM-specific properties that you may have previously added for a block.



3. Add or modify properties as required and click *OK*.

These properties are stored for the selected variant and are available for hierarchical BOM report generation. The following image illustrates a BOM report with the cost property.

## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL



Var Status	Subdesign Suffix	Block Variant	Text Code	Qty	Unit Price	Cost	PART_NUMB
ref	A	VARIANT1		1	100	100.00	
	B	VARIANT2		1	?	?	
				2	?	?	PRELMBE04 4155/121
				4		100.000	
					0	0.000	

### Custom Property for Hierarchical Variants

If you have multiple reuse blocks with variants in the design and you would like to use hierarchical variants for only a few selected blocks, you can specify a custom property for such blocks to identify them for hierarchical variants use. The custom property name can be configured using a project (.cpm) file directive, HIER\_BOM\_PROP, in the .cpm file.

For details about HIER\_BOM\_PROP, refer to the Creating Hierarchical Variant BOM Reports section in *Allegro Design Entry HDL Utilities User Guide*.

### Marking Block Instances as ‘Do Not Install’

When you include a variant from a lower-level block in the top-level variant, you can mark the complete block instance as a Do Not Install block. A block marked as DNI is displayed with a cross icon across it.

If you mark a block as a DNI instance, all the components in that block, under the hierarchy and up to the leaf level, are marked as DNI. For example, if a block, say MID, has two more blocks under it — LOW1 and LOW2 — and you mark MID as a DNI block, all the components in MID and in LOW1 and LOW2 are marked as DNI components.

### Refreshing Hierarchical Variants

There may be times when you modify reuse blocks. The variant information of these blocks may have also changed along with other changes. Because these blocks are reused, the

blocks need to be packaged again to update the packaging information. If there are packaging changes in the block, it is recommended that you open Variant Editor and save the variant database to sync the packaging changes with the variant data. This ensures that the packaging data and variant data are synchronized.

Once the reuse block is reintegrated in the top level design, package the top-level design. After the design has been packaged, the variant information from the lower-level reuse block should be refreshed. For example, if the reuse block MID has a variant and has been reused in the TOP block, package TOP then perform a refresh variant operation on the MID block.

The Refresh Hierarchical Variants option is activated only if and after you enable hierarchical variants for a block. To refresh a block, first select the block in the canvas then choose *Variants — Refresh Hierarchical Variants*, or right-click on the block and choose *Variants — Refresh Hierarchical Variants* from the pop-up menu.

### Generating Hierarchical Variants BOM Reports

With the inclusion of hierarchical variants in the Design Entry HDL schematic, you can now also generate hierarchical BOM reports. The hierarchical BOM report has a single entry for each reuse block instance, which has been identified for hierarchical variants. Components from block instances are excluded from the main BOM.

For more information about hierarchical variant BOM reports, refer to the Creating Hierarchical Variant BOM Reports section in *Allegro Design Entry HDL Utilities User Guide*.

### Marking Components for Variants

You can identify components for a variant and change or customize their values. You can mark components for a single variant or for multiple variants at a time. You can also select multiple components and mark them for one variant or multiple variants.

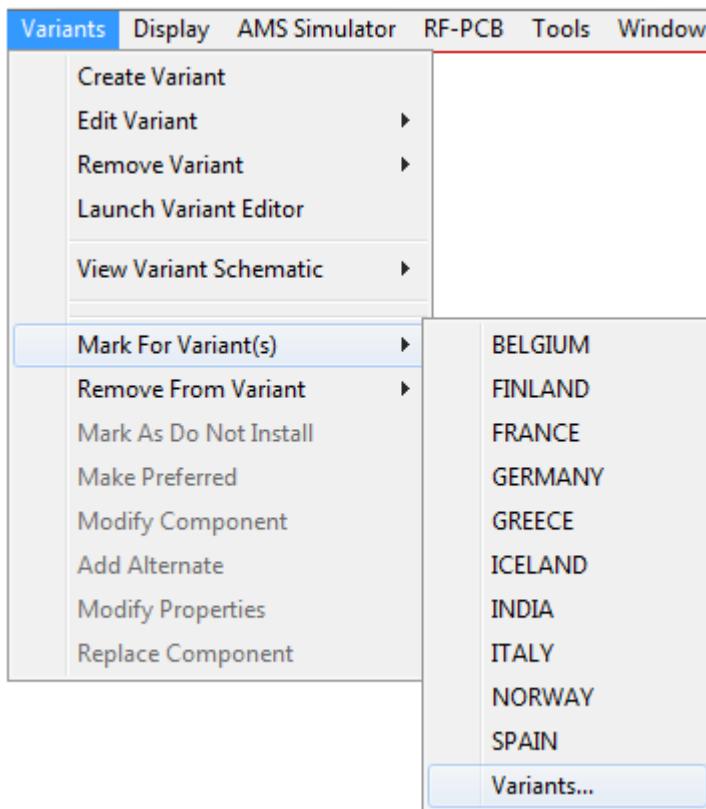
**1. Do one of the following:**

- To mark a component for a specific variant, choose *Variants — Mark for Variant(s)* — <variant name>.   
Select the component in the schematic that you want to mark for the selected variant.
- Select a component in the schematic, right-click and select *Variants — Mark for Variant(s)*.   
Select the variant for which you want to mark the selected component.

# Allegro Design Entry HDL User Guide

## Managing Variants in Design Entry HDL

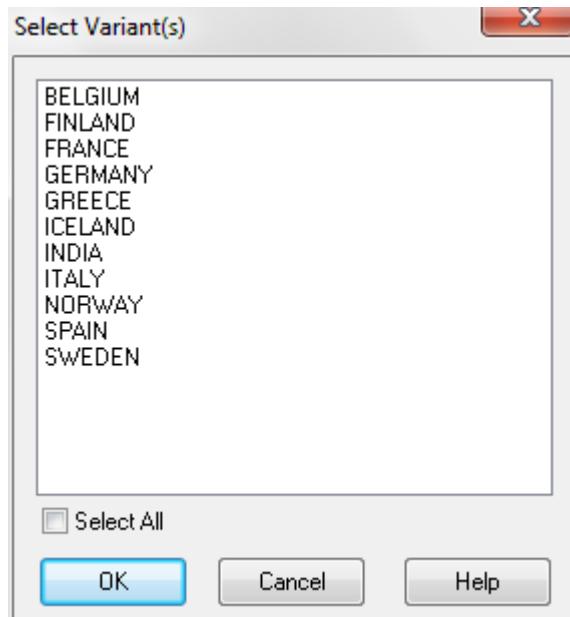
- ☐ To mark components for multiple variants at a time, choose *Variants — Mark for Variant(s) — Variants*.



## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL

The Select Variant(s) dialog box appears with the list of all the variants for the design.



To select non-consecutive items, click on a variant name, then, keeping the Ctrl key pressed, choose the other variants for which you want to mark a selected component.

To choose items serially, click on the first variant name. Keeping the Shift key pressed, choose the last variant. This selects all the variants from the first variant to the last clicked variant. Click *OK* in the dialog box.

You can also check the Select All box to select all the variants.

If you want to mark multiple components for a variant, choose the components in the schematic then do one of the following:

- ❑ Right-click on the selected components and select *Variants — Mark for Variant(s)* from the pop-up menu.
- ❑ Choose *Variants — Mark for Variant(s)* — <variant name> to mark the selected components for a specific variant.
- ❑ Choose *Variants — Mark for Variant(s)* — *Variants* then choose the variants for which you want to mark the selected components.

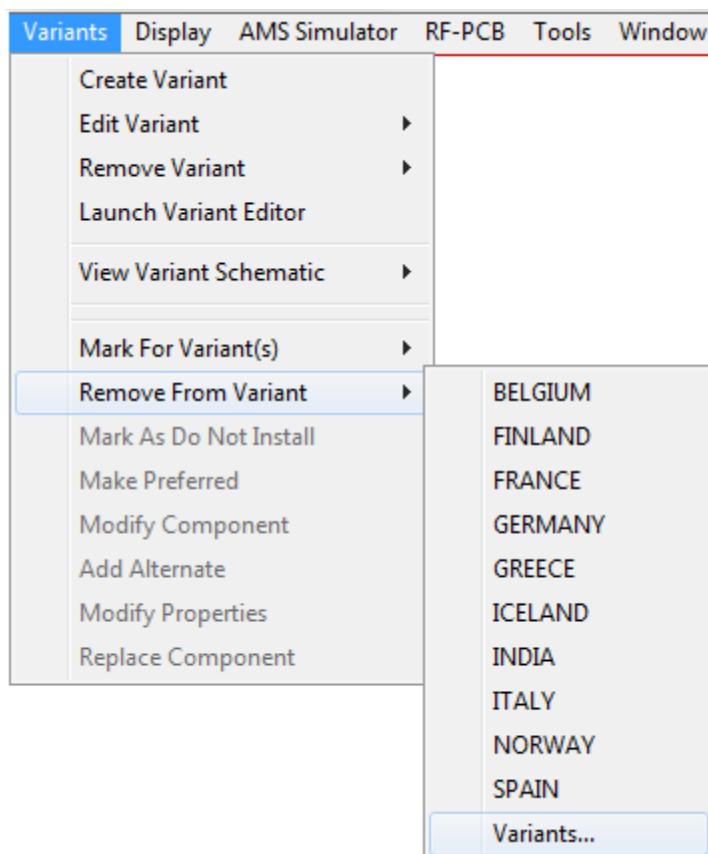
You can launch Variant Editor and view the details of the selected variant. All the components marked for the variant are visible in Variant Editor. For more information on marking components for variants, see the Design Variance User Guide.

## Removing Components from Variants

You can remove one or more components from selected variants. To do so, first ensure that you are in the required variant view.

1. To remove a component from a selected variant, do one of the following:

- Choose *Variants — Remove from Variant* then select the components you want to remove.



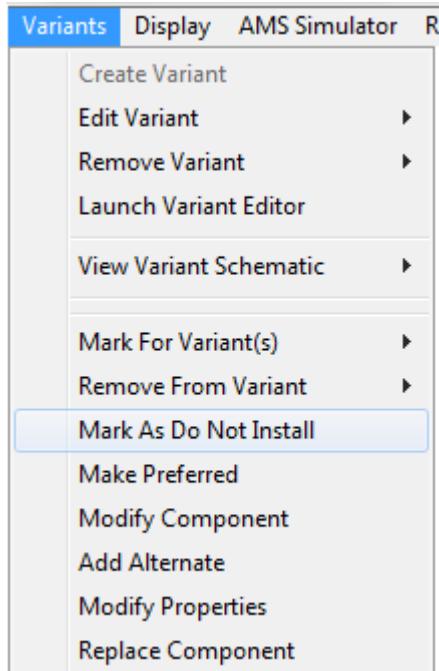
- Select one or more components in the schematic canvas, right-click, and select *Variants — Remove from Variant*.
2. In the confirmation box that appears, select Yes or No to remove the component from the selected variant.

For more information on removing components from variants, see the Design Variance User Guide.

## Marking Components as ‘Do Not Install’

In the variant schematic view, you can mark one or more components as Do Not Install (DNI) components.

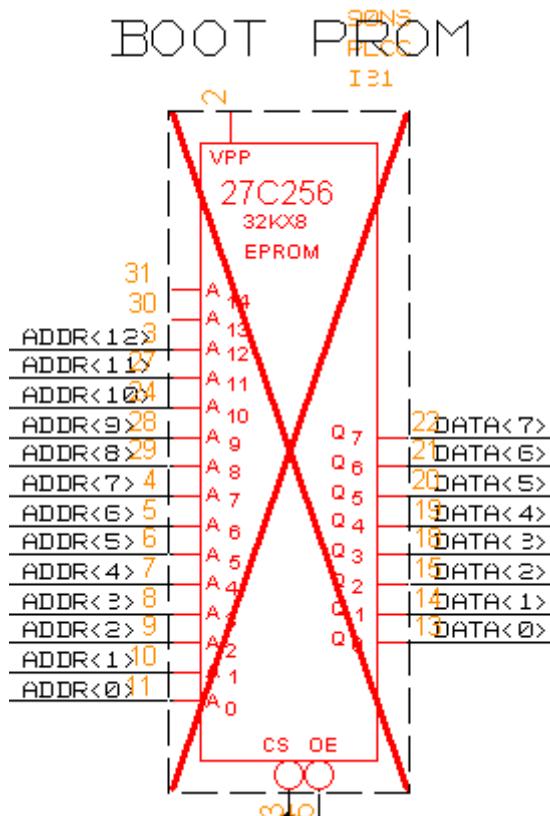
1. To mark a component as DNI, choose *Variants — Mark As Do Not Install*.



2. In the canvas, select the component that you want to mark as DNI.

**Note:** You can also first select one or more components in the schematic canvas, right-click, and choose *Variants — Mark As “Do Not Install”* from the pop-up menu.

The selected components are marked as DNI in the schematic and are displayed based on your settings in DE-HDL. For details on these settings, see the *Variant Overlay Options* section of the *Design Variance User Guide*.



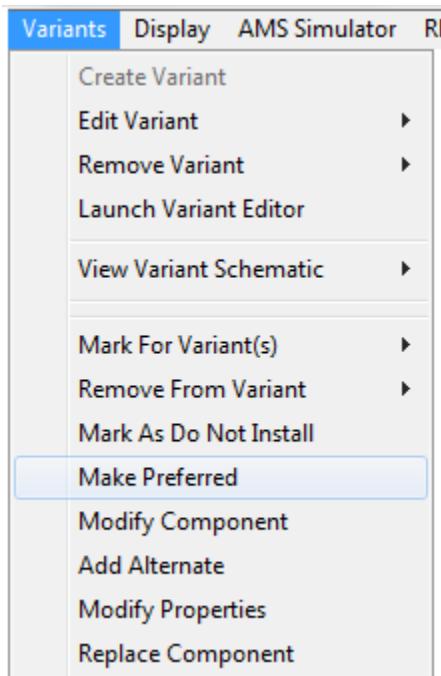
The X graphic that runs across the bounding box of the component is the default icon to indicate the DNI status, and cannot be modified.

For more information on marking components as DNI, see the *Design Variance User Guide*.

## Marking Components as Preferred Components

You can mark one or more components as preferred components in a design. If the selected component was previously marked as DNI or was already a preferred component with a different value, the selected component reverts to the base schematic value. The component also reverts to the display settings defined for the base schematic value (for details, see the *Variant Overlay Options* section of the *Design Variance User Guide*).

1. To mark a component as preferred components, ensure that you are in the variant schematic view then choose *Variants — Make Preferred*.



2. In the schematic canvas, select one or more components that you want to make preferred components.

**Note:** You can also first select components in the schematic canvas, right-click, and choose *Variants — Make Preferred* from the pop-up menu.

You can launch Variant Editor and view the details of the selected variant. All the components marked as preferred components for the variant are visible in Variant Editor. For more information on marking components as preferred, see the Design Variance User Guide.

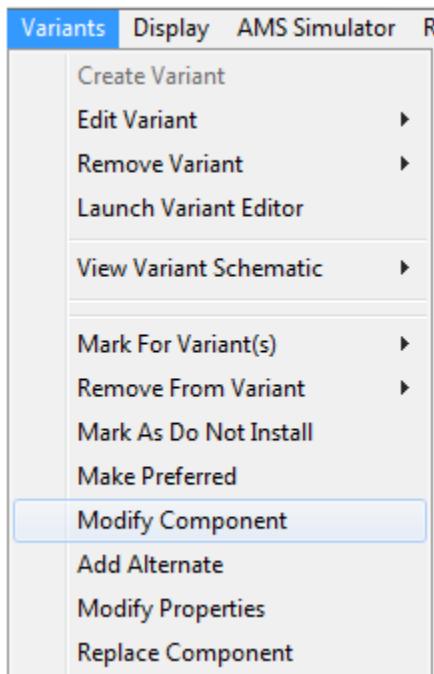
## Modifying Components

You can modify components by changing their attribute values. To modify components, ensure that you are in the required variant view.

# Allegro Design Entry HDL User Guide

## Managing Variants in Design Entry HDL

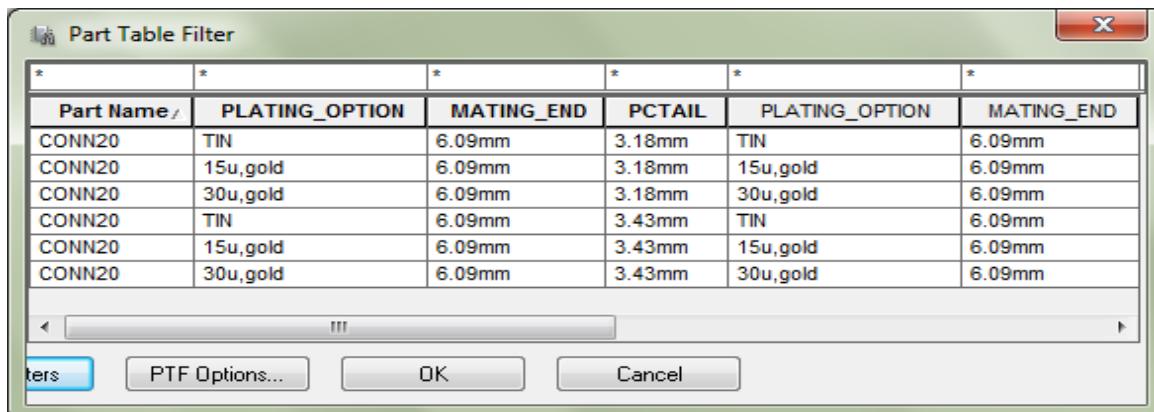
### 1. Choose *Variants — Modify Component*.



### 2. Select the components you want to modify.

**Note:** You can also first select one or more components in the schematic canvas, right-click, and choose *Variants — Modify Component* from the pop-up menu.

The Part Table Filter dialog box displays the PPT rows that match the footprint of the selected components. You can also define a list of compatible footprints and then use this command to select a row containing a compatible footprint.



### 3. Select the PPT row as required then click *OK*.

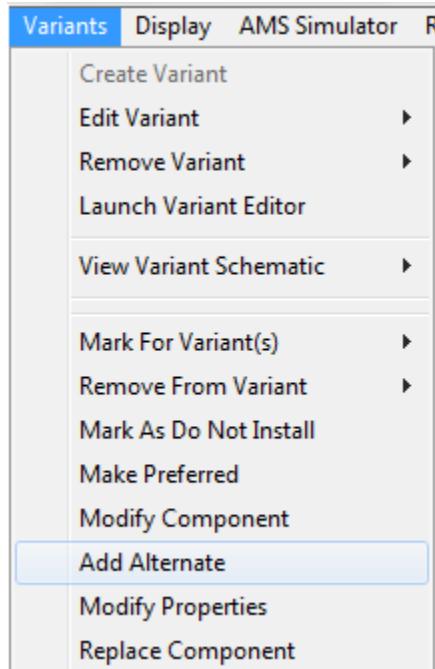
The Part Table Filter dialog box closes and the modified component displays in the color set for variant-specific components (see *Tools — Options — Variant Overlay Options*).

For more information on changing values for components, see the Design Variance User Guide.

## Adding Alternates for a Component

To add an alternate for a component in the schematic canvas, ensure that you are in the variant schematic view.

### 1. Choose *Variants — Add Alternate*.



### 2. Select the required component on the canvas.

**Note:** You can also first select a component in the schematic canvas, right-click, and choose *Variants — Add Alternate* from the pop-up menu.

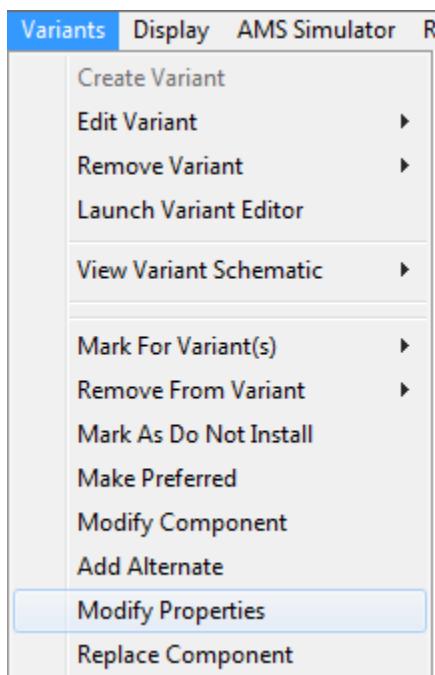
The Part Table Filter dialog appears with the PPT rows that match the footprint of the selected component. Choose the new preferred value and click *OK*. You can add multiple alternates for a component in the variant schematic view. To set the order of preference for the alternates, launch Variant Editor and perform the required operation.

For more information on adding alternates, see the Design Variance User Guide.

## Modifying Properties

You can add or modify the user-defined properties of one or more components for a specific variant.

1. Ensure that you are in the required variant view, then choose *Variants — Modify Properties*.



2. Select the components whose properties you want to modify.

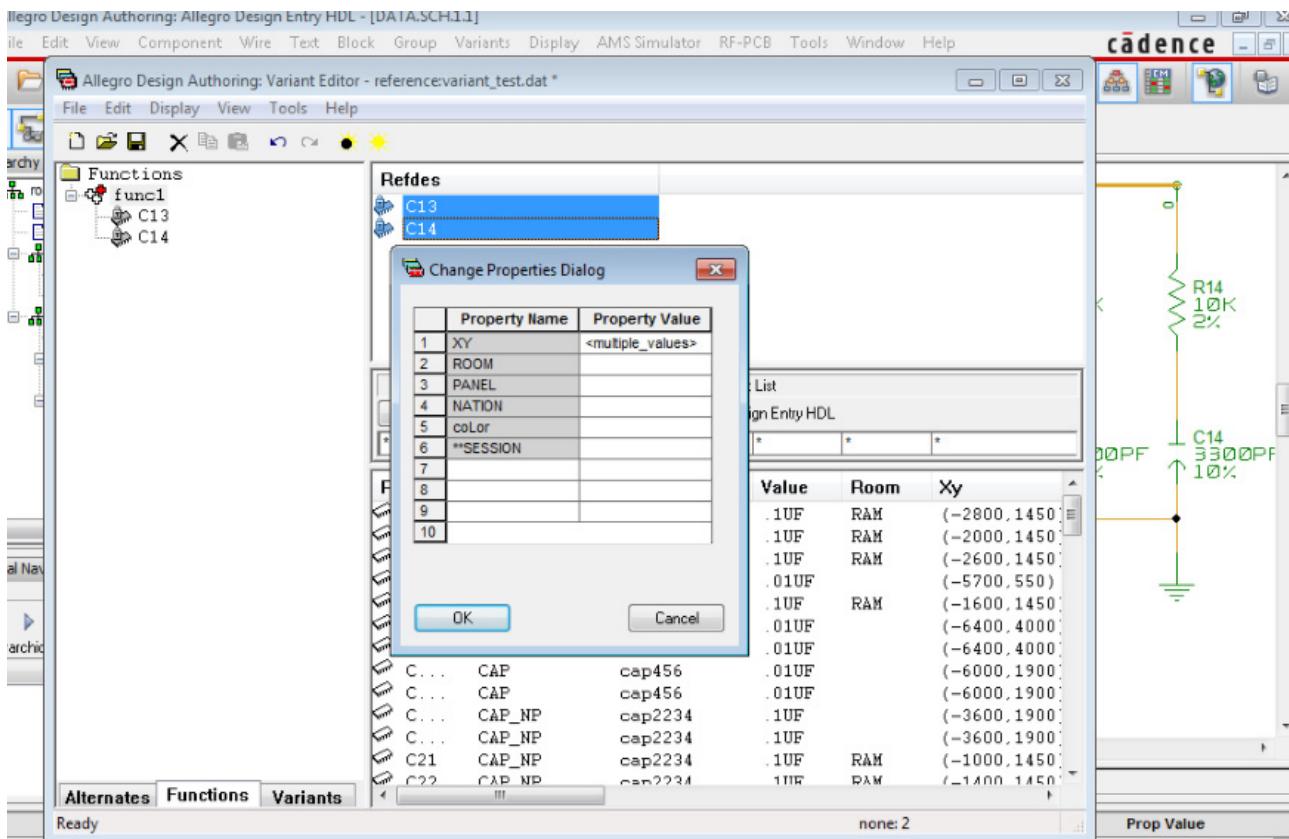
**Note:** You can also first select a component in the schematic canvas, right-click, and choose *Variants — Modify Properties* from the pop-up menu.

The Change Properties dialog box appears displaying user-defined properties for the selected component.

## Allegro Design Entry HDL User Guide

### Managing Variants in Design Entry HDL

If the components you select have the same properties but with different values, Variant Editor will display the value as <multiple\_values>.



To change properties for multiple components at a time, the components must have the same logical part names and compatible JEDEC\_TYPEs. If you want to use alternate values that do not have the same JEDEC\_TYPE, see the *Using Compatible JEDEC\_TYPEs* section in *Design Variance User Guide*.

All the properties that were configured as custom columns in Variant Editor are displayed in this dialog box. Since part table (PTF) properties cannot be modified, they are not displayed in this dialog box.

Custom columns are configured in Variant Editor using *View — Customize Columns*. See the *Customize Columns* section of the *Design Variance User Guide* for details.

**Note:** If you want to modify a component's PTF properties, use the Change Value (in Variant Editor) command or the Modify Component command (in the schematic).

3. Add or modify properties as required and click OK.

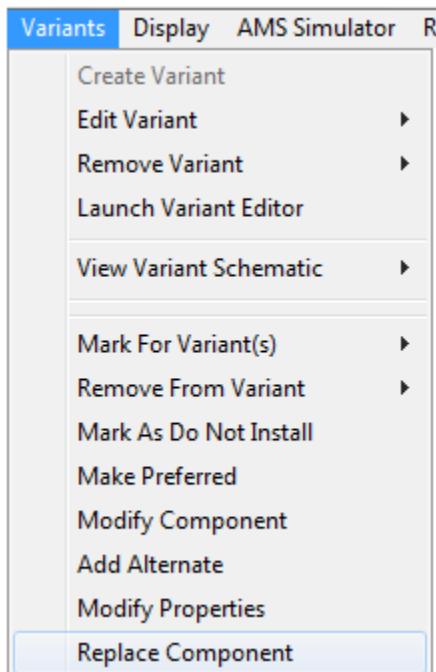
These properties are stored for the selected variant and are available for BOM generation.

## Replacing Components in Functions or Variants

After defining functions and variants, Variant Editor allows you to replace a component in a function or a variant with another component, if needed. If a component is part of an alternate group, it cannot be replaced. The replaced component can have a different name or properties, but it should have the same or compatible footprint (JEDEC\_TYPE property value).

To replace a component, do the following:

1. Ensure that you are in the required variant view, then choose *Variants — Replace Component*.



2. Select the required component on the canvas.

**Note:** You can also first select a component in the schematic canvas, right-click, and choose *Variants — Replace Component* from the pop-up menu.

The Replace Component dialog box displays all the libraries and components in your design. You can select any part available in your design library as a replacement.

## Allegro Design Entry HDL User Guide

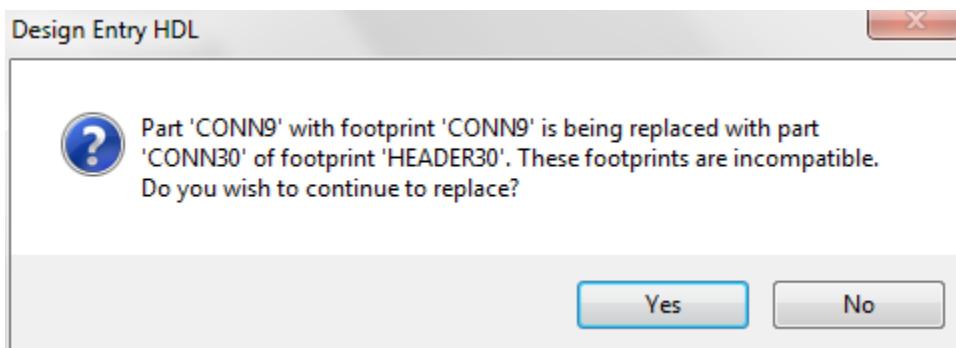
### Managing Variants in Design Entry HDL

- a. Select a cell in the Cells list box.

The physical part table (PPT) rows belonging to the selected cells appear in the Search Results pane.

- b. Choose the PPT row with which you want to replace the selected component.
- c. Click *Replace*.

When you replace a component in Variant Editor, the tool checks whether the two components—the component that is being replaced and the component that will replace the selected component—have the same or compatible footprints, that is, JEDEC\_TYPE properties. If the footprints are not compatible, a warning message is displayed.



It is recommended that a component not be replaced with another component with a non-compatible footprint. If you continue with the replace operation with a non-compatible footprint, all customized changes on the original component are lost.

If you click Yes, the Replace Component dialog box is closed and the component is replaced in the variant on the schematic. In Variant Editor, the preferred value of the former component (one that has been replaced) is displayed as a grayed out row.

**Note:** The Replace Component functionality is not backward compatible. If you replace a component in a variant then open the design in an older release, errors may appear.

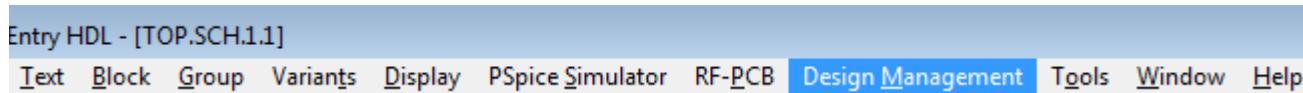
For more information on replacing components, see *Design Variance User Guide*.

---

## Design Management in Design Entry HDL

---

Using the Design Management option in DE-HDL, you can enable structured team design for a design to shorten your design cycle. You can manage the design on a file system, in a folder on SharePoint, or in a folder in PTC Windchill.



This functionality enables designers to manage a multiblock hierarchical, or flat, design with multiple users concurrently modifying portions of the logical and physical design. This helps design teams reduce the overall design time. Groups of engineers work together on individual portions of the design to develop a system and control how the portions of the system are integrated together.

The Design Management feature also allows you to manage changes to a design and version control (referred to as data management in the Design Entry HDL documentation).

With this feature, you can do the following:

- Prevent unintended modification of a design (control access)
- Provide version control and version history for all design changes
- Manage design modifications from multiple sources/sites
- Facilitate communication/notification among the design team
- Maintain the design data at a central location
- Integrate all design components into a released design



*Important*  
The design management (team design and data management) functionality is available to you depending on your licenses.

## **Allegro Design Entry HDL User Guide**

### Design Management in Design Entry HDL

---

For more information about design management (team management, version control, and Allegro Pulse features), refer to the following documents:

- Allegro Design Management User Guide*
- Allegro Pulse documentation

## Plotting Your Design

---

The plotting facility enables you to make hardcopies of your designs for debugging or documentation. You can use any of the plotters that are configured with your system. You can take plots on various global or local paper sizes. Select different orientations, scalings and other options. You can also customize plotting at the project level or at the site level.

Depending upon the methods used to plot and customize designs, Design Entry HDL provides the following plotting modes:

- Windows Plotting on Windows
- You should use only those paper sizes that are supported by the plotter you have chosen to plot the design on.
- Hierarchical Plotting

### Windows Plotting on Windows

This facility is referred to as Windows plotting because it uses Windows services to generate the plot output. Also, it uses the Windows way of storing information about plotters in the registry. You can customize the plotter settings according to your needs by making some changes in the registry.

Project and site level customization of some aspects of plotting can be done through project directives also.

**Note:** If you want to plot cross-referencing information on your design, first run the *Cross Referencer* tool on your design. The tool generates signal and part cross references for flat and hierarchical schematic drawings. It places the signal cross references directly on the page where the signals appear and creates text reports that contain the list of signal and part cross references. You can then plot the design to view all the additional information that has been attached to the design.

For plotting a design, follow these steps:

1. Setting Up Windows Plotting Options

### 2. Previewing the Design

### 3. Plotting the Design

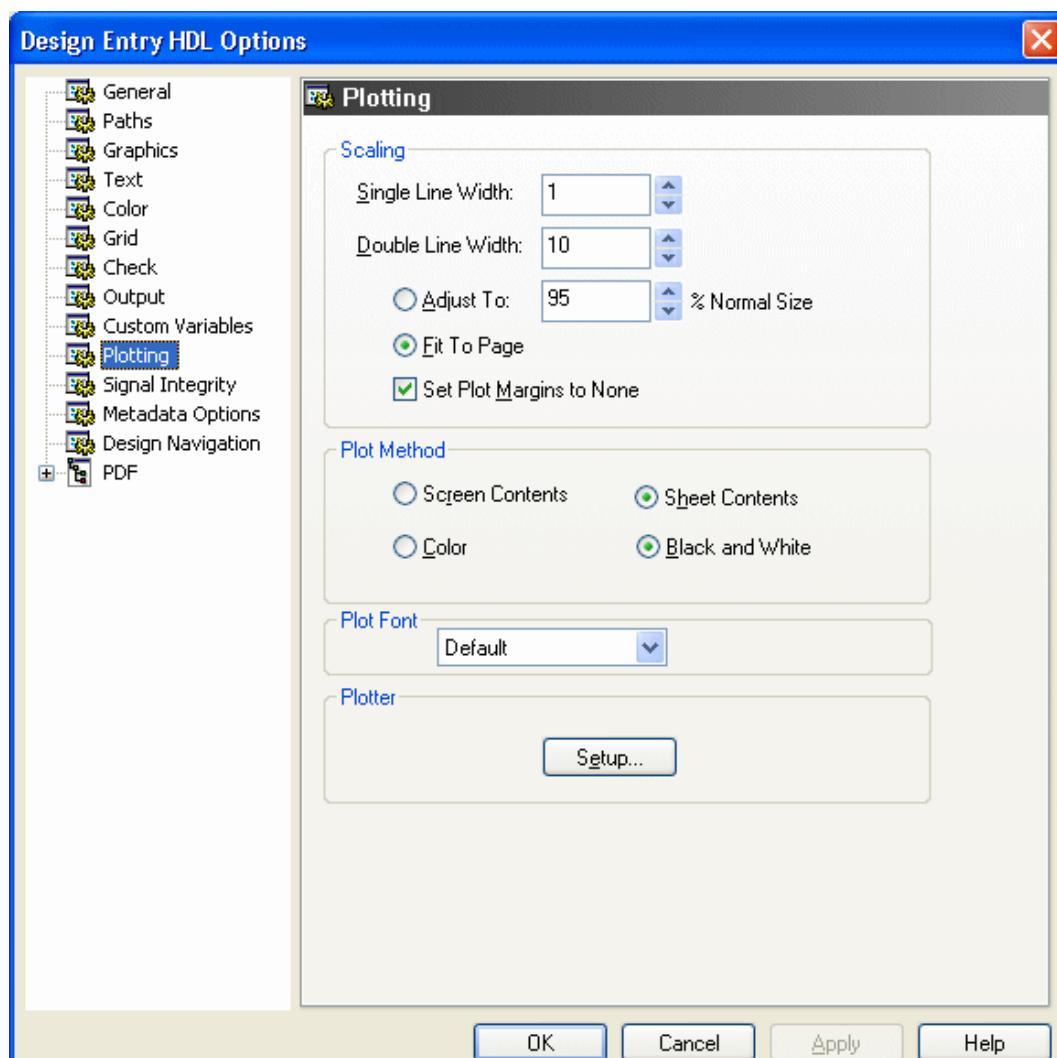
## Setting Up Windows Plotting Options

Before setting up the Windows plotting options, ensure that the plotter you want to use is configured properly.

You can setup the Windows plotting options in the *Plotting* tab of the *Design Entry HDL Options* dialog box.

### 1. Choose *File – Plot Setup* or *Tools – Options – Plotting*.

The *Design Entry HDL Options* dialog box appears.



## Allegro Design Entry HDL User Guide

### Plotting Your Design

**Note:** Select *Windows*.



2. Specify the width of single lines in the *Single Line Width* field.

This specifies the width of lines used to draw thin wires, boundaries of components and text on schematics. By default, the single line width is 1.

3. Specify the width of double lines in the *Double Line Width* field.

This specifies the width of lines used to draw buses and thick wires on schematics. By default, the double line width is 10.

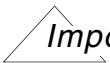
4. To adjust the plot size, choose *Adjust To* or *Fit To Page*.

- ❑ If you select *Adjust To*, specify the percentage by which to increase or decrease the size of the drawing. Design Entry HDL then plots the drawing on one or more papers of the specified size. The paper size can be specified by clicking the *Setup* button.

For example, if you have a drawing with Cadence A size page border, the percentage specified is 100, and the paper size selected is A4. The Cadence A size page border is bigger in size than A4. So, the schematic is plotted on more than one A4 paper.

- ❑ If you select *Fit To Page*, Design Entry HDL adjusts the size of the drawing so that it fits into one page of the specified paper size.

For example, you may have a drawing with Cadence A size page border, and the paper size selected is A4. Even though the Cadence A size page border is bigger in size than A4, the schematic is plotted so that it fits on one A4 paper.



#### *Important*

Paper sizes are stored in the MainWin registry file, which is in a binary format.

5. To set the margins on the paper for plotting, clear the *Set Plot Margins to None* check box. The check box is selected by default.
6. To select the plot method, choose

## Allegro Design Entry HDL User Guide

### Plotting Your Design

#### a. *Screen Contents or Sheet Contents.*

If you choose *Screen Contents*, Design Entry HDL plots the portion of the schematic that is displayed on the screen.

If you choose *Sheet Contents*, Design Entry HDL plots the entire page.

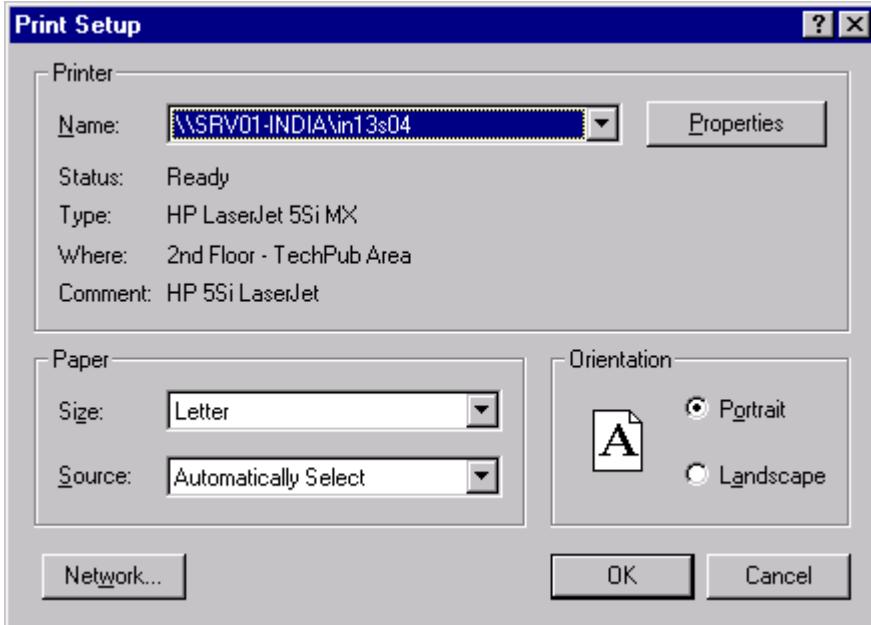
#### b. *Color or Black and White.*

If you choose *Color* and are using a color plotter, Design Entry HDL plots the drawing in color. It plots in gray scales if you are using a black and white printer.

If you choose *Black and White*, Design Entry HDL plots the drawing in black and white.

7. Select the font to be used for plotting the schematic in the *Plot Font* field. If you do not select any font, Design Entry HDL uses its *Default* font.
8. To set up the plotter, click *Setup*.

The *Print Setup* dialog box appears.



9. Choose the name of the plotter from the drop-down list.

The list shows the plotters that are configured with the system.

10. Click *Properties*.

A dialog box appears showing the system's default settings for plotting. Do not change any settings here.

**11.** Choose the paper size.

The default paper size is same as that of the system. You should use only those paper sizes that are supported by the plotter you have chosen to plot the design.

**12.** Choose the source of paper.

The default option is the same as that of the system.

**13.** To plot to a shared printer, click *Network*.

**Note:** The *Network* option is available only on Windows platform.

The *Connect to Printer* dialog box appears. It displays a list of network plotters from which you can select a plotter. The plot output is directed to the selected plotter.

**14.** Choose *Portrait* or *Landscape* as the orientation of the plot output.

**15.** Click *OK*.

The *Print Setup* dialog box closes.

**16.** Click *OK*.

The *Design Entry HDL Options* dialog box closes,\* and all the settings are saved in the project (.cpm) file.

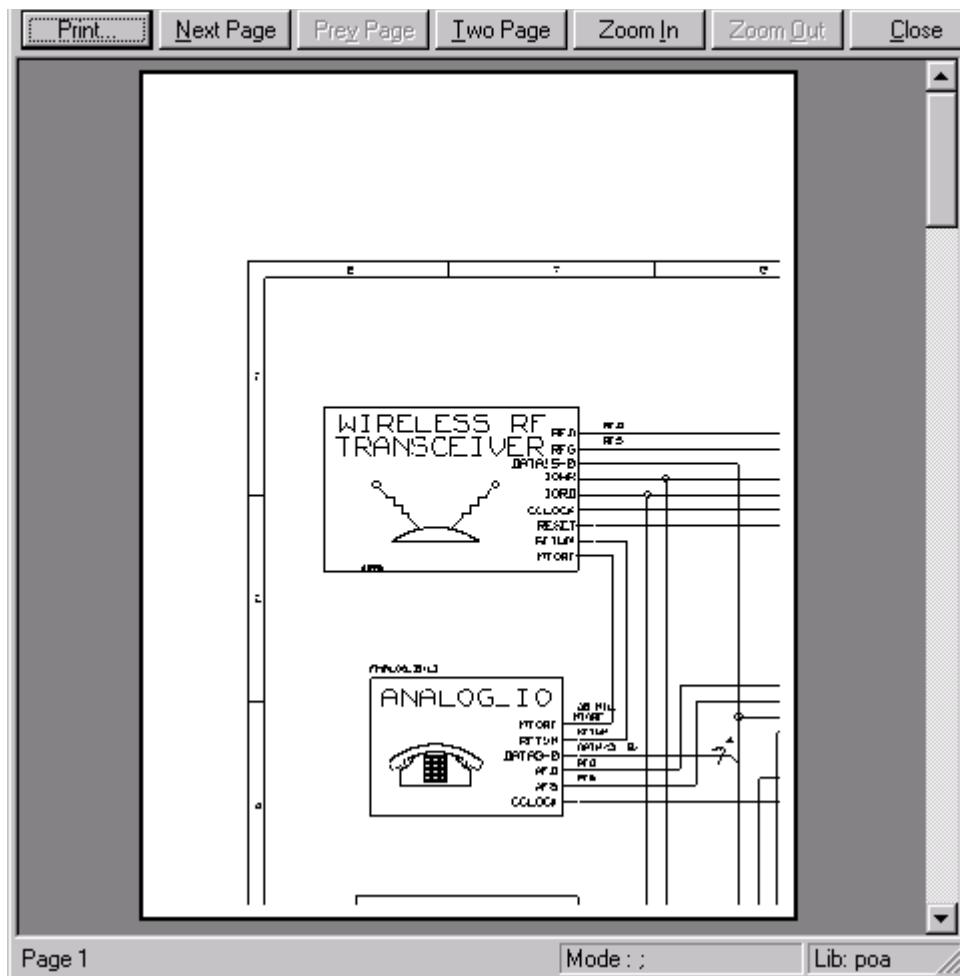
## Previewing the Design

**1.** Choose *File – Plot Preview*.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

The Preview window appears.

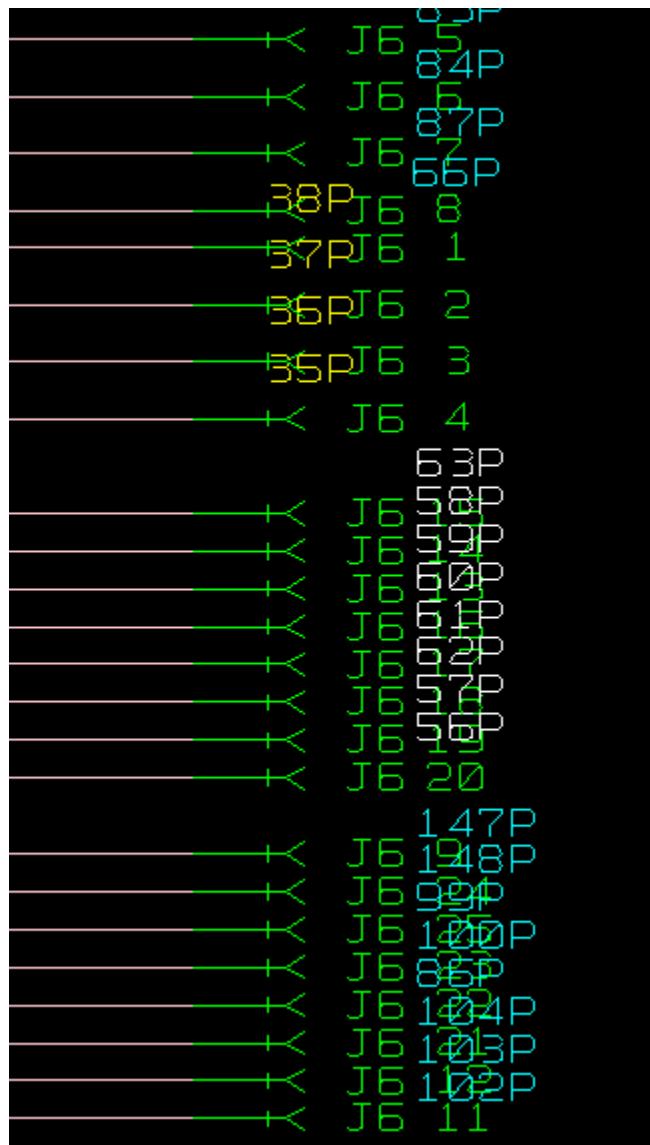


You can click the *Two Page* button to display two pages side by side, zoom in and zoom out the design.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

**Note:** If you have certain text (properties or notes) in the design such that two or more lines of text overlap as shown below:



## Allegro Design Entry HDL User Guide

### Plotting Your Design

---

In the plot of the design, some parts of the text can get truncated or hidden depending on the font size as shown below:

The following list shows various text labels with horizontal lines and arrows pointing to the right, illustrating font truncation:

- 83P
- J6 5
- 84P
- J6 6
- 8/P
- J6 7
- 66P
- 38P J6 8
- 37P J6 1
- 36P J6 2
- 35P J6 3
- J6 4
- 63P
- J6 58P
- J6 59P
- J6 60P
- J6 61P
- J6 62P
- J6 57P
- J6 56P
- J6 ..
- J6 20
- 147P
- J6 148P
- J6 99P
- J6 125)P
- J6 823>
- J6 122P
- J6 121P
- J6 112P
- ..

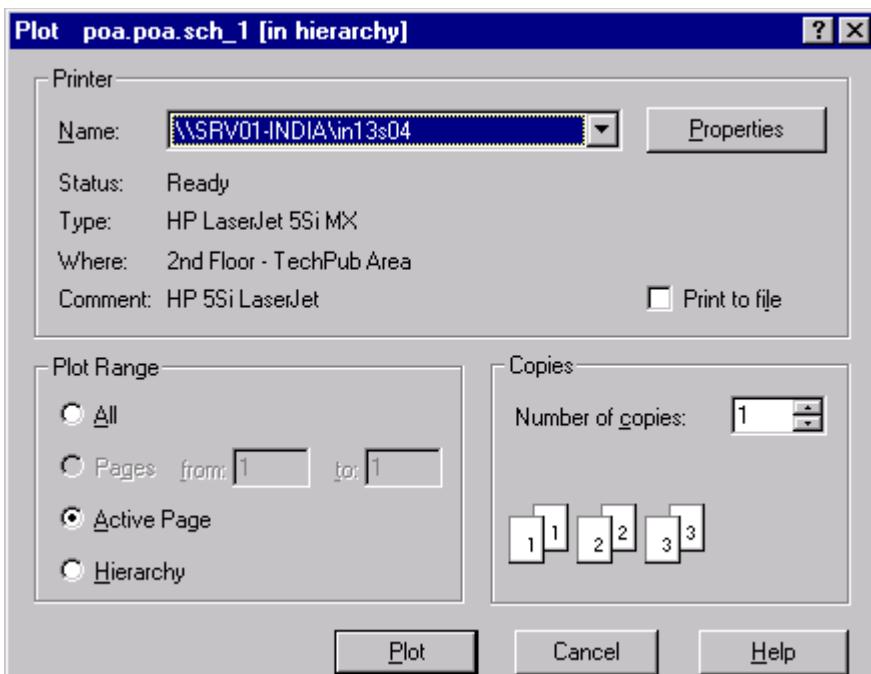
### Plotting the Design

1. Choose *File – Plot*.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

The *Plot* dialog box appears.



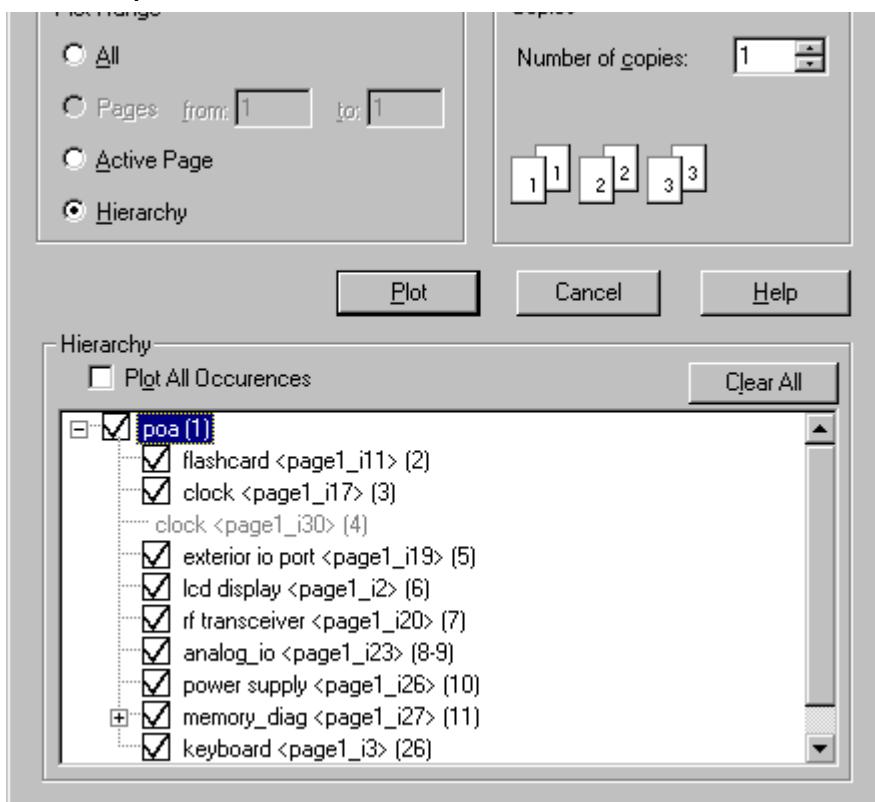
**Note:** The *Hierarchy* button does not appear if you are out of the hierarchy.

2. Choose the plotter name if you do not want to use the default plotter.
  3. Check *Print to File* to plot the drawing to a file. The name of the file can be given in the Print to File dialog box which appears when you click *Plot* if the *Print to File* button is checked.
  4. To select the plot range, choose *All*, *Pages from*, *Active Page*, or *Hierarchy*.
    - ❑ If you choose *All*, Design Entry HDL plots all the pages in your currently opened design in the active viewport.
- Note:** To plot all pages of the design, Design Entry HDL counts the number of .csb files. This gives the number of pages in the design.
- ❑ If you choose *Pages from*, specify the range of pages for plotting.
  - ❑ If you choose *Active Page*, Design Entry HDL plots the current page of the design opened in the active viewport.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

- If you choose *Hierarchy*, Design Entry HDL extends the Plot dialog box to display the hierarchical structure of the entire design. You can select or deselect sub-designs for plotting. You can click *Clear All* to clear all selections.



For more information on hierarchical plotting, see [Hierarchical Plotting](#) on page 599.

5. Click *Plot*.

Design Entry HDL plots the drawing.

## Plotting in Batch Mode

The plot settings can be set up in batch mode for:

- The current session
- All sessions



For setting options in the current session, see [Setup Commands](#) and for setting options for all sessions, see [Project File Directives](#).

### Console Commands on Windows

#### Setup Commands

Command	Capability
SETWPLT_SPOOLed	Sets the plot output to be directed to a file.
SET WPLT_FILE [<path>/] <file_name>	Sets the filename or path for the plot file you generate. If you do not set a filename, Design Entry HDL plots to a plot file named output.ps.  The <code>set wplot_spooled</code> command sets the print to file option as the default. To set this option for a given project in all sessions and to specify the name of the generated postscript file, add the following directives manually in the project file ( <code>&lt;project_name&gt;.cpm</code> ):  <code>PLOT_TO_FILE 'YES'</code> <code>PLOT_FILE_NAME 'output.ps'</code>  The format of the plot file depends on the plotter driver. The format can be PCL, PS, or any other.
SET WPLT_LOCAL	Sets the plot output to be directed to a plotter.
SET WPLT_THIN_width <width>	Sets the width of thin lines in plots. The default is 5.
SET WPLT_THICK_width <width>	Sets the width of thick lines in plots. The default is 10.
SET WPLT_SCREEN	Sets the plot method to plot screen contents, clipping the drawing outside the display screen.
SET WPLT_SHEET	Sets the plot method to plot sheet contents.
SET WPLT_ADJust	Sets the adjust to scale option.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

---

Command	Capability
SET WPLOT_SCALE <i>&lt;scale_value&gt;</i>	Adjusts the scale value of the plot to a given value.  <b>Note:</b> This command must be preceded by set wplot_adjust.  Example: If you have set the scale value to 50% in your project file, you can change the scale value to 100% by using the command:  <code>set wplot_adjust set wplot_scale 100</code>
SET WPLOT_FIT_to_page	Sets the plot to fit the paper size specified.
SET WPLOT_LANDscape	Sets the plot orientation to landscape
SET WPLOT_PORTrait	Sets the plot orientation to portrait
SET WPLOT_DEFault	Instructs Design Entry HDL to read and set defaults from the directives in the project file.
SET WPLOT_PAPER	Sets the paper size for plotting. To see the standard paper sizes, refer <a href="#">Paper Sizes Supported by Design Entry HDL</a> on page 582.  Example:  <code>set wplot_paper A4</code>  If you set a paper size not supported by the current printer, an error message is displayed.
SETW PLOT_PLOTTER <i>&lt;plotter_name&gt;</i>	Sets the plotter to the name specified. If the plotter name consists of any special characters, precede the first such character with a \ and put the plotter name within quotes.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

---

Command	Capability
SET FFACE [0 1 2 3 4 5]	Sets the font to be used while plotting the design.  The mappings of values and fonts are given below:  0 - Arial 1 - Helvetica 2 - Verdana 3 - Trebuchet MS 4 - Default  This is the font that Design Entry HDL used when the option of specifying fonts for Windows plotting was not available. If you do not specify any font, Design Entry HDL uses this font for plotting.  5 - Courier  <b>Example:</b> To set the font to Verdana, use <pre>set fface 2</pre>

### Project File Directives

These directives correspond to some fields in the Plot Setup dialog box. The directives are read and written by the Plot Setup dialog box. You can also change the values of these directives in the project file (.cpm) without invoking the Plot Setup dialog box.

---

Directive	Default Value	Description
PLOT_SINGLE_WIDTH	1	This directive corresponds to the single line width field in the Plot dialog box. This specifies the width of thin wires and buses. You can also control the text width by this field. On some plotters, if the plot output is very thin and does not show the text clearly, this width can be increased making the whole design, along with the text, thicker.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

Directive	Default Value	Description
PLOT_DOUBLE_WIDTH	10	This directive corresponds to the double line width field in the Plot dialog box. It specifies the width of thick wires and buses.
PLOT_SCALE	100	This directive corresponds to the scale value specified in the Adjust to field in Plot setup. It specifies the percentage by which to increase or decrease the plot size.
PLOT_FIT_TO_PAGE	OFF	This directive, when set to ON, directs Design Entry HDL to adjust the plot according to page size.
PLOT_SCREEN	OFF	This directive, when set to ON, directs Design Entry HDL to plot the portion of the schematic that is displayed on the screen.
PLOT_COLOR	OFF	This directive, when set to ON, directs Design Entry HDL to plot the drawing in color if you are using a color plotter, and in gray scales if you are using a black and white printer.
PAPER_ORIENTATION	1	This directive sets the orientation of the plot output. You can set it to 1 for Portrait or 2 for Landscape.
PAPER_SIZE	9	The number shows the index (0-based) number of the paper size selected in the combo box in the <i>Plot Setup</i> dialog box. This directive is useful if you want to take plot in one paper size only and want to retain this setting over multiple Design Entry HDL sessions.  It is recommended that you set the paper size in the <i>Plot Setup</i> dialog box. If you need to change it manually, ensure that you enter the correct index to map to the correct paper size in the list displayed in the combo box. All plotters may not support all the available sizes.

Directive	Default Value	Description
PAPER_SOURCE	4	<p>The number shows the index (0-based) number of the paper source selected in the combo box in the <i>Plot Setup</i> dialog box. This directive is useful if you want to use one paper source throughout the site or for all your designs.</p> <p>It is recommended that you change the paper size through the <i>Plot Setup</i> dialog box. If you need to change it manually, ensure that you enter the correct index to map to the correct paper source in the list displayed in the combo box.</p> <p><b>Note:</b> All plotters may not support all the available paper sources.</p>

## Plot Command

The syntax of the plot command is

```
plot [<lib>].[<cell>].[<view>].[<ver>]. [<page>]
```

Examples:

Command	Capability
Plot	Plots the currently opened drawing
Plot cache	Plots all pages in the cache
Plot cache.sym.1.1	Plots the symbol view in the cache
Plot cache.sym.1.2	Plots the page 2 schematic of the cache
Plot cache.sch.1.*	Plots all pages of version 1

where cache is the name of the drawing.

**Note:** Wildcard characters are supported for page numbers only.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

There are two directives for the plot console command. These are not read or written by plot setup of Design Entry HDL. You should always change them manually in the .cpm file.

Directive	Default Value	Description
PLOT_TO_FILE	NO	This directive tells the plot command to direct its plot output to a file. The file will by default be <code>output.ps</code> . Possible values: NO YES
PLOT_FILE_NAME		This directive is used in conjunction with the PLOT_TO_FILE directive to change the default file name for the plot output. You can also specify the full path of the file if you wish to direct output to a directory other than the project directory. By default the file name is <code>output.ps</code> and it is created in the project directory.

### Paper Sizes Supported by Design Entry HDL

The possible paper sizes that you can use are:

Paper Type	Paper Size	Paper Type	Paper Size
LETTER	8.5x11"	EXECUTIVE	7.5x10"
LEDGER	17x11"	STATEMENT	5.5x8.5"
A3	297x420mm	A4SMALL	210x297mm
B4	250x354mm	FOLIO	8.5x13"
10X14	10X14"	NOTE	8.5x11"
ENV_10	4.125x9.5"	ENV_C3	12.96x18.32"
CSHEET	17x22"	ENV_B6	7.04x5"
ENV_C5	6.48x9.16"	ENV_C4	9.16x12.96"
ENV_C65	4.56x9.16"	ENV_B4	10x14.12"
ENV_11	4.5x10.375"	DSHEET	22x34"

## Allegro Design Entry HDL User Guide

### Plotting Your Design

Paper Type	Paper Size	Paper Type	Paper Size
ENV_12	4x11"	ESHEET	34x44"
LETTERSMALL	8.5x11"	B5	182x257mm
LEGAL	8.5x14"	11X17	11x17"
A4	210x297mm	ENV_PERSONAL	3.625x6.5"
ENV_9	3.875x8.875"	A5	148x210mm
ENV_DL	4.4x8.8"	ENV_14	5x11.5"
FANFOLD_STD_GERMAN	8.5x12"	ENV_MONARCH	3.875x7.5"
FANFOLD_LGL_GERMAN	8.5x13"	ENV_B5	7.04x10"
FANFOLD_US	14.875x11"	ENV_ITALY	110x230mm
QUARTO	215x275mm	ENV_C6	4.56x6.48"

**Note:** You should use only those paper sizes that are supported by the plotter you have chosen to plot the design on.

## Setting up HPF Plotting Options

Before setting up the HPF plotting options, ensure that the plotter you want to use is configured properly. Also, ensure that the plotter is defined in the plotting configuration file .cdsplotinit.

You can set up the HPF plotting options in the *Plotting* tab of the *Design Entry HDL Options* dialog box. To access the *Plotting* tab of the *Design Entry HDL Options* dialog box for setting up HPF plotting options, do one of the following:

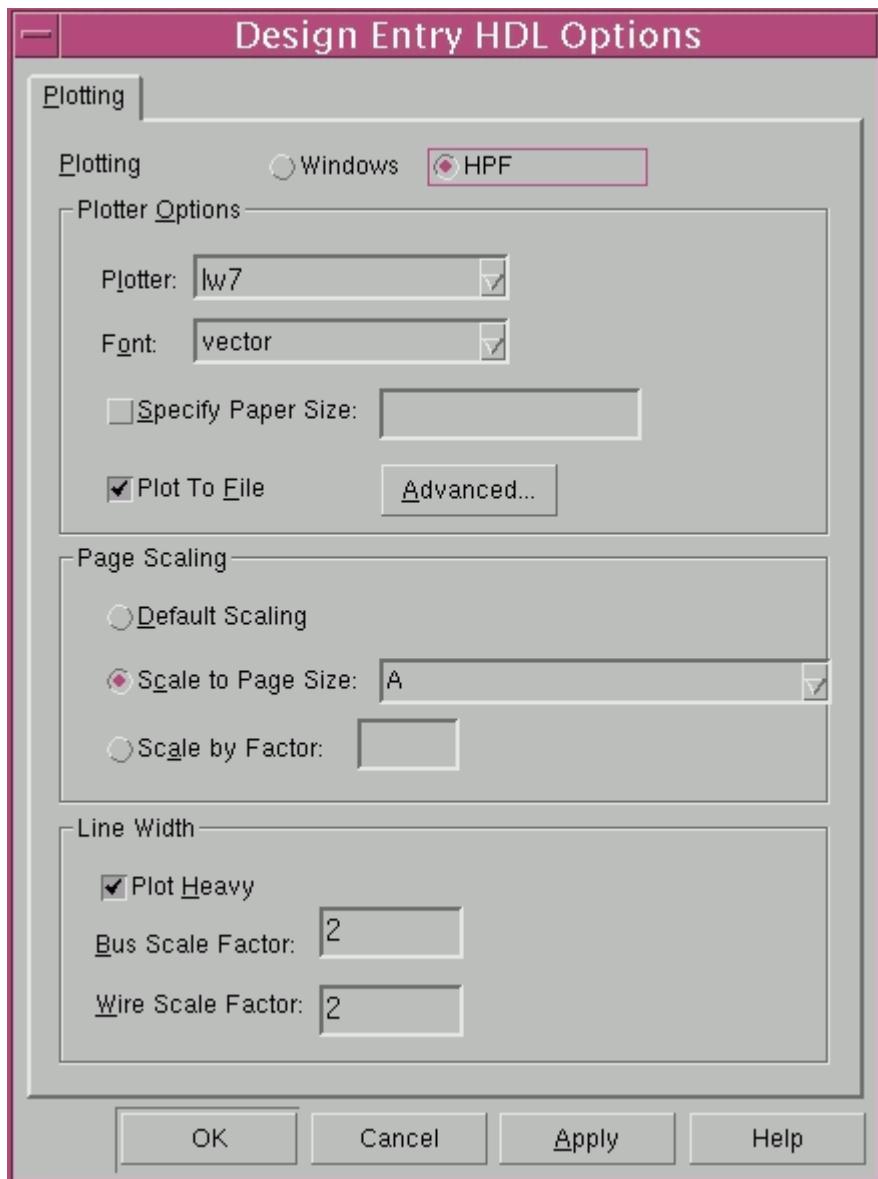
- From the *File* menu,
  - a. Choose *File – Plot Setup*.  
The *Plotting* tab of the *Design Entry HDL Options* dialog box appears.
  - b. Select *HPF* as the plotting facility.
- From the *Tools* menu,
  - a. Choose *Tools – Options*.  
The *Design Entry HDL Options* dialog box appears.

- b.** Select the *Plotting* tab.
    - c.** Select *HPF* as the plotting facility.
  - From the *HPF Plot* dialog box,
    - a.** Choose *File – Plot*.
- The *HPF Plot* dialog box appears.
- b.** Click *Setup*.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

The setup options for HPF plotting appears.



1. Select the plotter you want to use in the *Plotter* drop-down list.

The *Plotter* drop-down list displays all the plotters that are defined in the plotting configuration file `.cdsplotinit`. For example, if the `.cdsplotinit` file has the following entries for the Hewlett-Packard 7600 Series Electrostatic plotter, `hpgl12` will be displayed in the *Plotter* drop-down list.

```
hpgl12|Hewlett-Packard 7600 Series Electrostatic: \
:manufacturer=Hewlett-Packard: \
:type=hpgl12: \
```

## Allegro Design Entry HDL User Guide

### Plotting Your Design

---

```
:maximumPages#10: \
:resolution#1016: \
:paperSize="A" 9816 8236: \
:paperSize="D" 34544 22352: \
```

The first plotter defined in the .cdsplotinit file is the default plotter.

The plotter that you select in the *Plotter* drop-down list is written in the <project\_name>.cpm file by using the HPF\_PLOTTER '<plotter\_name>' directive. If you delete the entries for the plotter from the .cdsplotinit file, you will not be able to plot the drawing unless you select another plotter from the *Plotter* drop-down list.

#### 2. Select the font to be used for plotting in the *Font* drop-down list.

The text in the drawing is plotted using the selected font. The default font is VECTOR. The following fonts are supported:

- VECTOR
- CURSIVE
- NATIVE
- GOTHIC
- SYMBOL
- GREEK
- VALID
- MILSPEC

#### 3. Select the check box next to the *Specify Paper Size* field and specify the paper size.

If the paper size name has spaces, enclose it in parentheses. For example, if the paper size name is 22 inches wide, specify the paper size as "22 inches wide".

The paper size that you specify must be defined for the plotter in the .cdsplotinit file. For example, if the entries for the hpgl2 plotter in the .cdsplotinit file are as below, you can specify A, D, E, "22 inches wide" or "34 inches wide" as the paper size.

```
hpgl2|Hewlett-Packard 7600 Series Electrostatic: \
:manufacturer=Hewlett-Packard: \
:type=hpgl2: \
:maximumPages#10: \
:resolution#1016: \
:paperSize="A" 9816 8236: \
```

## Allegro Design Entry HDL User Guide

### Plotting Your Design

```
:paperSize="D" 34544 22352: \
:paperSize="E" 44704 34544: \
:paperSize="22 inches wide" 0 22352: \
:paperSize="34 inches wide" 0 34544:
```

You can define the paper size name for a plotter by using the `paperSize` option in the `.cdsplotinit` file.

**Note:** If you do not select the check box next to the *Specify Paper Size* field, the first paper size specified in the `.cdsplotinit` file for the specified plotter is taken as the default paper size. In the above example for the `hpgl2` plotter, `A` will be taken as the default paper size if you do not select the check box next to the *Specify Paper Size* field.

4. Select the *Plot to File* check box if you want to print the drawing to a file.
5. Click the *Advanced* button to specify plot to file options.



- a. Specify the *Location* of the file in which the design is to be plotted. The default location is the current working directory.
- b. Select the *Single File* option if you want to print all the pages of the design in a single file. Also specify the *Name* of the file in which to print the pages. The default file name is `vw.spool`.
- c. Select the *File Per Page* option if you want to print every page of the design in a separate file. Also specify the *Prefix* for the filenames in which pages of the design will be printed. The default prefix is `vw.spool`.
- d. Click *OK*.

**Note:** If the *Plot to File* check box is not selected, the drawing will be plotted.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

---

6. To scale the drawing, select *Default Scaling*, *Scale to Page Size*, or *Scale by Factor*

Select	To
<i>Default Scaling</i>	Plot the drawing as it is. In other words, the default scale factor 1 is used.
<i>Scale to Page Size</i>	Scale the drawing to be plotted to the page size you select. <b>Note:</b> The paper size that you select need not be defined for the plotter in the .cdsplotinit file. This option is similar to the <i>Fit To Page</i> option in the Windows plotting facility with the added feature of specifying the page size to which the drawing has to fit. For example, if you have used the C SIZE PAGE page border (17 x 22 inch) symbol in your drawing and want to plot the drawing in A paper size (8 1/2 x 11 inch), specify A in the <i>Specify Page Size</i> field and select A in the <i>Scale to Page Size</i> drop-down list. <b>Note:</b> If the paper size you select in the <i>Scale to Page Size</i> field is not the same or smaller than the paper size you specified in the <i>Specify Page Size</i> field, the drawing will be plotted in multiple sheets. For example, if you have used the C SIZE PAGE page border symbol in your drawing and plot the drawing to paper size B, the drawing will be plotted in multiple sheets of paper size B if you do not select paper size B or a smaller paper size in the <i>Scale to Page Size</i> drop-down list.
<i>Scale by Factor</i>	Specify the factor by which you want to scale the plot. For example, a scale factor of 0.5 will create a plot size that is half the drawing size. <b>Note:</b> If the scale factor results in a plot size that is larger than the paper size you have specified in the <i>Specify Page Size</i> field, the drawing will be plotted in multiple sheets of the paper size you specified in the <i>Specify Page Size</i> field.

7. Select *Plot Heavy* if you want to increase line widths of buses and wires.
8. Specify the scale factor in the *Bus Scale Factor* field to increase or decrease the line widths of thick wires (vectored signals) in plots.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

9. Specify the scale factor in the *Wire Scale Factor* field to increase or decrease the line width of thin wires (scalar signals) and thickness of text in plots.
10. Click *OK*.

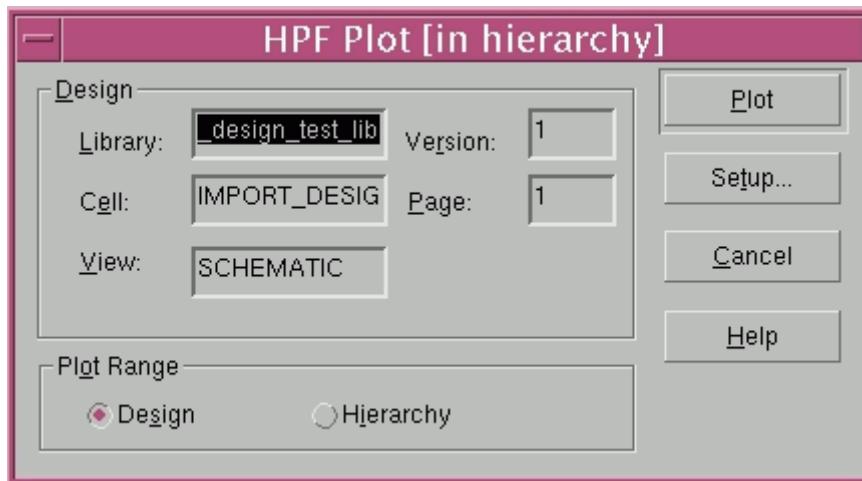
## Plotting the Design

To plot the design, you have to open the *HPF Plot* dialog box. The *HPF Plot* dialog box allows you to plot specific drawings in the design or to select sub-designs from a hierarchical design for plotting.

1. Choose *File – Plot*.

The *HPF Plot* dialog box appears if you selected *HPF* as the plotting facility in the *Plotting* tab of the *Design Entry HDL Options* dialog box. For more information, see [Setting up HPF Plotting Options](#) on page 583.

The default values are for the current drawing.



2. Click *Setup* if you want to change the HPF plotting options.

For more information, see [Setting up HPF Plotting Options](#) on page 583.

3. In the *Plot Range* group box.

- Select *Design* if you want to plot the drawing specified in the *Design* group box.
- Select *Hierarchy* if you want to perform hierarchical plotting.

Design Entry HDL extends the *HPF Plot* dialog box to display the hierarchical structure of the root design. You can select or deselect sub-designs for plotting. For more information on hierarchical plotting, see [Hierarchical Plotting](#) on page 599.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

---

If you select *Hierarchy*, all the fields in the *Design* group box are disabled.

4. Change the library name in the *Library* field, if required.

If you change the library name, ensure that the library is defined in the `cds.lib` file.

5. Change the cell name in the *Cell* field, if required.

If you change the cell name, ensure that the cell is present in the library you have Change the view name in the *View* field, if required.

#### Specify      If

**SCHEMATIC** You want to plot schematic drawings

**SYM** You want to plot symbol drawings

**SCHCREF\_1** You want to plot the schematic drawings generated by CRefer in the `schref_1` view

If you change the view name, ensure that the view is present in the cell you specified in the *Cell* field. You can use wildcards (\*) and (?) in this field.

6. Change the version number in the *Version* field, if required.

The version number indicates the version of the view you want to plot. For example, if you want to plot the schematic drawings in the `sch_1` view of a cell, specify **SCHEMATIC** in the *View* field and 1 in the *Version* field. If you want to plot the symbol drawings in the `sym_3` view of a cell, specify **SYM** in the *View* field and 3 in the *Version* field. To plot the schematic drawings generated by CRefer in the `schref_1` view of a cell, specify **SCHCREF\_1** in the *View* field and 1 in the *Version* field.

By default, the version number is 1. If you change the version number, ensure that the version of the view is present in the cell you specified in the *Cell* field. You can use wildcards (\*) and (?) in this field.

7. Change the page number in the *Page* field, if required.

By default, the page number is 1. If you change the page number, ensure that the page is present in the version of the view you specified in the *Version* field. You can use wildcards (\*) and (?) in this field.

8. Click *Plot*.

## Plotting the Design from the Console Window

Design Entry HDL allows you to plot the design in batch mode from the console window. You can setup the HPF plotting options and then plot the design by using the `hardcopy` console command. For more information, see [hardcopy Command](#) on page 595.

**Note:** The options for HPF plotting from the console window can be set up for:

- The current session only

The options you specify for HPF plotting in the *Plotting* tab of the *Design Entry HDL Options* dialog box are the default plotting options. You can override the default options only for the current session using the `set` console command. To set up HPF plotting options only for the current session, use [Setup Commands](#).

- All sessions

The options you specify for HPF plotting in the *Plotting* tab of the *Design Entry HDL Options* dialog box are the default plotting options. To set up the default HPF plotting options for all sessions, use [Project File Directives](#).

## Setup Commands

You can setup the following HPF plotting options using the `set` console command. The options that you set using the `set` console command override the default HPF plotting options (setup in the *Plotting* tab of the *Design Entry HDL Options* dialog box) for the current session.

- [Specifying the Font for Plotting](#) on page 591
- [Specifying the Default Paper Size](#) on page 592
- [Setting Plotting to a Plotter](#) on page 593
- [Setting Plotting to a File](#) on page 593
- [Specifying the Plotter](#) on page 594

### ***Specifying the Font for Plotting***

Console command syntax

```
set font <hpf_font_name>
```

where `hpf_font_name` can be one of the following:

- `VECTOR_FONT`

- CURSIVE\_FONT
- NATIVE\_FONT
- GOTHIC\_FONT
- SYMBOL\_FONT
- GREEK\_FONT
- VALID\_FONT
- MILSPEC\_FONT

### ***Specifying the Default Paper Size***

Console command syntax

```
set papersize <option>
```

where `option` is the name (string) indicating the paper size the plotter uses, including any offset.

Example:

```
set papersize A
```

If the paper size name has spaces, enclose it in parentheses. For example, if the paper size name is 22 inches wide, specify the paper size as “22 inches wide”.

The paper size that you specify must be defined in the `.cdsplotinit` file. For example, if the entries for the `hpgl2` plotter in the `.cdsplotinit` file are as below, you can specify A, D, E, “22 inches wide” or “34 inches wide” as the paper size.

```
hpgl2|Hewlett-Packard 7600 Series Electrostatic: \
  :manufacturer=Hewlett-Packard: \
  :type=hpgl2: \
  :maximumPages#10: \
  :resolution#1016: \
  :paperSize="A" 9816 8236: \
  :paperSize="D" 34544 22352: \
  :paperSize="E" 44704 34544: \
  :paperSize="22 inches wide" 0 22352: \
  :paperSize="34 inches wide" 0 34544:
```

You can define the paper size name for a plotter using the `paperSize` option in the `.cdsplotinit` file.

**Note:** The first paper size specified in the `.cdsplotinit` file for the specified plotter is taken as the default paper size. In the above example for the `hpgl2` plotter, `A` will be taken as the default paper size.

#### ***Setting Plotting to a Plotter***

Console command syntax

```
set local
```

or

```
set LOCAL_plot
```

The drawing will be plotted in a plotter.

#### ***Setting Plotting to a File***

Console command syntax

```
set spooled
```

OR

```
set SPOoled_plot
```

The drawing will be plotted to a file instead of being plotted on a plotter. The default filename is `vw.spool`, and it is created in the project directory. This file can be plotted later.

```
set hpfplot_file_location
```

The drawing will be plotted to a file at the specified location instead of being plotted instead of the default location.

```
set hpfplot_file_name
```

Set this option if you want to print all the pages of the design in a single file. The default filename is `vw.spool`. You can also specify another name if you want.

```
set multiple_spooled
```

Set this option if you want to print every page of the design in a separate file.

```
set hpfplot_file_per_page_prefix
```

If you have set the `multiple_spooled` option, specify the prefix for the filenames in which pages of the design will be printed. The default prefix is `vw.spool`.

### **Specifying the Plotter**

Console command syntax

```
set PLOTTER <plotter_name>
```

Where `plotter_name` is any plotter specified in the `.cdsplotinit` file.

### **Project File Directives**

The following directives are set in the project file (`<projectname>.cpm`) when you specify the options for HPF plotting in the *Plotting* tab of the *Design Entry HDL Options* dialog box.

Directive	Values	Description
HPF_BATCH	Yes	This directive corresponds to the <i>Plot to File</i> check box. It is used to direct Design Entry HDL to spool the plot to the <code>vw.spool</code> file.
	No	
HPF_PLOTTER	Name of the plotter	This directive corresponds to the <i>Plotter</i> field. It is used to specify the name of the plotter in the <code>.cdsplotinit</code> file.
HPF_SPEC_PLOT_PAGESIZE	Yes	This directive corresponds to the check box next to the <i>Specify Page Size</i> field. It is used to set the paper size.
	No	
HPF_PLOT_PAGESIZE	Page size	This directive corresponds to the <i>Specify Page Size</i> field. It specifies the paper on which to plot.
HPF_SCALETYPE	Default, Scale by factor, Scale to Page Size	This directive corresponds to the <i>Default</i> , <i>Scale by Factor</i> , and <i>Scale to Page Size</i> fields. It specifies the mode of scaling to be used for plotting.
HPF_PAGESIZE	A	This directive corresponds to the <i>Scale to Page Size</i> field. It specifies the standard page to which the design gets scaled.
	B	
	C	
	D	
	E	
	F	

## Allegro Design Entry HDL User Guide

### Plotting Your Design

Directive	Values	Description
HPF_BUS_SCALEFACTOR	A number	This directive corresponds to the <i>Bus Scale Factor</i> field. It specifies the width of buses in the design.
HPF_WIRE_SCALEFACTOR	A number	This directive corresponds to the <i>Wire Scale Factor</i> field. It specifies the width of wires, text, and component boundaries in the design.
HPF_FONT	A font name	This directive corresponds to the <i>Font</i> field. It specifies the font style to be used for plotting.
HPF_SCALEFACTOR	A number	This directive corresponds to the <i>Scale by Factor</i> field. It specifies the scaling factor.

### hardcopy Command

You can plot designs in batch mode using the `hardcopy` console command.

Before using the `hardcopy` command, ensure that the plotter you want to use is configured properly. Also, ensure that the plotter is defined in the plotting configuration file `.cdsplotinit`.

#### ***Using the hardcopy Command***

The `hardcopy` command takes two arguments, `scale_factor` or `paper_size` and `drawing_name`, in the following syntax:

```
hardcopy[scale_factor|scale_to_page] [drawing_name]
```

`scale_factor` A factor applied to the drawing to determine the final plot size. The default factor is 1.

Specify this option if you want to create a bigger or smaller plot size for the drawing.

- To create a plot size of half the drawing size, use a factor of .5.
- To plot twice the drawing size, use a factor of 2.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

---

*scale\_to\_page* A pre-determined plot size. The drawing to be plotted is scaled to the page size you specify.

For example, if you have used the C SIZE PAGE page border (17 x 22 inch) symbol in your drawing and want to plot the drawing to fit in A paper size (8 1/2 x 11 inch), specify A as the value for *scale\_to\_page*.

The paper size that you specify as the value for this option must be defined for the plotter in the .cdsplotinit file. The `paperSize` entry is used to specify the paper sizes that a plotter uses.

If you want the drawing to be scaled to the paper size you specify as the value of the *scale\_to\_page* option but want to plot the drawing on some other paper size, specify the default paper size using the `set console` command. For more information, see [Specifying the Default Paper Size on page 592](#).

For example, assume that you have entries A and D for the `paperSize` option for the hpgl2 plotter in the .cdsplotinit file, and you used the `set console` command to specify D as the default paper size. If you enter `hardcopy a`, the drawing will be scaled according to the `paperSize` entry A, but printed on the hpgl2 plotter using the paper size specified by the D entry.

If you do not specify a default paper size using the `set console` command, the first `paperSize` entry specified for the plotter in the .cdsplotinit file is taken as the default paper size.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

---

*drawing\_name*

This is the drawing you want to plot. The drawing does not have to be the one you are currently editing. If you do not specify a drawing name, the current drawing is plotted.

The drawing name is specified using the following syntax:

[<library> [<cell>.]<view>.]<version>.<page>]

- where `library` is the name of the library in which the drawing exists.

You need not specify the library if the `cell` is present in the library containing the root design for the project (also known as `worklib`). The library containing the root design for the project is specified in the *Global* tab of the *Project Setup* window. For more information, see [Setting Up a Project](#) on page 68.

- where `view` can be one of the following:
  - `sch` for the schematic view that contains schematic drawings
  - `sym` for the symbol view that contains symbol drawings
  - `schref_1` for the `schref_1` view that contains schematic drawings generated by CRefer
- where `version` is the version of the view you want to plot. For example, if you want to plot the schematic drawings in the `sch_1` view of a cell, specify the version as 1.
- where `page` is the number of the page you want to plot.

You can use wildcards to specify the drawing name. For a better understanding of the syntax used for specifying the drawing name, see [Sample hardcopy Commands](#).

#### *Important*

If you are in the Occurrence Edit mode, you can plot only the currently active drawing or drawings from the same version of the view in which the drawing exists. If you specify a different library, cell, view, or version number of the view, Design Entry HDL will display the following error message when you plot the drawing:

Only currently active drawing can be plotted in occurrence edit mode

**Sample hardcopy Commands**

The following examples assume that you have used the `set console` command to specify `B` as the default paper size. For more information, see [Specifying the Default Paper Size](#) on page 592.

Command	Capability
<code>ha</code>	Plots the current drawing at the default scale factor 1 on <code>B</code> size paper.
<code>ha b</code>	Scales the current drawing to paper size <code>B</code> and plots it on <code>B</code> size paper.
<code>ha 1 *</code>	Plots all the schematic drawings, symbol drawings, and the drawings generated by CRefer (in the <code>CREFOUT</code> and <code>SCHREF_1</code> views) that are present in the design. The drawings are plotted on <code>B</code> size paper.
<code>ha a cache.sch.1.1</code>	Scales the first schematic drawing present in the <code>sch_1</code> view of the <code>cache</code> cell to paper size <code>A</code> and plots the drawing on <code>B</code> size paper.
<code>ha 1 &lt;memory&gt;cache.sch.1.1</code>	Plots the first schematic drawing present in the <code>sch_1</code> view of the <code>cache</code> cell in the <code>memory</code> library on <code>B</code> size paper.
<code>ha 1 cache.schref_1.1.1</code>	Plots the first schematic drawing present in the <code>schref_1</code> view of the <code>cache</code> cell on <code>B</code> size paper.
<code>ha 1 &lt;poa&gt;cache.sch.2.3</code>	Plots the third schematic drawing present in the <code>sch_2</code> view of the <code>cache</code> cell in the <code>poa</code> library. The drawing is plotted on <code>B</code> size paper.
<code>ha a &lt;poa&gt;cache.sch.2.*</code>	Scales all the schematic drawings present in the <code>sch_2</code> view of the <code>cache</code> cell in the <code>poa</code> library to paper size <code>A</code> and plots the drawings on <code>B</code> size paper.
<code>ha 1 cache.sch.*</code>	Plots all the schematic drawings present in all the schematic views ( <code>sch_1</code> , <code>sch_2</code> , <code>sch_3</code> and so on) of the <code>cache</code> cell. The drawings are plotted on <code>B</code> size paper.
<code>ha 1 *.sch.*</code>	Plots all the schematic drawings that are present in all the schematic views ( <code>sch_1</code> , <code>sch_2</code> , <code>sch_3</code> etc.) of all cells in all <i>local</i> libraries used the project. The drawings are plotted on <code>B</code> size paper.

Command	Capability
ha 1 *.schcref_1.*	Plots schematic drawings generated by CRefer in the SCHREF_1 view. The drawings are plotted on B size paper.
ha 2 <lsttl>ls154.sym.1.1	Plots the symbol view of the ls154 part in the lsttl library on B size paper using a scale factor of 2.
ha 2 <lsttl>ls154.*	Plots all versions of the ls154 symbol (the symbol drawings present in the sym_1 and sym_2 views of ls154). The drawings are plotted on B size paper using a scale factor of 2.

---

## Hierarchical Plotting

Hierarchical plotting in Design Entry HDL allows you to selectively plot the schematics of cells that belong to a hierarchy of the design. Hierarchical Plotting is available in Windows Plotting. For details on Windows Plotting, refer to [Windows Plotting on Windows](#) on page 567. For details on Hierarchical Plotting, refer to [You should use only those paper sizes that are supported by the plotter you have chosen to plot the design on.](#) on page 583.

The following topics are discussed in this section:

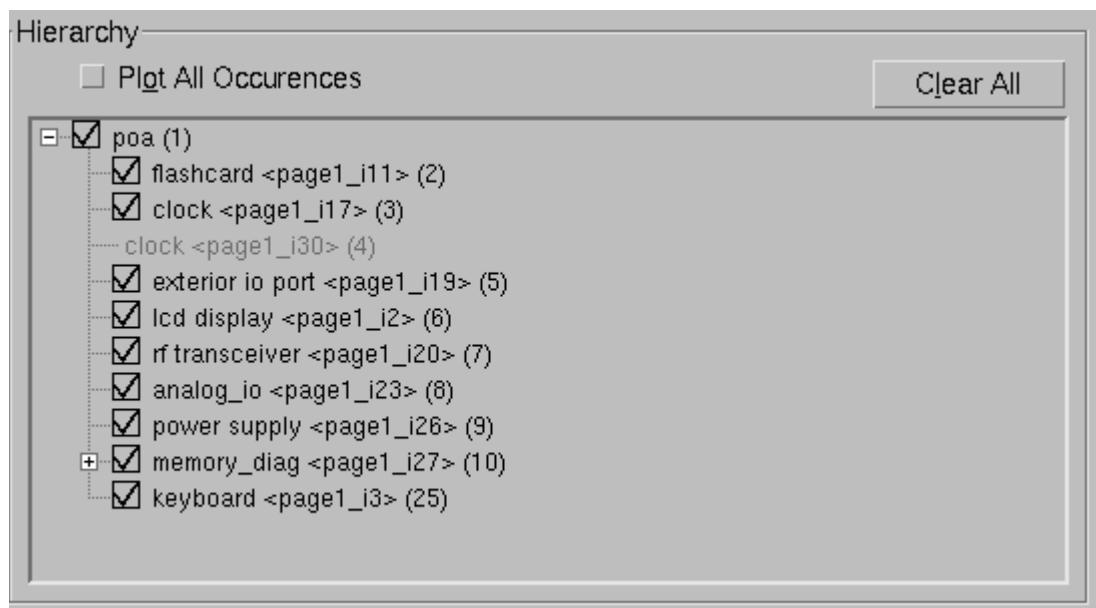
- [Changing the Order in Which Designs Are Plotted](#) on page 599
- [Plotting Hierarchical Designs](#) on page 601

### Changing the Order in Which Designs Are Plotted

You can modify the order in which designs are plotted. To do this, you must perform module ordering before plotting the design. If you exclude a module during module ordering, that module will not be displayed in the hierarchy. For more information on module ordering, see [Module Ordering](#) on page 466.

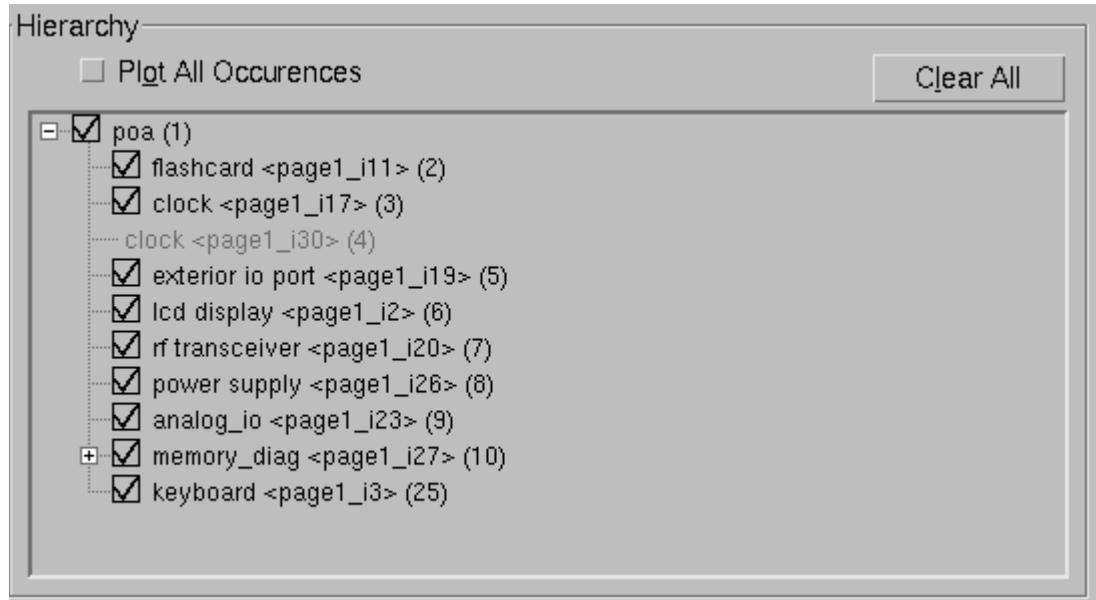
Let's take the following example to see how module ordering impacts hierarchical plotting:

**Figure 15-1 Hierarchy Before Module Ordering**



If you want to plot the `analog_io` design after the `power supply` design, perform module ordering to move the `analog_io` module after the `power supply` module. This is how the hierarchy will be displayed after you have performed module ordering:

**Figure 15-2 Hierarchy After Module Ordering**

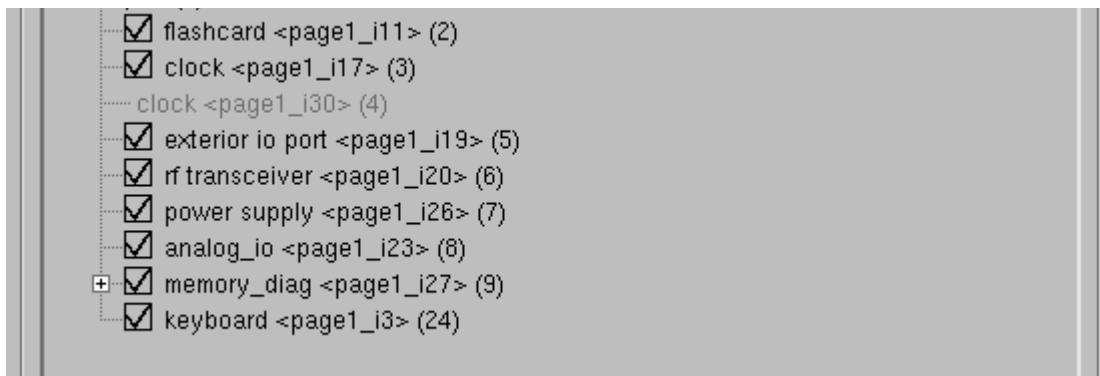


Notice that the schematic page numbers have also changed after module ordering. In [Figure 15-1](#) on page 600, the `analog_io` design had the page number 8 and the `power supply` design had the page number 9. After module ordering, the page number has changed to 8 for the `power supply` design and to 9 for the `analog_io` design. For more information on page numbering, see [Displaying and Working with Schematic Page Numbers](#) on page 487.

**Note:** If you perform page renumbering, the order in which the designs are plotted will not change. If you want the order in which the designs are plotted to change after you do page renumbering, you must perform module ordering. For more information, see [Page Renumbering, Module Ordering and Hierarchical Plotting](#) on page 495.

If you do not want the `lcd display` design to be plotted, you can clear the check box next to the design name or exclude the `lcd display` module by performing module ordering. If you exclude the `lcd display` module during module ordering, the `lcd display` design will not be displayed in the hierarchy. This is how the hierarchy will look after you exclude the `lcd display` module by performing module ordering.

**Figure 15-3 Hierarchy After Excluding the `lcd display` Module**



Note that the `lcd display` design is no longer displayed in the hierarchy. The page numbers of the designs have also changed.

## Plotting Hierarchical Designs

1. Choose *File – Save All*.
2. Choose *File – Plot*.

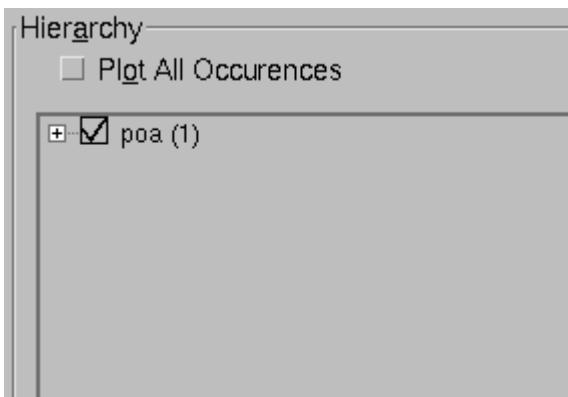
The *Plot* dialog box appears if you are performing Windows Plotting.

3. Select *Hierarchy* to extend the *Plot* dialog box.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

Design Entry HDL displays the hierarchical structure of the root design.



The root design is displayed as poa (1), where

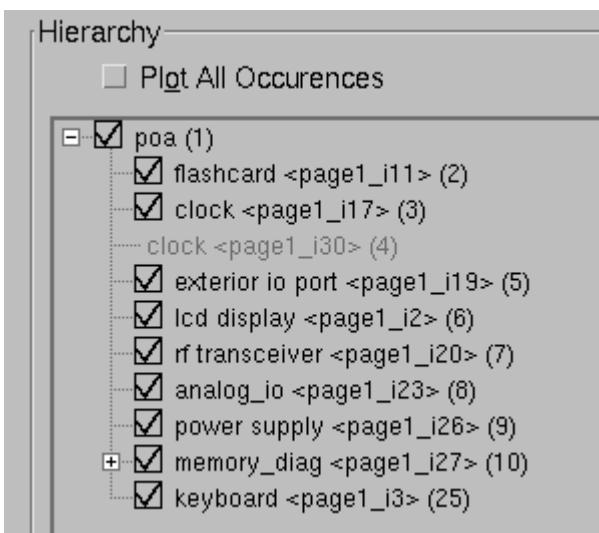
- ❑ poa is the name of the root design, and
- ❑ 1 is the number of the page in which the poa design will be plotted.

If the root design had three pages, the design name will be displayed as poa (1-3).

**Note:** The page number will not be plotted by default. If you want the page number to be plotted, you must use the CURRENT DESIGN SHEET custom text variable on the page in the design. For more information on page numbering, see [Displaying and Working with Schematic Page Numbers](#) on page 487.

The check box in a white background indicates that the design called poa and all sub designs under it will be plotted.

4. Click the + icon next to poa for Design Entry HDL to display all sub designs in it.



The first sub design is displayed as `flashcard <page1_i11> (2)`, where

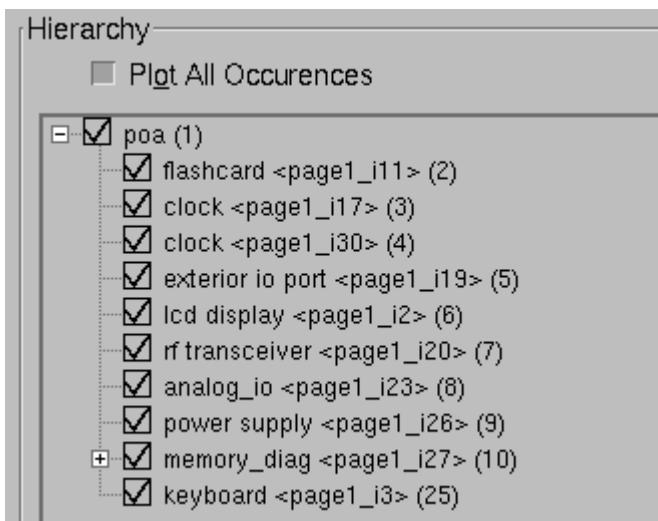
- ❑ `flashcard` is the name of the sub design,
- ❑ `<page1_i11>` indicates that the sub design is instantiated as instance `i11` on page 1 of the root design, and
- ❑ 2 is the number of the page in which the `flashcard` sub design will be plotted.

If the `flashcard` sub design had two pages, the design name will be displayed as `flashcard <page1_i11> (2-3)`.

**Note:** The page number will not be plotted by default. If you want the page number to be plotted, you must use the `CURRENT_DESIGN_SHEET` custom text variable on the pages in the design. For more information on page numbering, see [Displaying and Working with Schematic Page Numbers](#) on page 487.

Design Entry HDL displays the check boxes next to all sub designs under the design `poa` as selected. This means that all the sub designs will be plotted.

5. Select the *Plot All Occurrences* check box if you want to plot both the occurrences of the block `clock` in the design.



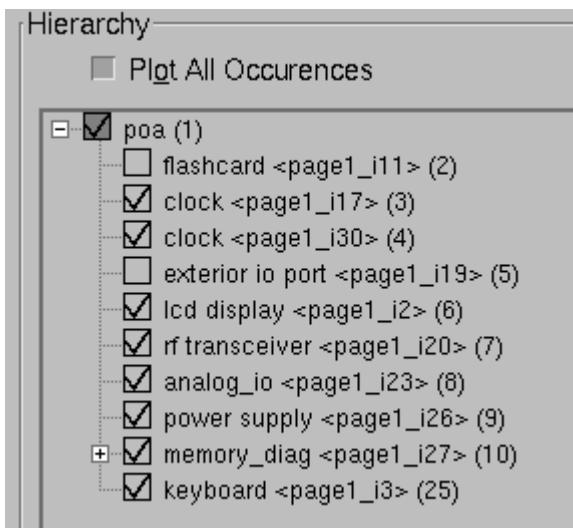
Both the occurrences of the block `clock` are selected for plotting.

You can select or deselect designs for plotting. To modify the order in which the designs are plotted, you must perform module ordering before plotting the design. For more information, see [Changing the Order in Which Designs Are Plotted](#) on page 599.

## Allegro Design Entry HDL User Guide

### Plotting Your Design

6. If you do not want to plot a sub design, clear the check box next to the sub design name.

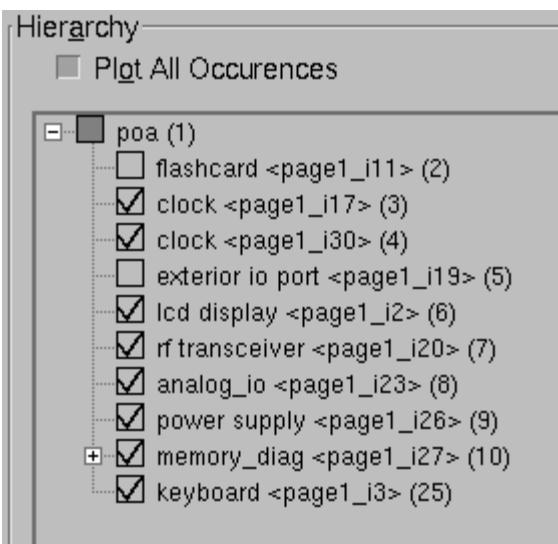


When you deselect some of the sub designs, Design Entry HDL displays the check box next to the design `poa` in a grey background. This indicates that Design Entry HDL will plot `poa` and only some sub designs in it.

To select a design for plotting, select the check box next to the design name.

**Note:** You can also click a design name to select or deselect the design for plotting. Each click on a design name selects or deselects the check box next to the design name.

7. Clear the check box next to `poa`.



The check box next to `poa` is displayed with a grey background. This indicates that Design Entry HDL will plot some sub designs in the design `poa`, but not `poa`. Even if you select all sub designs, Design Entry HDL will plot all the sub designs but not `poa`.

**Note:** Release 16.5 onwards, both occurrence and block-level properties are plotted. You cannot plot only master properties even if the "Plot All Occurrences" option is enabled or disabled.

## Frequently Asked Questions in Plotting

This section contains the answers to most frequently asked questions about plotting in Design Entry HDL. To view the answer to any question, click on that question in the list below.

[On Windows, how do I select a plotter that is on the network?](#)

[Can I change the thickness or font of the text on a schematic?](#)

[In HPF plotting, how can I plot a drawing so that it fits in a paper size that is smaller than the size of the drawing.](#)

[In HPF plotting, how can I plot a drawing so that it is scaled to fit in a paper size that is larger than the size of the drawing.](#)

[How do I select paper sizes in HPF plotting?](#)

[Can I exclude some design modules from plotting and cross referencing?](#)

[Do set console commands affect plotting through the Plot dialog box?](#)

[How can I plot a colored schematic on color plotters?](#)

[How can I view a spool file created by plotting?](#)

[How can I create a PDF output file of a Design Entry HDL schematic?](#)

[How can I plot a schematic for which I have only read-only permissions?](#)

[How do I plot all the pages in a flat schematic at the same time?](#)

[Can I plot hierarchical schematics?](#)

[Is previewing supported for HPF plotting?](#)

[In Windows plotting mode, can I preview all the plot pages together?](#)

From where can I select different setup options?

**Which are the plotters supported in HPF Plotting?**

Click the *Cadence Plotting Services* link in the Cadence Online Support (<https://support.cadence.com>) main page for information on the list of plotters that are supported for HPF plotting.

**On Windows, how do I select a plotter that is on the network?**

In Design Entry HDL, do the following:

1. Choose *File – Plot Setup*.

The *Plotting* tab in the *Design Entry HDL Options* dialog box is displayed.

2. Click *Setup*.

The *Print Setup* dialog box appears.

3. Click *Network*

4. Select the printer and click *OK*.

**Can I change the thickness or font of the text on a schematic?**

In the Windows plotting mode, the thickness of the text is always the same as the thickness of a thin line. You can adjust this thickness to increase or decrease the text thickness also.

1. Access the *Plotting (Windows)* tab of the *Design Entry HDL Options* dialog box.

2. Increase or decrease the size specified in the *Single Line Width* field.

The thickness of the text changes accordingly.

**Note:** You cannot change the font of the text in Windows plotting mode.

If you are using HPF plotting, do the following to change the thickness of the text.

1. Access the *Plotting (HPF)* tab of the *Design Entry HDL Options* dialog box.

2. Increase or decrease the scale factor specified in the *Wire Scale Factor* field.

The thickness of the text changes accordingly.

In the HPF plotting mode, you can use one of the following fonts for the text.

- VECTOR
- CURSIVE
- NATIVE
- GOTHIC
- SYMBOL
- GREEK
- VALID
- MILSPEC

For more information, see [You should use only those paper sizes that are supported by the plotter you have chosen to plot the design on.](#) on page 583.

#### In HPF plotting, how can I plot a drawing so that it fits in a paper size that is smaller than the size of the drawing.

For example, you have used the C SIZE PAGE page border (17 x 22 inch) symbol in your drawing and want to plot the drawing so that it fits in a single sheet of paper size A (8 1/2 x 11 inch).

Do the following:

1. Access the *Plotting (HPF)* tab of the *Design Entry HDL Options* dialog box.
2. Specify A in the *Specify Page Size* field and choose A in the *Scale to Page Size* drop-down list.

When you plot the drawing the entire drawing is plotted on a single sheet of A size paper.

Design Entry HDL allows you to scale a drawing to one of the five standard page sizes A, B, C, D, or E. In addition to the standard page sizes, you can define your own custom page sizes for the plotter in the .cdsplotinit file. However, you can scale a drawing to plot in a custom page size only with the hardcopy console command. For example, the command:

```
hardcopy MYPAGE
```

scales the currently open drawing to plot in page size MYPAGE (the custom page size you have defined in .cdsplotinit file).

#### In HPF plotting, how can I plot a drawing so that it is scaled to fit in a paper size that is larger than the size of the drawing.

For example, you have used the A SIZE PAGE page border (8 1/2 x 11 inch) symbol in your drawing and want to plot the drawing so that it is scaled to fit the complete area of paper size C paper size (17 x 22 inch).

Do the following:

1. Access the *Plotting (HPF)* tab of the *Design Entry HDL Options* dialog box.
2. Specify C in the *Specify Page Size* field and choose C in the *Scale to Page Size* drop-down list.

When you plot the drawing the entire drawing is plotted to fit the complete area of a single sheet of C size paper.

Design Entry HDL allows you to scale a drawing to one of the five standard page sizes A, B, C, D, or E. In addition to the standard page sizes, you can define your own custom page sizes for the plotter in the .cdsplotinit file. However, you can scale a drawing to plot in a custom page size only with the `hardcopy` console command. For example, the command:

```
hardcopy MYPAGE
```

scales the currently open drawing to plot in page size MYPAGE (the custom page size you have defined in the .cdsplotinit file).

#### How do I select paper sizes in HPF plotting?

For each plotter in the .cdsplotinit file, a number of paper sizes along with their dimensions are defined. You can specify any of these paper sizes in the *Specify Page Size* field of the *Plotting (HPF)* tab of the *Design Entry HDL Options* dialog box.

If you are using the `hardcopy` console command, you can use the `set papersize <size>` console command to specify the paper size.

**Note:** If an invalid paper size is specified, the first paper size defined for the plotter will be used.

#### Can I exclude some design modules from plotting and cross referencing?

Yes, in Hierarchical plotting you can choose to exclude some modules from plotting through the Hierarchy Viewer window or the xmodules.dat file. For more information on module ordering see [Module Ordering](#) on page 466.

#### **On Sun Solaris**

Set the LD\_LIBRARY\_PATH environment variable as below:

```
setenv LD_LIBRARY_PATH <your_install_dir>/tools/editor/lib $LD_LIBRARY_PATH
```

#### **On IBM AIX**

Set the LIBPATH environment variable as below:

```
setenv LIBPATH <your_install_dir>/tools/editor/lib $LIBPATH
```

#### **On HP-UX**

Set the SHLIB\_PATH environment variable as below:

```
setenv SHLIB_PATH <your_install_dir>/tools/editor/lib $SHLIB_PATH
```

### **Do set console commands affect plotting through the Plot dialog box?**

No. The `set` console commands affect plotting through the `plot` console command only. They do not have any effect on plotting through the *Plot* dialog box.

The `set` console commands do not change any directives in the `.cpm` file and are meant only for the current Design Entry HDL session. So, any change made through them is not visible in the dialog box settings.

### **How can I plot a colored schematic on color plotters?**

Do the following:

1. Access the *Plotting* (Windows) tab of the *Design Entry HDL Options* dialog box.
2. Select the Color option.

### **How can I view a spool file created by plotting?**

You can use the following third-party freeware utilities that are available on the Internet to view spool files:

- GSView for Windows
- Ghostview for SUN Solaris, IBM AIX and HP-UX
- Pageview for Sun Solaris.

#### How can I create a PDF output file of a Design Entry HDL schematic?

##### **On Windows**

To create a PDF output file of a Design Entry HDL schematic on Windows, you must have the Adobe Acrobat software from Adobe Systems Inc., installed on your machine.

In Design Entry HDL, do the following:

1. Choose *File – Plot*.

The *Plot* dialog box appears.

2. Select *Acrobat PDFWriter* or *Acrobat Distiller* in the *Printer Name* drop-down list.

3. Click *Plot*.

The *File Save As* dialog box appears.

4. Specify the PDF file name and click *Save*.

This generates the PDF output file for your Design Entry HDL schematic.

#### How can I plot a schematic for which I have only read-only permissions?

##### **Windows Plotting Mode**

On Windows, plotting is transparent. You only need to ensure that you have write permissions in the directory for the output file.

- If you are plotting through the *Plot* dialog box, you are prompted to select the directory and the file name for the output file. Select a directory in which you have write permissions.
- If you are plotting through the `plot` console command, you can specify the directory for the output file using the `set wplot_file` command as below:

```
set wplot_file <path>/<filename>.
```

Ensure that you have write permissions in the directory for the output file.

**Note:** If the plotter name consists of any special characters, precede the first such character with a \ (backslash character) and put the plotter name within quotation marks.

### ***HPF plotting Mode***

In HPF plotting, the spool file `vw.spool` is created in a temporary directory (`/tmp`) of the system from where Design Entry HDL was launched.

To create the spool file in another location, you need to set the environment variable `CDS_HPF_TMP` to a directory where you have write permissions.

**Note:** The spool file `vw.spool` is created in the directory specified using the `CDS_HPF_TMP` environment variable only if you are plotting a schematic for which you have read-only permissions. If you set this environment variable and plot a schematic for which you have write permissions, the spool file `vw.spool` is created in the project directory.

### **How do I plot all the pages in a flat schematic at the same time?**

You can use wild cards in both the `plot` and `hardcopy` console commands. Please refer to sections [Plot Command](#) and [Using the hardcopy Command](#) for details.

- If you are using Windows plotting, the following console command plots all the pages of the flat schematic `mydesign.sch.1`:  
`plot mydesign.sch.1.*`
- If you are using HPF plotting, the following console command plots all the pages of the flat schematic `mydesign.sch.1`:  
`ha a mydesign.sch.1.*`

### **Can I plot hierarchical schematics?**

Yes, you can plot hierarchical schematics in Windows plotting mode. Refer to [Hierarchical Plotting](#) for more details.

### **Is previewing supported for HPF plotting?**

Previewing is supported only in Windows plotting mode on Windows platform. It is not supported in the HPF plotting mode.

### **In Windows plotting mode, can I preview all the plot pages together?**

No, you can only preview the currently open page of the schematic.

You see multiple pages in the preview if you have selected the *Adjust to % Normal Size* option instead of the *Fit to Page* option in the *Plotting* (Windows) tab of the *Design Entry HDL Options* dialog box. If you have selected the *Adjust to % Normal Size* option, the schematic page may span a few plotter papers. All those pages are shown in the preview also. For example, if the schematic page spans four plotter papers, you can view four pages in the preview.

#### **From where can I select different setup options?**

You can select different setup options through the *Plot Setup* dialog box. This is available through the *File – Plot Setup* or the *Design Entry HDL Options* menu. In the *Plot Setup* dialog box and the *Plot* dialog box, the *Properties* button opens a dialog box showing the system level plot setup options.

It is not recommended to use *Properties* dialog box for setup options.

---

# **Design Techniques**

---

## **Introduction**

This chapter introduces the three basic design techniques: flat, structured, and hierarchical. One of these three techniques may best meet your needs:

- **Flat Design Technique**

The flat design technique is an efficient method for creating a design that is small and does not re-use portions of the circuitry. Flat designs are required for complete backannotation of the design and are more convenient for troubleshooting. Flat designs can include multiple drawing pages.

- **Structured Design Technique**

The structured design technique allows abbreviated bus structures and minimizes the required number of parts and interconnections. Structured design techniques using the SIZE property support designs that use large bused signals, register depth, and memory depth.

- **Hierarchical Design Technique**

The hierarchical design technique uses symbolic representations of circuitry for functions that are repeated throughout a design. Large designs that can be broken into functional modules or designs that re-use portions of circuitry can be efficiently created with a hierarchical technique.

Although all designs can be entered as flat drawings, choose the method most appropriate to your particular design. Design Entry HDL and the other system design tools are specially designed to operate efficiently with structured and hierarchical techniques.

## **Flat Designs**

The flat design method is the most straightforward technique for creating a design with the Cadence system design tools. In a flat design, all parts on the drawing come from Design

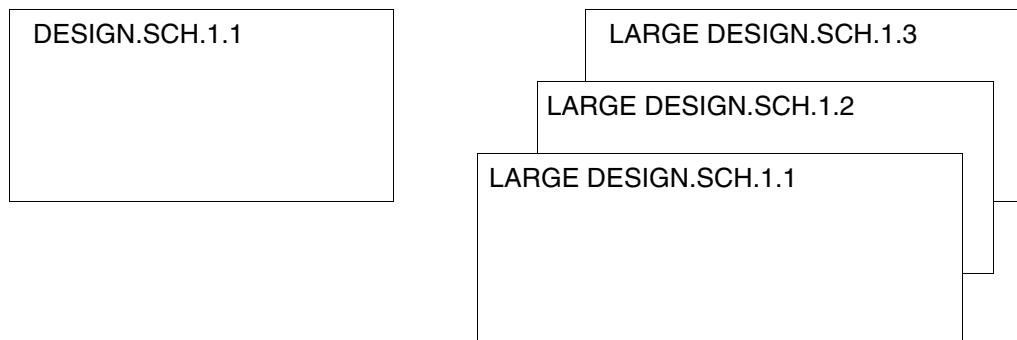
Entry HDL or user-defined libraries and are one-to-one logical representations of the physical parts. The entire interconnecting wiring within the design is entered pin-to-pin.

Flat designs are best suited for small designs that do not have sophisticated bus requirements and do not re-use portions of circuitry. Also, if the design must be completely backannotated with pin and physical location numbers, a flat drawing is required.

## Creating a Flat Design

Both single-page and multiple-page flat drawings can be created with Design Entry HDL and processed by the Cadence design analysis programs.

**Figure 16-1 Single and Multiple Drawing Pages**



Some designs are small enough to fit on one page of a drawing.

To create a single-page design,

1. Specify the drawing name with *File – Open*.
2. Use Design Entry HDL to draw the design on the screen.
3. Use *File – Save* to store the design on the disk.
4. Use *File – Export Physical* to package the design.

If the drawing is too large to fit on one page, create a multiple-page drawing.

To create a multiple-page drawing , do the following:

1. Specify the drawing name with *File – Open* and create page1 of the design.
2. Use *File – Save* to save page 1.
3. To begin page 2 of the drawing, use *File – Edit Page/Symbol – Insert Page*.

4. Use *File – Save* to save page 2.
5. Create subsequent pages of the drawing in the same way.

All pages of a multiple-page design have the same drawing name. The system links all drawings with the same name. If the names are different, each page is treated as a separate drawing.

Give signals that cross page boundaries the same signal name on subsequent pages. Signals with the same name have an implicit connection, even if they appear on different pages. For example, the signal SYSTEM CLK on pages 1 and 3 has the same effect as being on the same page with both instances wired together.

## Concurrent Engineering of Flat Designs

Design Entry HDL allows teams of designers to simultaneously work on different schematic pages in a flat design. When a designer is editing a page in the design, Design Entry HDL locks the page and does not allow other designers to modify the page. For more information on page locking, see [What is page locking?](#) on page 47.

Cadence recommends that you disable netlisting of the design if multiple designers are working on different schematic pages in a flat design. To disable netlisting of the design, do the following:

1. In Design Entry HDL, choose *Tools – Options*.
2. The *Design Entry Options* dialog box appears.
3. Select the *Output* tab.
4. Deselect the *Create Netlist* check box.

After all the designers have completed their changes, you can enable netlisting of the design and save the design.

## Considerations of Flat Designs

Keep these considerations in mind when you create a flat drawing:

- Flat designs take longer to create and process than structured and hierarchical designs.
- Flat designs tend to be cluttered and hard to read unless special care is taken to organize and layout the design.

- Troubleshooting errors in a large, multiple-page flat design is time-consuming and difficult.

## Structured Designs

The structured design method facilitates the entry and analysis of sophisticated designs that make use of bused signals, memory and, register depth. A structured design minimizes the number of interconnections and parts on the schematic.

### Creating a Structured Design

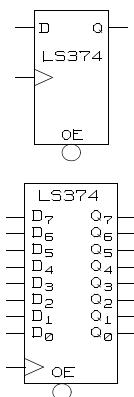
You use Design Entry HDL commands to enter and store your drawing. The main difference between a structured design and a flat design is the use of special library parts and the SIZE and TIMES properties.

#### SIZE Property

The SIZE property is attached to a symbol and is used to specify the width of pin names and signal names and to define size expansion.

For example, there are two versions of an LS374 octal register in the LSTTL library. Version 1 is a one-bit slice of the part. It accepts a vectored D input and produces a vectored Q output. Version 2 is the full-chip representation of the LS374 with all eight input and output bits explicitly shown.

*LS374 symbol  
Versions 1 & 2*

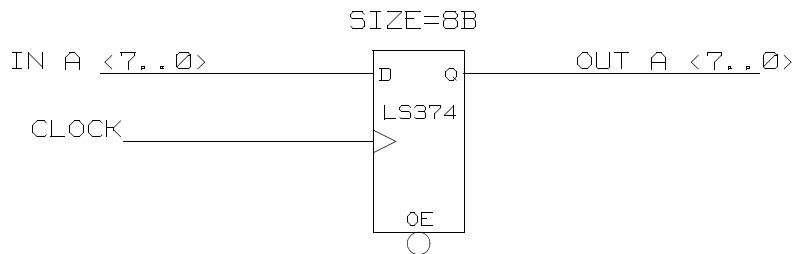


Version 1 is sizeable, which implies that you can specify the number of bits the part can represent. Library parts are generally developed with version 1 being sizeable. The *show vectors* command displays the pin names of a selected part allowing you to verify that a part is sizeable.

You attach the SIZE property to version 1 of the LS374 part to define the number of bits the pins D and Q represent. The signal syntax for bus notation is used to specify a range of bits for the input and output signals.

Figure 16-2 illustrates how you can use version 1 of the LS374 part in a structured design. In this example, the number of bits is set to 8 (SIZE = 8B) any number of bits can be specified to meet your requirements.

**Figure 16-2 Using the SIZE Property To Structure LS374**



Version 2 of LS374 is a flat representation of the part. Each pin on the drawing represents a pin in the physical package.

**Figure 16-3 Using Version 2 of LS374**

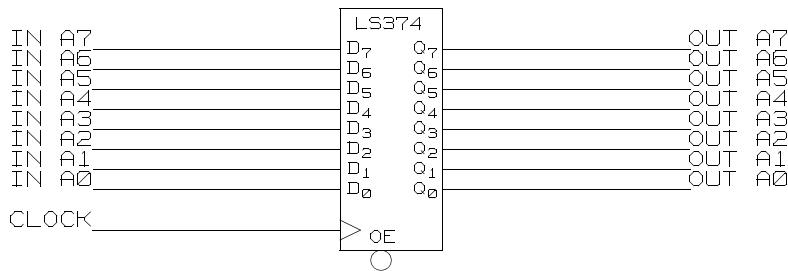
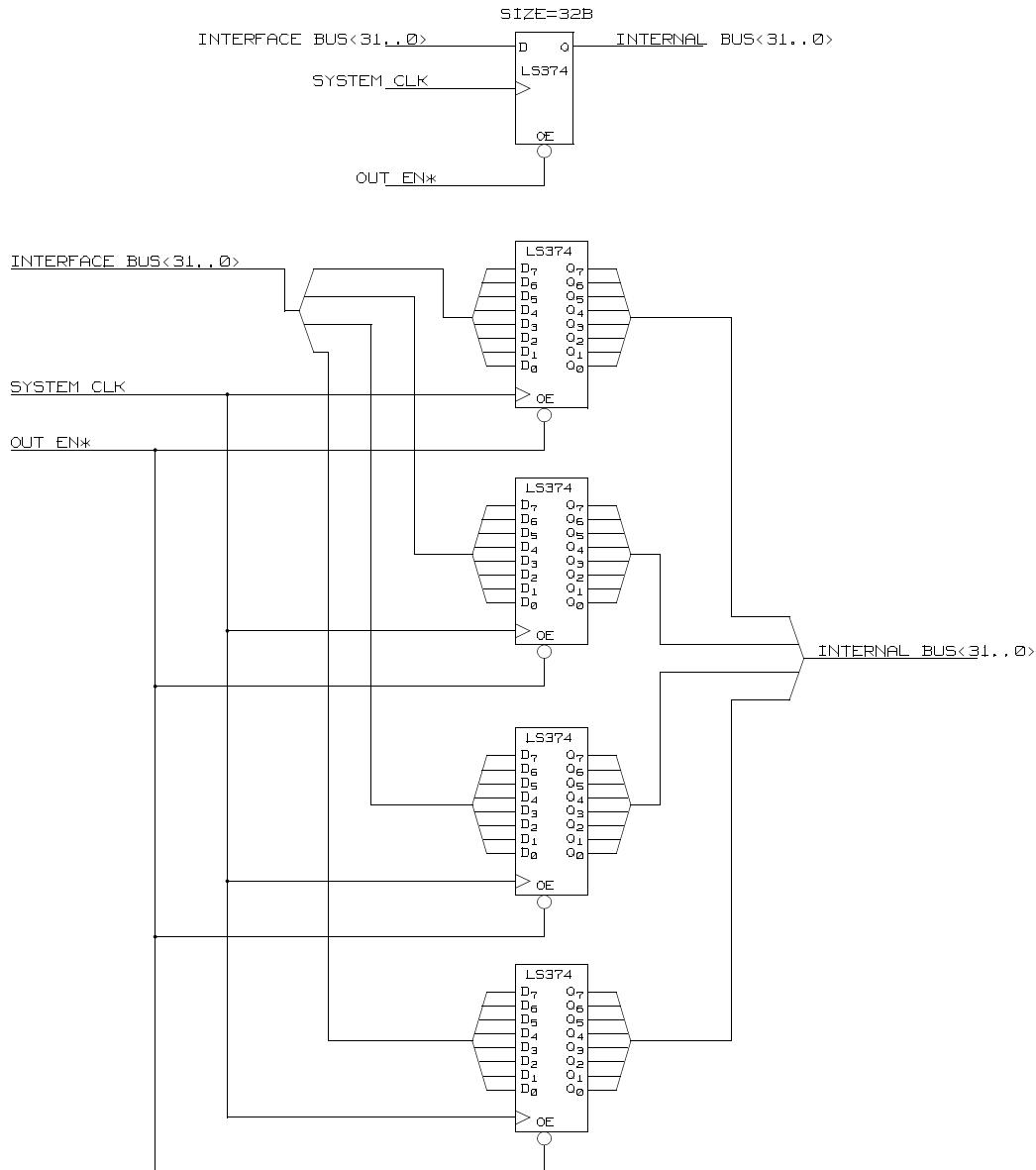


Figure 16-4 on page 618 illustrates the difference between the structured design and flat design techniques. Using the SIZE property can greatly minimize the number of parts and interconnections required. Also, you can avoid many possible entry errors.

## Allegro Design Entry HDL User Guide

### Design Techniques

**Figure 16-4 Structured and Flat Design Techniques**



You can control the mapping based on how the bus name on the sizeable pin is defined. For example, you can use the **SD<7..0>** or **SD<0..7>** as the bus name for a sizeable pin **D** with the size defined as eight.

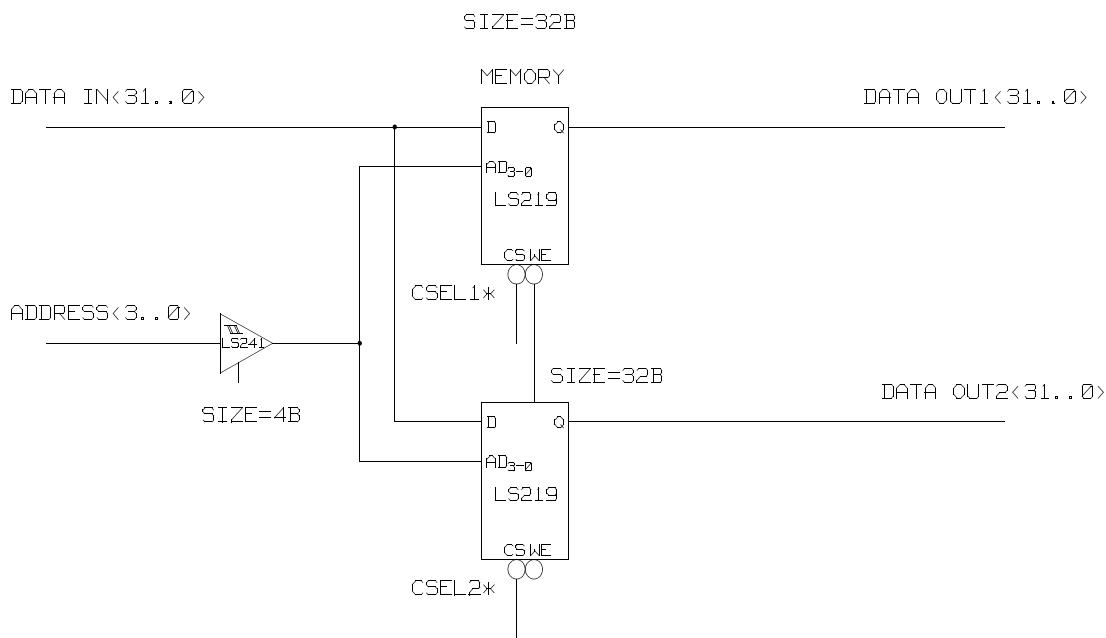
Packager-XL first assigns the MSB of the symbol pin, in this case the section from pin eight with the first bit of the signal name as defined on the schematic net. You can check the physical net name in the **pstxnet.dat** file, which will be transferred to the board layout. If

the signal name on the schematic is defined as SD<7..0>, then the SD<0> net name is mapped to the D<0> pin, and so on.

### **TIMES Property**

The TIMES property is used with the SIZE property on structured designs. TIMES allows you to create your structured design to data book specifications. TIMES can be used in cases where the SIZE property causes loading errors. For example, in Figure 16-5, a single part is driving too many inputs on SIZE-replicated parts.

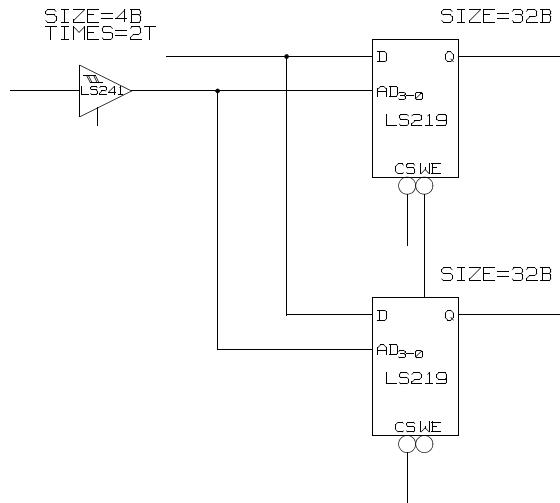
**Figure 16-5 Structured Design with the SIZE Property**



In this design, the 4-bit 3-state buffer drives 64 bits of memory. Four sections of LS241 do not have the drive capability to handle 16 memory packages; in such cases, Packager-XL reports a loading error.

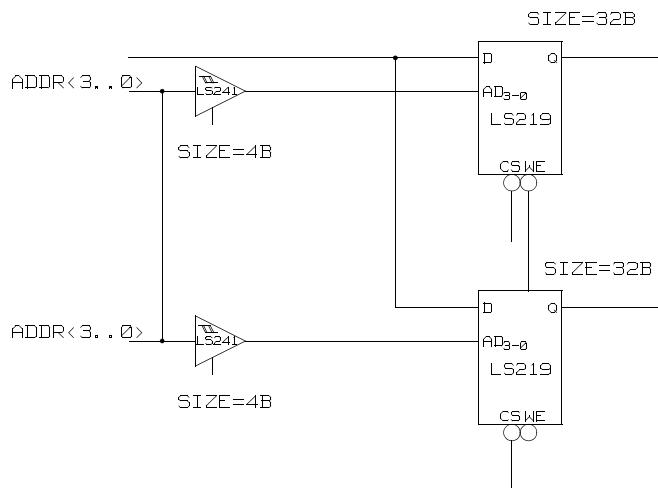
The TIMES property is used to correct loading violations in structured designs, as illustrated in Figure 16-6.

**Figure 16-6 Using the TIMES Property**



In this example, the TIMES property informs the system that two instances of a 4-bit, 3-state buffer are needed. The system checks the loading and balances the load between all the parts being driven. Using the TIMES property in this design is equivalent to adding another part and more interconnections, as illustrated in [Figure 16-7 on page 620](#).

**Figure 16-7 Manually Balancing Loads**



Using the TIMES property eliminates the need to manually balance the load and enter more data.

## The Standard Library

Cadence provides a Design Entry HDL library of standard parts that allow you to define and manipulate signals in a structured design. The Standard library is automatically associated with your search list of libraries so that you can conveniently use these parts in your designs. Although the bodies in the Standard library can be used for any of the design techniques, many of them are created especially for structured designs.

The library contains merge bodies for merging signals, tap bodies for tapping bits from buses, and several other special parts.

## Benefits of Structured Designs

Using a structured design technique has these advantages:

- Creating structured designs can dramatically decrease the design cycle time.  
The amount of data entered into Design Entry HDL is reduced, resulting in faster schematic entry. Also, the analysis tools run more efficiently on structured designs because they can process multiple bits in parallel.
- Errors in design entry are minimized because of the reduced number of parts and simplified interconnections.
- The resulting print is less cluttered, easier to read, and easier to understand.

## Considerations of Structured Designs

Packager-XL produces an easy-to-read cross-reference list for the logical-to-physical mapping of the design data. These lists are used with the structured print set for design troubleshooting. Members of the design team responsible for troubleshooting the structured design must be educated on how to read structured print sets and how to reference the physical information.

## Hierarchical Designs

The hierarchical design technique is an efficient approach to developing complex designs that can be organized into modules. This method is useful for designs that re-use many of the same circuit functions and for isolating portions of the design for teamwork assignments.

A hierarchical design results in print sets that are easy to read and produces modules that can be effectively debugged. Hierarchical designs, like structured designs, reduce the

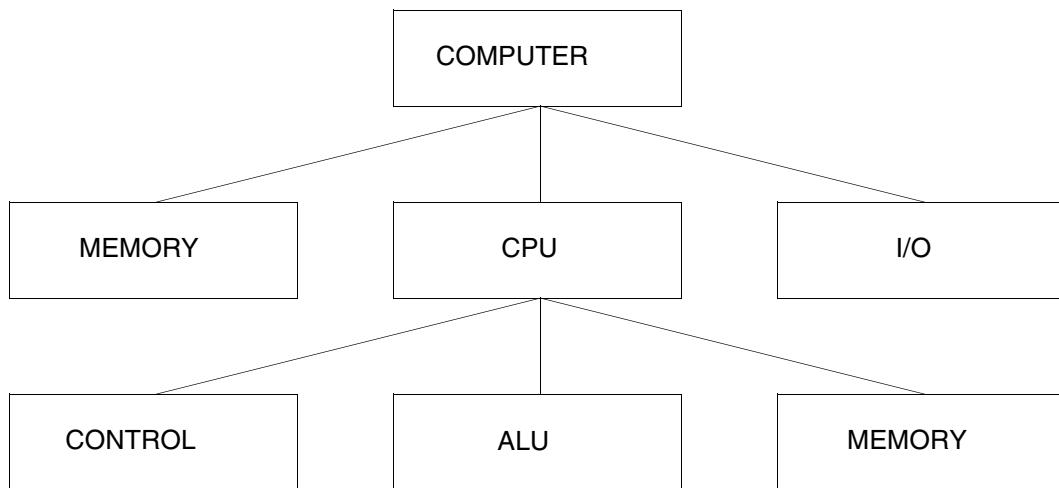
amount of data entry and interconnections required by the design, thereby reducing the chance for error. Also, all the design tools can be used to analyze partial designs (modules).

### Creating a Hierarchical Design

Creating a hierarchical design is a natural extension of the entire design process. If the design to be implemented is a computer, the design begins by planning the constituent parts of the computer.

The computer can be divided into the CPU, MEMORY, and I/O modules. The CPU module can be further divided into the ALU, MEMORY, and CONTROL modules. This represents three levels of hierarchy in the design. There are no limits to the number of levels you can include in a hierarchical design. [Figure 16-8](#) on page 622 shows the hierarchical levels of the computer.

**Figure 16-8 Levels of Hierarchy**



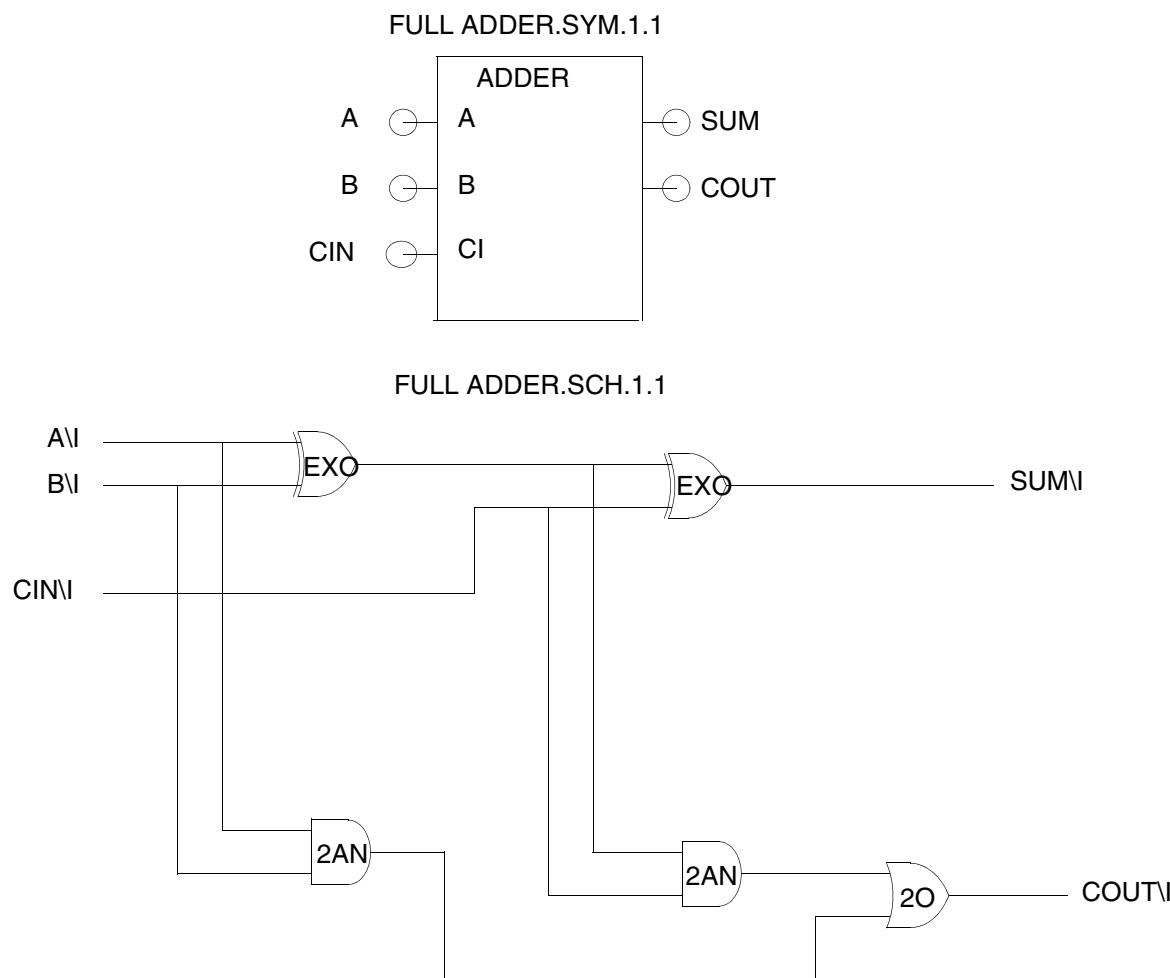
After you plan the modules of the design, you implement the design using the following basic procedure:

1. Create a schematic drawing that represents a functional portion (module) of your design (for example, counter, register file, memory unit, or control blocks of circuitry).  
You can start at the most detailed level of the design hierarchy.
2. Test that drawing, processing it with other system design tools to check its timing and logic functions.  
You can efficiently debug each module of the design as you work.

3. Create a symbol drawing to represent the design module.
  4. Create a new schematic drawing and add the required number of symbol representations to it, building a circuit using the modules.
- You have added a symbol that represents the functional module you created in Step 1. The symbol drawing acts as a pointer to the schematic definition of the circuit.
5. Continue to create the corresponding schematic/symbol representations for each of the defined modules in the design, working up the levels of hierarchy.

Figure 16-9 on page 623 illustrates the schematic and symbol drawings defined for use in a hierarchical design. Instead of having to wire together the gates of the Full Adder circuit whenever it is needed, you add the `Full Adder.body` drawing in its place.

**Figure 16-9 Full Adder Logic and Symbol Drawings**



Every level of hierarchy (except the lowest level) is made up of a schematic drawing and symbol drawing pair. The schematic drawing defines the functional circuitry for the design module. The symbol drawing is the picture or symbol that represents the logic function. The symbol points to the functional representation, but does not take up as much space in higher levels of the hierarchy. The result is a well organized and understandable design print set.

## Creating Symbols

When you create a hierarchical design, you draw simple blocks (also called symbols) to represent the specific logic for each element of the design. Design Entry HDL provides you with the tools to draw bodies and establish relationships between the symbol drawings and the logic drawings that they represent.

The pins on the bodies that correspond to signals in the logic drawing must have the same name. Additionally, these signals in the schematic drawing are given the interface signal property (`I`). This signal property is used to indicate an interface signal from a higher level drawing.

There are two ways to make a schematic drawing in Design Entry HDL.

- Use *Tools – Generate View* to automatically create a symbol drawing either from an existing schematic drawing or from a list of pins. See the Genview section of *Allegro Design Entry HDL Reference Guide* for details on this option.
- Use *File – New* to create a symbol by drawing.

## Editing Symbols

To edit a symbol drawing:

1. Use *File – Open* to edit a symbol drawing.

The *View Open* dialog box is displayed.

2. Specify the name of the *Cell*.

3. Select the *View as Symbol*.

A grid is displayed with a cross to mark the origin of the symbol. The default grid for symbol drawings is 0.05 inches, and every second grid point is displayed. This default grid is twice as fine as the default grid for logic drawings.

To change the grid for symbol drawings, use the `set command option default_symbol_grid`. To change the grid for schematic drawings, use the `set command option default_grid`. Both options require a numeric grid size argument.

## Allegro Design Entry HDL User Guide

### Design Techniques

---

4. Use the Edit options to modify the drawing. You can do any of the following to modify the drawing:

- a. Use *Edit – Move* to move the name away from the origin.

You can also move the origin, but be very careful if you do so. Do not place the origin at a connection point (pin end) for the symbol.

- b. Use *Wire – Draw* to draw the outline of the symbol around the origin symbol.

The grid is used as a guideline for the appropriate size and shape of the symbol.

- c. Add wire stubs for the pins.

If you add wire stubs for the pins, make them 0.1 inch (one grid segment) long and place them on visible grid lines, so that the symbol can be correctly wired on schematic drawings.

Be sure to place pins only at visible grid points on the symbol drawing. This guarantees that all of the symbol pins will be on the grid when the symbol is used in a logic drawing. Use the unmarked grid points on the symbol drawing only for placing notes and properties.

See “[Defining Low-Asserted Pins](#)” on page 627 for information about defining low-asserted pins.

- d. Use *Wire – Dot* to place a dot at the end of each pin.

Place dots on displayed grid intersection points. Use the right mouse button to ensure that the dot is properly placed at the end of the wire.

- e. Use *Wire – Signal Name* to add pin names (corresponding to the signal names in the related logic drawing) to each pin.

In a symbol drawing, the SIG\_NAME properties added by *Wire – Signal Name* are understood as PIN\_NAME properties. They can only be attached to pin connections. The name must match the corresponding name in the logic drawing, except for the omission of the interface property (I). Use *Display – Attachments* to ensure that all pin names are attached properly.

- f. Use *Text – Note* to place labels within the symbol drawing.

This makes the purpose of the symbol and each pin clear.

- g. Mark the clock signal with a wedge. Choose *Wire – Draw* and press the middle button to draw diagonal lines.

- h. Use *File – Save* to save the symbol drawing.

**Note:** Use *File – Save As* if you want to save the edited symbol drawing as a new symbol.

For information on editing split symbols, see “[Editing Split Symbols](#)” on page 416.

### The pinnames Command

When you create a hierarchical schematic drawing and symbol drawing pair, you can use the PIN NAMES symbol in the standard library to transfer the PIN\_NAME properties from the symbol drawing to the associated schematic drawing. When you add the PIN NAMES symbol to a schematic drawing, all the pin names in a symbol drawing of the same name are attached to the PIN NAMES symbol. You can reattach the names to appropriate signals in the schematic drawing. This eliminates the need for retyping signal names and reduces mislabeled signal names or missing interface scope (\\_I) signal properties.

The `pinnames` console window command adds a PIN NAMES symbol to a schematic drawing and attaches the pin names to the symbol. To do this,

1. Use *File – Open* to edit a symbol drawing.
2. Add pin names to the symbol using *Wire – Signal Name*.

When used in a symbol drawing, *Wire – Signal Name* attaches a PIN\_NAME property to the specified pin.

3. Save the symbol drawing using *File – Save*.
4. Create a schematic drawing by the same name as the symbol drawing.

For example, if the symbol drawing is `CLOCK.SYM.1.1`, type the following in the Design Entry HDL console window:

```
edit clock
```

The `CLOCK.SCH.1.1` drawing will contain the logic that the symbol represents. Place all the required parts and attach wires as required.

5. Type `pinnames` in the Design Entry HDL console window.
6. Click to add the PIN NAMES symbol to the `CLOCK.SCH.1.1` drawing.

Each pin name defined on the symbol drawing `CLOCK.SYM.1.1` appears in the schematic drawing attached to the PIN NAMES symbol. A \\_I suffix (scope = interface) is also suffixed to each signal name. When transferred to the schematic drawing, each pin name is identified with a SIG\_NAME property.

Do one of the following if you want to view the signal and property names:

## Allegro Design Entry HDL User Guide

### Design Techniques

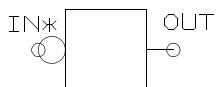
- ❑ Place the cursor on a pin name attached to the PIN NAMES symbol
- ❑ Choose *Text > Attributes* and click on the PIN NAMES symbol to view the signal and property names in the *Attributes* dialog box.

**Note:** If you choose *Component – Add* and add the PIN NAMES symbol on the schematic drawing, the pin names on the symbol drawing will not get automatically attached to the PIN NAMES symbol. Run the `check` console window command. The pin names on the symbol drawing are now attached to the PIN NAMES symbol on the schematic drawing.

7. Move the PIN NAMES symbol to an unused area of the schematic drawing.
8. Choose *Text – Reattach* to reattach the individual signal names from the PIN NAMES symbol to the appropriate signals on the schematic drawing.
9. Choose *Display – Attachments* to ensure that the signal names have been reattached to the appropriate signals.
10. For drawing clarity, choose *Edit – Move* to relocate the signal names near the associated signals.
11. Delete the PIN NAMES symbol.

### Defining Low-Asserted Pins

Use a circle instead of a wire to represent a low-asserted (bubbled) pin. You can use either *Edit – Circle* or *Edit – Arc* to add circles. The circle should be 0.1 inch in diameter. A dot is placed on the appropriate grid intersection point on the circumference of the circle to mark the connection point. The signal name should also be low-asserted (\*).



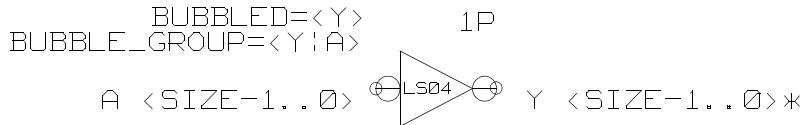
To define pins that can be either bubbled or unbubbled,

- Draw a symbol and represent the pins with both wires and circles.

There must be a line that extends across the diameter of the circle so that both representations are available. Be sure to place a dot at the connection point. You also attach the BUBBLE property to the origin of the symbol to define which pins are bubbled when the part is added to a drawing.

You can also define groups of pins that automatically change state when one of the pins in the group is bubbled. These are called bubble groups.

**Figure 16-10 Using Bubbleable Pins**



When you add the symbol to a drawing, use *Component – Bubble Pins* to toggle the pin from bubbled to unbubbled.

## Benefits of Hierarchical Designs

The benefits of creating hierarchical designs are similar to those of structured designs:

- Creating hierarchical designs can dramatically decrease the design cycle.  
Since symbol drawings act as pointers to schematic drawings, a large amount of data entry need not be repeated.
- The functional schematic drawing that a symbol represents can be compiled once and linked to all locations where the symbol is used.
- The number of entry errors is minimized because the amount of schematic entry is reduced.
- Each module can be fully tested before it is incorporated into higher levels of the design because functional modules are created when defining a hierarchical design.  
Testing can be performed incrementally rather than at the end of the design process.
- Hierarchical designs result in designs that are well organized and easy to read and understand.

## Comparing Design Techniques

The design methodologies discussed in this chapter (flat, structured, and hierarchical) are all appropriate for solving design problems. You must weigh the benefits and considerations of each technique before deciding which method to use.

There is no restriction against combining these methods in design drawings. Hierarchical and structured design techniques are often used together to provide maximum flexibility and

## **Allegro Design Entry HDL User Guide**

### Design Techniques

---

efficiency for the design engineer.

---

#### **Flat Designs**

Best suited for	Small designs Designs that do not re-use modules Designs that do not use buses
Benefits	Short learning curve
Considerations	Long design cycle time Cluttered print sets

#### **Structured Designs**

Best suited for	Designs that use sophisticated bus structures
Benefits	Shortened design cycle time Fewer errors during data entry Less cluttered print sets Print sets organized in the logical flow of the design Cross-reference listings
Considerations	Additional training required for design troubleshooters

#### **Hierarchical Designs**

Best suited for	Designs that re-use modules Large designs that can be organized into separate components
Benefits	Shortened design cycle Fewer data entry errors Easy-to-read print sets Cross-reference listings Effective debugging capability Print sets organized in logical (top-down) flow of design
Considerations	Additional training required for design troubleshooters

---

# **Allegro Design Entry HDL User Guide**

## Design Techniques

---

---

## **Simulating using PSpice Simulator**

---

### **What is PSpice?**

You can simulate designs created in Design Entry HDL using PSpice library components with the PSpice simulator and the Advanced Analysis add-on program.

PSpice is a simulation program that models the behavior of a circuit. PSpice simulates analog-only, mixed analog/digital, and digital-only circuits. Used with Design Entry HDL, PSpice is much like a software-based breadboard of your circuit. You can use it to test and refine your design before manufacturing the physical circuit board. For more information, see [PSpice User Guide](#).

Advanced Analysis is an add-on program for PSpice. Use these four Advanced Analysis tools to improve circuit performance, reliability, and yield:

- The Sensitivity tool allows you to examine the degree to which each component affects circuit behavior by itself and in comparison with other components. It also varies all tolerances to create worst-case (minimum and maximum) measurement values.
- The Optimizer tool allows you to analyze analog circuits and systems. It helps you modify and optimize analog designs to meet your performance goals. Optimizer fine tunes your designs faster than trial and error bench testing can. Use Optimizer to find the best component or system values for your specifications.
- The Monte Carlo tool allows you to predict the statistical behavior of a circuit when part values are varied within tolerance. Monte Carlo also calculates yield, which can be used for mass manufacturing predictions.
- The Smoke tool allows you to evaluate component stress due to power dissipation, increase in junction temperature, secondary breakdowns, or violations of voltage/current limits.

For more information, see [PSpice User Guide](#).

## Notes for Using Design Entry HDL with PSpice

Note the following when using Design Entry HDL with the PSpice Simulator and the Advanced Analysis add-on program:

- The PSpice Simulator and other PSpice-related menus are only available on Windows.
- You must use components from the PSpice libraries in your design if you want to simulate the design using PSpice. See the *Library List* for the list of PSpice library components.
- If you are creating a schematic that you want to simulate using PSpice, you must not use components from the `element` library. Instead, use components from the `pspice_elem` library.
- The components that you want to simulate using the Advanced Analysis add-on program must be from the Advanced Analysis libraries. See the *Advanced Analysis Library List* for the list of Advanced Analysis library components.
- If you are using the Advanced Analysis add-on program, ensure that the `templates` library is selected for your project. For more information on the procedure for selecting libraries for your project, see [Selecting Libraries for a Project](#) on page 72.

If you are not including the `cds.lib` file located at  
`<your_install_dir>\share\cdssetup\` in your project `cds.lib` file, you must define the `templates` library in your project `cds.lib` file as below:

```
DEFINE templates <your_install_dir>\share\library\templates
```

For more information on the `cds.lib` file, see the [Allegro Design Entry HDL Libraries Reference](#).

**Note:** The `templates` library contains models that are used by the Advanced Analysis add-on program.

## Conversion of Old PSpice Projects to New Project Format

When you open a PSpice project created in pre-14.2 versions of Design Entry HDL, the *Update Analog Project* dialog box appears.

The *Update Analog Project* dialog box allows you to convert your PSpice project to the current Design Entry HDL format. Converting your PSpice project to the latest Design Entry HDL format has the following benefits:

- The directory structure for PSpice projects makes it easier to manage the PSpice files for the project. For more information, see [New Directory Structure for PSpice Projects](#) on page 634.

## Allegro Design Entry HDL User Guide

Simulating using PSpice Simulator

---

- You can use the simulation profile features if your project is in the old format. For more information on the new simulation profile features, see the *PSpice Simulator Online Help*.

### **To convert a project to the new format**

1. Select the *Convert to new format* option to convert the project to the new project format.
2. Click *OK*.

The *Conversion Results* dialog box appears displaying the details of the conversion process.

3. Do one of the following:
  - Click *Save* to save the details of the conversion process in a file named <projectdirectoryname>\_convert.txt in the library where the root design is located.
  - Click *Cancel* to close the *Conversion Results* dialog box without saving the details of the conversion process.

The project in the new format is opened in Design Entry HDL.

### **To automatically convert projects to the new format**

1. Select the *Convert to new format* option to convert the project to the new project format.
2. Select the *Do not ask this question again* check box.

Any PSpice project created using pre-14.2 versions of Design Entry HDL that you open in later versions of Design Entry HDL will be automatically converted to the new format.

Later on, if you want to disable automatic conversion of old PSpice projects to the new project format, delete the following entry in the Windows registry:

HKEY\_CURRENT\_USER\Software\Cadence Design Systems\Concept-HDL\Analog Project Conversion\Convert

**Note:** The value of the registry entry can be 0 (indicating that projects should never be automatically converted) or 1 (indicating that projects should always be automatically converted).

## Allegro Design Entry HDL User Guide

Simulating using PSpice Simulator

---

### **To cancel conversion of the project to the new format**

- Select the *Do not convert* option and click *OK* if you do not want to convert the project to the new format.

The project in the old format is opened in Design Entry HDL.

If you do not want to be prompted again to convert any analog project to the new format, do the following:

1. Select the *Do not convert* option
2. Select the *Do not ask this question again* check box and click *OK*.

Later on, if you want to enable conversion of old PSpice projects to the new project format, delete the following entry in the Windows registry:

HKEY\_CURRENT\_USER\Software\Cadence Design Systems\Concept-HDL\Analog Project Conversion\Convert

**Note:** The value of the registry entry can be 0 (indicating that projects should never be automatically converted) or 1 (indicating that projects should always be automatically converted).



#### *Important*

If you do not convert a project to the new format, you cannot use the simulation profile features in Design Entry HDL. For more information on the new simulation profile features, see the *Allegro PSpice Simulator Online Help*.

## New Directory Structure for PSpice Projects

All files related to PSpice projects created using a pre-14.2 Design Entry HDL were maintained in the `cfg_analog` view of a design.

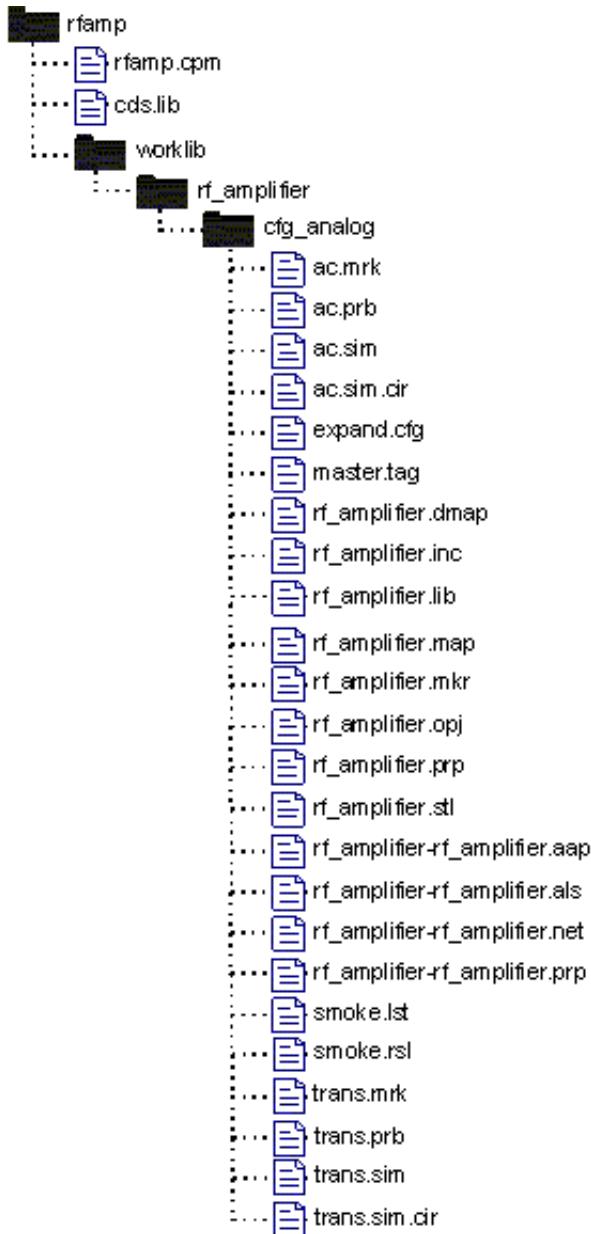
**Note:** When you open a project with the `cfg_analog` view in post-14.2 Design Entry HDL versions, the Update Analog Project dialog box appears. If PSpice is not in the path, the project might fail. You can set `CDS_CONCEPT_DISABLEPSpice=true` to avoid this.

## Allegro Design Entry HDL User Guide

Simulating using PSpice Simulator

---

**Figure 17-1 Directory structure for the files in cfg\_analog view of RF\_AMPLIFIER design in RFAMP project created using pre-14.2 Design Entry HDL**



In the preceding figure, the `rfamp` project has a design named `rf_amplifier`. All the PSpice related files for the design are stored in the `cfg_analog` view of the design.

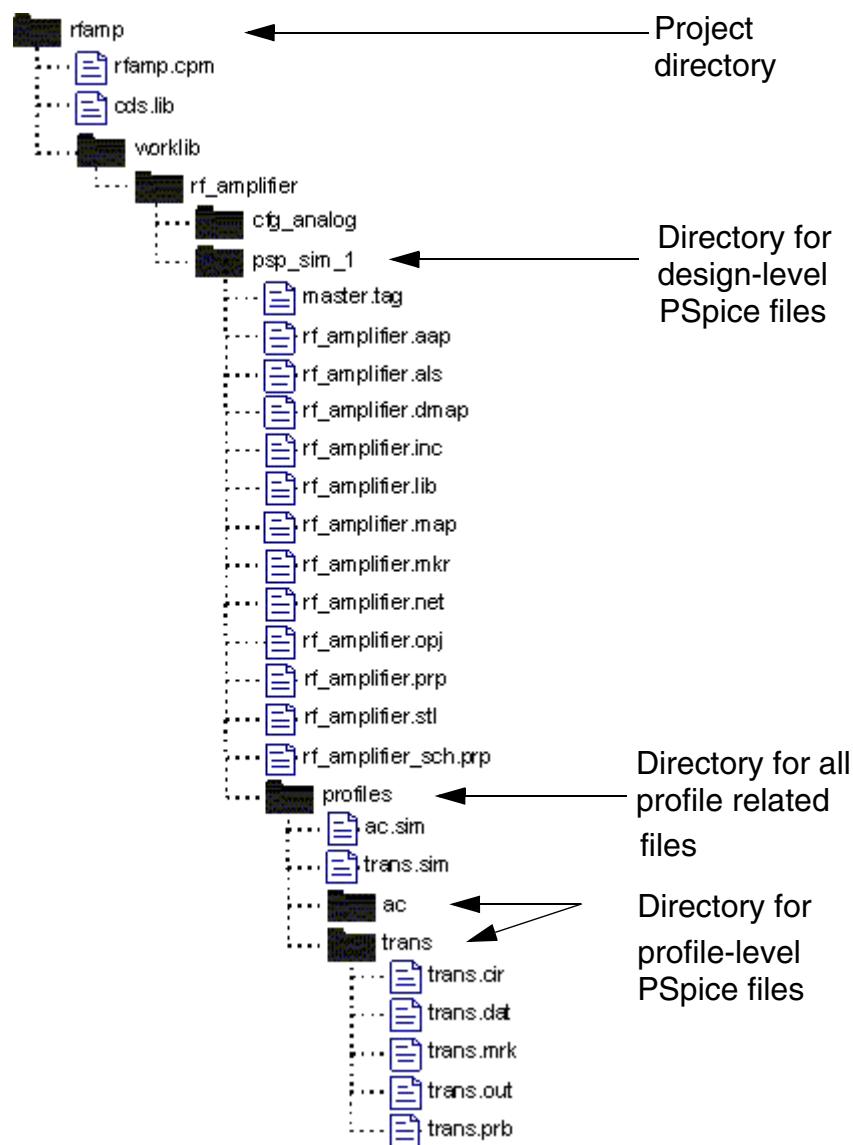
In the directory structure for PSpice projects, the design level and simulation profile-level PSpice files are organized in their respective directories. This makes it easier to manage the files for the project.

## Allegro Design Entry HDL User Guide

### Simulating using PSpice Simulator

**Note:** If you open a PSpice project that was created using Design Entry HDL version 14.2 or earlier, you will be prompted to convert the project to the current format. For more information, see [Conversion of Old PSpice Projects to New Project Format](#) on page 632.

**Figure 17-2 New directory structure for PSpice files related to RF\_AMPLIFIER design in RFAMP project**



In the new directory structure all the PSpice-related files for the `rf_amplifier` design are maintained in a view named `psp_sim_1`.

## Allegro Design Entry HDL User Guide

### Simulating using PSpice Simulator

---

- The PSpice files related to the design are maintained in the `psp_sim_1` view of the design. For more information, see [How are files configured at the design level maintained in the new directory structure for PSpice projects?](#) on page 637.
- The PSpice files related to all the profiles for the `rf_amplifier` design are maintained in the `profiles` folder in the `psp_sim_1` view of the design.
- The PSpice files related to each simulation profile are stored in a directory that has the same name as the profile. For example, the `.CIR` and `.DAT` PSpice files related to the `Trans` simulation profile are maintained in the `Trans` folder under the `profiles` folder in the `psp_sim_1` view of the `rf_amplifier` design.

The model libraries, stimulus files and include files configured for each profile are also stored in the directory that has the same name as the profile. For example, the model libraries, stimulus files and include files configured at the `Trans` profile level are stored in the `Trans` folder under the `profiles` folder in the `psp_sim_1` view of the `rf_amplifier` design. For more information, see [How are files configured at the profile level maintained in the new directory structure for PSpice projects?](#) on page 638.

- The `cfg_analog` view is used as the configuration view for design expansion.

**Note:** When you convert the project to the new format, Design Entry HDL retains the PSpice files that existed in the `cfg_analog` view at the time of conversion. The PSpice files in the `cfg_analog` view will not be updated after you convert the project to the new format.

### How are files configured at the design level maintained in the new directory structure for PSpice projects?

The model libraries, stimulus files and include files configured at the design level are stored in the `psp_sim_1` view of the design. For example, in Figure 17-2, the model libraries, stimulus files and include files configured at the design level are stored in the `\rfamp\worklib\rf_amplifier\psp_sim_1\` directory. The `rf_amplifier.stl` stimulus file in the `\rfamp\worklib\rf_amplifier\psp_sim_1\` directory is an example of a PSpice file related to the design.

**Note:** When you create a new simulation profile by importing the settings from another simulation profile that exists in another project, only the simulation settings are inherited from the source simulation profile. The files configured at the design level for the source simulation profile are not copied over to the `psp_sim_1` view of the design for which you are creating the new simulation profile.

## **Allegro Design Entry HDL User Guide**

Simulating using PSpice Simulator

---

### **How are files configured at the profile level maintained in the new directory structure for PSpice projects?**

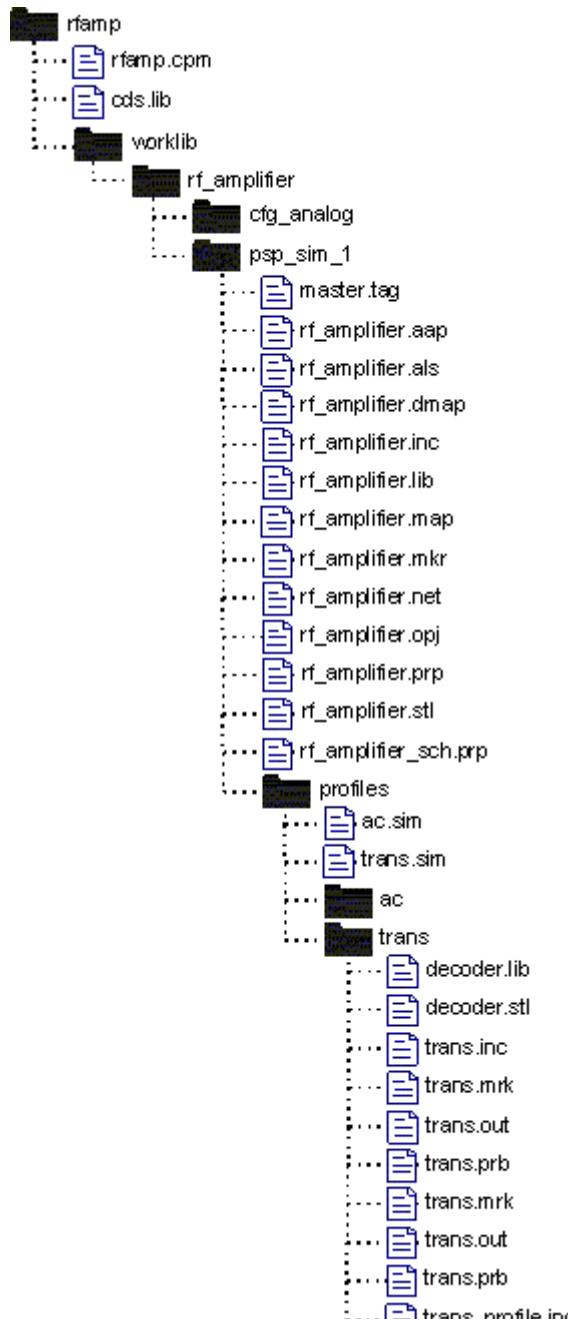
The PSpice files related to all the profiles for a design are maintained in the `profiles` folder in the `psp_sim_1` view of the design. For example, in Figure 17-3, the PSpice files related to all the profiles for the `rf_amplifier` design are maintained in the `\rfamp\worklib\rf_amplifier\psp_sim_1\profiles\` directory.

The model libraries, stimulus files and include files configured at the profile level are stored in a directory that has the same name as the profile. For example, in Figure 17-3, the PSpice files related to the Trans simulation profile are maintained in the `Trans` folder under the `\rfamp\worklib\rf_amplifier\psp_sim_1\profiles\` directory.

## Allegro Design Entry HDL User Guide

### Simulating using PSpice Simulator

**Figure 17-3 Directory structure directory for PSpice files related to RF\_AMPLIFIER design with files configured for the Trans profile**



An include file named `<profilename>_profile.inc` is created in the directory for the simulation profile. This file contains information on the model libraries, stimulus files and include files configured for that profile. For example, in Figure 17-3, the Trans profile directory contains a `Trans_profile.inc` include file that includes information on the `decoder.lib`

## Allegro Design Entry HDL User Guide

### Simulating using PSpice Simulator

---

model library, `decoder.stl` stimulus file and the `Trans.inc` include files configured for the Trans profile.

You must not delete the `<profilename>_profile.inc` file in the directory for a simulation profile.

**Note:** When you create a new simulation profile by importing the settings from another simulation profile that exists in the same project or in another project, the files configured at the profile level for the source simulation profile are copied to the directory for the new simulation profile. The files configured at the design level for the source simulation profile are not copied over to the `psp_sim_1` view of the design for which you are creating the new simulation profile.

### What should I do if the schematic for a converted PSpice project uses FILESTIM $n$ parts from the SOURCE library?

If you have specified only the name of the stimulus file as the value of the FILENAME property on a FILESTIM $n$  part, you must specify the path to the stimulus file in the value for the FILENAME property on the FILESTIM $n$  part.

# Design Entry HDL Bias Display with PSpice

## What is a bias point value?

PSpice generates bias information in form of bias voltage, bias current, and bias power. All these bias point values are influenced by the circuit components, their inter-connectivity, and the external excitations. Bias point values indicate the state of a circuit at the start of an analysis and are independent of the analyses performed on the circuit. By looking at the bias point data on your schematic, you can quickly focus in on potential problem areas of your design.

## Bias Display in Design Entry HDL

The Bias Display feature of Design Entry HDL, allows you to display the bias information on the schematic itself. Using this feature, the bias point value can be displayed as attributes of the nodes, devices, or pins.

In Design Entry HDL, bias point information is displayed as:

- currents through pins
- voltages on nodes
- power dissipation of the device

Using the Bias Display feature of Design Entry HDL, you can:

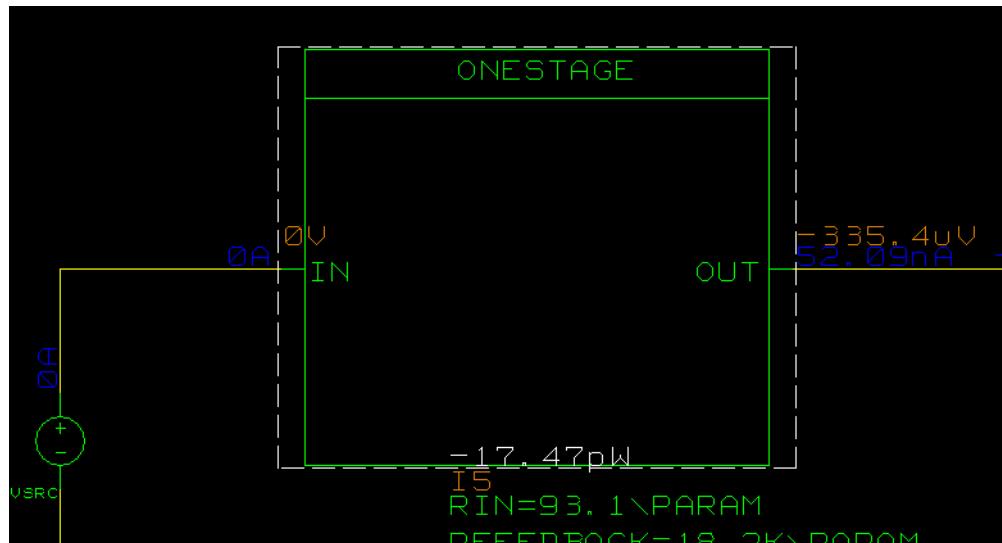
1. load bias information on the schematic.
2. set the color, size, and precision of the displayed bias values.
3. display or hide the bias voltage, bias current, and bias power on the schematic.

PSpice calculates and saves the bias point values for every simulation. After simulating with PSpice, you can display bias point information on your schematic page in Design Entry HDL. To display the bias point information on the schematic, you first need to load the bias point information, enable the display, and finally specify the bias values to be displayed.

## Allegro Design Entry HDL User Guide

Simulating using PSpice Simulator

Bias voltages are displayed next to their corresponding nets (nodes) and bias currents are displayed next to their corresponding device pins.



Bias point information is available for all analysis types except DC Sweep. Therefore, the bias point display feature is not applicable for DC Sweeps.

### ***Bias point data in multiple pages and reused hierarchical blocks***

Bias point data and the locations of the displayed values are saved as part of the schematic page. For a schematic containing multiple pages, the bias information is stored with each page. If you turn off bias point display, the display will be disabled for all the pages in the schematic.

If a page is reused (hierarchical subcircuits), the location of the bias point value will be stored with the page and will be the same for all occurrences of that page. Though the location of the bias point values will remain same across all instances of the reused page, the bias point values displayed on each instance will be different to accurately reflect the circuit values.

Distinct bias point information is saved for each simulation profile.

### ***Bias point data for multiple analyses***

If you have set up more than one analysis, Design Entry HDL displays bias information only for the last analysis. This means that if you perform a multi-run analysis like Parametric, Monte Carlo, Sensitivity/Worst-case or Temperature, you will see bias values for the last run only.

## Loading Bias Point Values

To display bias point values in Design Entry HDL, you first need to load the bias point values generated by PSpice and then display the values. To load the bias point values generated by PSpice, you can either annotate the bias point values or can use the Design Entry HDL settings to ensure that latest bias point values are automatically updated whenever the design is simulated.

You can load bias point values using any one of the following methods.

- Annotating bias point values
  - a. After you have simulated the design using PSpice, select *Bias Points* from the PSpice Simulator menu in Design Entry HDL.
  - b. Select *Annotate Bias Values* menu from the *Bias Points* submenu.
- Note: For a design with multiple pages, the bias point values are updated on all pages. Therefore, the process of annotating bias point values on PSpice may take some time.
- Automatically loading bias point values
  - a. Select *Bias Points* from the *PSpice Simulator* menu in Design Entry HDL.
  - b. Select *Preferences* from the *Bias Point* submenu.
  - c. In the Bias Point Preferences dialog box select the *Update Bias Point Information Automatically* check box.

Selecting this check box ensures that for all the future runs of PSpice, the bias point values are updated automatically every time you simulate the design. By default, this check box is not selected.

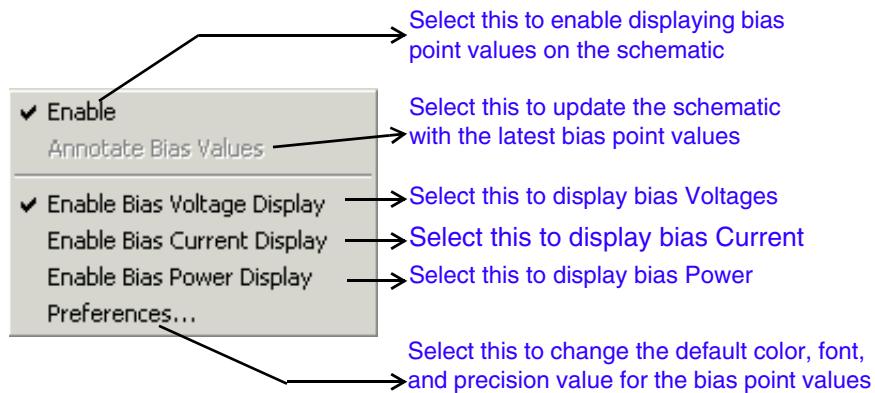
**Note:** For better performance, it is recommended that Auto Update feature should be turned off, especially in case of a design with multiple pages. This is because of the time required to open and update the bias point values on all the pages in the design.

## Displaying bias point values

To facilitate the display of bias points on the schematic, a new menu command, *Bias Points* has been added to the *PSpice* drop-down menu, see [Figure 17-4](#) on page 644.

### The Bias Points menu

To enable global viewing for the bias point values, select *Enable* from the *Bias Points* submenu. Unless the bias display feature is enabled, the bias point values will not be visible on the schematic even if the latest bias point values are loaded in Design Entry HDL.



**Figure 17-4 Bias Points submenu**

To display bias point values on the schematic, complete the following steps:

- From the *Bias Points* submenu, choose *Enable*.

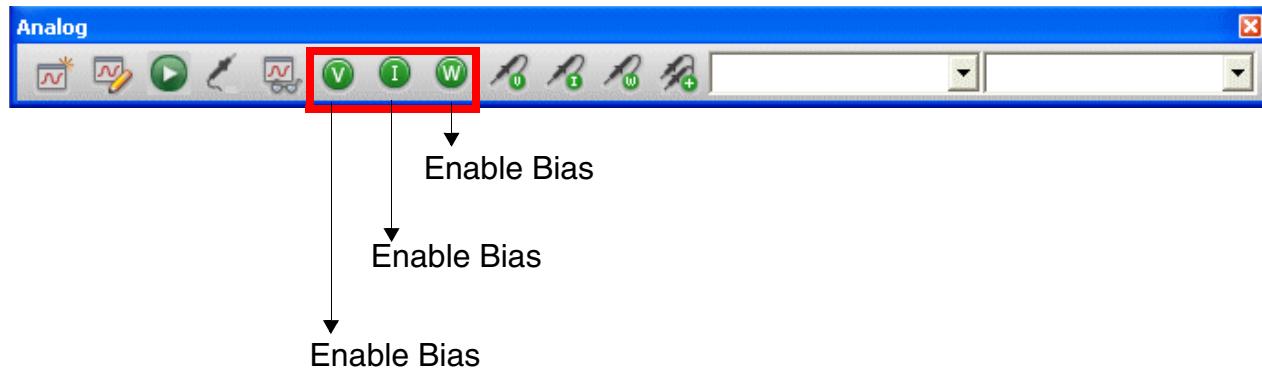
Menu options for displaying the bias Voltage, bias Current, and bias Power are enabled.

- To display bias voltages on the schematic, select *Enable Bias Voltage Display*.

To display bias currents on the schematic, select *Enable Bias Current Display*.

To display bias power values, select *Enable Bias Power Display*.

For the step b, where you specify the bias point values to be displayed, you can also use the buttons on the Analog toolbar. See [Figure 17-5 on page 645](#).



**Figure 17-5 The Analog**

**Note:** The bias point values displayed on the schematic are treated as instance properties. Therefore, as in case of any other instance property, you can use the Attributes dialog box to hide or display the bias point information on a particular component, pin, or net.

#### **Important**

Tough you can use the Attributes dialog box to edit the bias point values, it is strongly recommended that you do not edit these values. Once edited, these values are converted into hard properties and cannot be updated again.

#### ***Moving bias points values of an instance***

Like all other properties, you can change the location of the displayed bias point values. By default, the Bias points values are positioned near their corresponding wire or pin. You can select the bias point labels and relocate them to make the schematic page more legible.

To move a bias point label:

1. Click on the label you wish to move to select it.
2. While holding down the left mouse button, drag the label to the new location.

Once you move a label, the new location will be saved with the schematic page so the bias point will be displayed there again the next time you open that page.

## Bias Point Display Settings

The bias point display settings are stored in the `<projectname>.cpm` file. A section of the `.cpm` file listing the directives with the default values for the bias display settings is shown in the figure given below.

```
START_PSPICE
Auto_Update_Bias_Values '0'
Bias_Values_Enable '1'
Bias_Voltage_Values_Enable '1'
Bias_Current_Values_Enable '1'
Bias_Power_Values_Enable '1'
Bias_Voltage_Color 'RED'
Bias_Current_Color 'BLUE'
Bias_Power_Color 'WHITE'
Bias_Voltage_Font_Size '0.80'
Bias_Current_Font_Size '0.80'
Bias_Power_Font_Size '0.80'
Bias_Precision '4'

END_PSPICE
```

If required, you can modify the bias point display settings for a project by modifying the entries in the `projectname>.cpm` file. Alternatively, you can also modify the settings from the Design Entry HDL user interface.

### Modifying the display settings from Design Entry HDL

In Design Entry HDL, you can specify the number of digits of the bias point values to be displayed, and can also customize the color and the font of the displayed values. All these settings are defined using the Bias Point Preferences dialog box.

#### **To set the precision of the bias point values:**

1. From the Design Entry HDL *PSpice Simulator* menu, choose *Bias Points*.
2. From the submenu, select *Preferences*.

The Bias Point Preferences dialog box appears.

3. In the *Displayed Precision* text box, enter the number of digits you wish to display for the bias point values.

For example, if the value specified in the *Displayed Precision* text box is 5 and the bias point value calculated is 45.6789, the value displayed on the screen will be 45.679 (rounded off).

## Allegro Design Entry HDL User Guide

Simulating using PSpice Simulator

#### 4. Click **OK**.

This setting applies globally to all current and voltage bias points in your design. The default value displayed in the text box is 4. You can only specify the values between 2 and 5. This implies that a maximum of 5 digits can be displayed on the schematic. For example, even if the value specified in the *Displayed Precision* text box is 7, only a maximum of 5 digits will be displayed on the schematic.

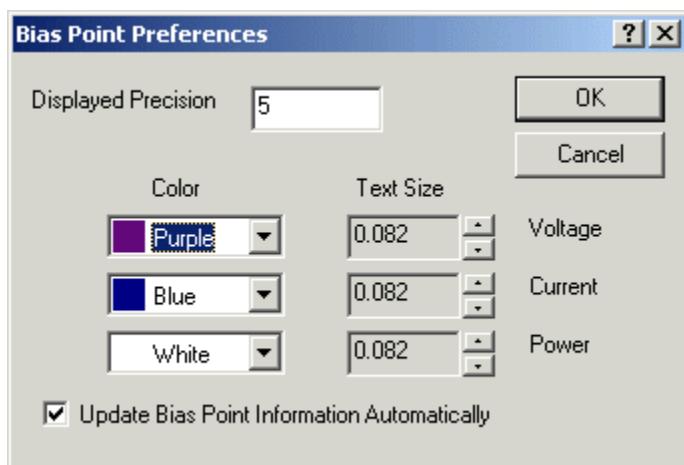


**Whenever you change the value in the Displayed Precision text box, you must resimulate the design for the change to take effect.**

#### To set the color of the bias point values:

1. From the Design Entry HDL *PSpice Simulator* menu, choose *Bias Points*.
2. From the submenu, select *Preferences*.  
The Bias Point Preferences dialog box appears.
3. Click in the Color box corresponding to either Current, Voltage, or Power and select the color you wish to display from the color palette.

For example, if you want the bias voltages to be displayed in Purple color, from the Voltage Color drop-down list box select Purple, as shown in the figure below.



#### 4. Click **OK**.

These settings apply globally to the bias points values displayed on all the pages in your design.



You cannot change the colors of the displayed bias point values in Occurrence Edit mode.

### **To set the fonts for the bias point values:**

1. From the Design Entry HDL *PSpice Simulator* menu, choose *Bias Points*.
2. From the submenu, select *Preferences*.  
The Bias Point Preferences dialog box appears.
3. Click in the Font box for either Current or Voltage and select the font type, style, and size you wish to display from the font dialog box.
4. Click *OK*.

These settings apply globally to the bias points values displayed on all the pages in your design. The default font for displayed the bias point values is Arial 7.



You cannot change the font of the displayed bias point values in Occurrence Edit mode.

### **Updating bias point values**

The bias point values displayed in Design Entry HDL reflect the data from the last simulation that was performed in PSpice. In some cases, these may not be the most recent values, depending on when they were last updated. If the bias point values are not updated, in the Design Entry HDL console window, you receive a message that the current bias values are stale.

You can update the bias point values displayed on the schematic using one of the following methods.

- Updating bias point data by resimulating
  - a. Simulate the design. From the Design Entry HDL *PSpice Simulator* menu, choose *Run*.
  - b. Update the schematic with latest bias values. For this use one of the methods described in the section [Loading Bias Point Values](#).

## Allegro Design Entry HDL User Guide

Simulating using PSpice Simulator

---

- Updating bias point data by changing profiles

- a. From the Analog toolbar in Design Entry HDL, click in the Profile Name text box.
- b. Select the profile you want to activate.

By opening a new simulation profile, the bias point values are to reflect the correct values that were last calculated for that profile.

**Note:** If the design has not been simulated using the selected profile, the stale bias point values will continue to display.

The bias point values are updated whenever you open a schematic page, when you resimulate, when you activate a different simulation profile, or when you change the display characteristics of the labels (such as color, font or precision).

To know more about using Design Entry HDL with PSpice, see *Allegro PSpice Simulator User's Guide (Design Entry HDL Version)*.

## Split Part

PSpice netlister combines the different sections of a split part and netlists them as a single instance based on the `SPLIT_INST` and `LOCATION` properties. The value of `SPLIT_INST` needs to be `TRUE` for a section to be recognized as belonging to a split part. The value of `LOCATION` identifies the sections belonging to one split part. Alternatively, the property `SPLIT_INST_NAME` can be used in place of the two properties `SPLIT_INST` and `LOCATION`. All sections of a split part needs to have the same value for `SPLIT_INST_NAME` if this property is used.

A requirement for netlister to be successful in recognizing split parts is that all sections should be in the same hierarchy level. However, the sections can be spread across different pages at the same level. In addition, the `PSPICETEMPLATE` property of each section must have the pins for all the sections of the split part.

The following conditions can result in either warning or an error:

- If `PSPICETEMPLATE` is missing from any of the sections, netlist is not created and an error message is displayed.
- If `PSPICETEMPLATE` has different values for the sections of a split part, netlist is not created and an error message is displayed.
- If one or more sections of the split part are missing, the pins of the missing part are treated as unconnected and a warning message is displayed.

- If one or more sections of a split part are instantiated more than one time, netlist is not created and an error message is displayed. This error is flagged only if the SEC property from symbol.css has the same value for one or more parts of different instances.

## Displaying PSpice Names

You can choose *PSpice Simulator–Display PSpice Names* to control the display of the \$PSPICE\_LOCATION value on the schematic. Selecting the Display PSpice Names option displays the \$PSPICE\_LOCATION value. If this option is not selected, the \$LOCATION value is displayed provided Attributes dialog box specifies the value to be shown.

---

## Glossary

---

**assertion level**

Part of a signal name, it describes the active state of the signal when asserted. By convention, a signal is active high for positive logic and active low for negative logic. An \* represents active low - for example, RESET\* is an active low signal. Two signals with the same name but different assertion levels are not the same signal.

**attribute**

Information that Design Entry HDL lets you attach to objects (components, wires/net, and pins) in a schematic. Attribute information is passed to other design programs for processing. An attribute consists of a name-value pair. Attributes are also called *properties*. See also *constraint*.

**attribute file**

A file that contains properties, their associated values, and some display information. Because different types of objects (components, wires, and pins) have different properties associated with them, they need to have different attribute files. A good way to add several properties to an object and ensure their names and values are correct is to use an attribute file as a template. See also *attribute*.

**automatic routing**

A Design Entry HDL function that automatically routes wires (*Wire – Route*) around objects in a schematic.

**backannotate**

The process of updating a Design Entry HDL schematic with information on new parts, connectivity, and properties from the Design Synchronization and Design Association tools. Usually, you backannotate the design after the first error-free run of Design Synchronization and then again after the design has been processed by a physical design system.

**block**

A hierarchical representation of a logical collection that can be reused in a schematic.

## Allegro Design Entry HDL User Guide

### Glossary

---

#### **body**

The symbolic representation of a component or design block. This is now called *symbol*.

#### **BODY drawing (symbol)**

The symbolic representation of a library component that you add to your design. This drawing defines the shape, pins, and general properties of the library component.

#### **bubble pins**

Low-asserted pins represented by circles on pins and indicated with a low-asserted signal name ( \* ).

#### **bus tap**

Tapping a subset of signals from a bus. See also *tap*.

#### **bus-through pins**

Special pins placed on a component to make it easier to wire a group of components together. Bus-through pins have the same name as the corresponding visible pin.

To find out if a component has bus-through pins, you can use *Display – Pins* to display an asterisk at every pin location.

#### **C-tap body**

The default bus tap provided in the Design Entry HDL standard library.

#### **category**

Refers to a group of components arranged hierarchically.

#### **cds.lib**

A file containing library definitions.

#### **cell**

Software representation of a component. Consider a cell to be a collection of views that describe an individual building block of a chip or a system.

#### **chips.prt**

A file containing physical information about a component.

#### **component**

Refers to the logical characteristics of a library part.

#### **Part Information Manager**

A dialog box in Design Entry HDL that lists active libraries and their contents, both drawings and components.

**component instance**

The placement of a component one or more times on a schematic.

**configuration**

A collection of views that control how a design is compiled and simulated.

**connectivity design data file**

A file that defines how all the components and nets connect together logically. This file is used by Design Entry HDL to generate the resulting VHDL or Verilog.

**constant signal**

A signal that has a numeric name. For example, a signal named 123.

See also, [non-constant signal](#)

**constraint**

A restriction on the physical implementation of a design object.

**cross probe**

The process of identifying corresponding parts, packages, and signals in the Design Entry HDL schematic and PCB Editor.

**design**

A schematic drawing created in Design Entry HDL.

**DRC**

Design Rule Checking.

**entity**

The view of a cell that contains the definition, including port (pin) definitions, for the current drawing (cell). Several checks are made to ensure that entity declarations, symbols, and schematics are in agreement.

**expand**

To build a complete design including all levels of the hierarchy based on views specified in the current expansion configuration.

**filter**

Screens file names, markers, and so on in the current directory and lists only those that match the filter. An asterisk ( \* ) or a blank field lists all the drawings or markers.

## Allegro Design Entry HDL User Guide

### Glossary

---

#### **flat design**

A design in which all parts of the drawing come from Design Entry HDL or user-defined libraries and are one-to-one logical representations of the physical parts. All of the interconnecting wiring within the design is entered pin-to-pin. Best suited for small designs that do not have sophisticated bus requirements or reuse portions of circuitry.

#### **grid**

Defines where wires and pins meet in the schematic. Design Entry HDL supports two grid types:

- Logic grid for schematic
- Symbol grid for symbol drawings

#### **hard property**

Properties that you add to the schematic to specify packaging assignments. Hard properties are included in the connectivity files and thus also in the Verilog/VHDL netlist. They differ from soft properties, which are essentially documentation properties on the schematic and are not included in the netlist.

#### **hierarchical design**

A design that is organized into modules to reuse many of the same circuit functions and isolate portions of the design for teamwork assignments. Using a block design lets you refer to a collection of logic without having to include the logic in the drawing. Hierarchical blocks simplify a drawing. This is also called *block design*.

#### **Hierarchy Editor**

A tool to create and edit configurations, which can be used in netlisting. You can also view the components of your design hierarchy using this tool.

#### **injected property**

A property that appears to the right of a PPT format definition or part row. Packager-XL passes these properties to Allegro in the physical netlist, - for example, company-specific part numbers, costs, or package types.

#### **in occurrence**

A drawing is *in occurrence* when

- Design Entry HDL understands the hierarchical location of the current drawing page being edited (*Tools – Expand Design*).
- Occurrence editing is enabled (*Tools – Occurrence Edit*) and occurrence properties are loaded into the drawing.

## Allegro Design Entry HDL User Guide

### Glossary

---

The title bar of the window shows the current page with the notation [in occurrence] - for example, *mycpu.SCH.1.1 [in occurrence]*.

#### **interface signal**

A signal property (\I) assigned to pins in block diagrams to indicate an interface signal from a higher level drawing. In a flat design, this is a signal in the schematic that corresponds to a pin in the symbol drawing.

#### **key property**

A property that appears to the left of a PPT format definition or part row. Packager-XL uses these properties to uniquely identify the physical part to use from the various table entries. For example, a resistor part table may use VALUE or TOLERANCE to select a specific physical part.

#### **library**

A collection of components from which you can select a component to place in a drawing.

#### **library properties**

Librarian-generated properties on symbols, chips, and in the Physical Part Table (PPT). Only the librarian can modify library properties.

#### **marker**

An error, warning, or information item that indicates a rule violation in your schematic. Markers are generated using the *Tools – Check* menu command, the Allegro Design Entry HDL Rules Checker utility, Design Synchronization, and Packager-XL.

#### **net**

A set of pins that are electrically connected.

#### **netlist**

An ASCII text file that describes the electrical connectivity (wires/nets and components) of a drawing.

#### **non-constant signal**

A signal that has an alphabetical or alphanumeric name. For example, ADDRESS, DATA1, 1CLOCK and so on.

See also [constant signal](#)

#### **NOT body**

Used to change the logic convention of a signal. If a signal is asserted low, it is considered to be a negative logic signal. If a signal is asserted high, it is considered to be a positive logic signal. The NOT body is used to change the logic convention of a

## Allegro Design Entry HDL User Guide

### Glossary

---

signal without introducing an actual logical inversion. This implies the state of the signal is not changed, it is just considered to be of the opposite logic convention.

#### **Orthogonal**

Bent to route around objects in a schematic. This is an alternative to direct (diagonal) placement.

#### **package**

(noun) In VHDL, a collection of types, constants, subprograms, and so on, usually intended to implement some particular service or to isolate a group of related items.

(verb) The process of translating a logical netlist into a physical netlist. Design Synchronization takes a logical representation of a schematic and applies the physical attributes necessary to allow physical layout.

#### **page**

Refers to a page in a design. If the amount of logic required to define a design does not fit on a single page, the drawing might extend to more than one page.

#### **part**

Refers to the physical symbol derived from the logical representation of a component or design block.

#### **physical**

Refers to the physical properties associated with a library component.

#### **Physical Part Table (PPT)**

Used to map logical parts in the schematic to physical parts for a layout.

#### **pin**

Conductors that protrude from packages. Pins allow the component to be connected logically to wires and other components in the logical design.

#### **placeholder property**

A temporary property assigned to the symbol drawing of a part. These properties serve as substitutes for part properties that will be assigned later in the schematic design. Placeholder properties let you predefined the location and text size of part properties through the part symbol drawing.

#### **placeholder value**

Substitutes a real property value. It is indicated with a ? value.

## Allegro Design Entry HDL User Guide

### Glossary

---

#### **PPT Browser**

Lets you select parts based on the properties defined in the PPT file, such as company part number or preferred status.

#### **primitive**

The symbol name in the *chips.prt* file.

#### **project**

The work area for a design, including all the views of the design, links to libraries, and setup information such as Physical Part Table, configuration, and expansion directives. Separate directories exist for each design project.

#### **property**

A logical characteristic of a design object. It is information that Design Entry HDL lets you attach to objects (components, nets, and pins) in a schematic. Property information is passed to other design programs for processing. A property consists of a name-value pair. Properties are also called *attributes*. See also *constraint*.

#### **ratsnest line**

In a design drawing, a line that shows a logical connection between two pins, connect lines, or vias. Elements connected by the same ratsnest line are part of the same net. The ratsnest shows the circuit logic and, for ECL circuits, the order in which pins are to be connected.

#### **reference designator**

The designator, or identification code, for a component.

#### **reference library**

A library containing cells that describe common components potentially used in many designs.

#### **root drawing**

The top-level drawing in your design. This is the drawing that Design Entry HDL opens by default when you start an editing session.

#### **route**

To autoroute a wire (*Wire – Route*). This is an alternative to manually drawing a wire (*Wire – Draw*) around objects on a schematic.

#### **rubberbanding**

A feature of interactive commands in which the lines that are attached to an element of the design drawing “stretch” as you move the element with the mouse.

### **rules-driven design**

User-defined design characteristics that can be specified by the schematic (as properties on components, pins, or nets) that are recognized by Allegro and determine processing results.

### **scalar signal**

A signal having a single bit.

### **SCH drawing**

A Design Entry HDL drawing that contains a *schematic*.

### **schematic**

The standard type of drawing created with Design Entry HDL to represent the logic of components or design blocks that make up a circuit. The symbolic drawing is generated in a physical layout tool. A schematic can contain library components and design blocks that represent other schematics.

### **schematic properties**

Modifiable properties that are defined when editing the schematic.

### **scope**

You can assign one of three different scopes to a signal:

**INTERFACE** Used on signals you want to access from a higher level of a hierarchy.  
Represented by \I.

**GLOBAL** Used on signals that you need to access on all levels of hierarchy.  
Represented by \G.

**LOCAL** Indicates that the signal is recognized only at its own level. No special characters are required because the local scope is the default.

### **script files**

Let you perform repetitive tasks in Design Entry HDL. You can build a script by editing a file and adding the commands in the sequence you want them to execute. You can use scripts to set up forms for routing, placing, and artwork or executing a series of check plots. Scripts can call other scripts.

#### **section**

Refers to a physical section on a logical component. Pin numbers are different in different sections of the component. You can section a component either before or after you package the design.

#### **signal**

Wire connections between components that support communication of dynamic data between components. Signals having the same name are interpreted as one signal; this is how signals are connected across multiple pages of a drawing.

#### **signal bits**

Signals can have a single bit (scalar signals) or multiple bits (vectored signals). The bit portion of the signal name is called the bit subscript and gives the bit information. Bit subscripts are enclosed in angle brackets, for example, <3..0>.

#### **SKILL**

A proprietary Cadence high-level interactive programming language based on the popular artificial intelligence language, LISP.

#### **soft property**

Properties that can change from one backannotation to the next. Soft properties are documentation properties on the schematic and are not included in the netlist. They differ from hard properties, which are included in the connectivity files and thus also in the Verilog/VHDL netlist.

#### **structured design**

Uses bus signals and memory and register depth. A structured design minimizes the number of interconnections and parts on the schematic.

#### **swap**

To exchange the locations of two logically identical pins within a function. This minimizes the average ratsnest crossings in a layout.

#### **SYM drawing**

A Design Entry HDL drawing that contains a *symbol*.

#### **symbol**

The symbolic representation of a library component that you add to your design. This drawing defines the shape, pins, and general properties of the library component.

#### **symbol properties**

Librarian-generated properties defined on a component through its symbol description and not by editing the schematic.

## Allegro Design Entry HDL User Guide

### Glossary

---

#### **system properties**

Non-modifiable schematic properties that Design Entry HDL adds.

#### **tap body**

Cadence-supplied taps found in the Design Entry HDL standard library: C-tap, tap.body, bustap.body, msbtap.body, and lsbtap.body.

#### **text**

Includes text can be signal names, properties, and notes.

#### **user-defined net**

A net with a signal name on it. Conversely, an unnamed net is one for which the user did not specify a net name and Design Entry HDL specifies the net name.

#### **vectorized signal**

A signal having multiple bits.

#### **version**

Different graphical but functionally equivalent representations of a component, all of which refer to the same logic drawing. If the version is not specified, Design Entry HDL assumes the version to be 1.

#### **VHDL**

VHSIC Hardware Description Language.

#### **view**

Designs are represented by these views in Design Entry HDL: schematic (or logic), symbol (or body), VHDL, and Verilog. Using *Tools – Generate View* in Design Entry HDL, you can generate one view of a design from another.

#### **visibility**

Refers to the amount of property or pin information displayed on a schematic.

#### **wire**

An electrical connection. A single wire can be an entire net, or, where there are many connections, a wire can be a segment of a net. This is also called *signal*.

#### **wire orientation**

Bent to route around objects in a schematic versus direct (diagonal).

---

## Working with Groups

---

A group is a collection of schematic objects. These objects can be symbols, properties, notes, wire segments, and dots. Groups can be used to group objects on the same schematic page. Pins cannot be added to a group.

For each defined group, Design Entry HDL assigns a letter of the alphabet as the name of the group. The current group name is displayed in brackets in the *Group* menu. Group contents are listed in the *Group Contents* dialog box that appears when you choose *Group – Show Contents*.

Design Entry HDL names each group you define with consecutive letters: a, b, c, and so on. You can have up to 26 groups at one time in each drawing. If you define more than 26 groups, the group name resets back to A, and the newly defined group A replaces the previously defined one.

**Note:** When defining a group using the *Group By Rectangle* or the *Group By Polygon* menu options to include symbols, do not specify the points of the group on the symbol.

### Creating a Group by Rectangle

When you create a group, you can either use Design Entry HDL's group identifier or assign the group identifier manually. If you assign the group identifier, you must set the current group before proceeding with these steps.

1. Choose *Group – Create – By Rectangle*.
2. Click in the upper left corner, and then in the lower right corner of the rectangle you want to use to define the group.
3. Move the cursor diagonally from where you clicked first.  
A rectangle encloses the objects to be included in the group.
4. Click again.
5. Objects in the group are highlighted.

## Creating a Group by Polygon

When you create a group, you can either use Design Entry HDL's group identifier or assign a group identifier. If you assign the group identifier, you must set the current group before proceeding with these steps.

1. Choose *Group – Create – By Polygon*.
2. Click near an object you want to group.

The point you click on will be the starting point of the polygon you will draw to define the group. The objects you want to group should lie inside the borders of the polygon.

3. Continue clicking to define a polygon that encloses objects to be included in the group.
4. Right-click and choose *Close Polygon* from the pop-up menu to close the polygon.

Objects in the group are highlighted.

## Creating a Group by Specifying an Expression

When you create a group, you can either use Design Entry HDL's group identifier or assign the group identifier yourself. If you assign the group identifier, you must set the current group before proceeding with these steps.

1. Choose *Group – Create – By Expression*.

Design Entry HDL places all occurrences of objects matching the pattern into a group.

2. Enter a pattern string.

The pattern is used to match component names, notes, property names, property values, or signal names. Properties can also be searched for by specifying both the name and the value separated by an equal sign.

Wildcards are allowed in the pattern. An asterisk matches any number of characters and a question mark matches any single character. The pattern is not case sensitive.

For example, let us suppose that a schematic includes the components LS00, LS03, LS04, and LS06. To include these components in a group, choose *Group – Create – By Expression*. In the *Pattern* dialog box, enter `ls0*`. This will include the above components in a group.

3. Choose *Group – Create – Next* to traverse to each object found in the search.

## Creating a Group by Including Objects

1. Choose *Group – Create – Include*.
2. Select one object after another to include them in the group.

## Grouping the Entire Schematic

1. Choose either *Group – Create – By Rectangle* or *Group – Create – By Polygon*.
2. Right-click and choose *All* from the pop-up menu.

## Including Additional Objects in a Group

1. Choose *Group – Show Contents*.
2. In the *Group Contents* dialog box, select a group A through Z.
3. Click *Show* under *Highlight*.  
The contents of the specified group are highlighted.
4. Choose *Group – Create – Include*.
5. Select an object to include it in the active group.  
The active group is shown in the *Group* menu options and on the *Group* toolbar.
6. In the *Group Contents* dialog box, reselect the group to confirm that the object was included.  
The object you included should appear highlighted.

## Excluding Objects from a Group

1. Choose *Group – Show Contents*.
2. In the *Group Contents* dialog box, select a group A through Z.
3. Click *Show* under *Highlight*.  
The contents of the specified group are highlighted.
4. Choose *Group – Create – Exclude*.
5. Select an object to exclude from the specified group.

6. In the *Group Contents* dialog box, reselect the group *A* through *Z* to confirm that the object was excluded.

The object you excluded should no longer appear highlighted.

## Setting the Current Group

1. Choose *Group – Set Current Group*.
2. Specify a group name.
3. Click *OK*.

The group name you specify appears in the square brackets next to several group menu commands. The group menu is also displayed on the *Group* toolbar.

## Viewing Group Contents

1. Choose *Group – Show Contents*.
2. In the *Group Contents* dialog box, select a group *A* through *Z*.
3. Click *Show* under *Highlight*.

The contents of the specified group are highlighted. Click *Clear* to turn off highlighting for the group.

## Moving a Group

1. Set the current group.
2. Choose *Group – Move [x]*.
3. Select the group to move.

By default, attachments to entries in the group do not move. Choose *Change Attachment*.

4. Right-click and choose *Change Attachment* from the pop-up menu to change attachments to either move with grouped objects or not at all.
5. Click at the new location.

## Rotating a Group

1. Set the current group.
2. Choose *Edit – Rotate*.
3. Press the middle mouse button on a three-button mouse or *Ctrl+left* mouse button on a two-button mouse and click on one of objects in the group to rotate the group.

## Rotating a Group of Properties

You can rotate a group that consists only of properties. The properties are rotated in place and not as a whole group.

1. Set the current group.
2. Choose *Edit – Rotate*.
3. Press the middle mouse button on a three-button mouse or *Ctrl+left* mouse button on a two-button mouse and click on one of properties in the group to rotate the properties in the group.

## Spinning a Group

1. Set the current group.
2. Choose *Edit – Spin*.
3. Press the middle mouse button on a three-button mouse or *Ctrl+left* mouse button on a two-button mouse and click on one of objects in the group to spin the group.

## Mirroring a Group

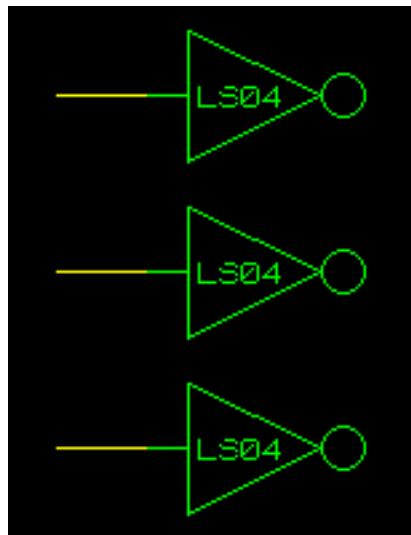
1. Set the current group.
2. Choose *Edit – Mirror*.
3. Press the middle mouse button on a three-button mouse or *Ctrl+left* mouse button on a two-button mouse and click on one of objects in the group.  
The group is attached to the cursor for you to place on the drawing.
4. Click in the same drawing or in another window to place the mirrored group.

## Allegro Design Entry HDL User Guide

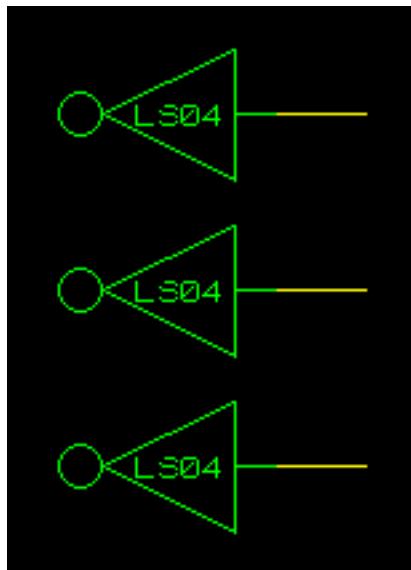
### Working with Groups

---

If the components in the group are in a horizontal position, the group is mirrored horizontally. For example, the three `ls04` components in a group are in a horizontal position.



If you mirror this group, it will be mirrored horizontally, as below:



If the components in the group are in a vertical position, you can mirror the group vertically by doing the following:

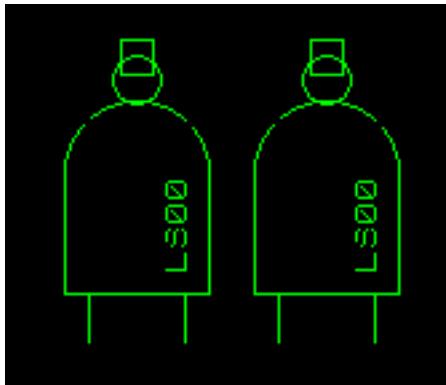
1. Set the current group.

## Allegro Design Entry HDL User Guide

### Working with Groups

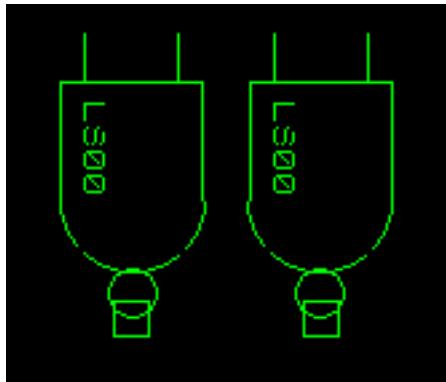
---

For example, the two 1s00 components in a group are in a vertical position.

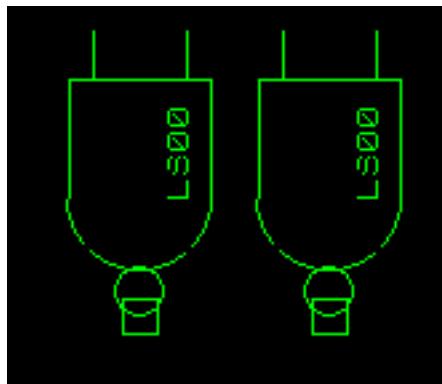


2. Rotate the group twice (by 180 degrees).

For more information, see [Rotating a Group](#) on page 665.



3. Choose *Edit – Mirror*.
4. Press the middle mouse button on a three-button mouse or *Ctrl+left* mouse button on a two-button mouse and click on one of objects in the group.



## Copying a Group

1. Set the current group.
2. Choose *Group – Copy [x]*.

The group is attached to the cursor for you to place on the drawing. To copy an object with its properties, right-click and choose *All* from the pop-up menu.

3. Click in the same drawing or in another window to place the copy.
4. If you're done placing copies but wish to remain in *Group – Copy* mode, choose *Terminate Selection* from the pop-up menu. To exit *Group – Copy* entirely, choose *Done* from the pop-up menu.

### ***To copy a group and its properties:***

1. Set the current group.
2. Choose *Group – Copy All [x]*.

The group is attached to the cursor for you to place on the drawing.

3. Click in the same drawing or in another window to place the copy.
4. When you have finished placing copies but wish to remain in *Group – Copy All* mode, choose *Terminate Selection* from the pop-up menu. To exit *Group – Copy All* entirely, choose *Done* from the pop-up menu.

**Note:** You can also copy properties in a group when you choose *Group – Copy* or *Group – Array*, right-click, and choose *All* from the pop-up menu.

### ***To make multiple copies of a group:***

1. Set the current group.
2. Choose *Group – Array [x]*.

The group is attached to the cursor for you to place on the drawing. To copy an object with its properties, right-click and choose *All* from the pop-up menu.

3. In the *Array Size* dialog box, specify the number of copies to make, and click *OK*.
4. Click in the same drawing or in another window to place the copies.

5. If you're done placing copies but wish to remain in *Group – Array* mode, choose *Terminate Selection* from the pop-up menu. To exit *Group – Array* entirely, choose *Done* from the pop-up menu.

## Deleting a Group

1. Set the current group.
2. Choose *Group – Delete [x]*.

Design Entry HDL deletes all objects in the specified group.

## Specifying Color for a Group

1. Set the current group.
2. Choose *Group – Color [x]*.

The *Color* toolbar is displayed.

3. Select a color from the toolbar.

Design Entry HDL colors all objects in the group according to your selection.

## Highlighting a Group on the Schematic

1. Set the current group.
2. Choose *Group – Highlight [x]*.

Design Entry HDL highlights all objects in the specified group.

## Replacing Components in a Group

1. Set the current group.
  2. Choose *Group – Components – Replace*.
- The *Replace Component* dialog box appears.
3. Select the component that should replace all components in the current group.

If you want to replace the components in the group with a component along with its physical properties, do the following:

- a. Click *Physical*.

The *Physical Part Filter* dialog box appears.

- b. Select the appropriate row of physical properties from the *Physical Part Filter* dialog box and click *Close*.

All the components in the current group are replaced with version 1 of the component that you selected in the *Replace Component* dialog box.

## Replacing Component Symbols in a Group (Versioning)

1. Set the current group.
2. Choose *Group – Components – Version*.

The version number of all components in the group will be incremented by 1. If the new version does not exist, the original will be replaced. If a component has two versions and the second is being used, this will change it to the first version.

## Breaking Up Components in a Group

1. Set the current group.
2. Choose *Group – Components – Smash*.

Design Entry HDL breaks all components in the group into individual elements such as lines, arcs, and dots.

## Adding Properties to a Group

To add user properties to a groups of components:

1. Set the current group.
2. Choose *Group – Add Property*.
3. In the Add Property - Group <group\_name> window:
  - a. Select the name of the group from the *Group Name* list box.
  - b. Specify the name of the property to be added to the group in the *Property Name* field.

- c. Specify the value of the property to be assigned to the selected group in the *Property Value* field.

## Specifying Property Display for a Group

1. Set the current group.
2. Choose *Group – Property Display – Name/Value/Both/Invisible*, depending on whether you want to:
  - display only the property Name
  - display only the property Value
  - display Both the property name and value
  - make properties Invisible

## Specifying Property Justification for a Group

1. Set the current group.
2. Choose *Group – Property Justification – Left Justified/Center Justified/Right Justified*, depending on whether you want to adjust the horizontal spacing so that the visible property values (text) of all the objects in the group are aligned towards:
  - left on the drawing
  - center on the drawing
  - right on the drawing

## Changing the Text Size in a Group

To change the size of the text (property name, property value, signal name, note, URL) of the objects in a group, use the following console command:

```
textsize <size in inches> <group name>
```

**Note:** You can specify a text size that has up to three decimal places. The minimum text size that you can specify is 0.008 inches and the maximum is 1.740 inches. The text size you specify should be a multiple of 0.002 inches.

For example, to change the size of all the text in a schematic page to 0.96 inches, create a group, say *A*, that covers all the objects in the schematic page (see [Grouping the Entire Schematic](#) on page 663) and enter the following command in the console window:

```
textsize 0.96 A
```

## Modifying Components with the Same Part Name in a Group

If a group contains components with the same part name, you can modify the properties of all the components using *Group – Components – Modify*.

To modify the components with the same part name,

1. Set the current group.
2. Choose *Group – Components – Modify*.

The *Physical Part Filter* appears.

3. Select the new row of properties and click *Close*.

Design Entry HDL replaces the existing components with the new selection along with the key properties.

# Index

---

## Symbols

[] in syntax [22](#)  
{} in syntax [22](#)  
| in syntax [22](#)

## A

abstract data types [241](#)  
add a library [258](#)  
add a pin [264](#)  
add library to search stack [259](#)  
Advanced Analysis [631](#)  
    Monte Carlo [631](#)  
    Optimizer [631](#)  
    Sensitivity [631](#)  
    Smoke [631](#)  
alias [209](#)  
alias symbols, rules [211](#)  
ALLOW\_POWER\_PINS [355, 369](#)  
ALLOW\_PROPERTY\_LOCKING  
    directive [305](#)  
arc command [627](#)  
ascend the hierarchy [127](#)  
assertion level  
    definition [651](#)  
    specifying [209](#)  
assign power pins [355](#)  
attachments [112](#)  
    verifying [625](#)  
attribute file [651](#)  
    creating [311](#)  
Attributes [305, 311](#)  
attributes [651](#)  
    OPTION\_SET\_ATTRIBUTES [191](#)  
    PROPERTY\_ATTRIBUTES [191](#)  
    PROPERTY\_SET\_ATTRIBUTES [192](#)  
Attributes drop-down list [307](#)  
    adding properties [307](#)  
Attributes form [151](#)  
automatic page borders [90](#)

## B

back annotate  
    design [466](#)  
backannotate [651](#)  
baselining a design [506](#)  
    metadata creation [507](#)  
    metadata folder [506](#)  
batch mode in Windows plotting [576](#)  
Bias display [641](#)  
bias display [641](#)  
    bias point values [643](#)  
    color [647](#)  
    cpm directives [646](#)  
    menu [644](#)  
    precision [646](#)  
    settings [646](#)  
bias point value [641](#)  
bias point values  
    display properties [647, 648](#)  
    displaying [643](#)  
    loading [643](#)  
    relocating [645](#)  
    setting precision [646](#)  
    updating [648](#)  
BIDIRECTIONAL property [267](#)  
blank page [485](#)  
block  
    definition [651](#)  
body  
    drawings  
        in hierarchical designs [623](#)  
    origin [625](#)  
    PIN NAMES [626](#)  
bounding box [150](#)  
braces in syntax [22](#)  
brackets in syntax [22](#)  
browse a library [258](#)  
bubble pins [652](#)  
BUBBLE property [627](#)  
bubbled pins  
    defining [627](#)  
bus  
    notation [617](#)  
bus-through pins [652](#)

## C

capturing an image [110](#)  
case sensitivity [349](#)  
`cds.lib` [65](#)  
`cds.lib` file [652](#)  
`cdsprop.paf` file [349](#)  
    description [76](#)  
    example [77](#)  
`cdsprop.tmf` file [376](#)  
cell  
    deleting [287](#)  
`chips.prt` file [652](#)  
    creating [268](#)  
choosing base signal, rules [212](#)  
circle command  
    defining bubbled pins [627](#)  
closing a window [128](#)  
color [96](#)  
    specify for an object [96](#)  
component  
    instance [653](#)  
    rotating [150](#)  
component information [112](#)  
components  
    modifying [276, 672](#)  
    modifying (in a group) [276](#)  
    replacing [276, 277](#)  
    sectioning [280](#)  
    unsectioning [286](#)  
compressing a page [482](#)  
concatenating signals  
    example [222](#)  
    use [221](#)  
connectivity design data file [653](#)  
console commands  
    about [40](#)  
console window, display [116](#)  
constant signal [653](#)  
constraint [653](#)  
controlling project settings [82](#)  
conventions  
    user-defined arguments [22](#)  
    user-entered text [22](#)  
copy and paste feature [99](#)  
    commands [104](#)  
    copying a group [104](#)  
    copying bitmap [105](#)  
    copying blocks [102](#)  
    copying components [100](#)

copying hyperlinks [104](#)  
copying page border [102](#)  
copying special characters [104](#)  
copying tables [104](#)  
copying text [103](#)  
copying wires [100](#)  
part of a schematic [99](#)  
Paste Special [100](#)  
`cpm` [60, 61, 66, 67](#)  
    bias display directives [646](#)  
    locking directives [82](#)  
`cpm` files [60, 61, 66, 67](#)  
create a symbol [263](#)  
creating a table of contents [511](#)  
creating Design Entry HDL parts [262](#)  
cross probe [653](#)  
    overview [464](#)  
cross-referencing the design [49](#)  
custom flow in Project Manager [67](#)  
custom text  
    definition [377](#)  
custom variables  
    case-sensitivity [384](#)  
    defining [382](#)  
    definition [378](#)  
    inbuilt [379](#)  
    types [378](#)

## D

default  
    body  
        drawing grid [625](#)  
        grid setting [625](#)  
default dpi [107](#)  
`DEFINE` symbol [374](#)  
defining  
    bubbled pins [627](#)  
    low-asserted pins [627](#)  
defining a default text editor [91](#)  
definitions  
    assertion level [651](#)  
    attribute file [651](#)  
    attributes [651](#)  
    automatic routing [651](#)  
    backannotate [651](#)  
    blank page [485](#)  
    block [651](#)  
    body [652](#)  
    constraint [653](#)

cross probe [653](#)  
entity [653](#)  
filter [653](#)  
flat design [654](#)  
grid [654](#)  
hard property [654](#)  
hierarchical design [654](#)  
injected property [654](#)  
interface signal [655](#)  
key property [655](#)  
library [655](#)  
library properties [655](#)  
netlist [655](#)  
non-constant signal [655](#)  
package [656](#)  
page gap [485](#)  
placeholder property [656](#)  
project [657](#)  
property [657](#)  
scalar signal [658](#)  
schematic [658](#)  
schematic properties [658](#)  
section [659](#)  
signal [659](#)  
soft property [659](#)  
structured design [659](#)  
symbol [652](#)  
symbol properties [659](#)  
system property [660](#)  
user-defined net [660](#)  
vectored signal [660](#)  
view [660](#)  
deleting a page [480](#)  
design  
    cross-referencing [49](#)  
Design Entry  
    font [579](#)  
Design Entry commands  
    set  
        default\_body\_grid [624](#)  
Design Entry HDL  
    about [23](#)  
    console command window [37](#)  
    getting started [27](#)  
    introduction [23](#)  
    menu bars [30](#)  
    overview [23](#)  
    panning [41](#)  
    starting [27, 172](#)  
    toolbars [32](#)  
    Unix [27](#)  
user interface [29](#)  
zooming [41](#)  
Design Entry HDL commands  
    about [40](#)  
    arc [627](#)  
    circle [627](#)  
    conventions [40](#)  
    dot [625](#)  
    move [625](#)  
    note [625](#)  
    pinnames [626](#)  
    show  
        vectors [617](#)  
    signature [625](#)  
    wire [625](#)  
    write [625](#)  
Design Entry HDL commands, enter [40](#)  
Design Entry HDL Options dialog box [148](#)  
    Design Navigation tab [475](#)  
Design Entry HDL paper sizes [582](#)  
Design Entry HDL variables [379, 380, 381](#)  
Design Entry HDL, running [27](#)  
design file structure [66](#)  
design name [64](#)  
design techniques  
    flat [613](#)  
    hierarchical [621](#)  
    overview [613](#)  
    structured [616](#)  
designs  
    flat design [654](#)  
    hierarchical design [654](#)  
    rules-driven design [658](#)  
    structured design [659](#)  
dialog boxes  
    Import Design [499](#)  
    Property Options [187](#)  
digital simulation  
    available simulators [524](#)  
    generate a netlist [525](#)  
    specify options for netlist [524](#)  
directives  
    ALLOW\_POWER\_PINS [369](#)  
    ALLOW\_PROPERTY\_LOCKING [305](#)  
    bias display [646](#)  
    RETAIN\_HARDLOCATION\_ON\_COPY  
        [100](#)  
    RETAIN\_PREVIOUS\_HILITE [117](#)  
    SYNC\_ON\_PAGE\_EDIT [295, 508](#)  
    SYNC\_ON\_STARTUP [295, 508](#)  
display console window [116](#)

display grid [116](#)  
display information on schematic [111](#)  
display status bar [116](#)  
display toolbars [115](#)  
dot command  
    in body drawings [625](#)  
dots per inch [107](#)  
drawing  
    BODY [623](#)  
    names  
        multiple page drawings [615](#)  
        pages [614](#)  
Drawings  
    adding PIN NAMES bodies [626](#)  
drawings [175, 287](#)  
    ascending [127](#)  
    deleting cells, views, and files [287](#)  
    descending [127](#)  
    LOGIC [623](#)  
    multiple page [615](#)  
    panning [125](#)  
    placing a border [175](#)

## E

edit text in console window [98](#)  
edit text in dialog box [98](#)  
edit text in dialog box, Windows [99](#)  
enabling metadata creation [508](#)  
entity [653](#)  
entity declaration  
    generating [265](#)  
entity, properties [265](#)  
errors in netlist for digital simulation [526](#)  
exiting Design Entry HDL [131](#)  
expand design  
    how to handle properties [76](#)

## F

files  
    cds.lib [652](#)  
        adding libraries [71](#)  
    cdsprop.paf [349](#)  
    cdsprop.tmf [376](#)  
    chips.prt [268, 652](#)  
    connectivity design data file [653](#)  
    created for new project [64](#)  
    deleting a file [287](#)

part table file(ptf) [270](#)  
ppt\_optionset.dat [187](#)  
script file [658](#)  
simNetlist.mkr [526](#)  
verilog.v [526](#)  
vhdl.vhd [526](#)  
xmodules.dat [470](#)  
filter [653](#)  
Filters [452](#)  
finding nets and cells [128, 423](#)  
flat design [654](#)  
flat designs [613](#)  
    creating [614](#)  
flattener  
    schematic [621](#)

## G

Genview [398](#)  
global find  
    limitations [424](#)  
global navigate  
    zooming to a region [151](#)  
global property display [317](#)  
global signals  
    shorting [213](#)  
    syntax [217](#)  
grid [654](#)  
    default settings for bodies [625](#)  
grid, display [116](#)  
group  
    creating [661, 662](#)  
    definition [661](#)  
    operations [661](#)  
guidelines for creating symbol [265](#)

## H

handle properties during expansion [76](#)  
hard property [654](#)  
hierarchical  
    designs [621](#)  
hierarchical design [654](#)  
    create [39](#)  
    module ordering [466](#)  
hierarchical design creation, steps [39](#)  
hierarchical designs [613](#)  
    creating [622](#)  
hierarchical plotting [599](#)

Hierarchy Editor [654](#)  
hierarchy viewer [466](#), [467](#)  
sheet names [472](#)  
highlighting [49](#)

## I

I interface signal property [626](#)  
image  
capturing [110](#)  
inserting [108](#)  
scaling [109](#)  
image, copying [105](#)  
Import Design dialog box [499](#)  
importing designs [497](#)  
block [500](#)  
blocks [501](#)  
multiple sheets [500](#)  
sheet [497](#)  
sheets [501](#)  
sheets containing blocks [501](#)  
in occurrence [654](#)  
inbuilt custom variables [379](#)  
injected property [654](#)  
INPUT\_LOAD property [267](#)  
inserting a page [476](#)  
inserting an image [108](#)  
instance [653](#)  
interface signal [655](#)  
interface signal property [624](#)  
interface signal property (I) [624](#), [626](#)  
introduction to Design Entry HDL [23](#)  
italics in syntax [22](#)

## K

key properties [304](#)  
key property [655](#)  
keywords [22](#)

## L

lib-cell-view [58](#)  
library  
add [258](#)  
add to search stack [259](#)  
browse [258](#)  
remove from search stack [259](#)

library clauses [268](#)  
library components  
deleting [287](#)  
library properties [655](#)  
LIBRARY property [268](#)  
limitation [201](#)  
limitations of global find [424](#)  
literal characters [22](#)  
location of netlist file [526](#)  
locking key properties [304](#)  
locking project file directives [82](#)  
log file [77](#)  
LOGIC drawings [623](#)  
logic type  
Verilog [231](#), [233](#)  
VHDL [234](#)  
low-asserted pins [627](#)

## M

marker [655](#)  
markers [118](#)  
maximum length  
property name [304](#)  
property value [304](#)  
merge symbol  
rules [227](#)  
types [224](#)  
use [224](#)  
MERGE\_NC\_PINS [355](#)  
MERGE\_POWER\_PINS [355](#)  
metadata folder [506](#)  
mode of a port [267](#)  
module ordering [466](#)  
hierarchy viewer [466](#)  
move  
grouped objects [664](#)  
move command [625](#)  
moving a page [482](#)  
moving a window [127](#)  
multiple  
page drawings  
flat designs [614](#)  
names [615](#)  
signals [615](#)

## N

names

drawing  
    multiple page drawings [615](#)  
    signal [615](#)  
navigating nets [130](#)  
NC [201](#)  
NC pin [201](#)  
NC\_PINS [355](#)  
net [655](#)  
netlist [655](#)  
    for analog and mixed signal simulation [529](#)  
    for digital simulation  
        generating [525](#)  
        simulators to use [524](#)  
        specify options [524](#)  
        view errors [526](#)  
    for PSpice simulation  
        generating [529](#)  
    for simulation [524](#)  
    for synthesis  
        generating [528](#)  
    PSpice [529](#)  
netlist files [526](#)  
netlist for digital simulation [525](#)  
netlist for hierarchical design [525, 526](#)  
netlist for PSpice simulation [529](#)  
netlist for synthesis [528](#)  
nets  
    navigating [130](#)  
    navigating in a design [425](#)  
    unnamed [220](#)  
non-constant signal [655](#)  
notation  
    bus [617](#)  
note [311](#)  
note command  
    labelling body drawings [625](#)

## O

occurrence property  
    add [203](#)  
Occurrence Property File (OPF) [656](#)  
operations on groups [661](#)  
option set [187](#)  
    copying [193](#)  
    removing [193](#)  
or-bars in syntax [22](#)  
origin  
    body [625](#)

OUTPUT\_LOAD property [267](#)  
OUTPUT\_TYPE property [267](#)

**P**

package [656](#)  
page [656](#)  
    renumbering [492](#)  
page border  
    adding [175](#)  
page borders  
    automatic [90](#)  
    types [175](#)  
page delete [480](#)  
page gap [485](#)  
page insert [476](#)  
page locking [47](#)  
page management [476](#)  
    deleting [480](#)  
        a set of pages [481](#)  
        current page [480](#)  
        page gaps [482](#)  
    GUI [476](#)  
inserting  
    a set of pages [477](#)  
    a single page [476](#)  
    beyond the end of the schematic [479](#)  
    page gaps [479](#)  
inserting pages [476](#)  
moving [482](#)  
    a set of pages outside the current range of pages [484](#)  
    non-contiguous pages to contiguous locations [484](#)  
    page before an existing page [483](#)  
removing page gaps [482](#)  
page move [482](#)  
page renumbering  
    about [492](#)  
    commands  
        page move [495](#)  
        using [494](#)  
pages  
    drawing [614](#)  
pan schematic [125](#)  
paper sizes  
    supported by Design Entry HDL [582](#)  
PARAM symbol [375](#)  
PARAMETER symbol [375](#)

Part [450](#)  
Part Information Manager [42](#), [178](#), [652](#)  
    adding a part [181](#)  
    modifying a part [182](#)  
    replacing a part [182](#)  
Part Manager  
    applying changes [456](#)  
    displaying canonical path [459](#)  
    highlighting part instances [459](#)  
    opening [447](#)  
    overview [447](#)  
    refreshing [458](#)  
    Resetting values of part instances [457](#)  
    updating multiple parts [457](#)  
    updating part instances [454](#)  
use model [447](#)  
user interface [448](#)  
    columns [449](#)  
    Filters [452](#)  
    left pane [448](#)  
    part status [450](#)  
    right pane [449](#)  
    Save Schematic [453](#)  
    status of logical parts [451](#)  
    summary [452](#)  
part status [450](#)  
part table file [187](#)  
    about [73](#)  
    example [270](#)  
part table file(ptf)  
    creating [270](#)  
parts  
    sizeable [617](#)  
    standard library [621](#)  
parts summary [452](#)  
Physical Part Table (PPT) [656](#)  
physical part table file [73](#)  
physical property options  
    configure [187](#)  
    defining [188](#)  
pin [656](#)  
pin name [264](#)  
PIN NAMES body [626](#)  
pin\_name property [264](#)  
pinnames command [626](#)  
pins  
    bubbled  
        defining [627](#)  
    changing states [279](#)  
    low-asserted [627](#)  
placeholder property [656](#)  
plot  
    font [570](#), [579](#)  
plot command [581](#)  
plotting  
    hierarchical plotting [599](#)  
    Windows plotting [567](#)  
port association [205](#)  
port mode [267](#)  
port names  
    adding from corresponding symbol [316](#)  
ports  
    custom [204](#)  
    rules [204](#)  
    Verilog logic type [203](#), [231](#), [233](#)  
    VHDL logic type [203](#), [234](#)  
power pin [355](#)  
power pins [355](#)  
POWER\_GROUP [355](#)  
    subtype [369](#)  
POWER\_PINS [355](#)  
ppt\_optionset.dat file [187](#)  
    sample [190](#)  
pre-select mode [89](#)  
primitive [657](#)  
project [657](#)  
project files [59](#)  
Project Manager  
    create project [62](#)  
    custom flow [67](#)  
    files [64](#)  
Project Manager, HTML browser [62](#)  
    Unix [62](#)  
project settings  
    controlling [82](#)  
Project Setup  
    add library [71](#)  
    change root design [69](#)  
    create new root design [69](#), [73](#)  
    edit cds.lib file [70](#)  
    select library [72](#)  
    specify expansion type [78](#)  
    specify log file [77](#)  
Project Setup, tasks [68](#)  
properties  
    BUBBLE [627](#)  
    case sensitive [349](#)  
    hard [654](#)  
    injected [654](#)  
    key property [655](#)  
    library [655](#)  
    placeholder property [656](#)

schematic property [658](#)  
signal  
    interface (I) [626](#)  
SIZE [616](#)  
soft property [659](#)  
symbol property [659](#)  
system property [660](#)  
    TIMES [619](#)

properties for Design Entry HDL [304](#)  
properties for entity [265](#)  
properties window [151](#)  
properties, definition [657](#)  
property [657](#)  
    BIDIRECTIONAL [267](#)  
    definition [303](#)  
    INPUT\_LOAD [267](#)  
    LIBRARY [268](#)  
    locking [304](#)  
    OUTPUT\_LOAD [267](#)  
    OUTPUT\_TYPE [267](#)  
    PATH [306](#)  
    pin\_name [264](#)  
    reattaching [316](#)  
    swap [316](#)  
    types [303](#)  
        vhdl\_mode [264, 266, 267](#)  
        vlog\_mode [264, 266, 267](#)  
property name, maximum length [304](#)  
property options  
    using [187](#)  
Property Options dialog box [187](#)  
property value, maximum length [304](#)  
property, add [201](#)  
PSpice simulation  
    generate a netlist [529](#)  
PSpice Simulator [631](#)  
    Advanced Analysis [631](#)  
    bias display [641](#)  
    directory structure [634](#)  
ptf  
    example [270](#)  
ptf file [73](#)  
    adding to a project [73](#)

## Q

QuickPick browser [119, 123](#)  
    filling information [124](#)  
QuickPick Browser window  
    adding a block [123](#)

adding a part [123](#)  
QuickPick toolbar [119, 122](#)  
    adding a block [123](#)  
    adding a library cell [122](#)  
    adding a part [123](#)

R

ratsnest line [657](#)  
reattach a property [316](#)  
redo [92](#)  
reference designator [657](#)  
reference library [657](#)  
remove library from search stack [259](#)  
removing page gaps [482](#)  
renumbering pages [492](#)  
replicate a signal [223](#)  
resizing a window [127](#)  
restriction on PATH property [306](#)  
root drawing [657](#)  
rotate a component [280](#)  
rotating a component [150](#)  
routing [657](#)  
rubberbanding [657](#)  
rules  
    design name [64](#)  
rules for signal naming [196](#)  
rules for specifying range of an object [238](#)  
rules-driven design [658](#)

## S

saving the design [243](#)  
scalar signal [658](#)  
scale a drawing [126](#)  
scaling an image [109](#)  
SCH drawing [658](#)  
schematic [658](#)  
    create [38](#)  
schematic creation, steps [38](#)  
schematic flattener [621](#)  
schematic properties [658](#)  
script files [658](#)  
scripts  
    how to run [462](#)  
    running [462](#)  
search stack  
    add library [259](#)  
    remove library [259](#)

section [659](#)  
sectioning  
  multiple components [281](#)  
set command  
  default\_body\_grid [624](#)  
sheet names  
  Design Navigation tab [475](#)  
  overview [472](#)  
sheet names in Design Entry HDL [472](#)  
show command  
  vectors [617](#)  
showing unconnected pins [152](#)  
signal [659](#)  
  names  
    multiple-page drawings [615](#)  
  properties  
    interface (I) [626](#)  
  syntax [617](#)  
signal bits [659](#)  
signal names  
  step size [199](#)  
signal naming  
  allowed characters [198](#)  
  characters not to be used [199](#)  
  limitations [201](#)  
signal naming limitation [201](#)  
signal naming, characters with special  
  meaning [196](#)  
signal with multiple drivers [239](#)  
signals  
  assertion level [209](#)  
  concatenate [221](#)  
  concatenate graphically [222](#)  
  concatenate textually [221](#)  
  global [213](#)  
  initial value [206](#)  
  naming [196](#)  
  properties  
    interface (I) [624](#)  
  unnamed [220](#)  
  Verilog logic type [231, 233](#)  
  VHDL logic type [234](#)  
signame command  
  in body drawings [625](#)  
simNetlist.mkr file [526](#)  
simulators for digital simulation [524](#)  
Single File Netlist [525, 526](#)  
single-page flat designs  
  creating [614](#)  
site [60, 61, 66](#)  
site.cpm [60, 61, 66, 67](#)

SIZE property [616](#)  
sizeable  
  parts [617](#)  
SKILL [659](#)  
SLASH [208](#)  
slice [228](#)  
soft property [659](#)  
specifying range  
  ascending [238](#)  
  descending [238](#)  
  rules [238](#)  
  unconstrained [239](#)  
  use [238](#)  
spin a component [280](#)  
Standard library  
  overview [257](#)  
standard library [621](#)  
standard port symbols [203](#)  
starting  
  Design Entry HDL [27, 172](#)  
status bar, display [116](#)  
status of logical parts [451](#)  
step size [199](#)  
strict entity check [527](#)  
strokes  
  guidelines [132](#)  
strokes, running commands [131](#)  
structured design [659](#)  
structured designs  
  about [613, 616](#)  
  creating [616](#)  
summary (Total/Defined/Undefined) [452](#)  
support for keyboard operations [148](#)  
swapping notes and properties [316](#)  
symbol [652, 659](#)  
  create [263](#)  
symbol properties [659](#)  
symbols  
  creating [265](#)  
synonym and alias, difference [210](#)  
syntax  
  signal name [617](#)  
synthesis  
  generate a netlist [528](#)  
  specify options for VHDL netlist [526](#)  
system properties [660](#)

## T

table of contents [511](#)

tap  
bus tap [652](#)  
C-tap body [652](#)  
tap body [660](#)

temp directory [75](#)  
text

adding [314](#)  
adding port names from corresponding symbol [316](#)  
changing the editor [316](#)  
editing in dialog boxes and the console window [98](#)  
modifying [314](#)  
resizing [315](#)

text editor  
changing [316](#)  
Unix [76](#)  
Windows [76](#)

text macro  
define in a file [376](#)  
defining [374, 375](#)  
definition [373](#)  
use [373](#)

TIMES property [619](#)

TOC [511](#)

toolbar  
QuickPick [119](#)

toolbars  
QuickPick [122](#)

toolbars, display [115](#)

tools setup [75](#)

transcribe [621](#)

type conversion functions  
restriction in Design Entry HDL [241](#)  
restriction in Verilog [240](#)  
use [240](#)

## U

unbubbled pins  
defining [627](#)  
unconnected pins [152](#)  
undo [92](#)  
unnamed net, specify size [208](#)  
unnamed signals or nets [220](#)  
unsectioning components [286](#)  
URL [311](#)  
usability changes [145](#)  
use  
clause [268](#)

property [268](#)  
user-defined net [660](#)

## V

variables  
CRef variables [381](#)  
drawing specific [379](#)  
global [380](#)  
parent variables [381](#)  
vectored signal [660](#)  
verbose output [527](#)  
verifying attachments [625](#)  
Verilog net type [231, 233](#)  
verilog.v file [526](#)  
version [660](#)  
vertical bars in syntax [22](#)  
VHDL netlist  
for synthesis  
specify options [526](#)  
VHDL scalar type [234, 236](#)  
VHDL vector type [234, 236](#)  
VHDL\_IN\_CONVERT [240](#)  
VHDL\_INIT [206](#)  
vhdl\_mode property [264, 266, 267](#)  
VHDL\_OUT\_CONVERT [240](#)  
vhdl.vhd file [526](#)  
VHSIC [660](#)  
view [660](#)  
deleting [287](#)  
view names  
changing [81](#)  
view pages of a drawing [115](#)  
viewing a design [505](#)  
vlog\_mode property [264, 266, 267](#)

## W

windows  
closing [128](#)  
windows mode [145](#)  
Windows Plotting  
font [570, 579](#)  
Windows plotting [567](#)  
batch mode [576](#)  
previewing [568, 571](#)  
setup [568](#)  
wire [660](#)  
wire command

marking clock signals [625](#)  
wiring ports together [203](#)  
worklib [65](#)

## X

xmodules.dat file [470](#)

## Z

zoom [41, 126](#)  
zoom by points [151](#)