Contents

Preface .................................................................................................................. 19
About This User Guide ......................................................................................... 19
Finding Information in This User Guide .............................................................. 19
Related Documentation ....................................................................................... 20
Design Entry HDL ................................................................................................. 20
Front-to-Back Flow ............................................................................................... 21
Related Tools and Flows ....................................................................................... 21
Typographic and Syntax Conventions ................................................................. 22

1
About Design Entry HDL ..................................................................................... 23
Design Entry HDL Features .................................................................................. 23
Design Entry HDL in the Design Flow .................................................................. 25

2
Getting Started ................................................................................................... 27
Starting Design Entry HDL .................................................................................... 27
Design Entry HDL User Interface ......................................................................... 29
Menu Bar .............................................................................................................. 30
Toolbars ............................................................................................................... 31
Status Bar ............................................................................................................ 33
Console Command Window .................................................................................. 33
Context-Sensitive Menus ...................................................................................... 34
Design Entry HDL Tasks ....................................................................................... 35
Creating a Schematic ............................................................................................ 35
Creating a Hierarchical Design .......................................................................... 37
Design Entry HDL Basics .................................................................................... 39
Where can I enter commands? .......................................................................... 39
Where are setup options? .................................................................................... 39
How do I pan drawings? ....................................................................................... 39
How do I zoom in and out of a drawing? ........................................... 40
How do I customize Design Entry HDL? ........................................... 40
What commands can I use to edit schematic text? ............................ 40
Are there menu shortcuts? ................................................................. 41
How do I browse drawings and components? .................................... 41
How do I add libraries? ................................................................. 41
How do I add notes? ................................................................. 42
How do I add parts? ................................................................. 42
How do I connect parts? ................................................................. 42
How do I name signals? ................................................................. 42
How do I add properties? ................................................................. 42
How do I add ports? ................................................................. 42
How do I check my drawing for errors? ............................................ 42
How do I save a design? ................................................................. 43
What is Page Locking? ................................................................. 43
How do I generate a Verilog or VHDL netlist from a schematic? .......... 43
How do I add additional pages? ............................................................. 43
How do I go to a specific page in a design? ............................................. 43
How do I plot a design? ................................................................. 44
What are groups? ................................................................. 44
What is different about working with groups? ...................................... 44
How do I locate parts and wires in a design? ....................................... 45
How do I generate a symbol view from a schematic? .............................. 45
How do I package my design? ................................................................. 45
How do I backannotate a design? .......................................................... 45
How do I highlight objects in a design? ................................................. 45
How do I cross-reference a design? ...................................................... 46
How do I archive a design? ................................................................. 46
How do I view the bias point values in Design Entry HDL? .................... 46

3
Project Creation and Setup ................................................................. 49
Introduction to Project Manager .......................................................... 49
Project Structure ............................................................................. 50
Project Files ..................................................................................... 51
Basic Editing Tasks
  Undoing an Operation
  Moving Objects
  Copying Objects
  Deleting Objects
  Changing the Color of Objects
  Drawing an Arc
  Drawing a Circle
  Splitting Overlaid Objects
  Displaying the Console Window
  Editing Text in Dialog Boxes and the Console Window
  Copying Parts of a Schematic Across Pages, Designs, and Projects
Displaying Information
  Displaying Schematic Information
  Displaying Toolbars and Other Parts of the Design Entry HDL Window
  Highlighting Objects
  Turning Off Highlighting
  Opening the Markers Control Window
  Displaying the Markers Toolbar
  Displaying the Error Status Bar
Using the QuickPick Browser
  Overview
  QuickPick Browser - Interface Overview
  Setting up QuickPick Browser
  Working with QuickPick Toolbar
  Working with the QuickPick Browser Window
  Filtering Information in the QuickPick Browser Window
Basic Navigation in Design Entry HDL
  Panning the Drawing
  Zooming In and Out of the Drawing
  Navigating the Drawing Hierarchy
  Moving a Window
  Resizing a Window
  Closing a Window
  How do I navigate a design?
  Finding Nets and Cells in Your Design
Navigating Nets in Your Design ................................................................. 123
Exiting Design Entry HDL ................................................................. 126
Running Commands with Strokes ...................................................... 126
  Guidelines for Strokes ................................................................. 127
Modes in Design Entry HDL ............................................................... 127
  In Hierarchy Mode ................................................................. 128
  Expanded Mode ................................................................. 128
  Occurrence Edit Mode ............................................................... 129
Usability Enhancements in Design Entry HDL in Release 16.2 .................. 130
  Windows Mode ................................................................. 130
  Reorganized Menus ............................................................... 132
  Design Entry HDL Options Dialog Box ........................................... 133
  Support for Keyboard Operations ............................................... 133
  Bounding Box on Components .................................................. 134
  Changes in the Global Navigate Window ........................................ 135
  New-Look Properties Window .................................................. 136
  Showing Unconnected Pins on Components ..................................... 136

5
Creating a Schematic ........................................................................... 139
Creating a Project ............................................................................. 139
Starting Design Entry HDL ................................................................. 140
Creating a Design Page .................................................................. 141
Adding Page Borders ....................................................................... 142
Using Component Browser .............................................................. 144
  Overview .................................................................................. 144
  Launching Component Browser ................................................... 144
  Searching Parts ....................................................................... 145
  Viewing Detailed Part Information ............................................... 149
  Adding a Part ............................................................................. 152
  Replacing a Part ....................................................................... 153
  Old Vs. New Component Browser - A Comparison ......................... 154
Configuring the Physical Property Options ........................................ 154
  Overview .................................................................................. 154
  Understanding Options Sets ....................................................... 155
# Allegro Design Entry HDL User Guide

**Getting Started with Property Options** ................................................................. 155  
**Customizing PPT Options Filters** ................................................................. 159  
**Connecting Parts** .......................................................................................... 160  
  **Drawing a Wire Manually** ........................................................................... 161  
  **Auto-Routing a Wire** .................................................................................. 161  
**Naming Signals** ............................................................................................. 161  
  **Signal Naming Conventions** ....................................................................... 162  
  **Step Size in Signal Names** ......................................................................... 165  
**Adding Properties** .......................................................................................... 167  
**Adding Ports** ................................................................................................ 169  
  **Using PORT Symbols** ............................................................................... 169  
  **Rules for Using Port Symbols** ................................................................. 170  
  **Port Association Restrictions** .................................................................... 171  
**Working with Ports and Signals** .................................................................... 171  
  **Setting the Initial Value of a Signal** ........................................................... 172  
  **Specifying the Size of Nets** ....................................................................... 174  
  **Specifying the Assertion Level of Pins and Signals** .................................... 175  
  **Creating an Alias for a Signal** ................................................................. 175  
  **Declaring a Base Signal** .......................................................................... 177  
  **Global Signals** .......................................................................................... 179  
  **Unnamed Signals** ...................................................................................... 186  
  **Signal Concatenations** ............................................................................. 187  
  **Signal Replication** .................................................................................... 190  
**Merge Symbols** .............................................................................................. 190  
**Signal Slices (Bit and Part Selects)** ............................................................... 194  
**Setting the Verilog Logic Type for Ports and Signals** ..................................... 197  
**Setting the VHDL Logic Type for Ports and Signals** ...................................... 200  
**Specifying Ranges for Ports, Signals and Aliases** .......................................... 204  
**Unconstrained Ranges for Ports, Signals, and Aliases** .................................. 205  
**Resolved Types and Resolution Functions** ................................................... 205  
**Type Conversion** .......................................................................................... 206  
**Abstract Data Types in VHDL** ...................................................................... 207  
**Using Iterated Instances** ............................................................................... 208  
**X-Replication** ............................................................................................... 208  
  **Using X-Replication** .................................................................................. 209  
  **Rules for Using X-Replication** ................................................................. 209
6
Working with Wires ................................................................. 215
About Signals and Connectivity .............................................. 215
About Bus Taps ................................................................. 216
About Bus Names .............................................................. 216
Drawing a Wire Manually .......................................................... 216
Auto-Routing a Wire ........................................................... 217
Stretching a Wire ............................................................... 217
Bending a Wire ................................................................. 218
Splitting a Wire ................................................................. 218
Snapping a Wire to the Nearest Pin .............................................. 218
Naming a Signal ................................................................. 218
Wiring Bus-Through Pins ...................................................... 219
Marking Wire Connections ...................................................... 219
Naming Signals on a Bus ........................................................ 220
Specifying a Tap Symbol ........................................................ 221
Attaching Values to Bus Taps ................................................... 222
Changing Wire Thickness and Pattern .......................................... 222

7
Working with Libraries and Components .......................................... 225
About the Standard Library ...................................................... 225
Working with Libraries .......................................................... 225
   Adding New Libraries ....................................................... 226
   Browsing Libraries .......................................................... 226
   Adding Libraries to the Search Stack ..................................... 226
   Removing Libraries from the Search Stack ................................. 227
   Defining Library Search Order ............................................ 227
Working with Components ...................................................... 228
# Allegro Design Entry HDL User Guide

## Contents

- Browsing the Component List ........................................... 229
- Creating Design Entry HDL Parts .................................... 229
- Creating a Symbol in Design Entry HDL ............................. 230
- Creating Entity Declarations from Symbols .......................... 232
- Creating the chips.prt File ........................................... 235
- Creating a Part Table File ............................................ 237
- Adding a Component ..................................................... 237
- Modifying Components .................................................. 238
- Replacing a Component .................................................. 239
- Breaking Up a Component ............................................... 240
- Changing Pin States on a Component ................................. 240
- Choosing a Version of a Component .................................. 241
- Mirroring Components or Blocks ..................................... 241
- Changing the Orientation of Components or Text ................. 241
- Sectioning a Component ................................................ 242
- Sectioning Multiple Components .................................... 243
- Unsectioning Components ............................................. 248
- Swapping Pins on a Component ....................................... 249
- Ways to Determine if a Component Has Bus-Through Pins ........ 249
- Deleting a Library Component (Cells, Views, and Files) ....... 249
- Creating a Page Border Symbol ....................................... 250
  - Customizing a Page Border in the Standard Library ............ 250
  - Creating a Page Border of Your Own ............................... 252
- Using Component Revision Manager ................................ 255
  - Overview .............................................................. 255
  - How Component Revision Manager Works .......................... 255
  - Getting Started with Component Revision Manager ............. 256
  - Using Component Revision Manager ................................ 258
  - Setting Default View .............................................. 263

## 8

### Working with Properties and Text

- About Properties ....................................................... 265
  - Locking Properties .................................................. 266
  - Copying Properties .................................................. 267
## Allegro Design Entry HDL User Guide

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving Properties</td>
<td>268</td>
</tr>
<tr>
<td>Adding Properties</td>
<td>269</td>
</tr>
<tr>
<td>Displaying and Modifying Property Attributes</td>
<td>272</td>
</tr>
<tr>
<td>Making an Attribute File</td>
<td>273</td>
</tr>
<tr>
<td>Adding a URL</td>
<td>274</td>
</tr>
<tr>
<td>Adding Text</td>
<td>276</td>
</tr>
<tr>
<td>Modifying Text</td>
<td>276</td>
</tr>
<tr>
<td>Changing the Text Editor</td>
<td>278</td>
</tr>
<tr>
<td>Adding Port Names from the Corresponding Symbol</td>
<td>278</td>
</tr>
<tr>
<td>Swapping Notes or Properties</td>
<td>278</td>
</tr>
<tr>
<td>Reattaching a Property from One Object to Another</td>
<td>279</td>
</tr>
<tr>
<td>Specifying Property Display for a Specific Object</td>
<td>279</td>
</tr>
<tr>
<td>Specifying Property Display for Objects Globally</td>
<td>279</td>
</tr>
<tr>
<td>Specifying Visibility of Pin Properties</td>
<td>281</td>
</tr>
<tr>
<td>Specifying Visibility of Pin Numbers</td>
<td>282</td>
</tr>
<tr>
<td>Case-Sensitivity in Property Names and Values</td>
<td>283</td>
</tr>
<tr>
<td>Case Insensitive Property Values</td>
<td>284</td>
</tr>
<tr>
<td>Case Insensitive Property Names</td>
<td>285</td>
</tr>
<tr>
<td>Case Sensitivity and PPTs</td>
<td>287</td>
</tr>
<tr>
<td>Constructing PPTs with Case-Sensitive Values</td>
<td>288</td>
</tr>
<tr>
<td>Assigning Power Pins</td>
<td>289</td>
</tr>
<tr>
<td>Viewing Properties on Power Pins</td>
<td>289</td>
</tr>
<tr>
<td>Using the Assign Power Pins Dialog Box</td>
<td>295</td>
</tr>
<tr>
<td>Viewing Properties in Occurrence Edit Mode</td>
<td>302</td>
</tr>
<tr>
<td>Controlling the Overwriting of POWER_PINS Property</td>
<td>304</td>
</tr>
<tr>
<td>Subtype Names in POWER_GROUP Property</td>
<td>307</td>
</tr>
<tr>
<td>Working with Text Macros</td>
<td>307</td>
</tr>
<tr>
<td>About Text Macros</td>
<td>308</td>
</tr>
<tr>
<td>How to use Text Macros</td>
<td>308</td>
</tr>
<tr>
<td>Defining Text Macros on a Drawing Using the DEFINE Symbol</td>
<td>309</td>
</tr>
<tr>
<td>Defining Text Macros Using \PARAMETER or \PARAM</td>
<td>310</td>
</tr>
<tr>
<td>Defining a Text Macro in a File</td>
<td>311</td>
</tr>
<tr>
<td>Working with Occurrence Properties</td>
<td>312</td>
</tr>
<tr>
<td>About Occurrence Editing</td>
<td>312</td>
</tr>
<tr>
<td>Viewing Occurrence Properties</td>
<td>313</td>
</tr>
<tr>
<td>Occurrence Property File (OPF)</td>
<td>313</td>
</tr>
</tbody>
</table>
Working with Custom Text ................................................................. 315
  Using Custom Text ...................................................................... 320

9
Working with Block Designs ............................................................. 325
About Blocks .................................................................................. 325
About View Generation in Hierarchical Designs .............................. 325
  Generating Views for Top-Down Design ...................................... 326
  Generating Views for Bottom-Up Design .................................... 326
Creating Hierarchical Designs ........................................................ 326
  Top Down Method ..................................................................... 327
  Bottom Up Method ................................................................... 329
  Adding a Block ......................................................................... 329
  Using Read-only Blocks in Your Design .................................... 332
  Navigating the Drawing Hierarchy ............................................. 332

10
Working with Groups ................................................................. 339
Creating a Group by Rectangle ......................................................... 339
Creating a Group by Polygon .......................................................... 340
Creating a Group by Specifying an Expression .............................. 340
Creating a Group by Including Objects ......................................... 341
Grouping the Entire Schematic ....................................................... 341
Including Additional Objects in a Group ....................................... 341
Excluding Objects from a Group .................................................... 341
Setting the Current Group ............................................................. 342
Viewing Group Contents ............................................................... 342
Moving a Group ........................................................................... 342
Rotating a Group ........................................................................ 343
Rotating a Group of Properties ...................................................... 343
Spinning a Group ........................................................................ 343
Mirroring a Group ....................................................................... 343
Copying a Group ........................................................................ 346
Deleting a Group ......................................................................... 347
Specifying Color for a Group .......................................................... 347
### Allegro Design Entry HDL User Guide

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highlighting a Group on the Schematic</td>
<td>347</td>
</tr>
<tr>
<td>Replacing Components in a Group</td>
<td>347</td>
</tr>
<tr>
<td>Replacing Component Symbols in a Group (Versioning)</td>
<td>348</td>
</tr>
<tr>
<td>Breaking Up Components in a Group</td>
<td>348</td>
</tr>
<tr>
<td>Adding Properties to a Group</td>
<td>348</td>
</tr>
<tr>
<td>Specifying Property Display for a Group</td>
<td>349</td>
</tr>
<tr>
<td>Specifying Property Justification for a Group</td>
<td>349</td>
</tr>
<tr>
<td>Changing the Text Size in a Group</td>
<td>349</td>
</tr>
<tr>
<td>Modifying Components with the Same Part Name in a Group</td>
<td>350</td>
</tr>
</tbody>
</table>

### 11 Working with Designs

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expanding Your Design</td>
<td>351</td>
</tr>
<tr>
<td>Finding Nets and Cells in Your Design</td>
<td>352</td>
</tr>
<tr>
<td>Navigating Nets in Your Design</td>
<td>354</td>
</tr>
<tr>
<td>Global Modification</td>
<td>355</td>
</tr>
<tr>
<td>Modifying Component, Pin, and Net Properties</td>
<td>356</td>
</tr>
<tr>
<td>Deleting Component, Pin, and Net Properties</td>
<td>360</td>
</tr>
<tr>
<td>Selecting the Component to Replace from the Physical Part Filter</td>
<td>363</td>
</tr>
<tr>
<td>Selecting the New Component from the Physical Part Filter</td>
<td>365</td>
</tr>
<tr>
<td>Selecting the Component to Replace from the Design (Schematic Pick)</td>
<td>367</td>
</tr>
<tr>
<td>Selecting the New Component from the Design (Schematic Pick)</td>
<td>368</td>
</tr>
<tr>
<td>Erroneous Conditions</td>
<td>372</td>
</tr>
<tr>
<td>Design Entry HDL Issues Affecting the Global Modify Solution</td>
<td>376</td>
</tr>
<tr>
<td>The Batch Mode Operation</td>
<td>376</td>
</tr>
<tr>
<td>Part Manager</td>
<td>378</td>
</tr>
<tr>
<td>The Part Manager Use Model</td>
<td>378</td>
</tr>
<tr>
<td>The Part Manager User Interface</td>
<td>379</td>
</tr>
<tr>
<td>Working with Part Manager</td>
<td>384</td>
</tr>
<tr>
<td>Updating Part Instances</td>
<td>385</td>
</tr>
<tr>
<td>Applying Changes to Part Instances</td>
<td>387</td>
</tr>
<tr>
<td>Resetting Changes Made to Part Instances</td>
<td>387</td>
</tr>
<tr>
<td>Updating Multiple Parts with a Single PTF Row</td>
<td>388</td>
</tr>
<tr>
<td>Refreshing Contents of Part Manager</td>
<td>388</td>
</tr>
<tr>
<td>Highlighting Part Instances on the Schematic</td>
<td>389</td>
</tr>
</tbody>
</table>
Displaying the Canonical Path ................................................. 390
Running Scripts ....................................................................... 390
  Running a Script ..................................................................... 392
  Stopping a Script .................................................................. 394
  Sample Scripts ...................................................................... 394
Highlighting (Cross-Probing) Objects ...................................... 394
Distributing Design Changes between Physical and Logical Designs ............................................. 395
Applying Connectivity Changes from the Physical Design to Your Schematic .................. 395
Back Annotating Your Design .................................................. 395
Module Ordering ...................................................................... 396
  Module Ordering in Design Entry HDL 15.x Releases .................. 396
Using the Hierarchy Viewer ....................................................... 397
Excluded Modules File ............................................................ 400
Sheet Names in Hierarchy Viewer ............................................. 401
Setting Up Page Numbers ...................................................... 403
Sheet Name Input in New Designs .......................................... 404
Performing Page Management Operations .................................. 405
  Inserting a Page ................................................................. 405
  Deleting a Page .................................................................... 409
  Compressing Pages ............................................................ 411
  Moving a Page ..................................................................... 411
  Definitions ......................................................................... 414
  Points to Remember ............................................................ 414
Displaying and Working with Schematic Page Numbers ......................... 416
  Displaying Page Numbers in a Schematic ......................... 417
  Modifying Custom Text for Page Numbers ....................... 420
  Updating Custom Text Variables for Page Numbers .......... 421
  Managing Schematic Pages .................................................. 422
Importing Designs ................................................................. 426
  Overview ............................................................................ 426
  Importing a Design Sheet .................................................... 426
Baselining a Design ................................................................ 433
  How Baselining Works ......................................................... 434
  Setting Preferences for Metadata Creation ....................... 434
  Baselining a Design ............................................................ 436
Creating the Table of Contents for a Design ................................. 438
Adding the Table of Contents to a Design ........................................ 439
Creating a Symbol for TOC .......................................................... 445

12
Netlisting Your Design ................................................................. 449
Netlisting for Packaging the Design ............................................... 450
  Disabling Netlisting of Designs ................................................. 452
  What Happens if Netlisting is Disabled and You Package the Design? 453
Netlisting for Simulation ............................................................... 454
  Netlisting for Digital Simulation ............................................... 454
  Netlisting for Synthesizing a Design in Synplify ............................ 456
  Netlisting for Analog and Mixed Signal Simulation ....................... 460
Netlisting Read-only Blocks Used in a Design ............................... 462

13
Plotting Your Design ................................................................. 465
Windows Plotting on Windows and UNIX Platforms ....................... 465
  Setting Up Windows Plotting Options ...................................... 466
  Previewing the Design .......................................................... 469
  Plotting the Design ........................................................... 472
  Plotting in Batch Mode ........................................................ 474
HPF Plotting on UNIX Platforms ................................................ 481
  Setting up HPF Plotting Options ............................................ 482
  Plotting the Design ........................................................... 487
  Plotting the Design from the Console Window ............................ 489
  Plotting Drawings from the UNIX Shell .................................... 498
Hierarchical Plotting ................................................................. 504
  Hierarchical Plotting in Hierarchy, Expanded, and Occurrence Edit Modes 504
  Changing the Order in Which Designs Are Plotted ...................... 506
  Plotting Hierarchical Designs .............................................. 508
Customization of Plotting on UNIX Platforms .............................. 512
  Customizing Windows Plotting on UNIX Platform ..................... 512
  Customizing HPF Plotting on UNIX Platform ............................ 518
Frequently Asked Questions in Plotting ...................................... 520
14
Design Techniques .............................................................. 531
Introduction ................................................................. 531
Flat Designs ................................................................. 531
    Creating a Flat Design ............................................. 532
    Concurrent Engineering of Flat Designs ....................... 533
    Considerations of Flat Designs ................................ 533
Structured Designs ....................................................... 534
    Creating a Structured Design .................................... 534
    Benefits of Structured Designs ................................ 539
    Considerations of Structured Designs ......................... 539
Hierarchical Designs ..................................................... 539
    Creating a Hierarchical Design ................................. 540
    Creating Symbols .................................................. 542
    Editing Symbols .................................................... 542
    Benefits of Hierarchical Designs ............................... 546
Comparing Design Techniques ........................................... 546

15
Simulating using AMS Simulator ....................................... 549
What is AMS? .............................................................. 549
Notes for Using Design Entry HDL with AMS ...................... 550
Conversion of Old AMS Projects to New Project Format ......... 550
    New Directory Structure for AMS Projects .................. 552
Design Entry HDL Bias Display with AMS ......................... 559
    What is a bias point value? ..................................... 559
    Bias Display in Design Entry HDL .............................. 559
    Loading Bias Point Values ..................................... 561
    Displaying bias point values ................................. 562
    Bias Point Display Settings ................................... 564
Split Part ......................................................................... 567
Displaying PSpice Names ................................................ 568
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:

<table>
<thead>
<tr>
<th>See...</th>
<th>For Information About...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1, “About Design Entry HDL”</td>
<td>Lists the various features of Design Entry HDL</td>
</tr>
<tr>
<td>Chapter 2, “Getting Started”</td>
<td>Describes the Design Entry HDL UI and basic steps for creating a schematic</td>
</tr>
<tr>
<td>Chapter 3, “Project Creation and Setup”</td>
<td>Introduction to creating and setting up design projects</td>
</tr>
<tr>
<td>Chapter 4, “Design Entry HDL Editing Environment”</td>
<td>Setting up and using the Design Entry HDL editing environment</td>
</tr>
<tr>
<td>Chapter 5, “Creating a Schematic”</td>
<td>Describes the tasks to be performed for creating a schematic</td>
</tr>
<tr>
<td>Chapter 7, “Working with Libraries and Components”</td>
<td>Describes how to use the default Design Entry HDL libraries and perform operations on components</td>
</tr>
<tr>
<td>Chapter 6, “Working with Wires”</td>
<td>Describes various operations that can be performed on wires in drawings</td>
</tr>
<tr>
<td>Chapter 8, “Working with Properties and Text”</td>
<td>Lists procedures for different types of properties and text in Design Entry HDL</td>
</tr>
</tbody>
</table>
Related Documentation

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

**Design Entry HDL**

- For information on the new features in 16.01, see *Allegro Design Entry HDL: What’s New in Release 16.01*.
- For learning Design Entry HDL, see *Allegro Design Entry HDL Tutorial*.
- For learning 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 the Packager-XL Reference.

- For information on 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.

- For learning how to create new Design Entry HDL projects and make various settings for them, see the Project Manager User Guide.

- For learning how to use the Design Entry HDL utilities - CRefer, Archiver and BOM, see the Design Entry HDL Utilities User Guide.

- For information on maintaining and modifying the Design Entry HDL digital libraries, see the 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 learning the Design Entry HDL Programmable IC flow, see Programmable IC Tutorial.

- For information on capturing electrical constraints in Constraint Manager, see the 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 about moving from SCALD to HDL flow, see *SCALD to HDL Evolution 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 the *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 the 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 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 CheckPlus, an advanced rule checking and rule development system.
- Integration with 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. For more information, see Importing Radio-Frequency Designs in the PCB Design Flows.

- Design Entry SKILL, the SKILL programming interface to Design Entry HDL.
Design Entry HDL in the Design Flow

- Project Creation and Setup
- Design Entry
- Packaging your Design
- PCB Layout
- Archiving your Project

Synchronizing Schematics and Boards

Library Management
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 35
- Design Entry HDL Basics on page 39

Starting Design Entry HDL

After you open the desired 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.

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

When you click the Design Entry image in Project Manager, the Design Entry HDL user interface appears (as shown in the following figure).

The interface consists of the following elements:
- Design window
- Menu bar
- Toolbars
- Status bar
Allegro Design Entry HDL User Guide
Getting Started

- Console command window
- Context-sensitive menus

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.

- Display
  For operations such as highlighting and de-highlighting components.

- AMS Simulator
For performing analog, digital and mixed-signal simulation using the AMS Simulator. This menu is visible only if you have installed AMS Simulator.

■ RF-PCB
Invokes the RF-PCB IFF Import UI which enables you to import radio frequency design into your schematic.

■ Tools
For operations such as setting up defaults, customizing, updating the 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
Invokes the Design Entry HDL help page and web resources such as Sourcelink and Education Services.

Toolbars

Design Entry HDL has eight toolbars. They are Standard, Block, Add, Edit, Color, Markers, Group, and QuickPick. If you have installed AMS Simulator A/D, the following six additional toolbars are available. For more information on these toolbars, see the Allegro AMS Simulator User’s Guide.

■ Analog
■ Passive
■ Source
■ Linear
■ Discrete
■ Misc

Standard Toolbar
The *Standard* toolbar has the standard functions that operate on a drawing (Open, Save, Print, Zoom, Previous Page, Next Page, Check and so on).

**Block Toolbar**

![Block Toolbar](image)

The *Block* toolbar lets you add blocks, add pins on blocks and draw wires to connect blocks.

**Add Toolbar**

![Add Toolbar](image)

The *Add* toolbar lets you add objects (components, wires, and text) and graphics such as dots and circles.

**Edit Toolbar**

![Edit Toolbar](image)

The *Edit* toolbar lets you perform edit operations such as copy, paste, delete, and spin.

**Group Toolbar**

![Group Toolbar](image)

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

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

Color Toolbar

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

QuickPick Toolbar

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

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.*
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

- Creating a Schematic on page 35
- Creating a Hierarchical Design on page 37

Creating a Schematic

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

- Create a Project
- Set up Design Entry HDL
- Add Page Borders
- Add Parts
- Connect Parts
- Name Signals
- Add Properties
- Add Ports
- Check the Design
- Fix Errors
- Save the Design
- Package the Design
- Cross-reference the Design
- Archive the Design
Creating a Hierarchical Design

The following figure illustrates the sequence of tasks you perform to create a hierarchical design.
Tasks in Creating a Hierarchical Design

1. Create a Project
2. Set up Design Entry HDL
   - Create a top-level schematic
   - Add blocks for sub-designs
   - Create sub designs
   - Check the Design
   - Fix Errors
   - Save the design
   - Create a low-level schematic
   - Generate symbol from schematic
   - Create a new schematic
   - Instantiate symbol in schematic
   - Check the Design
   - Fix Errors
   - Save the design
3. Package the Design
4. Cross-Reference the Design
5. Archive the Design
Design Entry HDL Basics

This section answers the basic 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:

- 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 setup options?

Global setup options are located in the Project Manager. You can access Design Entry HDL setup options both 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

- 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:
  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):

<table>
<thead>
<tr>
<th>To</th>
<th>Press</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move the cursor backwards</td>
<td>Left Arrow</td>
</tr>
<tr>
<td>Move the cursor forward</td>
<td>Right Arrow</td>
</tr>
<tr>
<td>Move the cursor to the beginning of the line</td>
<td>Home - or - Right-click and select Position at BOL.</td>
</tr>
</tbody>
</table>
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?

The capability to add and edit components used to be contained in a single browser. These are now separate functions.

- Choose File – Open to display a file browser from which you select the drawing you want to edit.

- Choose Component – Add to display the Component Browser from which you can select components to add to your drawing.

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.

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 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.

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

How do I generate a Verilog or VHDL netlist from a schematic?

When you save a design, Design Entry HDL generates a netlist if you have selected the option for generating a Verilog or VHDL netlist in Tools – Options – Output.

How do I add additional pages?


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


   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.

For more information on page numbering in Design Entry HDL, see *Displaying and Working with Schematic Page Numbers* on page 416.

### How do I plot a design?

You can plot a design using *File – Plot*. On UNIX, you have the option of using the HPF plotting utility also depending on the option you select (Windows plotting or HPF) using *Tools – Options – Plotting*.

### 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. It clearly shows the group that you are working with at any time by indicating the group name in brackets next to group 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 invoke Packager-XL using the Design Synchronization tool of the 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 the schematic with the 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 permissions 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 a plot of a schematic, it is often difficult to trace a signal or instances of a part. The Cross Referencer tool 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.

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 Libraries server is not available.

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

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

You can enable the bias display feature of Design Entry HDL to view the 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 you need to perform the following steps:

1. Load bias point values.
   
   From the AMS 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 AMS 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 the schematic, choose AMS Simulator – Bias Points > Enable Bias Voltage Display.

- To display bias currents on the schematic, choose AMS Simulator – Bias Points – Enable Bias Current Display.

- To display bias power values, choose AMS 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 AMS 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 15, “Simulating using AMS Simulator”.
Project Creation and Setup

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

- Introduction to Project Manager on page 49
- Project Structure on page 50
- Project Files on page 51
- Project Flows on page 54
- Creating a Project on page 54
- Setting Up a Project on page 61
- Locking Project File Directives on page 74

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 projects created by designers where the basic aim is creation of designs. Library projects are created by librarians for the sole purpose of library creation and management.

- 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.

- Import, export, and archive projects
Launch tools such as Design Entry HDL, PCB Editor, Allegro SI, Library Explorer, Part Developer, and PadStack Editor.

View project settings and libraries in a convenient tree form, and keep track of all 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.

**Project Structure**

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

<table>
<thead>
<tr>
<th>Lib</th>
<th>It is a directory that contains designs (cells).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell</td>
<td>It is the design directory. It contains all the data organized into views for that design. For example, the design cpu can have schematic, symbol, chips, packaged, and physical views. The configurations of a design are also stored as views in the design directory.</td>
</tr>
<tr>
<td>View</td>
<td>It 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 sch_n view (where n is the version number). All the information for a symbol representation of the design is in the sym_n view (where n is the version number). The packaged design data is in the packaged view. Configurations, including the four default configurations—cfg_package, cfg_verilog, cfg_pic, and cfg_vhdl—are also views under a cell.</td>
</tr>
</tbody>
</table>
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 three 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 the 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 15, “Simulating using AMS 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.
Project File (.cpm) Example

( Machine generated file created by SPI )
( Last modified was 11:38:31 Thursday, October 09, 1997 )
( 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 '0'
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

You create the site project file, called site.cpm, in the <your_inst_dir>/share/local/cdsssetup/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.

Installation Project File

The installation project file called cds.cpm is shipped by Cadence and is in the <your_install_dir>/share/cdsssetup/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 `$HOME/cdsssetup/projmgr/`, provided the environment variable `$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:** Currently, on UNIX platforms, Project Manager is not an HTML browser. You cannot use HTML files to define custom flows. Instead, use image maps to launch tools.

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/cdsssetup/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, perform the following steps:

1. Choose **File – New**.
   The **New Project Wizard** appears.

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

3. In the **Location** box,
Type the complete path of the folder in which you want to create the new project.

- or -

Click Browse, select a folder in the Choose Directory dialog box, and then click OK. The Location box is filled in with the path of the folder you selected.

**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 and 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,

- In the *Library List*, select the library in which you want to create the new design.
- In the *Design Name* box, type a name for the new design.

To select an existing design,

- In the *Library List*, select the library that contains the design.
- 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, the project location, the design library, the 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

```
*  (  
)  -  
+  `  
?  .  
~  `  
```

The following characters cannot be used in design names

```
!  @  
$  %  
\  .  
```
Files Created for Your New Project

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

- **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)**
  Project Manager creates four configurations for each new project - 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 the tools 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 `myproject` and a design `cpu` in `laptop.lib`, you will have the following file structure:

```
myproject
  laptop.cpm  laptop_lib  cds.lib  temp
    (worklib)
      cpu
        cfg_package  cfg_verilog  cfg_vhdl  cfg_pic
          expand.cfg  expand.cfg  expand.cfg  expand.cfg
```

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,

2. In each tab of the `Project Setup` dialog box, specify the default setup information you want for all projects. For information about the setup options, click the `Help` button in the dialog box.
3. Click `Apply` to save your changes.
6. In the `Export Project` dialog box,
Type site.cpm in the File Name box.

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.

Ensure that the Save File as Type box displays Project Files (*.cpm).

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 your CDS_SITE = 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 over 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 recommended below:

<table>
<thead>
<tr>
<th>Files and Descriptions</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>concepthdl_menu.txt</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/concept/ to $CDS_SITE/concept/ .</td>
</tr>
<tr>
<td>(Design Entry HDL menus)</td>
<td></td>
</tr>
<tr>
<td>cref.dat</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/creferhdl/ to $CDS_SITE/cdssetup/creferhdl/ .</td>
</tr>
<tr>
<td>(template options of CRefer)</td>
<td></td>
</tr>
<tr>
<td>concepthdl_key.txt</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/concept/ to $CDS_SITE/concept/ .</td>
</tr>
<tr>
<td>(Design Entry HDL shortcut keys)</td>
<td></td>
</tr>
<tr>
<td>bom.callouts</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/ to $CDS_SITE/cdssetup/ .</td>
</tr>
<tr>
<td>(mechanical parts to be added in the BOM reports)</td>
<td></td>
</tr>
<tr>
<td>cdsinfo.tag</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/ to $CDS_SITE/cdssetup/ .</td>
</tr>
<tr>
<td>(project-specific information, including the name of the data management system, if any, used in the project)</td>
<td></td>
</tr>
<tr>
<td>cdsprop.paf</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/ to $CDS_SITE/cdssetup/ .</td>
</tr>
<tr>
<td>(information about properties)</td>
<td></td>
</tr>
<tr>
<td>cdsprop.tmf</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/ to $CDS_SITE/cdssetup/ .</td>
</tr>
<tr>
<td>(information about text macros)</td>
<td></td>
</tr>
<tr>
<td>cjedectype.txt</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/ to $CDS_SITE/cdssetup/ .</td>
</tr>
<tr>
<td>(compatible JEDEC types in the Variant Editor tool)</td>
<td></td>
</tr>
<tr>
<td>propflow.txt</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/ to $CDS_SITE/cdssetup/ .</td>
</tr>
<tr>
<td>(the default property flow setup in Packager Setup)</td>
<td></td>
</tr>
<tr>
<td>template.bom</td>
<td>Copy from &lt;your_inst_dir&gt;/share/cdssetup/ to $CDS_SITE/cdssetup/ .</td>
</tr>
<tr>
<td>(the default template for BOM reports)</td>
<td></td>
</tr>
</tbody>
</table>
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
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.
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.
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.

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

Editing the cds.lib File

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:

- A directive to include the installed Cadence libraries. (For example: INCLUDE <your_install_dir>/share/cdsssetup/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

1. Open the project.


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:/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 librarypath
```

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

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 librarypath. 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
```

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
```

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
```

**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.
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. To add one library, select the library in the `Available Libraries` list and click `Add`.
7. To add all the libraries in the `Available Libraries` list, click `Add All`.
8. To remove one library, select the library in the `Project Libraries` list and click `Remove`.
9. To remove all the libraries in the `Project Libraries` list, click `Remove All`.
10. 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.
11. To move a library one level up, select the library and then click `Up`. 
12. To move a library one level down, select the library and then click **Down**.

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

**Available Libraries and Project Libraries**

Available Libraries

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

Project Libraries

They 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 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 data you need to add or modify the physical properties of a symbol. The *.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, then Packager-XL does not read it if the INCLUDE_PPT directive is set. To include the cell-level *.ptf* file, you will have to add it in the INCLUDE_PPT directive.

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.
The following steps are required 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. 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.
      —or—
      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**.
   - Select the **Merge Physical Part Table Files** check box to merge the information in all included Physical Part Table files.
   - Select the **Perform Case Sensitive Row Match** check box to perform a case-sensitive match of 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 Allegro, 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:

- 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.

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:


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 on Windows and vi on UNIX.

To select a text editor

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,

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.
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, and MPS sessions and hosts.

If you want to maintain a log file for the 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

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.

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);
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:

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. 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:
You can select a different configuration for each expansion type.

To select the configuration for each expansion type:

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:

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

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 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:

<table>
<thead>
<tr>
<th>Type of View</th>
<th>View Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packaged</td>
<td>packaged</td>
</tr>
<tr>
<td>Chips</td>
<td>chips</td>
</tr>
<tr>
<td>Part Table</td>
<td>part_table</td>
</tr>
<tr>
<td>Physical</td>
<td>physical</td>
</tr>
<tr>
<td>Constraints</td>
<td>constraints</td>
</tr>
<tr>
<td>Board Design</td>
<td>sch_1</td>
</tr>
<tr>
<td>Programmable IC Design</td>
<td>sch_1</td>
</tr>
<tr>
<td>Verilog Simulation</td>
<td>sim_sch_1</td>
</tr>
<tr>
<td>VHDL Simulation</td>
<td>sim_sch_1</td>
</tr>
</tbody>
</table>
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.

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 would get reflected in all the projects you open, irrespective of the settings present in the project’s `.cpm` file.

**Controlling Project Settings at Different Levels**

Until now, project settings of a Design Entry HDL project were configured in the Cadence default `cpm` file (`cds.cpm`), the CDS_SITE area (`site.cpm`), and in the local project (`<project>.cpm`).

Unlike Allegro PCB Editor which supports an env file in the user’s home account, the Design Entry HDL solution does not provide a way for users to customize project settings for all projects they open.

This means that you have to change the individual settings each time you open a project because there is no provision for user 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.
To address this issue in Design Entry HDL, the CPM file directive locking feature provides control over the list of directives which you can configure at the user level and which would reflect in all the projects irrespective of the project settings. This is achieved by the introduction of another level, user.cpm, where user-specific settings are defined. The new user.cpm file is located at $HOME/cdssetup/projmgr/, provided the environment variable $HOME is set. 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 and is present in a new section of 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 present in project.cpm. If it is not present in project.cpm, the directive will be honored from site.cpm or cds.cpm as the case may be. However, the directive if present in user.cpm will not be honored.

### Example

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

The following is an existing section:

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

To lock the directive, the following new section is added 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 shown in the following:

```
START_CONCEPTHDL_CONTROL_SETTINGS
```

```
ALLOW_USER_CPM
```

```
END_CONCEPTHDL_CONTROL_SETTINGS
```
Locking Support for Directives at Different Levels

The directive locking mechanism 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 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 present in user.cpm will be honored only when 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.

**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 the settings made at different levels impact your project:
CPM File | Description
--- | ---
user.cpm | Contains the user-level settings
Directives defined in the `user.cpm` file are honored if:
- They are not locked at the `<project>.cpm`, `site.cpm`, or `cds.cpm` levels
- The `ALLOW_USER_CPM` keyword is present at `site.cpm` or `cds.cpm` for the directive
Modified directives honored in `user.cpm` are written to `user.cpm` and not to `<project>.cpm`.

`<project>.cpm` | Contains settings that are local to a project
Directives defined in the `<project>.cpm` file are honored if:
- They are not honored in `user.cpm`
- They are not locked at `site.cpm` or `cds.cpm`
Modification to a directive locked at `<project>.cpm` is allowed and the new value is written in the `<project>.cpm`.
The modified directive in the `<project>.cpm` file does not have the `LOCK` keyword associated with it.

`site.cpm` | Contains site-level settings
Directives defined in the `site.cpm` file are honored if:
- They are not honored in `user.cpm` or `<project>.cpm`
- They are not locked at `cds.cpm`
Modifications to the directives are not honored if they are locked in `site.cpm` itself. Otherwise, all modifications to the directives are written to the `<project>.cpm` file.
A modified directive in the `<project>.cpm` file does not have the `LOCK` keyword associated with it.
Example

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

<table>
<thead>
<tr>
<th>CPM File</th>
<th>Directive Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>cds.cpm</td>
<td>Contains install-level settings</td>
</tr>
<tr>
<td></td>
<td>Directives defined in the cds.cpm file are honored if they are not honored in user.cpm, &lt;project&gt;.cpm, or site.cpm.</td>
</tr>
<tr>
<td></td>
<td>Modifications to the directives are not honored if they are locked in the cds.cpm itself. Otherwise, all the modifications to the directives are written to the &lt;project&gt;.cpm file. The modified directive in the &lt;project&gt;.cpm file does not have the LOCK keyword associated with it.</td>
</tr>
<tr>
<td>user.cpm</td>
<td>START_CONCEPTHDL</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>PINNUMBER_SIZE 0.075</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>END_CONCEPTHDL</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>&lt;project.cpm&gt;</td>
<td>START_CONCEPTHDL</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>PINNUMBER_SIZE 0.072</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>END_CONCEPTHDL</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>site.cpm</td>
<td>START_CONCEPTHDL</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>PINNUMBER_SIZE 0.090</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>END_CONCEPTHDL</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>START_CONCEPTHDL_CONTROL_SETTINGS</td>
</tr>
<tr>
<td></td>
<td>PINNUMBER_SIZE LOCK</td>
</tr>
<tr>
<td></td>
<td>END_CONCEPTHDL_CONTROL_SETTINGS</td>
</tr>
</tbody>
</table>
Here, the directive `PINNUMBER_SIZE` with a value of `0.090` (in `site.cpm`) will be honored. Modification to this value is not allowed.

<table>
<thead>
<tr>
<th>CPM File</th>
<th>Directive Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>cds.cpm</td>
<td>START_CONCEPTHDL</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>PINNUMBER_SIZE 0.082</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>END_CONCEPTHDL</td>
</tr>
</tbody>
</table>
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 81
- Basic Editing Tasks on page 84
- Displaying Information on page 103
- Using the QuickPick Browser on page 111
- Basic Navigation in Design Entry HDL on page 118
- Running Commands with Strokes on page 126
- Modes in Design Entry HDL on page 127
- Usability Enhancements in Design Entry HDL in Release 16.2 on page 130

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


2. Click an option on the left pane to display the setup options you want to view.

3. For example, click Grid to display the grid options.

4. Click OK.
Enabling the Pre-Select Menu Use Model

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

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 new mode of Design Entry HDL.

Important

If you customize Design Entry HDL menus in 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.

Now, for operations on schematics, select the object first, and 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 menu option Edit – Rotate.

Note: In the new Windows Mode of Design Entry HDL, these operations are much easier to handle. See Usability Enhancements in Design Entry HDL in Release 16.2 on page 130 for more details.

1. Select the component.
2. Choose Edit – Rotate.
   OR

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:

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:*

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 vi on UNIX and 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 85
Undoing an Operation

To undo an operation:

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

To redo an operation:

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

1. Select the object.
2. Move the object to a new location and click again.
   OR
3. Choose *Edit – Move*.
4. Click on the object.
   
   **Note:** When moving a wire, select the middle of the wire. Selecting the wire at the end stretches it.

5. Move the object to a new location and click again.

To move a wired component

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 LMB near the open edge of the wire.**
   
   Only the wire moves, the component attached to it does not move.

   **Click LMB on the wire segment near the component pin.**
   
   Both the component and the wire attached to it move.

   **Click RMB and select the context menu item Change Attachment.**
   
   There are three possible states of the move operation:

   - The first time selection of the *Change Attachment* menu item moves the component along with all the wires connected to it. The attached wires are directly routed.
   - The second time selection of the *Change Attachment* menu item disconnects the wire segment from the pin of the component and moves the wire segment.
The third time selection of the Change Attachment menu item 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

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, click the right mouse button 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:

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.

   **Tip**

   You can also use 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 specifying 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 are done placing copies but wish to remain in the Edit – Copy mode, choose Terminate Selection from the pop-up menu. To exit the 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.

To copy an object and its properties:

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 specifying 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 wish 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:

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 wish 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

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, click the right mouse button 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 92.

### Deleting Objects

To delete an object:

1. Choose Edit – Delete.

   **Tip**

   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
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, click the right mouse button 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:

1. Choose Edit – Color.
2. Click on the objects whose color you want to change.

**Drawing an Arc**

To draw an arc:

1. Choose Edit – Arc.

   OR

   Choose 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:*

1. Choose Edit – Circle.

   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:**

1. Choose *Edit – Arc*.
   Choose *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 object(s).

**Displaying the Console Window**

➤ Choose *View – Console Window*.

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

**Editing Text in Dialog Boxes and the Console Window**

<table>
<thead>
<tr>
<th>To</th>
<th>Press</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move the cursor left, right, up, or down</td>
<td>Arrow keys</td>
</tr>
<tr>
<td>Move the cursor to the beginning of line</td>
<td>Home</td>
</tr>
<tr>
<td>Move the cursor to the end of line</td>
<td>End</td>
</tr>
<tr>
<td>Select text (or extend selection) to the left, right, up, or down</td>
<td>Shift + Arrow keys</td>
</tr>
<tr>
<td>Select text (or extend selection) to the end of line</td>
<td>Shift + End</td>
</tr>
</tbody>
</table>
### Copying Parts of a Schematic Across Pages, Designs, and Projects

Design Entry HDL provides support for 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:

<table>
<thead>
<tr>
<th>To</th>
<th>Press</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select text (or extend selection) to the beginning of line</td>
<td>Shift + Home</td>
</tr>
<tr>
<td>Delete the previous character</td>
<td>Backspace</td>
</tr>
<tr>
<td>Delete the next character</td>
<td>Delete</td>
</tr>
<tr>
<td>Delete selected text</td>
<td>Delete</td>
</tr>
<tr>
<td>Copy selected text</td>
<td>Ctrl + Insert - or - Ctrl + C - or - Copy (UNIX only)</td>
</tr>
<tr>
<td>Cut selected text</td>
<td>Ctrl + X - or - Cut (UNIX only)</td>
</tr>
<tr>
<td>Paste</td>
<td>Shift + Insert - or - Ctrl + V - or - Paste (UNIX only)</td>
</tr>
<tr>
<td>Undo the last cut or paste</td>
<td>Alt + Backspace</td>
</tr>
</tbody>
</table>

**Note:** On Windows, you can also use commands from the pop-up menu to edit text in dialog boxes. These commands are not available on UNIX platforms.
1. Select the part on the schematic.

2. Choose Edit – Copy.

   Alternatively, you can 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. Choose Edit – Paste and click at the target location on the schematic canvas.

   Alternatively, you can right-click at the target location and select Paste from the pop-up menu.

5. Click repeatedly if you want to place multiple instances of the copied part on the schematic.

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

Note: You need to inactivate 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 project cpn 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 choose Paste Special from the pop-up menu.
The following dialog box appears where you can paste copied parts on to the schematic by selecting the *Paste Schematic* option or change 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 containing current signal names and Design Entry HDL-recommended signal names appears.

You can choose to edit these signal names or leave a signal unnamed. You can add a prefix or a suffix to the existing signal name also.
3. To edit signal names, enter new value in the New Signal Name field and click OK. The signal names are changed and the copied part appears selected with the mouse cursor.

4. Click at the target location on the schematic canvas.

5. Click again to invoke 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 present 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.

![Warning dialog box](image)

**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 such that the border is not included in the project
- A primitive is not defined in any library
- 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:

1. Select the text in the text editor.

2. Choose Edit – Copy.
   Alternatively, you can right-click on the text and select copy from the pop-up menu.

3. Click the location on 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 original text casing, make changes in the Design Entry HDL Options - Text page.

4. To display text setup options, click Tools – Options and select Text in the left pane.

5. To retain original text casing, clear the Upper-case Input check 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 Design Entry HDL 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:

<table>
<thead>
<tr>
<th>When you copy a ...</th>
<th>Design Entry HDL ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>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 are ignored while copying text. Every paragraph of text is added as a separate note.</td>
</tr>
<tr>
<td>Hyperlink</td>
<td>Copies it as an active link in the form of a note. If you double-click on this note, the appropriate link would open in a Web browser.</td>
</tr>
<tr>
<td>Special character</td>
<td>Copies it as a note.</td>
</tr>
</tbody>
</table>

**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:

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 the frequently used commands in the console window. You select a piece of text appearing in the console window and copy and paste it for executing it again. If Design Entry HDL correctly resolves 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:

1. Select the command/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 can use this functionality to paste a company logo to a page border.

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

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

**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’`.

**Tip**

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, the table-based formatting would be lost in
the text paste operation. You can grab an image of the datasheet and then paste it on a schematic page. An example of datasheet copied as an image is shown below.

![Image of datasheet](image.png)

**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 part 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:

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 gets 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:

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.

Size and rotate: Width  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 ratio  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), the tool automatically adjusts the values of the other three options based on your input for the selected option. Deselect this check 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 the image of a 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:

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 104
- Displaying Toolbars and Other Parts of the Design Entry HDL Window on page 108
- Highlighting Objects on page 109
- Turning Off Highlighting on page 110
- Opening the Markers Control Window on page 110
- Displaying the Markers Toolbar on page 111
Displaying Schematic Information

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

Design Entry HDL also displays information about these items:

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

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.


   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

2. Click a wire (net).
   The selected net is displayed in red.

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:*

1. Choose *View – Toolbars*.

2. Click the check boxes next to the toolbar you want to display:
   
   - Standard
   - Add
   - Block
   - Markers
Design Entry HDL places a check in the box next to the toolbar you want to display. If a toolbar is already checked (on) 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**

- 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 (on), Design Entry HDL turns it off and removes the check.

**Highlighting Objects**

To highlight an object:

1. Choose **Display – Highlight**.
2. Click an object.

To highlight multiple objects
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, click the right mouse button and choose Exclude to exclude components, properties or wires from the selected objects.

2. Click the right mouse button and choose Highlight to highlight the objects.

**Retaining Successive Highlighting**

To retain successive highlighting of different components on the same page, set the RETAIN_PREVIOUS_HILITE directive to ON in the project (.cpm) file. By default, this directive is set to OFF, which results in dehighlighting of previous highlighted instance when a new instance is selected in Allegro PCB Editor or the Global Navigate window in Design Entry HDL.

**Turning Off Highlighting**

To de-highlight an object:


2. Click the pin, wire (net), or component that you want to de-highlight.

To de-highlight multiple objects

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, click the right mouse button 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:

- Choose Tools – Markers.

  or

  Click in the Markers toolbar.
Displaying the Markers Toolbar

   The Toolbars dialog box is displayed.
2. Check Markers.
3. Click OK.

Displaying the Error Status Bar

➤ Choose View – Error Status Bar.
   or
   Click in the Markers toolbar.

To enlarge the drawing:

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 may have designs requiring the use of common parts and frequently-used library components. In such cases, searching and opening the same part repeatedly using the Component Browser window can turn to be a time-consuming task impacting the productivity and efficiency. To help you save time and quickly add frequently-used parts and components to your design, Allegro Design Entry HDL has introduced the QuickPick Browser feature.

Using QuickPick Browser, you can:

- Add frequently-used parts quickly in a design.
Add commonly-used standard library components directly without using the Component Browser window.

Add previously-used local blocks in a design.

**Note:** The QuickPick Browser is available only with the Component Browser.

### QuickPick Browser - Interface Overview

You can access the QuickPick Browser feature using a toolbar, which 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 to appear (View – Toolbars). You can use the QuickPick toolbar to quickly add commonly-used cells, parts, and local blocks to the design.

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

<table>
<thead>
<tr>
<th>Icon..</th>
<th>Lets you..</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Power Icon" /></td>
<td>Add a cell listed under the Power category icon.</td>
</tr>
<tr>
<td><img src="image" alt="Ground Icon" /></td>
<td>Add a cell listed under the Ground category icon.</td>
</tr>
<tr>
<td><img src="image" alt="Ports Icon" /></td>
<td>Add a cell listed under the Ports category icon.</td>
</tr>
<tr>
<td><img src="image" alt="Taps Icon" /></td>
<td>Add a cell listed under the Taps category icon.</td>
</tr>
</tbody>
</table>
Setting up QuickPick Browser

Before using the QuickPick toolbar in Allegro Design Entry HDL, ensure that you specify standard library cells to appear 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:

1. Open an instance of the command window.
2. Type qpsetup and press ENTER.
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 containing 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.

![QPSetup dialog box](image)

---

**Important**

For automatic populating this field, you need to 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.

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. 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:

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.

2. Choose a cell from the drop-down list.

3. Place the cell on the schematic.

To add a part or block:
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 block. All such cells appear as blocks in the drop-down list (beside the Blocks icon).

   **Note:** If you're 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, QuickPick feature is available, irrespective of the license you have.

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

**Working with the QuickPick Browser Window**

You can use the QuickPick Browser window to view the description 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:

1. Double-click a category icon in the toolbar.
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 Component Browser. 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:

1. Enter a string in the filters to filter cell, part or block names, and press Enter.
2. Alternatively, 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 118
- Zooming In and Out of the Drawing on page 119
- Navigating the Drawing Hierarchy on page 120
- Moving a Window on page 121
- Resizing a Window on page 121
- Closing a Window on page 121
- How do I navigate a design? on page 121
- Finding Nets and Cells in Your Design on page 122
- Navigating Nets in Your Design on page 123
- Exiting Design Entry HDL on page 126

Panning the Drawing

To pan using the mouse:

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:**

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:**

■ 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.

1. Click slightly above and to the left or right of the objects.
2. Move the cursor down diagonally from where you first clicked.
3. 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:
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:
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:

   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:
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.

How do I navigate a design?

■ Choosing either 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, double-clicking objects, or 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

   OR
   Choose View – Expand Design in the Windows Mode.

   OR
   Choose Edit – Global Find in the Windows Mode.

   The Global Find dialog box appears.

3. In the Name box, do one of the following:
   - Type the name of the net or cell to be located. For example, typing ls04 will find all
     instances of 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.
     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. To optionally restrict the search by property, perform the following actions 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
     having 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 search process begins. The Find button becomes the Stop Find button, 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 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.
    ❑ Click Close.

Navigating Nets in Your Design

The Global Navigation window helps you navigate your design. 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.

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 invoke Design Entry for the first time. When you dock the window and invoke it again or access Design Entry HDL the next time, the previous settings prevail and the window appears docked in the same location.

- You are not prompted to expand the design before docking the Global Navigation window.
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, a message box appears prompting you to expand the design:
  - If you select the "Expand without prompt" check box, Design Entry will highlight the net. As a result of selecting this check box, you will be able to highlight the nets later also even without expanding the design.
  - If you select "Yes", the design is expanded and 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.

To navigate nets in your design:

   OR
   Choose *View – Global Navigate* in the *Windows Mode*.
   The *Global Navigation* window appears.

2. Select a net in your design.

3. Right-click and choose *Highlight* from the pop-up menu.
   A message box appears prompting you to expand the design.

   ![Message Box](image)

4. Click *Yes* to expand the 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 located net instances.

5. To view a search result, select a net instance in the status area box.

When you select a result to view, the page containing the object appears with the object highlighted.

To navigate to a different net, highlight another net in your design.

### 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.

**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
- change
- copy
- delete
- exit
- move

- note
- property
- route
- select
- undo
- version

- zoom
- zoom fit (world view)
- pan down
- pan up
- pan left
- 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 stroke off
```

To turn the feature on, enter the following command:

```
set stroke on
```

Modes in Design Entry HDL

Depending on the level of design you have opened and the type of editing you want to perform on a design, Design Entry HDL offers the following three modes:

- In Hierarchy Mode
- Expanded Mode
Occurrence Edit Mode

In Hierarchy Mode

When you open a design project in Design Entry HDL, the top-level drawing is displayed and the title bar displays the design name with *in hierarchy* written within parentheses. This means that Design Entry HDL recognizes the design with all its pages and levels. If you have made any changes in the drawing, but have not saved the drawing, Design Entry HDL shows a * sign in the title bar.

In Hierarchy mode, Design Entry HDL does not allow you to perform actions such as global find and global navigate. To perform these actions, you must expand the design.

You can ascend or descend into the various pages and levels in a design. You can also choose File – Return to return to the previous page you had viewed.

**Note:** Choose View – Return in the Windows Mode.

When you open any other drawing that is not used in the design and not in the hierarchy of the schematic, Design Entry HDL opens the drawing with just the design name in the title bar. You cannot use the File – Return command in this case.

Expanded Mode

Design Entry HDL expands a design to read all pages and levels and to enable communication with other tools. Drawing expansion can take a while to complete, depending on the complexity of design.

To expand your design, choose Tools – Expand Design. When you have expanded the design, Design Entry HDL displays the design name with the word *expanded* written within parentheses. Now if you change the drawing on the root design, Design Entry HDL displays *needs expansion* written within parentheses followed by a * sign in the title bar of the root design. If you change the drawing on any other page of the expanded design, Design Entry HDL displays *needs expansion* on the title bar of the root design.

Once you have expanded your design, you can:

- View properties from net synonyms that are in the current drawing or in other drawings.
Choose **Text – Attributes** to view net synonyms in expanded drawings. Or choose **Edit – Attributes** in the **Windows Mode**.

Navigate the drawing hierarchy based on the current expansion.

Find all synonyms of a specified net or all instances of a specified cell in a hierarchical design or multipage schematic. Synonyms are used to specify that two signals with different names are actually the same. You can use the **SYNONYM** symbol in the Standard library to define a synonym.

**Occurrence Edit Mode**

While netlisting a design, Design Entry HDL stores the design data in two types of files:

- **Connectivity** (*verilog.v* or *vhdl.vhd*)
- **Properties** (*viewprps.prp*)

In a scenario where there are multiple instances of a drawing in a design and you want to specify different properties to each instance in the design, you can use occurrence properties. Occurrence properties can be specified for drawings, components, nets, and pins. The Occurrence Property File (*props.opf*) stores the properties of multiple instances. To edit this file, you have to be in the Occurrence Edit mode. Choose **Tools – Occurrence Edit** to shift to occurrence edit mode. Or choose **View – Occurrence Edit** in the **Windows Mode**. You can use the **Attributes** dialog box in the Occurrence Edit mode to edit the *props.opf* file.

**Note:** It is not recommended to use the **POWER_GROUP** property in occurrence edit mode as it may cause unpredictable results.

Issuing a command puts Design Entry HDL in that particular command mode. To end a command (exit a command mode):

- Right-click in the drawing area and choose **Done** in the pop-up menu.
- Type **;** in the console window.
- Press **Esc**.

You can copy and paste text in the console window so that you do not have to recopy commands.
Usability Enhancements in Design Entry HDL in Release 16.2

In release 16.2, several usability enhancements have been done in Design Entry HDL so that it conforms to the general usability standards of Windows-based applications, such as Adobe Reader and Microsoft Office applications. The main goal of these changes is to provide support for common Windows commands and operations in order to make Design Entry HDL more user-friendly.

The following changes have been made to Design Entry HDL for enhanced ease of use:

- Reorganized menus that conform to Windows standards
- A new-look Design Entry HDL Options dialog box similar to the Preferences dialog box of Adobe Reader
- Support for keyboard operations, such as cut, copy, paste, delete, and so on
- Introduction of the bounding box as a component selection indicator with handles to move and rotate components as in Microsoft Powerpoint
- New-look Attributes form renamed as the Properties window
- Introduction of Next, Previous, and Zoom to Points icons in the Global Navigate window
- Support for showing unconnected pins on components

Windows Mode

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, Windows mode is disabled. To enable Windows mode:

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

![Preferences](image)

**Note:** Notice that the *Ctrl+LMB Select and Drag* check box is disabled in the Windows mode implying that you need to press the Ctrl or Shift keys along with the LMB action in order to make a *stroke* on the drawing.

3. Click *OK*.

As you switch to the Windows mode a message pops up informing you that you will find differences in the menus in the Windows mode.

![Message](image)

**Note:** This message does not appear if you have customized Design Entry HDL menus and the *concepthdl_menu.txt* file is present 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 non-windows mode and the *concepthdl_menu_win.txt* file is present in the *HOME* directory, you will continue to see the menus in the Windows mode.

To see the menus in the Windows mode as per the original settings (factory 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
Reorganized Menus

In the Windows mode, the most prominent change you notice is the reorganization of menus.

All menu items are organized in such a way that they now fit into 12 menus in the Windows mode as compared to the 14 menus in the normal 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.

**Note:** You may want to print a quick reference card to have a quick look at the changes made to the menus in Windows mode.

**Important**

If you customize menus in the pre-select mode, changes are not reflected in the post-select mode. Similarly, the 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 the support for keyboard functions, such as CTRL+C for copy and CTRL+V for
You can copy objects from the Windows Clipboard to a schematic through keyboard shortcuts. Other keyboard operations supported are listed in the following table:

<table>
<thead>
<tr>
<th>Keyboard Combination</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl + C</td>
<td>Copy</td>
</tr>
<tr>
<td>Ctrl + V</td>
<td>Paste</td>
</tr>
<tr>
<td>Ctrl + A</td>
<td>Select All</td>
</tr>
<tr>
<td>Ctrl + X</td>
<td>Cut</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete</td>
</tr>
<tr>
<td>Home</td>
<td>Go to the first sheet within a module</td>
</tr>
<tr>
<td>End</td>
<td>Go to the last sheet within a module</td>
</tr>
<tr>
<td>Shift + Up Arrow</td>
<td>Move component up to the next grid point</td>
</tr>
<tr>
<td>Shift + Down Arrow</td>
<td>Move component down to the next grid point</td>
</tr>
<tr>
<td>Shift + Left Arrow</td>
<td>Move component left to the next grid point</td>
</tr>
<tr>
<td>Shift + Right Arrow</td>
<td>Move component right to the next grid point</td>
</tr>
<tr>
<td>Arrow Keys</td>
<td>Move objects in small increments</td>
</tr>
<tr>
<td>Page Up</td>
<td>Move one page up</td>
</tr>
<tr>
<td>Page Down</td>
<td>Move one page down</td>
</tr>
</tbody>
</table>

**Bounding Box on Components**

A bounding box with anchor points or handles around a component appears when you click a component on a schematic in the Windows mode. The bounding box acts as a selection indicator.
You can rotate a component by pressing $Ctrl+R$ key combination. Each time you press this key combination the component is rotated anti-clockwise.

Changes in the Global Navigate Window

The Global Navigate window displays three icons: Next, Previous, and Zoom by points. If the Zoom by points 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.
New-Look Properties Window

In 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 information of the object – component or a net. The window has icons to add, delete, and save properties and to load and save a property (.att) file.

![Attributes window](image)

Object Type and Name:

Symbol <parts_lib>I80386S

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Visible</th>
<th>Align</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LOCATION</td>
<td>U2</td>
<td>Value</td>
<td>Left</td>
</tr>
<tr>
<td>SEC</td>
<td>1</td>
<td>None</td>
<td>Left</td>
</tr>
<tr>
<td>ROOM</td>
<td>CPU</td>
<td>None</td>
<td>Left</td>
</tr>
<tr>
<td>SEC_TYPE</td>
<td>&lt;&lt;NULL&gt;&gt;</td>
<td>None</td>
<td>Left</td>
</tr>
<tr>
<td>NEEDS_NO_SIZE</td>
<td>TRUE</td>
<td>None</td>
<td>Left</td>
</tr>
<tr>
<td>MAX_POWER_DIS</td>
<td>90</td>
<td>None</td>
<td>Left</td>
</tr>
<tr>
<td>PATH</td>
<td>13P</td>
<td>None</td>
<td>Left</td>
</tr>
</tbody>
</table>

Showing Unconnected Pins on Components

This option is made available through the **Edit – Component – Unconnected Pins** menu (**Component – Unconnected Pins** menu in the non-Windows mode), which 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
```
When this option is on, all the unconnected pins on the components of a drawing are marked with pink dots.
Creating a Schematic

You have to perform the following tasks to create a schematic in Design Entry HDL:

- Create a project.
- Start Design Entry HDL.
- Create a design page.
- Add a page border.
- Add parts using component browser.
- 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 (<name of project>.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
Allegro Design Entry HDL User Guide
Creating a Schematic

- cds.lib file
- Project file (.cpm file)

For more information on creating a project using Project manager see “Creating a Project” on page 54.

Starting Design Entry HDL

After you create the 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.

Click the Design Entry icon to invoke Design Entry HDL.

About Drawing Names

The drawing name identifies your design. Drawing names can have several fields separated by periods:

```
```

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.
Creating a Schematic

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,

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 schematic:

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 for documenting 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.

The Standard library supplied by Cadence contains the following symbols that can be used as page borders:

<table>
<thead>
<tr>
<th>Page Name</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Size Page</td>
<td>8.5 x 11 inch border</td>
</tr>
<tr>
<td>B Size Page</td>
<td>11 x 17 inch border</td>
</tr>
<tr>
<td>Cadence A Size Page</td>
<td>8.5 x 11 inch border with the Cadence Design Systems logo</td>
</tr>
<tr>
<td>Cadence B Size Page</td>
<td>11 x 17 inch border with the Cadence Design Systems logo</td>
</tr>
<tr>
<td>C Size Page</td>
<td>17 x 22 inch border</td>
</tr>
<tr>
<td>D Size Page</td>
<td>22 x 34 inch border</td>
</tr>
<tr>
<td>E Size Page</td>
<td>34 x 44 inch border</td>
</tr>
<tr>
<td>F Size Page</td>
<td>44 x 68 inch border</td>
</tr>
</tbody>
</table>

### 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.
   The Component Browser 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.

**Using Component Browser**

**Overview**

Allegro Design Entry HDL lets you search and select library parts defined in the `<projectname>.cpm` file of your design. Enhancing the Component Browser of earlier versions, Design Entry HDL 15.5 or later lets you browse libraries and cells and helps you add and replace components in a schematic.

Component Browser enables you to perform the following tasks:

- Search for parts
- Browse for parts
- View details of parts including symbols and footprint
- Add and replace parts

**Note:** Before using the Component Browser, ensure that your metadata has been generated in the current release. Cadence recommends you not to use metadata created using earlier versions.

**Launching Component Browser**

You can launch Component Browser in two modes using:

- Design Entry HDL
- Standalone Component Browser utility

**Using Design Entry HDL**

Within Design Entry HDL, invoke Component Browser using any of the following methods.

- Choose *Component – Add.*
- Choose *Component – Replace*

**Note:** The choice of the method depends on the tasks you wish to perform. For example, you might invoke Component Browser for adding or replacing parts.
Using the Standalone Component Browser utility

1. Open the command prompt window.
2. Type `UniversalBrowser` and press Enter.
   The Component Browser window appears.
3. Specify the design project in the `Project Path` field.
4. Click `OK`.
   The Component Browser window appears.

Searching Parts

You can search classifications or libraries for parts. You can specify attributes of a part as the parameters of your search. The attributes can include `PART_NUMBER`, `DESCRIPTION`, `JEDEC_TYPE`, `PACK_TYPE`, `VENDOR`, `STATUS`, and all physical properties.

Searching Parts by Family or Allegro Design Entry HDL Libraries

To search for parts by classification or project libraries:

1. In the `Tree View` pane of the Component Browser, select either a classification node or a library node.

   **Note:** For a quick and specific part search in classifications, you can select multiple cells from one or many classification nodes by using Shift + Click and Shift + Ctrl. Similarly, you can select multiple cells from one or many library nodes (cells).

   **Note:** The nodes you select in the `Tree View` pane appear in form of a path at the top of the Search/Details pane.

   **Caution**

   *Cadence recommends you not to perform search by selecting the root nodes, Libraries or Classifications. This will trigger an exhaustive component search which will take long to execute.*

   **Important**

   You cannot select a combination of classification and library nodes.
2. In the *Search/Details* pane, enter an attribute or select the name of the attribute using the first drop-down list box.

3. Enter or select a logical operator (<, >, =, =>, =!, or <=) from the second drop-down list box.

4. In the third drop-down list box, enter a value (numeric or string) for the selected attribute (in the first drop-down list box). You can also use the asterisk (*), a wildcard character, to make the value generic.

Many a times, physical properties of components are expressed in units representing power-of-10 multiples. In such cases, an effective search requires you to include units along with the values.

Using Component Browser, you can perform search not only by providing values for attributes but also including power-of-10 multiples with it. Some of the power-of-10 multiples you can use are: pico (10^{-12}), nano (10^{-9}), micro (10^{-6}), milli (10^{-3}), kilo (10^{3}), mega (10^{6}), giga (10^{9}), and tera (10^{12}).

The Component Browser is equipped to automatically handle the inter-conversion of units and display all the components meeting the search criteria. For example, if you want to search for a component with resistance less than 1000 ohms, you can specify <= (logical operator) in the second drop-down list box, and 1K in the third drop-down list box. This will provide search results containing components with resistance in milliohms and micro ohms as well.

To get the desired results, ensure that you specify the correct power-of-10 multiple acronym (such as K= Kilo, Meg= Mega, m/M=mili, p=pico, and so on) after the value. For searches requiring M to be interpreted as Meg, make sure that M is not followed by any unit. For example, you should use “10M” instead of “10MOhm.”

**Important**

Defining a string-based search that contains wildcard (*) characters, and using either the equality (=) or inequality (!=) operator only provides the desired results.

Using other logical operators (<, >, =>, and <=) will evaluate the search by using the equivalent ASCII codes of the string moving from left to right.

For example, if you perform a search, where ‘Description’ is ‘less than’ a string say, ‘ZY’, then ‘Z’ will be converted into its equivalent ASCII code (90) and ‘Y’ will be converted into its equivalent ASCII code (89). The search will be performed for all numeric values less than 9089. Finally, in search results, the strings whose ASCII equivalents are less 9089 will appear. Since, this search mechanism does not work with wildcard characters. Hence, you must not use wildcard characters with these logical operators.
5. You can specify multiple attributes in your part search. To do so, click More. A new row of attribute with three drop-down list boxes appears.

6. Perform steps 2, 3, and 4 to specify the search parameters for the new attribute(s). You can specify a maximum of six attributes.

7. To remove an attribute row, click Fewer. In case of multiple rows, clicking Fewer removes the last row of the attribute.

   **Note:** To remove an attribute other than the last attribute, delete the attribute name from the corresponding first drop-down list box. This will exclude the attribute from the search criteria.

8. Select the Match All radio button to perform a search that meets all the specified parameters. It is similar to using the AND operator. To perform a search that meets any of the criteria, select the Match Any radio button. It is similar to using the OR operator.

9. Click Search.

   **Note:** To stop a search that is in progress, click Stop Search in the search progress window.

10. The search results appear in the Search Results pane. The Search Results pane consists of physical part rows for all the cells that satisfy the search criteria.

   **Note:** Grayed cells for a property indicate that they do not have any value associated with them.

   **Note:** To reload the part table for a part, right-click on the part table row and choose Reload Part Tables from the pop-up menu.

**Browsing Parts by Allegro Design Entry HDL Libraries**

Additionally, if you prefer to browse the libraries and cells for parts, you can do so using the Component Browser. To browse for a part by Allegro Design Entry HDL libraries:

1. In the Tree View pane of the Component Browser, select the Browse Libraries node.

   The Search/Details pane appears containing the Library list box and the Cells list box. The Library list box contains all the Allegro Design Entry HDL libraries listed in the cds.lib file of your schematic.

   The Cells list box contains the cells available in a library you select. By default, all the Allegro Design Entry HDL libraries and their cells appear in the Search/Details pane. Alternatively, select a library in the Library list box to display corresponding cells in the Cells list box.

   **Note:** To select all the libraries under the Library list box, press Ctrl+A.
2. Select a cell in the *Cells* list box.

3. To filter the cells listed in the *Cells* list box, enter the initials of the cell name in the text box above the *Cells* list box. You can also use wildcard characters (*) to filter the cell name.

4. The physical part table rows belonging to the selected cell(s) appear in the *Search Results* pane.

**Refreshing Libraries**

You can update the libraries while browsing for parts in the Component Browser. This is helpful when you add or remove cells from libraries, and want the changes to be reflected immediately in the Component Browser.

To refresh all the libraries that appear in the Tree view, choose *View – Refresh Library*. However, to refresh a specific library, in the *Library* list box, right-click the library and choose *Refresh Lib* from the pop-up menu.

However, if you're copying a cell from an external library to your design library, run the command, `lib <library name>` in the console window, select the library in the *Library* list box of Component Browser, and choose *Refresh Lib* from the RMB pop-up menu.
Refining Search Results

You can refine search results by entering a filter string in the appropriate header locations of the Search Results pane. To do so:

1. Enter the filter string and press Enter.

2. The physical part table rows meeting the filter string appear in the Search Results pane.

   Example: If you have four part table rows for a appearing in the Search Results pane and you want to filter out and use only those physical parts with a value of 0.1, enter 0.1 in the Filter field (corresponding to the Value column) and press enter. Component Browser displays only those physical part table rows where part type is 0.1.

Viewing Detailed Part Information

You can view detailed part information by including the part into the Tree View pane and the Search/Details pane. To view the part information, click a part table row in the Search Results pane. Alternatively, right-click a part table row and choose Show Details from the RMB menu. A tab with the part number label beside the standard Part tab appears. Viewing the part information includes:

- Viewing the properties
- Viewing the symbols and footprints
Viewing the Properties

You can view the key and injected properties for a part. To do so, click the Properties tab in the Search/Details pane.

The property information appears under the following column headers: Name, Value, and Type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part Name</td>
<td>CAP</td>
<td>KEYPROPERTY</td>
</tr>
<tr>
<td>PACK_TYPE</td>
<td>SMDCAP</td>
<td>KEYPROPERTY</td>
</tr>
<tr>
<td>VALUE</td>
<td>3300pf</td>
<td>KEYPROPERTY</td>
</tr>
<tr>
<td>TOLERANCE</td>
<td>10%</td>
<td>KEYPROPERTY</td>
</tr>
<tr>
<td>PART_NUMBER</td>
<td>cap789</td>
<td>INJECTED_PROPERTY</td>
</tr>
<tr>
<td>VALUE</td>
<td>3300pf</td>
<td>INJECTED_PROPERTY</td>
</tr>
<tr>
<td>TOLERANCE</td>
<td>10%</td>
<td>INJECTED_PROPERTY</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>cap 3300pf 10%</td>
<td>INJECTED_PROPERTY</td>
</tr>
<tr>
<td>JEDEC_TYPE</td>
<td>SM_0805</td>
<td>INJECTED_PROPERTY</td>
</tr>
</tbody>
</table>

Viewing the Symbols and Footprints

Component Browser also lets you view the schematic symbol and the footprint of a part. To do so:

1. Click a part table row in the Search Results pane; the <Part Name> tab appears. By default, when you click the <Part Name> node (the root node in the Tree View pane), the schematic symbol or footprint model of the part appears in the Search/Details pane.

   **Note:** The availability of schematic and footprint viewers depend on the model viewer configurations for the Component Browser.
2. Alternatively, to view a symbol, select a schematic model in the *Models – Schematic* node.

3. Similarly, to view the footprint, select a footprint model in the *Models – Footprint* node. The following image shows the *Graphics* tab containing both viewers, Symbol and Footprint.

**Note:** In order to display the footprints correctly, ensure that:

- You have defined psmpath and padpath for the footprints using the `enved` utility.
- Your footprints contain the quickview data that the footprint viewer uses. To generate this data, use the `qvupdate` utility.

**Note:** If you have defined the psmpath and padpath in the cpm file of your project, then only those paths will be read for footprints. However, if they are not defined in the cpm, only then the paths will be read from the environment settings.

4. For viewing schematic symbols, you can use the icons available at the top of the viewer to control the zoom functions. Table 5-1 lists the zoom functions.

**Note:** Component Browser displays the symbol versions (in the *Symbol* drop down box) depending on the PACK_TYPE of the part selected. Also, it shows the highest symbol index.
which is associated with the selected PACK_TYPE. However, it does not show the total number of symbols available in the cell.

**Table 5-1 Zoom functions**

<table>
<thead>
<tr>
<th>Using the Icon...</th>
<th>Lets you...</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Zoom In" /></td>
<td>Zoom into the symbol</td>
</tr>
<tr>
<td><img src="image" alt="Zoom Out" /></td>
<td>Zoom out of the symbol</td>
</tr>
<tr>
<td><img src="image" alt="Zoom Fit" /></td>
<td>Fit the symbol in the screen</td>
</tr>
<tr>
<td><img src="image" alt="Zoom Points" /></td>
<td>Stretch a rectangle around the area you want to zoom in</td>
</tr>
</tbody>
</table>

**Adding a Part**

To add a part:

1. In Allegro Design Entry HDL window, choose **Component – Add**.
   
The Component Browser appears.

2. Search for a part.
   
   For information on how to search for parts, see **Searching Parts**.

3. Click a part to choose in the **Search Results** pane.
   
The <Part Name> tab appears with the part information.

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 the Component Browser.

2. Search for a part. For information on how to search for parts, see *Searching Parts*.

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 the Component Browser. Alternatively, choose *Replace* from the RMB pop-up menu.

3. Select the component that should replace all the components.
Old Vs. New Component Browser - A Comparison

The following table lists the main differences between the old (shipped in releases prior to SPB 15.5) and the new Component Browser.

<table>
<thead>
<tr>
<th>Old Component Browser</th>
<th>New Component Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displays the classifications and library views as separate tabs.</td>
<td>Displays the classifications and library views as separate folders in a tree-like hierarchy.</td>
</tr>
<tr>
<td>Lets you browse and select components to add in the design</td>
<td>Lets you search, browse and select components to add using the values of the desired attributes.</td>
</tr>
<tr>
<td>Does not support the display of part information.</td>
<td>Displays information such as key properties, injected properties, attribute values, and classification hierarchy for the part selected. Also, lets you preview footprints and symbols.</td>
</tr>
<tr>
<td>Displays physical part table file information using the Physical Part Table Filters dialog box.</td>
<td>Displays the physical part table file information meeting the search criteria in the Search Results pane (Component Browser window).</td>
</tr>
<tr>
<td>Lets you open multiple instances of the Component Browser window to view the contents of more than one library at once.</td>
<td>Does not let you open more than one instance of the Component Browser window; a hierarchical view of the libraries eliminates the need to open multiple windows of the Component Browser.</td>
</tr>
</tbody>
</table>

**Note:** For more information on the user interface changes in the new Component Browser, see the chapter *Component Browser User Interface* in the *Allegro Design Workbench Interface Overview* guide.

Configuring the 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` does not appear on the schematic.
Using the property options feature, you can:

- Configure various physical property options for parts.
- Control the appearance of PPT columns in the Component Browser 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 invokes, when you:

- Click PPT Options in the Modify Component dialog box.
- Choose PPT Options from the pop-up menu that appears when you right-click on a PPT row in the Component Browser window.

**Defining Physical Property Options**

1. Select a physical property listed under the Property Name column, and click Up or Down to modify the order in which the property appears in the Property Options dialog box and on the schematic.

2. For example, if you move the PACK_TYPE property to the lowest level in the list, the PPT row in the Component Browser 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.

3. Use the Filter Pattern column to filter physical property values based on the string you enter.

4. For example, if you want the Search Results pane in the Component Browser window to display only that row of the PPT 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.

5. 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.

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.

6. 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.

⚠️ **Important**

You can perform numeric sort on property columns in the Search Results pane of the Component Browser provided that all the part table rows have the same part name.

⚠️ **Important**

You cannot use numeric sort on a property value (in the part tables) that contains strings.

7. Select the *Hide Column* check box to hide the selected property column in the *Component Browser* window.

8. Click *OK* or *Apply* to apply the options you define.

9. Click *Save* to save all options to the `pptoptionset.dat` file. You can load these options in later sessions using the *Load* button. To save all the options in a new file, click *Save As*.

10. Click the *Load* button to load the option sets and the options from the `pptoptionset.dat` file.

### Sample `ppt_optionset.dat` File

The syntax of the `ppt_optionset.dat` file is as follows:

```plaintext
( "VERSION 3.0"
```
OPTION_SET_ATTRIBUTES has the following four fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>optionsetname</td>
<td>A character string to identify the option set</td>
</tr>
<tr>
<td>totalprops</td>
<td>Total number of properties in the option set</td>
</tr>
<tr>
<td>Keyprops</td>
<td>Number of key properties in the set</td>
</tr>
<tr>
<td>Injprops</td>
<td>Number of injected properties in the set</td>
</tr>
</tbody>
</table>

PROPERTY_ATTRIBUTES has the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Lets you</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proptype</td>
<td>Define the type of property. Use 1 for key property and 0 for the injected property.</td>
</tr>
<tr>
<td>Propname</td>
<td>Specify the name of the property.</td>
</tr>
<tr>
<td>Filterpattern</td>
<td>Define a filter pattern to be used for the property.</td>
</tr>
<tr>
<td>SortType</td>
<td>Use 1 to specify numeric sort; use 0 to ensure that numeric sort is not applied</td>
</tr>
<tr>
<td>SortOrder</td>
<td>Specify the order of properties in the optionset. 0 is used for the first order.</td>
</tr>
<tr>
<td>Annotate</td>
<td>Define 0 for not using annotation, 1 for name, 2 for value, 3 for both, and 4 for invisible.</td>
</tr>
<tr>
<td>Visible</td>
<td>Use 1 to make a property visible in the Component Browser window; use 0 to hide a property in the Component Browser window.</td>
</tr>
<tr>
<td>Propcol</td>
<td>Specifies the column number of the property. Use 0 for the property in the first column.</td>
</tr>
</tbody>
</table>

Given below is a sample ppt_optionset.dat file with the following option sets:
20L10
    DG419

( "VERSION 3.0"

( ("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.

<table>
<thead>
<tr>
<th>Field</th>
<th>20L10</th>
<th>DG419</th>
</tr>
</thead>
<tbody>
<tr>
<td>optionsetname</td>
<td>20L10</td>
<td>DG419</td>
</tr>
<tr>
<td>totalprops</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Keyprops</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Injprops</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

The following table explains the first row of PROPERTY_SET_ATTRIBUTES for the two option sets.

<table>
<thead>
<tr>
<th>Field</th>
<th>20L10</th>
<th>DG419</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proptype</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Copying an Option Set

To copy an option set:

1. Select a name in the Option Set list.
2. Change the name in the Name/Pattern field.
3. Click Add.

Removing an Option Set

To remove an option set:

1. Select a name in the Option Set list.
2. Click Remove.

Note: You cannot remove the current option set that you are working with.

Customizing PPT 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 Component Browser, select a part.
2. Right-click a row in the Search Results window and choose PPT Options from the pop-up menu.
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 on 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 search is performed on a part for which PPT 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

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

2. Right-click and choose Signal Name… from the pop-up menu.
3. Type a signal name in the Signal Name box.
4. Click OK.
5. Click wherever you want the wire to bend, or click a pin on another component.

Auto-Routing a Wire

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.
To name a wire when you draw it:

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:

1. Names must start with a letter.
2. Names cannot be VHDL and Verilog keywords.
3. 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*.

& When the *Multi-format Vectors* option in the *General* tab of the *Design Entry HDL Options* dialog box is selected, an ampersand represents concatenation and needs an operand on both sides. When the *Multi-format Vectors* option is not selected, 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, select the *Multi-format Vectors* option in the *General* tab of the *Design Entry HDL Options* dialog box. It 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 177.

- **G** - For *Global*. Implies the signal is a global signal. For more information, see [Global Signals](#) on page 179.

- **I** - For *Interface*. Implies the signal is a port. Cadence recommends that you use the port symbols INPORT, OUTPORT, or INOUT instead of the \ suffix. The \ 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 169.

- **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.

( )  
When the *Multi-format Vectors* option in the *General* tab of the *Design Entry HDL Options* dialog box is selected, the parentheses indicate that the signal is a bus. Must be matched correctly and contain either a parameter or an integer. When the *Multi-format Vectors* option is not selected, the parentheses have no special meaning and can be used anywhere in the signal name.
When the *Multi-format Vectors* option in the *General* tab of the *Design Entry HDL Options* dialog box is selected, the square brackets indicate that the signal is a bus. Must be matched correctly and contain either a parameter or an integer. When the *Multi-format Vectors* option is not selected, the square brackets have no special meaning and can be used anywhere in the signal name.

* When used at the end of a signal name or a pin name, an asterisk 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 175.

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. For more information, see *Specifying the Assertion Level of Pins and Signals* on page 175.

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

`[]` When the *Multi-format Vectors* option in the *General* tab of the *Design Entry HDL Options* dialog box is selected, the square brackets indicate that the signal is a bus. Must be matched correctly and contain either a parameter or an integer. When the *Multi-format Vectors* option is not selected, the square brackets have no special meaning and can be used anywhere in the signal name.

* When used at the end of a signal name or a pin name, an asterisk 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 175.

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. For more information, see *Specifying the Assertion Level of Pins and Signals* on page 175.
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

<table>
<thead>
<tr>
<th>Subscript</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;31..0:2&gt;</td>
<td>30 28 26 ... 6 4 2 0</td>
</tr>
<tr>
<td>&lt;11..0:4&gt;</td>
<td>8 4 0</td>
</tr>
<tr>
<td>&lt;9:1:3&gt;</td>
<td>7 4 1</td>
</tr>
<tr>
<td>&lt;0..31:-1&gt;</td>
<td>31 30 29 ... 3 2 1 0</td>
</tr>
<tr>
<td>&lt;15..0:20&gt;</td>
<td>0</td>
</tr>
<tr>
<td>&lt;0..6:-2&gt;</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>&lt;0..7:-2&gt;</td>
<td>0 2 4 6</td>
</tr>
</tbody>
</table>
Consider the following examples:

![Diagram showing B<60..0:4> and B<15..0>]

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].

![Diagram showing Z<31..0:2> and A<20..45>]

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:

```plaintext
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.

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.

   Note: Spaces are not permitted in the values of LOCATION and $LOCATION properties. If you use spaces while specifying LOCATION and $LOCATION properties, the Property dialog box exits with an error message. Similarly, colon (:) should not be used in reference designator values, otherwise it returns an error condition.

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:

   \[ \text{property} \]

   For more information on strokes and a list of available stroke patterns, see Running Commands with Strokes on page 126.
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*.

**To add occurrence properties**

2. Choose *Tools – Occurrence Edit* or type `attribute` in the console window.
3. Choose *Text – Attributes*.
4. Select an object.
   - The object you select is highlighted, and the *Attributes* dialog box displays attributes for the object.
5. In the *Attributes* dialog box, click *Add*.
   - An empty row appears.
6. Enter a property name in the *Name* box and a property value in the *Occurrence Value* box.
   - You can also replace the occurrence value for an existing property. However, you cannot change or delete the schematic values that are shaded gray.
7. Adjust property visibility and alignment as needed.
8. Click *OK*.
9. Click near the object to indicate where to display the property information.
By default, Design Entry HDL displays only the property value. Choose Text – Property Display to modify how properties are displayed.

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.
   
   The Component Browser 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 the Component Browser.
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 \texttt{STD\_LOGIC} for scalar ports and \texttt{STD\_LOGIC\_VECTOR} for vectored ports. The default Verilog logic type for all ports in Design Entry HDL schematics is \texttt{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 197 and Setting the VHDL Logic Type for Ports and Signals on page 200 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 \texttt{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 205.

- 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 \texttt{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 206.

- 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 207.

### 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.

<table>
<thead>
<tr>
<th>Formal Port</th>
<th>Actual Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>IN, INOUT, BUFFER</td>
</tr>
<tr>
<td>OUT</td>
<td>OUT, INOUT</td>
</tr>
<tr>
<td>INOUT</td>
<td>INOUT</td>
</tr>
<tr>
<td>BUFFER</td>
<td>BUFFER</td>
</tr>
</tbody>
</table>

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 172
- Specifying the Assertion Level of Pins and Signals on page 175
- Creating an Alias for a Signal on page 175
- Global Signals on page 179
- Unnamed Signals on page 186
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.

```vhdl
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:

```vhdl
global VCC_ARROW: std_logic ;
BEGIN
  VCC_ARROW  <= '1' ;
```

The vhdl.vhd file then contains the initial value for signal VCC_ARROW:
**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:

```
signal UNNAMED_1_54ALS00_I1_Y: std_logic;
```

You can see that the `vhdl.vhd` file does not contain the initial value for the unnamed signal.

---

**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`.
   
   The `Component Browser` 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 the `Component Browser`.

6. Attach the `SLASH` symbol to the net you want to size.

7. From the `Text` menu, choose `Attributes`. 

---
8. Click on the \texttt{SLASH} symbol to display the \textit{Attributes} dialog box.

9. Specify the size of the net by changing the value of the \texttt{SIZE} property. The default value of the \texttt{SIZE} property is 1.

\textbf{Note:} If you are using a \texttt{SLASH} symbol only for documentation and the width of the signal is already set, the value of the \texttt{SIZE} property on the \texttt{SLASH} symbol must match the actual width of the signal.

10. Click \textit{OK} to save the changes and close the \textit{Attributes} dialog box.

\section*{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 \texttt{\_N} or \* suffix or a \texttt{-} prefix. Cadence recommends that you use a \texttt{\_N} suffix to indicate a low-asserted signal or pin.

\textit{To specify a low-asserted signal or pin}

\begin{itemize}
  \item Suffix \texttt{\_N} or \* to the signal name or pin name.
\end{itemize}

\textbf{Example}

\begin{itemize}
  \item \texttt{ENABLE\_N} is an active low scalar signal.
  \item \texttt{DATA<15..0>_N} is an active low vector signal.
  \item \texttt{ENABLE\*} is an active low scalar signal.
  \item \texttt{DATA<15..0>*} is an active low vector signal.
\end{itemize}

Any signal or pin that does not have the \texttt{\_N} or \* suffix is assumed to be an active high signal.

\textbf{Note:} \texttt{DATA\_N<15..0>} is also supported for low vector signal names.

\section*{Creating an Alias for a Signal}

Use the \texttt{ALIAS} symbol in the Standard library to specify another name for a signal.
For example, the following ALIAS

```
instr(7:5) ALIAS opcode(0 to 2)
```

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.
   
   The Component Browser 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.

```
 ALIAS
```

5. Close the Component Browser.
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.
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.

Important

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 synonymed, 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 synonymed to any other signal or if it is selected as the base signal.

You may want the name of a particular aliased or synonymed signal to be passed to PCB Editor as the physical net name. You can force Design Entry HDL to use a particular aliased or synonymed signal by declaring it as a base signal, as below:
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.

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 synonymed signals at higher schematic levels always supersede aliased or synonymed 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 synonymed signal at the highest schematic level at which the signal exists.

Note: If a global signal is aliased or synonymed 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 178.

Note: If a power symbol vcc is aliased or synonymed 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 179.

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 synonymed 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 `glbl` 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:
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 synonymed 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.
b. Type **CLOCK** in the *Signal1* field and **SWITCH** in the *Signal2* field next to the **CLOCK** signal.

![Design Entry HDL Options](image)

The *Allowed Global Shorts* list allows you to add pairs of global signals that you want to remain shorted in the design. Click 

![Add Row](image) to add a row. Select a row and click 

![Delete Row](image) to delete the row.

3. 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
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 454.

### Supported Syntax in the Allowed Global Shorts List

The syntax that is supported in the *Allowed Global Shorts* list is explained below using examples:

<table>
<thead>
<tr>
<th>Example</th>
<th>Enter the following in Signal1 field of the Allowed Global Shorts list</th>
<th>Enter the following in Signal2 field of the Allowed Global Shorts list</th>
</tr>
</thead>
<tbody>
<tr>
<td>■ To allow shorting of a scalar global signal <em>CLOCK\G</em> with another scalar global signal <em>DATA\G</em></td>
<td><em>CLOCK</em></td>
<td><em>DATA</em></td>
</tr>
<tr>
<td>■ To allow shorting of any bit of a vectored global signal <em>CLOCK&lt;3..0&gt;\G</em> with any bit of another vectored global signal <em>DATA&lt;4..0&gt;\G</em></td>
<td><em>CLOCK</em></td>
<td><em>DATA</em></td>
</tr>
</tbody>
</table>
To allow shorting of two vectored global signals INT<3..0>G and DATA<4..7>G that have the same width.

This means that Design Entry HDL allows shorting of bit INT<3> with bit DATA<4>, INT<2> with DATA<5>, INT<1> with DATA<6> and INT<0> with DATA<7>.

**Note:** Design Entry HDL displays an error message if the width of the shorted signals specified in the *Allowed Global Shorts* list is not the same.

To allow shorting of the third bit of a vectored global signal CLOCK<3..0>G with the fifth bit of another vectored global signal DATA<7..0>G.

The shorting of any other bit of CLOCK with any other bit of DATA, except CLOCK<3> and DATA<5> is reported as a global signal short error.

To allow shorting of any bit of a vectored global signal DATA<3..0>G with a scalar global signal VCC\G.

You can also use the following syntax to allow shorting of any bit of a vectored global signal CLOCK<3..0>G with a scalar global signal DATA\G.

To allow shorting of a bit of global signal CLOCK(3)\G with another bit of a global signal DATA[5]\G.

<table>
<thead>
<tr>
<th>Example</th>
<th>Enter the following in Signal1 field of the Allowed Global Shorts list</th>
<th>Enter the following in Signal2 field of the Allowed Global Shorts list</th>
</tr>
</thead>
<tbody>
<tr>
<td>To allow shorting of two vectored global signals INT&lt;3..0&gt;G and DATA&lt;4..7&gt;G that have the same width</td>
<td>INT&lt;3..0&gt;</td>
<td>DATA&lt;4..7&gt;</td>
</tr>
<tr>
<td>To allow shorting of the third bit of a vectored global signal CLOCK&lt;3..0&gt;G with the fifth bit of another vectored global signal DATA&lt;7..0&gt;G.</td>
<td>CLOCK&lt;3&gt;</td>
<td>DATA&lt;5&gt;</td>
</tr>
<tr>
<td>To allow shorting of any bit of a vectored global signal DATA&lt;3..0&gt;G with a scalar global signal VCC\G.</td>
<td>DATA&lt;3..0&gt;</td>
<td>VCC</td>
</tr>
<tr>
<td>You can also use the following syntax to allow shorting of any bit of a vectored global signal CLOCK&lt;3..0&gt;G with a scalar global signal DATA\G.</td>
<td>CLOCK</td>
<td>DATA</td>
</tr>
<tr>
<td>To allow shorting of a bit of global signal CLOCK(3)\G with another bit of a global signal DATA[5]\G.</td>
<td>CLOCK&lt;3&gt;</td>
<td>DATA&lt;5&gt;</td>
</tr>
</tbody>
</table>
Important

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 `pxl.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:
Suppose that the Multi-format Vectors check box in the General tab of the Design Entry HDL Options dialog box is deselected and you have a scalar global signal CLOCK(3)\G aliased or synonymed to another scalar global signal INT[4]\G in your schematic. If 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 now select the Multi-format Vectors check box, 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 Vectors check box is selected).

For more information on the Multi-format Vectors option, see Multi-format Vectors.

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 $pinname are substituted with the actual values. Therefore, the netname appears as UNS11$CAPACITOR$157$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 (:).
  
  If you select the Multi-format Vectors option in the General tab of the Design Entry HDL Options dialog box, you can also use a comma (,) or an ampersand (&) to concatenate two signals. When the Multi-format Vectors option is not selected, ampersands and commas do not represent concatenation and have no special meaning in a signal name. For more information, see Naming Signals on page 161.

  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)`.

```
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*.

   The *Component Browser* 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 the *Component Browser*.

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 \(a(0 \to 9)\), the scalar signal \(b\), and the vectored signal \(c(9 \downarrow 0)\).

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:

\[ a(0 \to 9) \& b \& c(9 \downarrow 0) \]

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.

```
```

■ Use the CONCAT10 symbol.

For more information, see CONCAT symbols (standard libraries).

■ Use a combination of text and graphics.

```
GND & GND
```

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 x 5 x 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*.
   
   The *Component Browser* 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, click the right mouse button and choose *Version*. The next version of the component is displayed. Repeat this step until the version you want is displayed.

   **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 the *Component Browser*.

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.
   
   The Component Browser 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, click the right mouse button and choose Version. The next version of the component is displayed. Repeat this step until the version you want is displayed.
   
   **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 the Component Browser.

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 \texttt{addr<15..0>} into four signals, name the output signals \texttt{addr<8..0>}, \texttt{addr<10..9>}, \texttt{addr<12..11>}, and \texttt{addr<15..13>}. 
**Example**

Version 4 of the 4 MERGE symbol below is used to separate a vectored signal into four signals.

![Diagram](image1.png)

The following example shows how the 4 MERGE symbol given above is used to separate a vectored signal into many signals.

![Diagram](image2.png)

**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 174.

- 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 do not select the Multi-format Vectors option in the General tab of the Design Entry HDL Options dialog box, you must be careful about the syntax you use to specify the width of the signal. See Naming Signals on page 161 for more information.

**To create a slice graphically**

1. In Design Entry HDL, choose Component – Add.

   The Component Browser 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 207 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 172.

**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 207 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 `<>`.

  **Note:** You can also use ( ) or [ ] if you select the `Multi-format Vectors` option in the `General` tab of the `Design Entry HDL Options` dialog box.

- Specify a descending range with a colon (:) or an 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 (:) or an ellipsis (..), 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 `downto`
- `<0:10>` Colon (:) also works like `to`
- `<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. `to` must be an ascending range.
- `<0 downto 10>` Illegal. `downto` must be a descending range
**Important**

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:

```vhdl
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:

```vhdl
definition 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:

```vhdl
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 INPORT or OUTPORT 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-vectored 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-vectored 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:

\[ \text{PATH} = \text{I3}<15..0> \]

The generated instance labels in Verilog and VHDL are:

\[ \text{I3\_GEN\_15} \]
\[ \text{I3\_GEN\_14} \]
\[ \text{I3\_GEN\_13} \]
\[ \ldots \]
\[ \text{I3\_GEN\_0} \]

If you have selected the Multi-format Vectors option in the General tab of the Design Entry HDL Options dialog box, 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]

X-Replication

X-replication, also called size replication, is a powerful mechanism for creating concise representations of a design. X-replication uses the SIZE property to duplicate parts and specify how they will be connected. To use X-replication, you must add a DEFINE symbol to the design and at least one net on the design must have an X variable. The DEFINE symbol must have the X_FIRST, X_STEP, and X_SIZE properties on it.
The number of times the design is instantiated depends on the values of the X_SIZE, X_FIRST and X_STEP properties and on the expression in which the x variable is used. In the first instance of the design, the value of x is X_FIRST. In the second instance, the value of x is X_FIRST+X_STEP, and so on until x is less than or equal to the value of X_SIZE. The number of times x can be evaluated this way is the number of times the design is instantiated.

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.

Using X-Replication

1. Add a DEFINE symbol to the schematic. The DEFINE symbol must have the following three properties on it: X_SIZE, X_FIRST, and X_STEP.

2. Specify the values of these properties. The values must be constants; they cannot be variables.

Rules for Using X-Replication

- The design must have a DEFINE symbol and at least one net on the design must have the X variable.
- There can be only one DEFINE symbol in a drawing.
- The DEFINE symbol must have these three properties: X_SIZE, X_FIRST, and X_STEP.
- The values of the X_SIZE, X_FIRST, and X_STEP properties must be constants; they cannot be variables.

Saving a Design

Design Entry HDL displays a * sign in the title bar to show that the current page needs to be saved. In the hierarchy mode, you see [in hierarchy]*, and in the expanded mode, you see [needs expansion]*.

Note: In the occurrence edit mode, Design Entry HDL does not display * sign to indicate that the drawing needs to be saved.
When you save the design, Design Entry HDL writes the current design on the disk. The * sign disappears from the title bar.

In the expanded mode, if you change any drawing of the expanded design, Design Entry HDL displays "needs expansion" on the title bar of the page. This signifies that the design needs to be re-expanded. In the occurrence edit mode, if you make any change and then come back to the expanded mode, Design Entry HDL displays "needs expansion" on the title bar. If you re-expand the design, the message on the title bar changes to "expanded."

**Note:** When you package the design using Packager-XL, Packager-XL adds certain soft properties to the design, but Design Entry HDL does not display "needs expansion" on 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.

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 211

■ Recovering a Drawing on page 212

■ Reverting to the Previous Saved Version of a Drawing on page 212

Opening a Drawing


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.

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:


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.
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.

  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>.

  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.

<table>
<thead>
<tr>
<th>Bus Name</th>
<th>Associated Signal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A&lt;3..0&gt;</td>
<td>A&lt;3&gt;, A&lt;2&gt;, A&lt;1&gt;, A&lt;0&gt;</td>
</tr>
<tr>
<td>A&lt;0..3&gt;</td>
<td>A&lt;0&gt;, A&lt;1&gt;, A&lt;2&gt;, A&lt;3&gt;</td>
</tr>
<tr>
<td>A&lt;0&gt;</td>
<td>A&lt;0&gt;</td>
</tr>
<tr>
<td>A&lt;7..0:2&gt;</td>
<td>A&lt;6&gt;, A&lt;4&gt;, A&lt;2&gt;, A&lt;0&gt;</td>
</tr>
</tbody>
</table>

**Drawing a Wire Manually**

To draw a wire without naming it
1. Choose **Wire – Draw**.

2. Click a pin on a component.

3. To change the orientation of the wire as you draw it, click right 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

1. Choose **Wire – Draw**.

2. Click right and choose **Signal Name…** from the pop-up menu.

3. Type a signal name in the **Signal Name** box.

4. Click **OK**.

5. Click wherever you want the wire to bend, or click a pin on another component.

**Tips for Drawing Wires**

<table>
<thead>
<tr>
<th>To...</th>
<th>Do this...</th>
</tr>
</thead>
<tbody>
<tr>
<td>End a wire at a pin, dot, or other wire</td>
<td>Click left</td>
</tr>
<tr>
<td>Snap a wire to the nearest pin</td>
<td>Click Ctrl+right</td>
</tr>
<tr>
<td>Change the bend of the wire</td>
<td>Click Ctrl+left or click right and choose <strong>Orientation</strong> from pop-up menu</td>
</tr>
<tr>
<td>End a wire in a free space</td>
<td>Click twice at the final point</td>
</tr>
</tbody>
</table>

**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**

2. Begin drawing a wire.
3. Click right 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**

2. Press *Ctrl +* click right.
   - 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
1. Choose *Wire – Signal Name*.

   The *Signal Name* dialog box appears.

2. Type one or more signal names on separate lines.

3. Select the wire(s) you are naming in the same order you entered them in the *Signal Name* dialog box.

To name a wire when you draw it


2. Click right 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.

### 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**

![Diagram of bus-through pins](image)

### Marking Wire Connections


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
- 1 as the increment

In this example, the first click is above the top bit.
Specifying a Tap Symbol


**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.
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.
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 AMS 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 the Allegro Design Entry HDL Reference Guide.

Working with Libraries

This section describes the procedures for working with libraries.

- Adding New Libraries on page 226
- Browsing Libraries on page 226
- Adding Libraries to the Search Stack on page 226
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 *Component Browser* 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**.
   
   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 229
- Creating Design Entry HDL Parts on page 229
- Creating a Symbol in Design Entry HDL on page 230
- Creating Entity Declarations from Symbols on page 232
- Creating the chips.prt File on page 235
- Creating a Part Table File on page 237
- Adding a Component on page 237
- Modifying Components on page 238
- Replacing a Component on page 239
- Breaking Up a Component on page 240
- Changing Pin States on a Component on page 240
- Choosing a Version of a Component on page 241
- Mirroring Components or Blocks on page 241
- Changing the Orientation of Components or Text on page 241
- Sectioning a Component on page 242
- Swapping Pins on a Component on page 249
Browsing the Component List

To browse logical components one library at a time

1. Click Component – Add.
   
The Component Browser 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 may follow these procedures and guidelines for creating parts if you do not have PCB Librarian installed. If you have installed PCB Librarian, you may 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.

Given below is the directory structure that has the files and directories that define the part lso0.

In the above figure, lstt1 is the library name, ls00 is the part name (cell level directory), and the directories underneath contain the view files for lso0.

To create a Design Entry HDL part:

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.

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.


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 on the right of a
symbol as an input pin, add the `vlog_mode` property to the pin with value as `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 \texttt{SEL(1)} and pins \texttt{SEL(0)} represent the single port \texttt{SEL(1:0)} in the entity declaration.

The following are some examples of pin names:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{A}</td>
<td>single-bit pin</td>
</tr>
<tr>
<td>\texttt{SEL(1:0)}</td>
<td>two-bit wide pin</td>
</tr>
<tr>
<td>\texttt{SEL(1)}</td>
<td>one bit of a two-bit wide vectored port</td>
</tr>
<tr>
<td>\texttt{SEL(0)}</td>
<td>one bit of a two-bit wide vectored port</td>
</tr>
<tr>
<td>\texttt{I(size-1:0)}</td>
<td>parameterized pin width</td>
</tr>
<tr>
<td>\texttt{I(SIZE-1:0)}</td>
<td>SCALD style parameterized pin width</td>
</tr>
</tbody>
</table>

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 200.
- Declaring the Verilog Logic Type of Ports
  
  For more information, see Setting the Verilog Logic Type for Ports and Signals on page 197.
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`

If you want to read the value of an `OUT` port inside an architecture:

- Declare the port as an `INOUT`.

  Or

- 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.

   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:

\[ \text{LIBRARY}_n = \text{libname} \]

where \( n \) is a unique number and \( \text{libname} \) 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:

\[ \text{USE}_n = \text{libname} \]

where \( n \) is a unique number and \( \text{libname} \) 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=' COMPILATION 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 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.
```

November 2008 235 Product Version 16.2
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.

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.
   The Component Browser 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 Component Browser window.

   The Property Options dialog box appears.

**Note:** See Defining Physical Property Options on page 155 for more information on defining physical property options.

### Modifying Components

**To modify a single component,**

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.

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.

   The Physical Part Filter dialog box appears.
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*.
   
The *Component Browser* 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 the *Component Browser*.

   **Tip**
   
   To replace with another version of component, double-click the selected component in Component Browser and select another version in *Version* field.

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 the *Component Browser*.

3. Select the component that should replace all the components.

**To replace a component along with its physical properties**

1. Choose *Component – Replace*.
   
The Component Browser appears.

2. Select a component from the *Library*.

3. Right-click on a PPT row in the Component Browser window and choose *PPT Options*.
   
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 155 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 Component Browser 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**


2. Click a component in your drawing.

3. Select the discrete pieces that made up the component.

**Changing Pin States on a Component**


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 BUBBLED and BUBBLE_GROUP properties attached:

\[ \text{BUBBLED} = (B) \]
\[ \text{BUBBLE\_GROUP} = (A \ | \ B) \]

Because BUBBLED=(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

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:

\[ \text{\textasciitilde \textbackslash} \]

For more information on strokes and a list of available stroke patterns, see Running Commands with Strokes on page 126.

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.
If you section a component in the Occurrence Edit mode in Design Entry HDL, the text "<<ERROR>>" might be displayed against a pin on the component. If you package a design that has a component with this error, packaging fails with the following error:

ERROR(1134): PN <n> does not belong to SEC <n>.

To correct this problem, swap the pins in the Occurrence Edit mode or section the component in the Hierarchy or Expanded mode in Design Entry HDL.

**Sectioning Multiple Components**

You can assign pin numbers to multiple logical part instances, simultaneously. The multiple sectioning feature is available as a menu option in the Component – Section menu. This feature helps you avoid section each part instance, individually. In case you realize that you have incorrectly sectioned the part instances, you can also unsection multiple part instances, simultaneously. The pin numbers you specify could either be numeric or alphanumeric.

To section multiple components:

1. Choose Component – Section – Multiple Sections.

You will be prompted to select the components to section.
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.
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 if you want to assign alphabet or numeric increment.

- For the 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.

All the selected part instances will be sectioned with pin numbers as per your specifications.

**Examples**

The following examples explain the behavior of 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
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 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 how sectioning takes place for the components with multiple pin per section.

Case 1
- Initial pin number = 2
- Pin increment = 3
Result - The first pin, pin A in the first instance of the component F245, will be assigned the initial pin number as 2. For the same pin in the next instance, three pin numbers (as 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 as 2A. For the same pin in the next instance, two pin numbers (as
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 back to the original state by unsectioning the part instances. To unsection
multiple part instances:

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, $SPN$ becomes the hard property $PN$.

**Note:** Swapping pins with the `HAS_FIXED_SIZE` property in Design Entry HDL stores pin names as $SPN$ (soft property) instead of $PN$ (hard property) resulting in conflicting values in OPF. Therefore, it is recommended that you first close Constraint Manager, if running, and then run Export Physical after swapping pins.

### 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*. 
Caution

You must not delete cells, views or files from Windows Explorer or the UNIX 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 83.

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 250
- Creating a Page Border of Your Own on page 252

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.

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.
   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 253.

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 254.

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 253.

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 \(<your\_install\_dir>/share/cdssetup/creferhdl/\).

**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 315.

**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:

\[ \text{Page } <\text{CON\_PAGE\_NUM}> \]

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, \(\text{Page 1}\) or \(\text{Page 2}\).

See Adding Custom Text on page 322 for the procedure for adding custom text.

**Tip**

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 ls04 component in the above schematic as:

\[ \text{<value of LOCATION property> 74LS04 <cell_name> [ <page_number>D2 ] } \]

For example, if the ls04 component that has the $LOCATION=Ul property is located in zone D2 on a schematic page ANALOG_IO.SCH.1.8, the Crefparts report will display the location of the ls04 component as:

\[ \text{Ul 74LS04 ANALOG_IO [ 8D2 ]} \]
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, Allegro Design Entry HDL has introduced Component Revision Manager.

Component Revision Manager enables you to perform the following tasks:

- Receive notification for differences between the schematic and reference library cells.
- Update 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 SPB 15.5. Cadence strongly recommends you not to use metadata created using earlier versions.
Getting Started with Component Revision Manager

The Component Revision Manager invokes, when you:

- 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.

2. Select the *Metadata Options* tab.

**Figure 7-1 Design Entry HDL Options dialog box — Metadata Options page**

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:

   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.

<table>
<thead>
<tr>
<th>Directive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNC_ON_STARTUP ‘ON’</td>
<td>Checks for differences between the schematic and library cells at Design Entry HDL startup.</td>
</tr>
<tr>
<td>SYNC_ON_PAGE_EDIT ‘ON’</td>
<td>Checks for differences between the schematic and library cells every time you move from one page to another in design.</td>
</tr>
</tbody>
</table>

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 stating that the differences exist between the schematic cells and the library cells.
The **Schematic Consistency Check** dialog box contains the following buttons:

- **Update**: Click this button if you want to save all the schematic pages with updated library information without showing the differences. The revision history of each page is also updated.

- **Details**: Click this button if you want to launch Component Revision Manager, which shows details of the differences between the schematic and the libraries. You can highlight a changed component to view the changes. You can also update the revision history of schematic sheets with appropriate library information.

- **Ignore**: Click this button if you want no action to be taken. The revision history of schematic pages is not updated. The changes are visible in the schematic if they are on a symbol, but they will not be incorporated until you save the design. The Schematic Consistency Check message reappears on subsequent launch of Design Entry HDL.

### 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
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.

**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

<table>
<thead>
<tr>
<th>Sch. Version</th>
<th>Status</th>
<th>Page(s) Impacted</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0.0</td>
<td>Major Update</td>
<td>@sync1_lib.sync1(sch_1):page1</td>
</tr>
<tr>
<td>4.0.0</td>
<td>Major Update</td>
<td>@sync1_lib.sync1(sch_1):page2</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Sch. Version</th>
<th>Status</th>
<th>Instance(s) Impacted</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.0</td>
<td>Minor Update</td>
<td>@sync1_lib.sync1(sch_1):page1_3</td>
</tr>
<tr>
<td>3.2.0</td>
<td>Minor Update</td>
<td>@sync1_lib.sync1(sch_1):page1_4</td>
</tr>
<tr>
<td>3.2.0</td>
<td>Minor Update</td>
<td>@sync1_lib.sync1(sch_1):page1_7</td>
</tr>
<tr>
<td>3.2.0</td>
<td>Minor Update</td>
<td>@sync1_lib.sync1(sch_1):page1_8</td>
</tr>
</tbody>
</table>
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.

Figure 7-5 Block-wise view of the Schematic Cells pane containing a block named “sync1”

<table>
<thead>
<tr>
<th>Sch. Version</th>
<th>Status</th>
<th>Block(s) Impacted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0.0</td>
<td>Major Update</td>
<td>sync1</td>
</tr>
</tbody>
</table>

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 cell(s) on the schematic. To do so:

1. Select a row in the Schematic Cells pane.

Figure 7-6 Choosing Highlight Instance highlights an instance on the schematic

<table>
<thead>
<tr>
<th>Sch. Version</th>
<th>Status</th>
<th>Page(s) Impacted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0.0</td>
<td>Major Update</td>
<td>@sync1_lib.sync1(sch_1):page1</td>
</tr>
</tbody>
</table>
| 4.0.0        | Major Update | @sync1_lib.sync1(sch_1):page2 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:

1. Select a row in the Schematic Cells pane.

Figure 7-7 Choosing Update Design updates all the instances of the schematic

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.
## Figure 7-8  Sample syncStatus.txt file

<table>
<thead>
<tr>
<th>Cell Name</th>
<th>Cell Version in Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>as27</td>
<td>1.1.0</td>
</tr>
<tr>
<td></td>
<td>Cell Version used in Schematic = 1.0.0</td>
</tr>
<tr>
<td></td>
<td>Status = 'Minor Update'</td>
</tr>
<tr>
<td></td>
<td>Instance(s) Impacted :-</td>
</tr>
<tr>
<td></td>
<td>Instance = '@parts_lib1.b1k_ro(sch_1):page1_sp'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cell Name</th>
<th>Cell Version in Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>f153</td>
<td>2.0.0</td>
</tr>
<tr>
<td></td>
<td>Cell Version used in Schematic = 1.0.0</td>
</tr>
<tr>
<td></td>
<td>Status = 'Major Update'</td>
</tr>
<tr>
<td></td>
<td>Instance(s) Impacted :-</td>
</tr>
<tr>
<td></td>
<td>Instance = '@parts_lib1.b1k_ro(sch_1):page1_i15'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cell Name</th>
<th>Cell Version in Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>perr</td>
<td>2.0.0</td>
</tr>
<tr>
<td></td>
<td>Cell Version used in Schematic = 1.0.0</td>
</tr>
<tr>
<td></td>
<td>Status = 'Major Update'</td>
</tr>
<tr>
<td></td>
<td>Instance(s) Impacted :-</td>
</tr>
<tr>
<td></td>
<td>Instance = '@parts_lib1.b1k_ro(sch_1):page2_i121'</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Using the directive...</th>
<th>Lets you...</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>NAVIGATION_OPTION 'page'</code></td>
<td>Show the occurrences of schematic cells (in the <em>Schematic Cells</em> 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 <em>Page(s) Impacted</em> header column.</td>
</tr>
<tr>
<td><code>NAVIGATION_OPTION 'instance'</code></td>
<td>Shows the occurrences of schematic cells (in the <em>Schematic Cells</em> pane) on a per-instance basis. When you specify this directive, the page number and the instance name of the cell appears under the <em>Instance(s) Impacted</em> header column.</td>
</tr>
<tr>
<td><code>NAVIGATION_OPTION 'block'</code></td>
<td>Shows the occurrences of schematic cells (in the <em>Schematic Cells</em> pane) on a per-block basis. For example, you 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 <em>Block(s) Impacted</em> header column.</td>
</tr>
</tbody>
</table>

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*. 
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 four 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).
- **Occurrence properties** - these are properties assigned to schematic objects to define different values for the same property.

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**

The key properties of a component are locked by default. You cannot edit them in the schematic through any of these — Attributes dialog box, change command, text editor, scripts, and the Global Modification dialog box.
In the Attributes dialog box, the key property names and values appear grayed. For example, for component COUPON_RESISTOR, the Attributes dialog box appears as follows:

You can see that the names and values of key properties for COUPON_RESISTOR — VALUE, TOL, POWER, and PACK_TYPE, are non-editable (appearing grayed). And the values of injected properties — SIM, PATH, BODY_REV, PART_REV, TITLE, ABBREV, 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 the Component Browser. This ensures that the components on 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.
Properties you cannot copy are:

- pin properties
- wire properties
- unnamed signals
- PATH properties

To copy ops properties of a component moved from one page to another,

1. Switch to the occurrence edit mode.
2. Choose Text – Attributes and click on the component.
3. Choose File – Save Attributes from the Attributes dialog box menu.
   The Save As dialog box appears.
4. Enter a name for the attribute file and click Save.
5. Switch to the hierarchy mode.
6. Move the component to the required page.
7. Switch to the occurrence edit mode.
8. Choose Text – Attributes and click on the component.
9. Choose File – Load Attributes from the Attributes dialog box menu.
   The Open dialog box appears.
10. Enter a name for the attribute file and click Open.
    The properties from the attribute file are copied to the component.

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 126.

You can also add a property using the property command.

Note: To propagate the new properties to the board, you have to define the properties in Allegro as well.


   This invokes the Define User Properties dialog box.

2. Define the properties so that Allegro 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.

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 270) or a net (Figure 8-2 on page 271). 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.

   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.

![Attributes dialog box](image)

**Figure 8-1**
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 have to define the property in Allegro as well.

1. Choose *Setup – Property Definitions* option in Allegro.

   This invokes the *Define User Properties* dialog box.

2. Define the property so that Allegro 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.

To add occurrence properties


2. Choose *Tools – Occurrence Edit* or type attribute in the console window.

3. Choose *Text – Attributes*.

**Figure 8-2**

![Attributes dialog box](image)
4. Select an object.
   The object you select is highlighted, and the Attributes dialog box displays the properties for the object.

5. In the Attributes dialog box, click Add.
   An empty row appears.

6. Enter a property name in the Name box and a property value in the Occurrence Value box.

7. You can also replace the occurrence value for an existing property. However, you cannot change or delete schematic values that are shaded gray.

8. Adjust property visibility and alignment as needed.

9. Click OK.

10. 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.

### 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:

\[\text{\textit{\texttt{ attribute}}}\]

For more information on strokes and a list of available stroke patterns, see Running Commands with Strokes on page 126.

### To display net properties from net synonyms


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.
Important

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

**Example 1**

Ctrl + click (LMB) on the following property value opens the Cadence website.

![Attributes dialog box](image)

Ctrl + click opens the Cadence website.
Example 2

Double clicking on the following note opens the Cadence PCB website.

www.cadencepcb.com

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 | SIR COUP RESISTOR www.cadence.com
```

Property LINK = www.cadence.com

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:

   ![note](image)

   For more information on strokes and a list of available stroke patterns, see Running Commands with Strokes on page 126.

To add text from a 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.

   Note: 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.

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:

   ![Stages]

   For more information on strokes and a list of available stroke patterns, see *Running Commands with Strokes* on page 126.

---

**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*).

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

2. Click the text line whose size you want to reduce.

Setting the Text Size

To set the text 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

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

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.

3. 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.

4. Select the required visibility from the Property Visibility field. You can choose from Invisible, Name, Value, and Both.

5. Define the scope of the visibility by selecting the relevant option. You can choose from Design, Module, Page, and Page Range.

6. Select the **Save after Change** check box if you want to save the changes to the design after changing the visibility.

![Global Property Visibility Change](image)
7. Click OK.

**Specifying Visibility of Pin Properties**

Design Entry HDL lets you to 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:

1. Choose *Tools – Options*.
   
The *Design Entry HDL Options* dialog box appears.
2. Select the *Text* tab.
3. Select the pin property visibility option.

<table>
<thead>
<tr>
<th>Select</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Invisible</em></td>
<td>Make all pin properties invisible when you add a component in your schematic. 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 <em>Name, Value or Both</em>.</td>
</tr>
<tr>
<td><em>Defined By Component</em></td>
<td>Make all pin properties visible or not depending on how the property visibility is defined on the pin on the symbol for the component.</td>
</tr>
</tbody>
</table>

4. Clock *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. After you package the design, pin numbers are displayed in the Occurrence Edit mode in Design Entry HDL. After you backannotate the design, pin numbers are displayed in the Hierarchy and Expanded mode in Design Entry HDL.

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 282
- Making Pin Numbers Invisible by Default on page 282

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.

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, occurrence property file, 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

<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
- 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.

Important
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
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
- PACK_SHORT
- PART_NAME
- PATH
- PINCOUNT
- PIN_GROUP
- PIN_NUMBER
- POWER_GROUP
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.

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

```plaintext
: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

```plaintext
: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 loaded by any of the tools. This restriction exists because the physical part name resulting from any PPT row has to be unique after upercasing 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 chapter “Preparing your Schematic for Packaging” 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.
  
  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 the `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 Example 3.

For further details on how the `POWER_PINS` and `POWER_GROUP` properties work together, refer chapter “Preparing your Schematic for Packaging” of the Packager-XL User Guide.

The `Assign Power Pins` dialog box can be invoked 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 menu. The `Assign Power Pins` dialog box reads the properties existing on the power and NC pins of a component and appears as follows:

![Assign Power Pins dialog box](image)

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,**

- 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.
Tip

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.

For example, if the following case, the maximum length of property value is specified as 5 characters:

![Assign Power Pins dialog box](image-url)
So, different `POWER_PINS` properties are created for the instance as follows:

![Attributes window showing multiple instances of POWER_PINS]

If *Separate property for each pin name* is selected, Design Entry HDL creates a new `POWER_PINS` property for a pin name.
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:
To change the power source for pin G3 to global signal A,

➤ 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.
2. Right-click to bring up the pop-up menu as follows

3. Choose Power Name.
   
   The Assign Power Pins dialog box appears.
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
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

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Power</th>
<th>Power Names</th>
<th>NC Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>F9</td>
<td>GND</td>
<td>C</td>
<td>✓</td>
</tr>
<tr>
<td>F3</td>
<td>GND</td>
<td>C</td>
<td>✓</td>
</tr>
<tr>
<td>J6</td>
<td>GND</td>
<td>C</td>
<td>✓</td>
</tr>
<tr>
<td>C6</td>
<td>GND</td>
<td>C</td>
<td>✓</td>
</tr>
<tr>
<td>G9</td>
<td>VCC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G3</td>
<td>VCC</td>
<td>A</td>
<td>✓</td>
</tr>
</tbody>
</table>

**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)
On the instance:

\[
\text{POWER\_GROUP} = (\text{GND} = \text{GND1})
\]

The *Assign Power Pins* dialog box appears as follows

![Assign Power Pins dialog box](image)

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 `.cpx` 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
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.
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.

Viewing Properties in Occurrence Edit Mode

In the occurrence edit mode, you can only view the properties on power and NC pins. And before you view these properties, Design Entry HDL prompts you to delete any occurrence properties that exist on the schematic.
Example

Suppose you have a component `VECTOR_PART` with the following properties:

![Attributes window](image)

To view the *Assign Power Pins* dialog box,

1. Choose *Text – View Power Pins* and click on the component.

   The following warning appears

   ![Warning dialog](image)

   2. Select the *Delete the `opf` property* option.

   You cannot invoke the *Assign Power Pins* dialog box without selecting this option.

   3. Click *OK*. 
The *Assign Power Pins* dialog box appears as follows

![Assign Power Pins dialog box]

You can see that all the columns are greyed out; the properties are read only.

**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 `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 stating 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.

![Assign Power Pins dialog box](image)

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**

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 3 sections of a split part on a page of a schematic. And 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 comprising of the 3 sections.

2. Choose Group – Assign Power Pins[A].

   The Assign Power Pins dialog appears.

3. For pin AE9, select the check box for NC Pins.

4. For pin AC21, select the name GND 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 chapter Preparing your Schematic for Packaging 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.

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.

Consider the example of a hierarchical block, CNTR, which has the instances ls00 and ls04 inside it. Property LOCATIONS = U%'MY_LOC’1 is attached to ls00, and LOCATION = U%'MY_LOC’2 is attached to ls04. If the property MY_LOC = 5\parameter or MY_LOC = 5\param is attached to the hierarchical block CNTR, then running Packager-XL will cause the property LOCATION to have a values U51 and U52 for the instances ls00 and ls04, respectively.

Note: Text macro substitution takes place only when Packager-XL is run in the forward mode.

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.

Important

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 generation of Verilog.

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)

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:

\begin{verbatim}
vlog_parml=SFX:string
SFX=E
\end{verbatim}

**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:

\begin{verbatim}
macro_identifier = 'macro_definition'
\end{verbatim}

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 Occurrence Properties**

This section describes the procedures for working with occurrence properties in Design
Entry HDL.

**About Occurrence Editing**

The netlist generated for any design specified in Design Entry HDL has two distinct aspects:

- Connectivity
- Properties

Connectivity represents the definition of the structure of the design and is specified in Verilog
using the verilog.v file. This file is generated by Design Entry HDL every time the
schematic is saved.

A property acts as a directive to the downstream tools (Packager-XL, Simulation Flow,
Allegro, and so on.), and aids in controlling their functions. Properties can be attached to
objects (primitive instances, pins on instances, and nets) in two modes:

- The schematic mode
  
The properties attached in the schematic mode are saved in the viewprps.prp file.
- The occurrence mode
  
The properties attached in the occurrence mode are saved in the props.opf file.

The schematic and occurrence modes differ in the way they treat hierarchical blocks.

Packager-XL reads both the verilog.v file and the viewprps.prp file to generate the pst*
files. The properties specified on instances are stored in the pstxprt.dat file, and the
properties specified on nets and pins are stored in the pstxnet.dat file.
**The Schematic Mode**

The schematic mode is the default mode in Design Entry HDL. In this mode, all properties attached to instances, nets, and pins are stored in the `viewprps.prp` file. This file is located in the schematic view (sch_1) of the cell in the design library.

**The Occurrence Mode**

To add properties to a specific occurrence of an object, you can use the occurrence mode of property editing in Design Entry HDL.

**Note:** The POWER_GROUP property cannot be used in the occurrence edit mode

**Viewing Occurrence Properties**

To view occurrence edit properties:


This enables occurrence editing. Occurrence properties are displayed on the schematic if they have corresponding schematic or placeholder properties. If there are no corresponding properties, you can view occurrence properties from the Attributes dialog box when occurrence editing is enabled.

**To display occurrence property attributes with occurrence editing enabled**

1. Choose Text – Attributes.
2. Select an object.

   The object you select is highlighted, and the Attributes dialog box displays the Winning values for the object.

**Note:** To disable occurrence editing, choose Tools – Occurrence Edit.

**Occurrence Property File (OPF)**

The occurrence property file is a database that stores properties for all objects (instances, pins, and nets). It is accessed in read-write mode from both Design Entry HDL and Packager-XL.
Creating an occurrence property file

The OPF is created in two ways:

- When you invoke the occurrence edit mode in Design Entry HDL
  Design Entry HDL creates an OPF, if it does not already exist (that is created by Packager-XL) and writes all the properties added in the occurrence edit mode into it.

- When you package your design
  Packager-XL creates an OPF if it does not already exist (and is created by Design Entry HDL). Packager-XL always reads and writes properties into the OPF both in the forward and feedback mode.

If a design has been packaged at least once, the OPF is loaded by Design Entry HDL in the read-write mode wherein all (relevant) packaging information is accessible. You can modify by using the Attribute menu in the occurrence mode.

Location of the Occurrence Property File

The OPF is located in the opf view of the root cell.

Any property that is attached to an object in the occurrence mode is stored in the OPF at the root level. This is regardless of where they are located in the design hierarchy. The root cell is the design name in the .cpm file and can be defined through the Project Manager Setup.

Hierarchy of Occurrence Property Files

Since an OPF always resides at the root level of a design that is being edited or packaged, you can have a hierarchy of OPFs by packaging the hierarchical blocks individually.

OPFs, when arranged in a hierarchy where one OPF exists at the top level and the other OPFs exists at a lower level, are referred to as Overlaid OPFs and Subdesign OPFs, respectively. The properties in the subdesign OPFs are visible at the top level. In case of a conflict such as when a property has different values at the top and lower levels, the value at the topmost level takes precedence. This is unlike inheritance where the lowermost level always takes precedence.

Property Precedence in an OPF

There are two rules for setting the precedence of properties in an OPF.
1. If the same occurrence of an object has a schematic value and an OPF property value, the OPF property value is preferred over the schematic value.

2. If the same occurrence has multiple OPF property values stored at different levels of the hierarchy (in different OPF files), then the OPF property value at a higher level gets preference over the OPF property value at the lower level.

**Inheritance of Properties in an OPF**

Inheritance is used to propagate properties down in the hierarchy. An OPF handles inheritance in the same way as a schematic.

A property, whether schematic or OPF, is propagated to all lower level cells present in the hierarchy. If the same property is present in any lower level cell, the existing property at the lower level cell still gets preferred over the propagated property.

**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 320.

  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
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.

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 418.

- **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 418.

- **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 418.

- **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 418.

**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

It is recommended that you add custom text to the symbol of a page border. To add custom text to symbols, attach it to the origin. The symbol just shows the format string. When the page border is instantiated, the values of custom variables are substituted. For example, if you have the following custom text on the page border symbol:

Page <CON_PAGE_NUM>

When the page border gets instantiated, the custom variable CON_PAGE_NUM would 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:

The Design Entry HDL Options dialog box appears.

2. Click on the Custom Variables tab.
3. Enter the name of the variable.
4. Enter the value of the variable.
   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 button.

Important
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 a `$` 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* get updated depending upon the current page and the custom text gets 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 gets attached to the cursor.

10. Right click and select the Done option.

**Note:** You can add multiple custom text to the same object.

**Case-Sensitivity in Custom Text**

If you add a variable in the Unix environment, the case of the variable is very important. You can set the option of storing input in uppercase in Design Entry HDL using Design Entry HDL Options – Text and enabling the check box Upper-case Input. While setting a variable, if this check box is checked, the variable name is stored in uppercase. So, while adding the variable to custom text, you would have to enter the name of the environment variable in uppercase.

**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:** The operations like delete, move, copy, rotate and spin that can be performed on properties can be done on custom text also.
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.

**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. Click right 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 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 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 pin(s) where you specify.

   **Note:** To toggle the pin type before you place the pin on the block, click the right mouse button 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 pin(s) 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 Allegro and maintain it in a reference library. You can then use the block as a read-only block in other designs. For more information on creating reusable blocks and using them in other designs, see the Design Reuse chapter of the Allegro PCB Design Flows user guide.

For information on netlisting read-only blocks, see Netlisting Read-only Blocks Used in a Design.

**Navigating the Drawing Hierarchy**

To view a block diagram from the top-level schematic

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.

To descend into drawings while in In Hierarchy mode, you can set the environment variable CONCEPT_DESCEND_EDIT_LIST variable.
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

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 to and the order in which they will be accessed by choosing Display – Return.

Note: Release 15.7 onwards, UNC path names are supported while invoking Genview.

Generating a Design 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:

<table>
<thead>
<tr>
<th>Select the type</th>
<th>If you have selected the following view</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schematic</td>
<td>sch_1</td>
</tr>
<tr>
<td>Symbol</td>
<td>sym_1</td>
</tr>
<tr>
<td>VHDL</td>
<td>vhdl_1</td>
</tr>
<tr>
<td>Verilog</td>
<td>vlog_1</td>
</tr>
</tbody>
</table>

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.

**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 that you have a schematic `TOP.SCH.1.1` as below:

1. Generate the symbol for the schematic. A symbol named `TOP` will be created as below:

![Symbol for TOP schematic]

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, as below:

![Updated schematic with added `INT<1..0>` pin]
3. Generate the symbol for the schematic again with the Retain Graphics check box selected. The symbol TOP will be created as below:

![Symbol](image1)

Note that the placement of the pins CLOCK and RESET on the symbol have not changed.

Suppose that you have instantiated the symbol TOP on some other schematic page and have connected a wire to the pin CLOCK on the symbol. The connectivity between the wire and the pin CLOCK 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 symbol TOP will be created as below:

![Symbol](image2)

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 have 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 below:

![Symbol Diagram]

Note that the placement of the pins on the symbol have changed. This is because the graphic for the symbol is regenerated when the *Retain Graphics* check box is not selected. The placement of the pins on the symbol is done by Design Entry HDL using its internal algorithms.

Suppose that 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 the Component Browser.

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 place holder 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 Genview behaves 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 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.
Important

Manually changing the block symbol pin_name 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 INPORT. 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 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. Click right 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. Click right 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.
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**

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. Click right 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.
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.
For example, the two \texttt{ls00} 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 343.

3. Choose \textit{Edit} – \textit{Mirror}.

4. Press the middle mouse button on a three-button mouse or \textit{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, click right 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, click right, 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, click right 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:
Working with Groups

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:

\texttt{textsize <size in inches> <group name>}

\textbf{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 341) 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.
Working with Designs

This section describes the following tasks:

Expanding Your Design on page 352
Finding Nets and Cells in Your Design on page 352
Navigating Nets in Your Design on page 354
Global Modification on page 355
Part Manager on page 378
Running Scripts on page 390
Highlighting (Cross-Probing) Objects on page 394
Applying Connectivity Changes from the Physical Design to Your Schematic
Back Annotating Your Design
Module Ordering on page 396
Performing Page Management Operations on page 405
Displaying and Working with Schematic Page Numbers on page 416
Importing Designs on page 426
Baselining a Design on page 433
Creating the Table of Contents for a Design on page 438
Expanding Your Design

When you start a design, it is in hierarchy. This means that multiple drawings in a design are connected in a tree-like fashion. Before expansion, schematic views are obtained from the default specified in Project Manager Setup (in the Schematic box of the Views tab). Once a design is expanded, you can navigate up and down the drawing hierarchy using the Hierarchy Editor or by clicking on the Descend or Ascend icons as needed.

Expanding the design enables communications between Design Entry HDL and other tools and lets you associate an expansion configuration with the design.

When you create an expansion configuration using the Hierarchy Editor (Tools – Hierarchy Editor) and then expand the drawing (Tools – Expand Design), schematic views are obtained from the current configuration and not from the default specified in the Project Manager Setup.

With design expansion, Design Entry HDL reads all levels of the design so that you can:

- Highlight the objects (cross-probe) between tools using Display – Highlight and Display – Dehighlight.
- View properties from net synonyms that are in the current drawing or in other drawings.
- Navigate the drawing hierarchy based on the current expansion configuration.
- Find all synonyms of a specified net or all instances of a specified cell in a hierarchical design or multipage schematic.

To expand a design

➤ Choose Tools – Expand Design.

Design Entry HDL expands the design to read all pages and levels and to enable communication with other tools.

Note: Drawing expansion takes some time to complete, depending on the complexity of your design.

Finding Nets and Cells in Your Design

To find nets and cells:

The *Global Find* dialog box appears.

3. In the *Name* box, 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.

4. Select the object type to be located as either *Net* or *Cell*.

5. To optionally restrict the search by property, do both of 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 a * (asterisk) to locate all objects having 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.

6. Select how you want the search results to be listed - by full Hierarchical Names or by Library Location.

7. Click *Find* to begin the search.

   The search process begins. The *Find* button becomes the *Stop Find* button, 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.

8. Specify 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. Click on another search result to view in the design.

    or

    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 Global Find**

- Objects with the following properties:

<table>
<thead>
<tr>
<th>VHDL_SLICE</th>
<th>VHDL.Concat</th>
<th>TIE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDL_SCHEMATIC</td>
<td>HDL_POWER</td>
<td>HDL_CONCAT</td>
</tr>
<tr>
<td>HDL_REPLICATE</td>
<td>COMMENT_BODY</td>
<td>PLUMBING_BODY</td>
</tr>
<tr>
<td>FLAG_BODY</td>
<td>PINNAME_BODY</td>
<td>HDL_PORT</td>
</tr>
<tr>
<td>VHDL_PORT</td>
<td>VHDL_ALIAS</td>
<td></td>
</tr>
</tbody>
</table>

- Objects named:

<table>
<thead>
<tr>
<th>PIN NAMES</th>
<th>DRAWING</th>
<th>DEFINE</th>
<th>MENU</th>
</tr>
</thead>
</table>

- 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

**Navigating Nets in Your Design**


   The Global Navigation window appears.

2. Select how the search results are to be viewed - by full Hierarchical Names or by Library Locations.

3. Select a net in your design.

4. A message box appears prompting you to expand the design:
If you select "Yes", the design is expanded and 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.

If you select the Expand without prompt check box, you will also be able to navigate nets later without expanding the 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.

5. If you want to zoom in on a search result, select Zoom to Object, otherwise, go to step 6.

6. To view a search result, select a net instance in the status area box.

When you select a result to view, the page containing the object appears with the object highlighted.

7. Select another net in your design.

8. Use the Back and Next buttons to move between different nets that you selected.

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.

Important

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 SUNW10f package.

To open the Global Modification window:

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 input fields for controlling the modification of properties.

**Property Change**  Use this tabbed page to change properties of components, pins, and nets across a design.

**Property Delete**  Use this tabbed page to delete 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 a batch mode. To know more, please see the The Batch Mode Operation section below.

**Modifying Component, Pin, and Net Properties**

To change a net, pin, or component property,

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.

You can also preserve the source property name by selecting the first option in the drop-down menu, ++Preserve Source Name++. This option cannot be used in conjunction with the ++Preserve Source Value++ option, otherwise no modification is made.
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 in the GUI for the original property name and value. The * is always handled as a wildcard in the original property name. The Enable Wildcard check box controls whether the * in the original property value is handled as a wildcard or as a literal. To know more about the usage 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 only to the current page, select the Current Page radio button.
- 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 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.

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 * character is always treated as a literal in the new property value field.

9. To specify that the schematic sheet be saved after a modification is made, select the Save after Change check box.

   After specifying the changes to be made, the Property Change page should appear as
shown below:

![Global Modification Window](image)

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 displaying the current status.
After all the changes are made based on the defined scope, the Results box displays the Successful Completion status as shown in the picture below.

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:

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 either 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 in the GUI for the delete property name and value. The * 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 only to the current page, select the *Current Page* radio button.
- 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 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.

6. To specify that the schematic sheet be saved after a modification is made, select the *Save after Change* check box.

After specifying the changes to be made, the Property Delete page should resemble the
7. To start the processing and close the Global Modification window, click the OK button.
The **Confirm Request** message box appears.

![Confirm Request](image)

8. In the **Confirm Request** message box, click **Continue**.

The **Results** dialog box appears, first displaying the current status and then the final status as in the Modifying component, pin, and net properties procedure.

### Selecting the Component to Replace from the Physical Part Filter

The **Component Change** tab in the GUI can be used to perform functions like **global delete**, **global modify**, **global replace**, and **global refresh**. While globally replacing components, you can choose the new component either 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. These are:

- Selecting the component from the physical part filter
- Selecting the component from the design (Schematic Pick)

There is not wildcard support in the **Component Change** tab. An * is allowed in the original property value and new property value fields, but they are taken literally.

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`.
   
   Ensure that the *Select Physical Component* check box is selected. Selecting this check box ensures 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. The probability of making changes by mistake is high when wildcards are used both in source and destination.

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 as shown in the picture below with the properties of the selected component listed in the *Properties* list below the *Original*
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 * is allowed in the Original Property Value and New Property Value fields, but they are taken literally.

Selecting the New Component from the Physical Part Filter

To select the new component from the physical part filter:

1. Click the Change button in the New Component group.
   The Select New Component dialog box appears.
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.

   ![Global Modification Window](image)

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,:

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 as shown below with the properties of the selected component listed in the Properties list below the Original Component
Selecting the New Component from the Design (Schematic Pick)

To select the new component,: 

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.
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 as shown below.

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.

Recovering a Component

The features of a component replace are as follows:

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.
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:

1. 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.

2. Blank property names in the original and new property fields.

3. 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.

4. 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, the Component Browser 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. This command takes a single argument, 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 rage 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
;; -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
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:

- The Part Manager Use Model on page 378
- The Part Manager User Interface on page 379
- Working with Part Manager on page 384

The Part Manager Use Model

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.

**Invoking Part Manager**

You can invoke 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*.
- 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.
### Table 11-1

<table>
<thead>
<tr>
<th>Column Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schematic Name</td>
<td>This column shows the block name of the part instance.</td>
</tr>
<tr>
<td>RefDes</td>
<td>This column shows the reference designator of the component. If the location property is not available, a question mark (“?”) is displayed in this column.</td>
</tr>
<tr>
<td>Part Status</td>
<td>This column shows the status of parts represented by icons. The values that this column can show are listed in Table 11-2 on page 381.</td>
</tr>
</tbody>
</table>

### Table 11-2

<table>
<thead>
<tr>
<th>Icon</th>
<th>Part Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Green" alt="Checkmark" /></td>
<td>MATCHED</td>
<td>This part instance was added in logical mode (no ptf rows used) and the PXL directive <code>FORCE_PTF_ENTRY</code> is not set. Part Manager recognizes a part as logical if the part does not have a ptf associated with it.</td>
</tr>
<tr>
<td><img src="Green" alt="Checkmark" /></td>
<td>MATCHED</td>
<td>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.</td>
</tr>
</tbody>
</table>
Working with Designs

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.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Part Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>INJECTED</td>
<td>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.</td>
</tr>
<tr>
<td>✔️</td>
<td>MISMATCH</td>
<td>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.</td>
</tr>
<tr>
<td>✗</td>
<td>NOT MATCHED</td>
<td>This part instance does not match any row in any part table file.</td>
</tr>
<tr>
<td>IGNORED</td>
<td>IGNORED</td>
<td>This status is shown for the parts which have a PACK_IGNORE property.</td>
</tr>
</tbody>
</table>

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,
RATED\_POWER and PKG and two injected properties, VALUE and TOL. Notice 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/Undefiend)

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
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
- Applying Changes to Part Instances
- Resetting Changes Made to Part Instances
- Updating Multiple Parts with a Single PTF Row
- Refreshing Contents of Part Manager
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(s)*.

Alternatively, you can select *Update Instance(s)* from the *Options* menu. This brings up the Part Table Filter window.
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.

3. 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.
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(s) and Apply Changes** commands put together.

**Note:** In case the selected row is part of a reuse block, Part manager will show a warning stating: "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 row(s) 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*.

   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.

**Note:** The *Update Instance(s)* and *Update and Apply* commands also read the Occurrence Property File (OPF). The OPF values, if present, override the schematic property values. In such cases, Part Manager shows the OPF value. To distinguish OPF properties from other properties, the values coming from OPF appear in bold. Part Manager only reads the OPF values and cannot write on to the file. You need to manually update these properties in Design Entry HDL in case of any errors.

**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(s)* 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.

![Warning message]

**Note:** If you save the schematic and then move the focus to Part Manager, it prompts you to refresh Part Manager.

![Refresh prompt]

**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.

![Application prompt]

**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*.

   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.

![Part Manager window with Hierarchical Path column](image)

Notice 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 391.

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 392.

### 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** | **Use**
--- | ---
Redirect all the messages to the command line | `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 392
- **Running a script from the Design Entry HDL console window** on page 393
- **Running a script from the Design Entry HDL console window** on page 393
- **Running a script from the UNIX or Windows command prompt** on page 393

**Running a script from the Tools menu**

1. Choose **Tools – 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:

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 the *Allegro Design Entry HDL Reference Guide*.

### Running a script from the UNIX or Windows command prompt

➤ Use the following command:

```bash
nconcepthdl -proj <project_name>.cpm -scr <script_file_name>
```

For more information on the *nconcepthdl* command, see *Nongraphical Design Entry HDL* in the *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.

For more information on highlighting and dehighlighting objects, see Highlighting Objects on page 109 and Turning Off Highlighting on page 110.

Distributing Design Changes between Physical and Logical Designs

   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

   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 `pstback.dat` file containing the physical part and adds the information to the design.

**Note:** To know more about the `pstback.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 15.x Releases

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 UI 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 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. The 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

```plaintext```
("<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

```plaintext```
("capacitor" "gnd")
```

Important

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.

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 Entry HDL XL
- 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

- 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:
   
   b. Select the Design Navigation tab.
   c. Type `<property name>` in the Property on page border for page name.
   d. Select the casing for the page name. For example, `Preserve`.
   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

In the pre-15.1 releases of Design Entry HDL, inserting a new page between existing pages required scripts to create a page gap, so that the new page could be inserted. This process was tedious and error-prone, potentially resulting in lost or incorrectly ordered pages. In release 15.1, _PAGE console commands were introduced to enable you to insert, delete, or move pages in a schematic with ease. Release 15.2 onwards, Design Entry HDL provides a graphical user interface to these commands to help you perform various page management operations with the click of a button. When you perform various page management operations on a page or a set of pages using the page management GUI or the _PAGE commands, all page numbers are automatically adjusted.

For a detailed description of the corresponding console commands, refer to PAGE commands in the 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 414.

Inserting a Page

You either insert a single page or a set of pages in a schematic using the new 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:

A confirmation box is displayed.

![Confirmation Box](image.png)

2. Click **OK** to confirm.
   
   A blank page is inserted and a success notification is displayed.

![Success Notification](image.png)

**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**.
The Insert Pages dialog box is displayed

![Insert Pages dialog box](image)

In this dialog box, you can specify the number of pages to insert and the location from where you want to insert the page(s). 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.

![Design Entry HDL](image)

**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:

2. In the Insert Pages dialog box, select 2 in the Insert 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:

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.
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:

   A confirmation box is displayed.

2. Click OK to confirm.
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:


   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 page(s) being deleted. This will create page gaps for the pages you delete. For this example, leave this option unchecked.

4. Click OK.
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:

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:

   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.
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:

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.
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.

**Checking the Page Number Mismatch Option**

You must ensure that the *Page Number Mismatch* option is selected in the Check tab of the Design Entry Options dialog box. When you save the schematic, Design Entry HDL will report and correct the PAGE_NUMBER directive conflicts in the ASCII and binary files for all the pages of the design.

**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. Therefore, before you perform a page management operation, you must save all the affected pages with unsaved changes.
For example, if you delete pages 3-4 in a schematic with pages 3, 4 and 5 having 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 page(s) 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, then 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 417
- Modifying Custom Text for Page Numbers on page 420
- Updating Custom Text Variables for Page Numbers on page 421
- Managing Schematic Pages on page 422
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 43.

- 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 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 315.

- 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 may add the CON_PAGE_NUM and CON_PARENT_CNAME variables to your design. For details, refer the quickplace command in the PCB Sytems 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.

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.

- 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
   - 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 gets 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:** The operations like delete, move, copy, rotate and spin that can be performed on properties can be done on the custom text also.

### 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.

For example, if you have added only the `CURRENT_DESIGN_SHEET` custom text variable in the 20th page in a hierarchical design that has 25 pages, the schematic page will display the page number as **20**. Now 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.

   ![Custom Text dialog box]

3. Select the `TOTAL_DESIGN_SHEETS` variable from the **Variables** drop-down list.

4. Modify the text in the **FORMAT string** field to:
5. Click OK.

The page numbering information on the schematic changes to PAGE 15 of 20.

**Note:** The page number format is always integer. Design Entry HDL does not allow you to change the page number format to i, I, a, etc.

### 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 422.

**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**

1. 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 the *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 423.

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
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.

**Important**

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:

```
poa
  clock <page1_i1>
  flashcard <page2_i3>
```
The \textit{Plot} dialog box displays the following design hierarchy tree:

\begin{itemize}
  \item \texttt{poa[1:2]}
  \begin{itemize}
    \item \texttt{clock <page1_i1> (3)}
    \item \texttt{flashcard <page2_i3> (4)}
  \end{itemize}
\end{itemize}

Now swap page 1 with page 2 using the \texttt{page swap 1 2} Design Entry HDL console command.

**Hierarchy Tree in the Hierarchy Viewer after Page Swap**

\begin{itemize}
  \item \texttt{poa}
    \begin{itemize}
      \item \texttt{clock <page2_i1>}
      \item \texttt{flashcard <page1_i3>}
    \end{itemize}
\end{itemize}

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 426.

**Hierarchy Tree in Plot Dialog Box after Page Swap**

\begin{itemize}
  \item \texttt{poa[1:2]}
    \begin{itemize}
      \item \texttt{clock <page2_i1> (3)}
      \item \texttt{flashcard <page1_i3> (4)}
    \end{itemize}
\end{itemize}

Note that the page number for the block \texttt{clock} has not changed from 3 to 4 and the page number for the block \texttt{flashcard} has not changed from 4 to 3 in the \textit{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 \texttt{POA} that has the block \texttt{clock} instantiated on it
2. Page 1 of block \texttt{POA} that has the block \texttt{flashcard} instantiated on it
3. Pages of block \texttt{clock}
4. Pages of block \texttt{flashcard}

However, the pages are plotted in the following order:

1. Page 1 of block \texttt{POA} that has the block \texttt{clock} instantiated on it
2. Page 1 of block \texttt{POA} that has the block \texttt{flashcard} instantiated on it
3. Pages of block \texttt{clock}
4. Pages of block \texttt{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:

```
- pca
  - flashcard <page1_i3>
  - clock <page2_i1>
```

**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 etc.

Using the Import Design feature, you can perform the following tasks:

- 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.

**Note:** The Import Design feature is only available in Allegro Design Entry HDL XL.

**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 place holders on the schematic for the cells which are not found in the destination design.
To import a design sheet,


   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 sheet(s).

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 populated to the Projects list box and only absolute paths appear in the list box. If you have assigned a relative path in a file, its absolute path appears in the list box. You can place the preferred_projects.txt file at home or home/cdsssetup or at the project level.
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. This would open Allegro Design Entry HDL 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:
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. On Solaris systems, if you select a sheet and display the context sensitive menu then the Design Import dialog box hides behind Allegro Design Entry HDL and only context sensitive menu appears. You can use the View Sheet, Import and Reset buttons in the Design Import dialog box for performing 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. Notice 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.

8. To import the selected sheets, click Import.

**Note:** If the selected sheet(s) have blocks that are already imported in the design, then a dialog box appears indicating the presence of those blocks in the selected design. You can click the Re-import button to re-import the blocks or No to prevent the re-import.
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:

a. 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.

b. 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 the Library field.

c. 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 the 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 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.

![Import Design: Signal Name Clash](image)

You can change signal names by using a combination of following ways:

a. Edit a signal name in the **New Signal Name** field.

b. Select the **New Signal Name** field for few signal(s) 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 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 to the schematic. This happens because .csa files do not exist for the pages created by Crefer.

**Viewing a Design**

Design Entry HDL 15.5 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 perform the following activities 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,

   
   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 toolbar 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.

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.

**Important**

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:

2. Select the Metadata Options tab.

![Design Entry HDL Options](image)

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 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.

<table>
<thead>
<tr>
<th>Directive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNC_ON_STARTUP 'ON'</td>
<td>Checks for differences between the schematic and library cells at Design Entry HDL startup.</td>
</tr>
</tbody>
</table>
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:

1. In Allegro Design Entry HDL, choose *File – Save and Baseline*.

   The *Baseline: <design name> Current Baseline Version* dialog box appears.

   ![Baseline dialog box](image)

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.

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.

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 stating that there are no changes in the schematic, and the baseline operation is cancelled.

**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.

### Important

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. 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. TOC also helps in navigating the design when the schematic designs are plotted or published.

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 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.

### 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.

![Custom Text dialog box](image)

### 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.

3. **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 is configurable.
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.

   ![TABLE OF CONTENTS](image)

   The TOC symbol is added to page1.

7. Click a page in the hierarchy viewer other than page 1 and then click page 1 again.

<table>
<thead>
<tr>
<th>SHEET NUMBERS</th>
<th>SHEET DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>page1</td>
</tr>
<tr>
<td>2</td>
<td>Index</td>
</tr>
<tr>
<td>3</td>
<td>Main Block Diagram</td>
</tr>
<tr>
<td>4</td>
<td>Processors</td>
</tr>
</tbody>
</table>

The TOC is generated.
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.

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.
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.

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
```
The TOC shows page ranges in the Sheet Number column wherever applicable.

<table>
<thead>
<tr>
<th>35</th>
<th>Decoupling Caps.</th>
</tr>
</thead>
<tbody>
<tr>
<td>35-37</td>
<td>EGRESS .QDRII .SRAM .CHANNEL 0.</td>
</tr>
<tr>
<td>38-39</td>
<td>EGRESS .QDRII .SRAM .CHANNEL 1.</td>
</tr>
<tr>
<td>40-41</td>
<td>EGRESS .QDRII .SRAM .CHANNEL 2.</td>
</tr>
<tr>
<td>42-43</td>
<td>EGRESS .QDRII .SRAM .CHANNEL 3.</td>
</tr>
</tbody>
</table>

**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.
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.
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.
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.

The TOC is updated, and the page numbers appear as sheet ranges where appropriate.
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:

- Packaging the design.
  
  For more information, see Netlisting for Packaging the Design on page 450.

- Digital simulation
  
  For more information, see Netlisting for Digital Simulation on page 454.

- Programmable IC simulation
  
  For more information, see Netlisting for Synthesizing a Design in Synplify on page 456.

- Analog and Mixed Signal simulation
  
  For more information, see Netlisting for Analog and Mixed Signal Simulation on page 460.

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 the Allegro Design Entry HDL Reference Guide.

- Performs entity declaration checking for instantiated components.

  For more information, see Entity Declaration Checking for Instantiated Components in the Allegro Design Entry HDL Reference Guide.
Netlisting for Packaging the Design

Design Entry HDL generates the Verilog netlist that is used for packaging the design when you save the design. You can specify the options for generating the netlist for packaging the design.

To specify the options for generating the Verilog netlist used for packaging the 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.
   Verilog netlisting is enabled by default. You can select the VHDL check box if you want to generate the VHDL netlist for the design.

3. Click the Options button next to the Verilog check box.
   The Verilog Netlist dialog box appears.

4. Select the Verbose Output check box to log the debug messages of the netlisting process in the hdldir.log file located in the <project_directory>/temp/ directory.

5. Select the Analyze on Save check box if you want the netlist to be analyzed by ncvlog.exe (NC Verilog compiler) every time you save the design in Design Entry HDL.
   This check box is enabled only if you have selected the NC Verilog simulator in the Tools tab of the Project Setup dialog box.

6. 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."
7. 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.

8. Specify the timescale directive for the Verilog module of the schematic in the Time Scale field. The default value is 1ns/1ns.

9. Specify the Verilog logic type for all nets in the design. You can use any legal Verilog net type, such as WIRE, WAND, and WOR. The default value is WIRE.

   Note: The specified net type applies to all drawings in the design. You can override the net type for individual drawings by using the VLOG_NET_TYPE property on a VERILOG_DEC symbol. For more information on specifying the Verilog logic type of ports and signals, see Setting the Verilog Logic Type for Ports and Signals on page 197.

10. Specify the signal names for the Verilog net type Supply 0.

11. Specify the signal names for the Verilog net type Supply 1.

12. Click OK to save the settings and close the Verilog Netlist dialog box.

13. Click OK to close the Design Entry HDL Options dialog box.

To generate the Verilog netlist used for packaging the design

➤ In Design Entry HDL, choose File – Save.

   The Verilog netlist file verilog.v is generated in the following directory:

   <project_directory>/worklib/<design_name>/sch_1/

Important

   If you are working on a large design, saving the design in Design Entry HDL takes a long time because the netlist for the entire design is regenerated every time you save the design. You can avoid this by disabling netlisting of the design. For more information, see Disabling Netlisting of Designs on page 452. When you package your design, the Verilog netlist for packaging the design is generated even if netlisting is disabled. For more information see What Happens if Netlisting is Disabled and You Package the Design?

The Verilog netlisting errors that are identified during the netlisting process are displayed in the Markers dialog box in Design Entry HDL. The VHDL netlisting errors are displayed only if the VHDL check box is selected in the Output tab of the Design Entry HDL Options dialog box. 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 the file in the Markers dialog box. The hdldir.log file located in the <project_directory>/temp/ directory contains the details of the netlisting process.

**Note:** The Verilog netlist for packaging the design is also regenerated when you package the design. During the packaging process only the Verilog netlisting errors, if any, are displayed. The VHDL netlisting errors, if any, are not displayed even if VHDL netlisting is enabled in the Output tab of the Design Entry HDL Options dialog box. If there are any Verilog netlisting errors, the packaging process will stop. You have to correct the errors in the design and restart the packaging process. If there are any warnings in the design, the packaging process will not stop. The Progress dialog box that appears when the packaging process is run will display the warnings in the design and the packaging process will continue.

**Note:** When you perform the File — Save operation on a design, all the pages are processed by HDL Direct for generating netlist even though you intend to save only one page. If you do not want the overhead of netlisting on each page, you can set HDL Direct Netlisting to OFF through the Tools — Options — Output tab in Design Entry HDL.

**To view netlisting errors in the packaging netlist**

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.

For more information on the errors displayed in the Markers dialog box, see Netlisting Errors in the Allegro Design Entry HDL Reference Guide.

**Disabling Netlisting of Designs**

If you are working on a large design, saving the design in Design Entry HDL takes a long time because the netlist for the entire design is regenerated every time you save the design. You can avoid this by disabling netlisting of the design.

**To disable Verilog and VHDL netlisting**

1. In Design Entry HDL, choose Tools — Options.
The Design Entry HDL Options dialog box appears.

2. Select the Output tab.

3. Deselect the Create Netlist check box.

If both Verilog and VHDL netlisting are disabled the Verilog and VHDL netlisting errors will not be displayed in the Markers dialog box when you save the design.

**Note:** When you package your design, the Verilog netlist for packaging the design is generated even if netlisting is disabled. For more information see What Happens if Netlisting is Disabled and You Package the Design?

**To disable VHDL netlisting**

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. Deselect the VHDL check box.

If VHDL netlisting is disabled the VHDL netlisting errors will not be displayed in the Markers dialog box when you save the design.

**What Happens if Netlisting is Disabled and You Package the Design?**

When you package the design, the Verilog netlist for packaging the design is generated even if netlisting of the design is disabled. The Verilog netlisting errors, if any, are displayed in the Markers dialog box. If there are any Verilog netlisting errors, the packaging process will stop. You have to correct the errors in the design and restart the packaging process. If there are any warnings in the design, the packaging process will not stop. The Progress dialog box that appears when the packaging process is run will display the warnings and the packaging process will continue.

**Note:** If you want to package a design that has netlisting errors, select the Continue expansion even if netlisting errors check box in the Expansion tab of the Project Setup dialog box. If this check box is selected, the packaging process will continue even if the design has netlisting errors. For more information, see Chapter 3, “Project Creation and Setup.”

**Note:** When you package a design, only the Verilog netlisting errors, if any, are displayed. The VHDL netlisting errors in the design, if any, are not displayed.
Netlisting for Simulation

This section describes the following:

- Netlisting for Digital Simulation on page 454
- Netlisting for Synthesizing a Design in Synplify on page 456
- Netlisting for Analog and Mixed Signal Simulation on page 460

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.
   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 the 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 the 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.

   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.

**Netlisting for Synthesizing a Design in Synplify**

Design Entry HDL allows you to generate the Verilog and VHDL netlist that will be used to pass the entire design for synthesis to Synplify, the synthesis tool from Synplicity. You can specify the options for generating the netlist for synthesizing a design.

**To specify the options for generating the Verilog 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. Click the Options button next to the Verilog check box.

   The Verilog Netlist dialog box appears.

4. Select the Verbose Output check box to log the debug messages of the netlisting process in the `hdldir.log` file located in the `<project_directory>/temp/` directory.

5. Select the Analyze on Save check box if you want the netlist to be analyzed by `ncvhdl.exe` (NC VHDL compiler) or `cv.exe` (Leapfrog compiler) every time you save the design in Design Entry HDL.

   This check box is enabled only if you have selected the NC VHDL or Leapfrog simulator in the Tools tab of the Project Setup dialog box. If the NC VHDL simulator is selected, the netlist is analyzed by `ncvhdl.exe`. If the Leapfrog simulator is selected, the netlist is analyzed by `cv.exe`.

6. Select the Check InstanceVs 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."

7. 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 exceed the number specified here, Design Entry HDL will not generate the netlist.

8. Specify the timescale directive for the Verilog module of the schematic in the Time Scale field. The default value is `1ns/1ns`.

9. Specify the logic type for all nets in the design. You can use any legal Verilog net type, such as WIRE, WAND, and WOR. The default value is WIRE.

   Note: The specified net type applies to all drawings in the design. You can override the net type for individual drawings by using the `VLOG_NET_TYPE` property on a VERILOG_DECS symbol.

10. Specify the signal names for the Verilog net type Supply 0.

11. Specify the signal names for the Verilog net type Supply 1.
12. Click OK to save the settings and to close the Verilog Netlist dialog box.

13. Click OK to close the Design Entry HDL Options 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 hdldir.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 InstanceVs 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 \texttt{STD\_LOGIC} and \texttt{BIT}. The default value is \texttt{STD\_LOGIC}.

\textbf{Note:} The specified scalar type applies to all drawings in the design. You can override the scalar type for individual drawings by using the \texttt{VHDL\_SCALAR\_TYPE} property on a \texttt{VHDL\_DECS} symbol. For more information on specifying the VHDL logic type of ports and signals, see \textit{Setting the VHDL Logic Type for Ports and Signals} on page 200.

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, \texttt{IEEE} will be used as the default library.

\textbf{Note:} The default library (\texttt{IEEE}) is not used if you specify any other library. If you want to use the \texttt{IEEE} library along with other libraries, you must explicitly add the \texttt{IEEE} library.

You can also add libraries for a drawing by using the \texttt{LIBRARY} property on a \texttt{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, \texttt{IEEE.STD\_LOGIC\_1164.ALL} will be used as the default.

\textbf{Note:} The \texttt{IEEE.STD\_LOGIC\_1164.ALL} use clause will not be used if you add any other use clauses. If you want to use \texttt{IEEE.STD\_LOGIC\_1164.ALL} along with any other use clause, you must explicitly add \texttt{IEEE.STD\_LOGIC\_1164.ALL}.

You can also add use clauses for a drawing by using the \texttt{USE} property on a \texttt{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 \textit{OK} to save the settings and close the \textit{VHDL Netlist} dialog box.

14. Select the \textit{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 \textit{Programmable IC Tutorial}.

15. Click \textit{OK} to close the \textit{Design Entry HDL Options} dialog box.

\textit{To generate the netlist for synthesizing a design}

➤ In Design Entry HDL, choose \textit{File – Save}.  

November 2008 459 Product Version 16.2
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/
```

⚠️ **Important**

If you are working on a large design, saving the design in Design Entry HDL takes a long time because the netlist for the entire design is regenerated every time you save the design. You can avoid this by disabling netlisting of the design. For more information, see [Disabling Netlisting of Designs](#) on page 452.

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 the file in the `Markers` dialog box. The `hdldir.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 14.0 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-design_name>.net` will be generated in the following directory:

```plaintext
<project_directory>\worklib\<design_name>\cfg_analog\n
```

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 `hdldir.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.
Netlisting Read-only Blocks Used in a Design

When the design is netlisted, Design Entry HDL generates files in the libraries used in the design. If you use blocks from read-only libraries in your design, Design Entry HDL may try to write files in the read-only libraries when the design is netlisted. If this occurs, the following error message is displayed:

```
NTL_ERROR : Need to analyze the design for cell : <cell name> in library : <library name>" root->cellName, root->libName
Cell : <cell name> in library : <library name> is marked as read only", root->cellName, root->libName
Please provide write permissions in the cell for analysing the design.
You can use the TMP directive in cds.lib file. Please create the cell in the TMP library.
```

Do one of the following:

- Provide write permissions in the cell containing the read-only block
- Use the TMP attribute in the cds.lib file to specify a temp location for the library containing the read-only block. The TMP attribute is used in the cds.lib file as below:

  ```
  ASSIGN <name of library containing read-only block> TMP <name of temp directory>
  ```

The usage of the TMP attribute is explained using the following figure.

Directory Structure of a Project

The library ic_lib is a read-only library. The cell ic_reset contains the read-only block used in the design ic_design. When the design is netlisted, if Design Entry HDL displays the error message listed above, use the TMP attribute in the cds.lib file as below:

```
DEFINE ic_lib ./ic_lib
ASSIGN ic_lib TMP ./ic_lib_tmp
```

The first entry defines the ic_lib library. The second entry assigns the attribute TMP to the library defined as ic_lib. The value of TMP is ./ic_lib_tmp.
The following figure displays the directory structure after the design is re-netlisted.

- A directory `ic_lib_temp`  
  This is the temp location for the read-only library `ic_lib`.  
- a directory `ic_reset` under `ic_lib_temp`  
  This is the temp location for the cell `ic_reset` that contains the read-only block used in the design.  
- a directory `sch_1` under `ic_reset`  
  This is the sch_1 view for the cell `ic_reset`. All the files generated by Design Entry HDL when the read-only block `ic_reset` is netlisted are written to the specific views.
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 and UNIX Platforms
- HPF Plotting on UNIX Platforms
- Hierarchical Plotting

Windows Plotting on Windows and UNIX Platforms

Although this facility is available on both Windows and UNIX platforms, it 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. On UNIX, you can configure plotting information through the Add Printers Wizard.

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.


The Design Entry HDL Options dialog box appears.
**Note:** On UNIX, additional options are given, to choose between Windows and HPF plotting. 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**

From 15.5 onwards, 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
   
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, which was the only font available prior to the 15.0 release.

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. On UNIX, this button should be used to specify a filename in case you want to plot to a file.

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**

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.
Note: If you have certain text (properties or notes) in the design such that two or more lines of text overlap as shown below:
In the plot of the design, some parts of the text can get truncated or hidden depending on the font size as shown below:

```
| 83P | J6 | 5 |
| 84P | J6 | 6 |
| 87P | J6 | 7 |
| 64P | J6 | 8 |
| 38P | J6 | 1 |
| 37P | J6 | 2 |
| 36P | J6 | 3 |
| 35P | J6 | 4 |
| 63P | J6 | 5 |
| 58P | J6 | 6 |
| 59P | J6 | 7 |
| 60P | J6 | 8 |
| 61P | J6 | 9 |
| 62P | J6 | 10 |
| 57P | J6 | 11 |
| 56P | J6 | 12 |
| 52P | J6 | 13 |
| 48P | J6 | 14 |
| 44P | J6 | 15 |
```

**Plotting the Design**

1. Choose *File – Plot.*
The *Plot* dialog box appears.

![Plot dialog box](image)

**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.

   **Note:** On UNIX, this check box does not appear. To plot to a file on UNIX, click the *Properties* button. Select the *File* or *Encapsulated PostScript File* option and specify the name of the file.

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, 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.
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 504.

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

---

**Important**

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 and UNIX

### Setup Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET WPLOT_SPOOLed</td>
<td>Sets the plot output to be directed to a file.</td>
</tr>
<tr>
<td>SET WPLOT_FILE [&lt;path&gt;/]</td>
<td>Sets the filename or path for the plot file you generate.</td>
</tr>
<tr>
<td></td>
<td>If you do not set a filename, Design Entry HDL plots to a plot file named output.ps.</td>
</tr>
<tr>
<td></td>
<td>The set wplot_spooled command sets the print to file option as the default.</td>
</tr>
<tr>
<td></td>
<td>To set this option for a given project in all sessions and to specify the name of the</td>
</tr>
<tr>
<td></td>
<td>generated postscript file, add the following directives manually in the project file</td>
</tr>
<tr>
<td></td>
<td>(project_name.cpm):</td>
</tr>
<tr>
<td></td>
<td>PLOT_TO_FILE ‘YES’</td>
</tr>
<tr>
<td></td>
<td>PLOT_FILE_NAME ‘output.ps’</td>
</tr>
<tr>
<td></td>
<td>The format of the plot file depends on the plotter driver. The format can be PCL, PS,</td>
</tr>
<tr>
<td></td>
<td>or any other.</td>
</tr>
<tr>
<td>SET WPLOT_LOCAL</td>
<td>Sets the plot output to be directed to a plotter.</td>
</tr>
<tr>
<td>SET WPLOT_THIN_width</td>
<td>Sets the width of thin lines in plots. The default is 5.</td>
</tr>
<tr>
<td></td>
<td>Sets the width of thick lines in plots. The default is 10.</td>
</tr>
<tr>
<td>SET WPLOT_SCREEN</td>
<td>Sets the plot method to plot screen contents, clipping the drawing outside the display</td>
</tr>
<tr>
<td></td>
<td>screen.</td>
</tr>
<tr>
<td>SET WPLOT_SHEET</td>
<td>Sets the plot method to plot sheet contents.</td>
</tr>
<tr>
<td>SET WPLOT_ADJust</td>
<td>Sets the adjust to scale option.</td>
</tr>
</tbody>
</table>
### Allegro Design Entry HDL User Guide
### Plotting Your Design

<table>
<thead>
<tr>
<th>Command</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET WPLOT_SCALE &lt;scale_value&gt;</td>
<td>Adjusts the scale value of the plot to a given value.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This command must be preceded by set wplot_adjust.</td>
</tr>
<tr>
<td></td>
<td>Example: If you have set the scale value to 50% in your project file, you</td>
</tr>
<tr>
<td></td>
<td>can change the scale value to 100% by using the command:</td>
</tr>
<tr>
<td></td>
<td>set wplot_adjust</td>
</tr>
<tr>
<td></td>
<td>set wplot_scale 100</td>
</tr>
<tr>
<td>SET WPLOT_FIT_to_page</td>
<td>Sets the plot to fit the paper size specified.</td>
</tr>
<tr>
<td>SET WPLOT_LANDscape</td>
<td>Sets the plot orientation to landscape</td>
</tr>
<tr>
<td>SET WPLOT_PORTrait</td>
<td>Sets the plot orientation to portrait</td>
</tr>
<tr>
<td>SET WPLOT_DEFault</td>
<td>Instructs Design Entry HDL to read and set defaults from the directives in</td>
</tr>
<tr>
<td></td>
<td>the project file.</td>
</tr>
<tr>
<td>SET WPLOT_PAPER</td>
<td>Sets the paper size for plotting.</td>
</tr>
<tr>
<td></td>
<td>To see the standard paper sizes, refer Paper Sizes Supported by Design Entry</td>
</tr>
<tr>
<td></td>
<td>HDL on page 480.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>set wplot_paper A4</td>
</tr>
<tr>
<td></td>
<td>If you set a paper size not supported by the current printer, an error</td>
</tr>
<tr>
<td></td>
<td>message is displayed.</td>
</tr>
<tr>
<td>SET WPLOT_PLOTTER &lt;plotter_name&gt;</td>
<td>Sets the plotter to the name specified. If the plotter name</td>
</tr>
<tr>
<td></td>
<td>consists of any special characters, precede the first such character</td>
</tr>
<tr>
<td></td>
<td>with a \ and put the plotter name within quotes.</td>
</tr>
</tbody>
</table>
**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

**Example:** To set the font to Verdana, use

```
set fface 2
```

---

**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.

<table>
<thead>
<tr>
<th>Directive</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLOT_SINGLE_WIDTH</td>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>PLOT_DOUBLE_WIDTH</td>
<td>10</td>
<td>This directive corresponds to the double line width field in the Plot dialog box. It specifies the width of thick wires and buses.</td>
</tr>
</tbody>
</table>
### Directive Default Value Description

<table>
<thead>
<tr>
<th>Directive</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLOT_SCALE</td>
<td>100</td>
<td>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.</td>
</tr>
<tr>
<td>PLOT_FIT_TO_PAGE</td>
<td>OFF</td>
<td>This directive, when set to ON, directs Design Entry HDL to adjust the plot according to page size.</td>
</tr>
<tr>
<td>PLOT_SCREEN</td>
<td>OFF</td>
<td>This directive, when set to ON, directs Design Entry HDL to plot the portion of the schematic that is displayed on the screen.</td>
</tr>
<tr>
<td>PLOT_COLOR</td>
<td>OFF</td>
<td>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.</td>
</tr>
<tr>
<td>PAPER_ORIENTATION</td>
<td>1</td>
<td>This directive sets the orientation of the plot output. You can set it to 1 for Portrait or 2 for Landscape.</td>
</tr>
<tr>
<td>PAPER_SIZE</td>
<td>9</td>
<td>The number shows the index (0-based) number of the paper size selected in the combo box in the Plot Setup 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 Plot Setup 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.</td>
</tr>
</tbody>
</table>
Allegro Design Entry HDL User Guide
Plotting Your Design

Plot Command

The syntax of the plot command is

plot [<lib>].[<cell>].[<view>].[<ver>].[<page>]

Examples:

<table>
<thead>
<tr>
<th>Command</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plot</td>
<td>Plots the currently opened drawing</td>
</tr>
<tr>
<td>Plot cache</td>
<td>Plots all pages in the cache</td>
</tr>
<tr>
<td>Plot cache.sym.1.1</td>
<td>Plots the symbol view in the cache</td>
</tr>
<tr>
<td>Plot cache.sym.1.2</td>
<td>Plots the page 2 schematic of the cache</td>
</tr>
<tr>
<td>Plot cache.sch.1.*</td>
<td>Plots all pages of version 1</td>
</tr>
</tbody>
</table>

where cache is the name of the drawing.

Note: Wildcard characters are supported for page numbers only.

Directive Default Description

PAPER_SOURCE 4 The number shows the index (0-based) number of the paper source selected in the combo box in the Plot Setup dialog box. This directive is useful if you want to use one paper source throughout the site or for all your designs.

It is recommended that you change the paper size through the Plot Setup 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.

Note: All plotters may not support all the available paper sources.
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.

<table>
<thead>
<tr>
<th>Directive</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLOT_TO_FILE</td>
<td>NO</td>
<td>This directive tells the plot command to direct its plot output to a file. The file will by default be output.ps. Possible values: NO, YES</td>
</tr>
<tr>
<td>PLOT_FILE_NAME</td>
<td></td>
<td>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 output.ps and it is created in the project directory.</td>
</tr>
</tbody>
</table>

**Paper Sizes Supported by Design Entry HDL**

The possible paper sizes that you can use are:

<table>
<thead>
<tr>
<th>Paper Type</th>
<th>Paper Size</th>
<th>Paper Type</th>
<th>Paper Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LETTER</td>
<td>8.5x11&quot;</td>
<td>EXECUTIVE</td>
<td>7.5x10&quot;</td>
</tr>
<tr>
<td>LEDGER</td>
<td>17x11&quot;</td>
<td>STATEMENT</td>
<td>5.5x8.5&quot;</td>
</tr>
<tr>
<td>A3</td>
<td>297x420mm</td>
<td>A4SMALL</td>
<td>210x297mm</td>
</tr>
<tr>
<td>B4</td>
<td>250x354mm</td>
<td>FOLIO</td>
<td>8.5x13&quot;</td>
</tr>
<tr>
<td>10X14</td>
<td>10X14&quot;</td>
<td>NOTE</td>
<td>8.5x11&quot;</td>
</tr>
<tr>
<td>ENV_10</td>
<td>4.125x9.5&quot;</td>
<td>ENV_C3</td>
<td>12.96x18.32&quot;</td>
</tr>
<tr>
<td>CSHEET</td>
<td>17x22&quot;</td>
<td>ENV_B6</td>
<td>7.04x5&quot;</td>
</tr>
<tr>
<td>ENV_C5</td>
<td>6.48x9.16&quot;</td>
<td>ENV_C4</td>
<td>9.16x12.96&quot;</td>
</tr>
<tr>
<td>ENV_C65</td>
<td>4.56x9.16&quot;</td>
<td>ENV_B4</td>
<td>10x14.12&quot;</td>
</tr>
<tr>
<td>ENV_11</td>
<td>4.5x10.375&quot;</td>
<td>DSHEET</td>
<td>22x34&quot;</td>
</tr>
<tr>
<td>ENV_12</td>
<td>4x11&quot;</td>
<td>ESHEET</td>
<td>34x44&quot;</td>
</tr>
</tbody>
</table>
Note: You should use only those paper sizes that are supported by the plotter you have chosen to plot the design on.

### HPF Plotting on UNIX Platforms

From Design Entry HDL, you can either plot a drawing directly or create a plot file to be printed at a later time or to be physically transferred to another system. In either case, you use the `hardcopy` command.

Design Entry HDL calls a utility called `hpfhdl` to perform its plots. You can also run `hpfhdl` separately from a UNIX shell window.

⚠️ **Important**

Images are not plotted in HPF plotting.

You can plot drawings in the HPF plotting mode:

- From Design Entry HDL using the *HPF Plot* dialog box
  
  For more information, see *Plotting the Design* on page 487.

- From Design Entry HDL using the *hardcopy* console command
  
  For more information, see *Plotting the Design from the Console Window* on page 489.

- From a UNIX shell using the `hpfhdl` utility

---

<table>
<thead>
<tr>
<th>Paper Type</th>
<th>Paper Size</th>
<th>Paper Type</th>
<th>Paper Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LETTERSMALL</td>
<td>8.5x11&quot;</td>
<td>B5</td>
<td>182x257mm</td>
</tr>
<tr>
<td>LEGAL</td>
<td>8.5x14&quot;</td>
<td>11X17</td>
<td>11x17&quot;</td>
</tr>
<tr>
<td>A4</td>
<td>210x297mm</td>
<td>ENV_PERSONAL</td>
<td>3.625x6.5&quot;</td>
</tr>
<tr>
<td>ENV_9</td>
<td>3.875x8.875&quot;</td>
<td>A5</td>
<td>148x210mm</td>
</tr>
<tr>
<td>ENV_DL</td>
<td>4.4x8.8&quot;</td>
<td>ENV_14</td>
<td>5x11.5&quot;</td>
</tr>
<tr>
<td>FANFOLD_STD_GERMAN</td>
<td>8.5x12&quot;</td>
<td>ENV_MONARCH</td>
<td>3.875x7.5&quot;</td>
</tr>
<tr>
<td>FANFOLD_LGL_GERMAN</td>
<td>8.5x13&quot;</td>
<td>ENV_B5</td>
<td>7.04x10&quot;</td>
</tr>
<tr>
<td>FANFOLD_US</td>
<td>14.875x11&quot;</td>
<td>ENV_ITALY</td>
<td>110x230mm</td>
</tr>
<tr>
<td>QUARTO</td>
<td>215x275mm</td>
<td>ENV_C6</td>
<td>4.56x6.48&quot;</td>
</tr>
</tbody>
</table>
You cannot plot occurrence properties or hierarchical drawings using the `hpfhdl` utility. For more information, see Plotting Drawings from the UNIX Shell on page 498.

**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,
    
    The Plotting tab of the Design Entry HDL Options dialog box appears.
  
  - b. Select HPF as the plotting facility.

- From the Tools menu,
  
    
    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,
  
    
    The HPF Plot dialog box appears.
  
  - b. Click Setup.
The setup options for HPF plotting appear.

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, `hpgl2` will be displayed in the *Plotter* drop-down list.

```plaintext
hpgl2|Hewlett-Packard 7600 Series Electrostatic: \
:manufacturer=Hewlett-Packard: \
:type=hpgl2: \
```
The first plotter defined in the .cdsplotinit file is the default plotter. For more information on the .cdsplotinit file, see Customizing HPF Plotting on UNIX Platform on page 518.

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.

```plaintext
hpgl2|Hewlett-Packard 7600 Series Electrostatic: \\
:manufacturer=Hewlett-Packard: \\
:type=hpgl2: \\
:maximumPages#10: \\
```
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.

**Plot To File Options**

- **Location:** Enter the location of the file in which the design is to be plotted. The default location is the current working directory.

- **Single File**
  - **Name:** Enter the name of the file in which to print the pages. The default file name is `vw.spool`.

- **File Per Page**

**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.

6. To scale the drawing, select Default Scaling, Scale to Page Size, or Scale by Factor.

<table>
<thead>
<tr>
<th>Select</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Scaling</td>
<td>Plot the drawing as it is. In other words, the default scale factor 1 is used.</td>
</tr>
<tr>
<td>Scale to Page Size</td>
<td>Scale the drawing to be plotted to the page size you select.</td>
</tr>
</tbody>
</table>

Note: The paper size that you select need not be defined for the plotter in the .cdsplotinit file.

This option is similar to the Fit To Page 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 Specify Page Size field and select A in the Scale to Page Size drop-down list.

Note: If the paper size you select in the Scale to Page Size field is not the same or smaller than the paper size you specified in the Specify Page Size 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 Scale to Page Size drop-down list.

<table>
<thead>
<tr>
<th>Scale by Factor</th>
<th>Specify the factor by which you want to scale the plot.</th>
</tr>
</thead>
</table>

For example, a scale factor of 0.5 will create a plot size that is half the drawing size.

Note: If the scale factor results in a plot size that is larger than the paper size you have specified in the Specify Page Size field, the drawing will be plotted in multiple sheets of the paper size you specified in the Specify Page Size 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.
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.

**Note:** If you want to plot occurrence properties, you must change to the Occurrence Edit mode before plotting the schematic. For more information, see *Occurrence Edit Mode* on page 129.

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 482.

   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 482.

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 504.

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.
   
   This field is disabled if you are in the Occurrence Edit mode. 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.

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 specified in the *Library* field. You can use wildcards (* and ?) in this field.
   
   This field is disabled if you are in Occurrence Edit mode. In 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.

6. 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 *schcref_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.
   
   This field is disabled if you are in the Occurrence Edit mode. 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.

7. 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.

This field is disabled if you are in the Occurrence Edit mode. 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.

8. 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.

9. 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 494.

**Note:** If you want to plot occurrence properties, you must change to the Occurrence Edit mode before plotting the schematic. For more information, see `Occurrence Edit Mode` on page 129.

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 490
- Specifying the Default Paper Size on page 490
- Setting Plotting to a Plotter on page 491
- Setting Plotting to a File on page 492
- Specifying the Plotter on page 492

**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:

```plaintext
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.

```plaintext
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**

```plaintext
set local
```

or

```plaintext
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.

<table>
<thead>
<tr>
<th>Directive</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPF_BATCH</td>
<td>Yes, No</td>
<td>This directive corresponds to the Plot to File check box. It is used to direct Design Entry HDL to spool the plot to the vw.spool file.</td>
</tr>
<tr>
<td>HPF_PLOTTER</td>
<td>Name of the plotter</td>
<td>This directive corresponds to the Plotter field. It is used to specify the name of the plotter in the .cdsplotinit file.</td>
</tr>
<tr>
<td>HPF_SPEC_PLOT_PAGESIZE</td>
<td>Yes, No</td>
<td>This directive corresponds to the check box next to the Specify Page Size field. It is used to set the paper size.</td>
</tr>
<tr>
<td>HPF_PLOT_PAGESIZE</td>
<td>Page size</td>
<td>This directive corresponds to the Specify Page Size field. It specifies the paper on which to plot.</td>
</tr>
<tr>
<td>HPF_SCALETYPE</td>
<td>Default, Scale by factor, Scale to Page Size</td>
<td>This directive corresponds to the Default, Scale by Factor, and Scale to Page Size fields. It specifies the mode of scaling to be used for plotting.</td>
</tr>
<tr>
<td>HPF_PAGESIZE</td>
<td>A, B, C, D, E, F</td>
<td>This directive corresponds to the Scale to Page Size field. It specifies the standard page to which the design gets scaled.</td>
</tr>
<tr>
<td>HPF_BUS_SCALEFACTOR</td>
<td>A number</td>
<td>This directive corresponds to the Bus Scale Factor field. It specifies the width of buses in the design.</td>
</tr>
<tr>
<td>HPF_WIRE_SCALEFACTOR</td>
<td>A number</td>
<td>This directive corresponds to the Wire Scale Factor field. It specifies the width of wires, text, and component boundaries in the design.</td>
</tr>
</tbody>
</table>
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 ]
```

<table>
<thead>
<tr>
<th><strong>scale_factor</strong></th>
<th>A factor applied to the drawing to determine the final plot size. The default factor is 1.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specify this option if you want to create a bigger or smaller plot size for the drawing.</td>
</tr>
<tr>
<td></td>
<td>➤ To create a plot size of half the drawing size, use a factor of 0.5.</td>
</tr>
<tr>
<td></td>
<td>➤ To plot twice the drawing size, use a factor of 2.</td>
</tr>
</tbody>
</table>

**Directive Values Description**

- **HPF_FONT**
  - **Values**: A font name
  - **Description**: This directive corresponds to the *Font* field. It specifies the font style to be used for plotting.

- **HPF_SCALEFACTOR**
  - **Values**: A number
  - **Description**: This directive corresponds to the *Scale by Factor* field. It specifies the scaling factor.
**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 490.

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.
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 61.

- 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
  - schcref_1 for the schcref_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 490.

<table>
<thead>
<tr>
<th>Command</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>ha</td>
<td>Plots the current drawing at the default scale factor 1 on B size paper.</td>
</tr>
<tr>
<td>ha b</td>
<td>Scales the current drawing to paper size B and plots it on B size paper.</td>
</tr>
<tr>
<td>ha 1 *</td>
<td>Plots all the schematic drawings, symbol drawings, and the drawings generated by CRefer (in the CREFOUT and SCHCREF_1 views) that are present in the design. The drawings are plotted on B size paper.</td>
</tr>
<tr>
<td>ha a cache.sch.1.1</td>
<td>Scales the first schematic drawing present in the sch_1 view of the cache cell to paper size A and plots the drawing on B size paper.</td>
</tr>
<tr>
<td>ha 1 &lt;memory&gt;cache.sch.1.1</td>
<td>Plots the first schematic drawing present in the sch_1 view of the cache cell in the memory library on B size paper.</td>
</tr>
<tr>
<td>ha 1 cache.schcref_1.1</td>
<td>Plots the first schematic drawing present in the schcref_1 view of the cache cell on B size paper.</td>
</tr>
<tr>
<td>ha 1 cache.sch.*</td>
<td>Plots all the schematic drawings present in all the schematic views (sch_1, sch_2, sch_3 and so on) of the cache cell. The drawings are plotted on B size paper.</td>
</tr>
<tr>
<td>ha 1 <em>.sch.</em></td>
<td>Plots all the schematic drawings that are present in all the schematic views (sch_1, sch_2, sch_3 etc.) of all cells in all local libraries used the project. The drawings are plotted on B size paper.</td>
</tr>
</tbody>
</table>
## Plotting Your Design

### Plotting Drawings from the UNIX Shell

You can plot drawings from outside Design Entry HDL (from a shell window) using the `hpfhdl` utility. The `hpfhdl` utility uses the same input file format as the `hardcopy` command uses within Design Entry HDL. So, you can plot

- ASCII vectorized format files
- ASCII component (body) files
- Design Entry HDL binary format files

**Important**

The `hpfhdl` utility does not allow you to plot custom text and occurrence properties. For more information on custom text and occurrence properties, see Chapter 8, “Working with Properties and Text.”

See the following sections for more information:

- Setting Up the `hpfhdl` Utility on page 499
- Using the `hpfhdl` Utility on page 499
- Header File Format on page 500
- hpfhdl Command Syntax on page 503

<table>
<thead>
<tr>
<th>Command</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>ha 1 <em>.schcref_1.</em></td>
<td>Plots schematic drawings generated by CRefer in the SCHREF_1 view. The drawings are plotted on B size paper.</td>
</tr>
<tr>
<td>ha 2 &lt;lsttl&gt;ls154.sym.1.1</td>
<td>Plots the symbol view of the ls154 part in the lsttl library on B size paper using a scale factor of 2.</td>
</tr>
<tr>
<td>ha 2 &lt;lsttl&gt;ls154.*</td>
<td>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.</td>
</tr>
</tbody>
</table>
Setting Up the hpfhdl Utility

The system location of the hpfhdl utility is `<your_install_dir>/tools/editor/lib/`. To run the hpfhdl utility from a UNIX terminal, you have to set environment variables as given below:

**On Sun Solaris**

```
set path (<your_install_dir>/tools/editor/lib $path)
setenv LD_LIBRARY_PATH <your_install_dir>/tools/lib $LD_LIBRARY_PATH
```

**On IBM AIX**

```
set path (<your_install_dir>/tools/editor/lib $path)
setenv LIBPATH <your_install_dir>/tools/lib $LIBPATH
```

**On HP-UX**

```
set path (<your_install_dir>/tools/editor/lib $path)
setenv SHLIB_PATH <your_install_dir>/tools/lib $SHLIB_PATH
```

Using the hpfhdl Utility

You can create a file that you can plot later, or send directly to print. For using the hpfhdl utility, do the following:

1. In your design directory, use a text editor to create a header file with information on the plotter, the font, the scale, and the file format. These header fields are described in **Header File Format**.

   **Sample Header File**

   ```
   hp7580
   1 vector_font
   B 500
   B
   <path to project_dir>/worklib
   <your_install_dir>/share/library/standard
   <your_install_dir>/share/library/fsttl
   ```

   **Note:** There must be no extra lines in the header file after the last directory path for hpfhdl to work properly.

2. Identify the name of the binary drawing file that you want to plot.

   Each drawing directory contains a drawing file in the binary format. The filename is `pageN.csb`. For example, if you want to plot the first page of a schematic `mylogic`, the binary file for the schematic page is `page1.csb`, which is located at:
In this example, we will plot the binary file `/<project_dir>/worklib/mylogic/sch_1/page1.csb`.

3. Enter the `hpfhdl` command with parameters as follows:
   - To create a plot file named `myoutputfile` to print later:
     ```
     hpfhdl -f myoutputfile -2 myheadername.header worklib/mylogic/sch_1/page1.csb
     ```
     where `myheadername.header` is the name of the header file.
   - To print directly to the plotter specified in the header file:
     ```
     hpfhdl -2 myheadername.header worklib/mylogic/sch_1/page1.csb
     ```

**Header File Format**

The input file that `hpfhdl` reads has a special header containing information about the type of the plotter, the line width, the scale, and the format of the graphical information. If the file is binary, the header includes a list of paths to the directories where the `hpfhdl` utility searches to find the referenced symbols in the binary file for the drawing.

Header fields and options are

- **Plotter name**
  - Any plotter name in the `.cdsplotinit` file

- **Line weight**
  - NORMAL_WEIGHT lines (1)
  - HEAVY_WEIGHT lines (2)

- **Font type**
  - The font type parameter appears on the same line as the line weight parameter. Font type is optional. If you do not specify a font type, the `vector_font` type is used for plotting. The following font types are available:
    - `vector_font` (default)
    - `cursive_font`
    - `valid_font`
    - `greek_font`
    - `milspec_font`
Plotting Your Design

- symbol_font
- gothic_font
- native_font (defaults to Courier)
- Courier
- Times New Roman
- Helvetica

⚠️ Important
If you use resident (native) fonts, the .cdsplotinit file must include the following entry:
:residentFonts:

■ Scale
The scale can be
- An ASCII positive real number string
- Any paper size specified for the plotter in the .cdsplotinit file

■ Coordinates-per-inch
These are the plotter coordinates and they appear on the same line as the scale parameter. Coordinates-per-inch is an ASCII integer; hpfhd1 multiplies the incoming coordinates by the scale to get correct plotter coordinates.

Design Entry HDL uses the following plotter coordinates:

<table>
<thead>
<tr>
<th>Coordinates-Per-Inch</th>
<th>For Plot Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>Decimal</td>
</tr>
<tr>
<td>508</td>
<td>Metric</td>
</tr>
<tr>
<td>400</td>
<td>Fractional</td>
</tr>
</tbody>
</table>

■ Encoding Type
An encoding type can be

V  Vectorized
An illegal encoding type causes the `hpfhdl` program to terminate.

**Directory Paths to Libraries**

A list of directory paths follows the encoding type only if the encoding type is binary (B). Specify the absolute directory paths to all the libraries used in your design. The directory paths are used to find the referenced symbols in the binary file for the drawing. For example, if you have used the `MERGE` symbol from the `standard` library in your schematic page, the `MERGE` symbol is referenced in the binary file for the schematic page. The `hpfhdl` utility will be able to plot the `MERGE` symbol in the schematic only if you have given the path to the `standard` library in the header file as below:

```
<your_install_dir>/share/library/standard
```

If you do not give the path to the `standard` library in the header file, `hpfhdl` displays the following error message when you plot the schematic page and the `MERGE` symbol will not be plotted correctly.

```
Error locating body ‘MERGE’ version 1
```

**Note:** Do not use environment variables to specify the directory path to a library. For example, if you use the `$INST_DIR` environment variable to point to the installation directory for Cadence tools, do not specify the path to the `standard` library in the header file as below:

```
$INST_DIR/share/library/standard
```

**Caution**

*There must be no extra lines in the header file after the last directory path or `hpfhdl` will not work properly.*

The following table shows three sample headers.

<table>
<thead>
<tr>
<th>Headers</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>hp7580</td>
<td>HP 7580 pen plotter</td>
</tr>
<tr>
<td>1 milspec_font</td>
<td>NORMAL lines, Milspec font</td>
</tr>
<tr>
<td>D 500</td>
<td>scaled to D size, 500 units per inch</td>
</tr>
<tr>
<td>V</td>
<td>vectorized format file</td>
</tr>
</tbody>
</table>
Note: You must not include comments in a header file. Comments are included in these examples for your information only.

### hpfhdl Command Syntax

hpfhdl [-f|-v outputfile] [-o] [-p papersize] [-2 <headerfile> <path_to_drawing>]

- **-f**  
  Writes the data to a new version of the output file

- **-v**  
  Writes a vector format file to a new version of the output file

- **outputfile**  
  If no output file is specified, the output is sent to the printer or the plotter

- **-o**  
  This parameter operates if you specify the -f option, which implies that hpfhdl overwrites the new version of the output file (-f specification) rather than append data to the file specified, which is the default behavior.

- **-p**  
  Specifies the paper size. This value must already be defined for the plotter in the .cdsplotinit file. The default is the first paper size entry defined for the plotter in the .cdsplotinit file.

- **headerfile**  
  The name of the header file. For more information, see Header File Format
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 both Windows Plotting and HPF Plotting. For details on Windows Plotting, refer to Windows Plotting on Windows and UNIX Platforms on page 465. For details on Hierarchical Plotting, refer to HPF Plotting on UNIX Platforms on page 481.

The following topics are discussed in this section:

- Hierarchical Plotting in Hierarchy, Expanded, and Occurrence Edit Modes on page 504
- Changing the Order in Which Designs Are Plotted on page 506
- Plotting Hierarchical Designs on page 508

Hierarchical Plotting in Hierarchy, Expanded, and Occurrence Edit Modes

Hierarchical plotting is supported in Hierarchy, Expanded, and Occurrence Edit modes. In the Occurrence Edit mode, occurrence properties are also plotted. When you plot a hierarchical design with multiple instantiations of a block in the Occurrence Edit mode, Design Entry HDL plots every occurrence of the block. When you plot the same design in the Hierarchy mode or Expanded mode, Design Entry HDL plots the block only once.

The following figure shows how the hierarchy is displayed in the Hierarchy and Expanded modes:
Hierarchy in the Hierarchy and Expanded Modes

Notice that two instances of the block `CLOCK` is displayed in the hierarchy. This means that the block `CLOCK` is instantiated twice in the design. Note that the second instance of the block is grayed out. This is because, in the Hierarchy or Expanded mode, only one occurrence of a block is plotted by default. Select the `Plot All Occurrences` check box if you want to plot all occurrences of a block.

The following figure shows how the hierarchy is displayed in the Occurrence Edit mode:
Hierarchy in Occurrence Edit Mode

In the Occurrence Edit mode, both the occurrences of the block **CLOCK** are enabled. This is because, when you plot in the Occurrence Edit mode, Design Entry HDL plots each occurrence of a block. The occurrence properties specified on each instance of a block are also plotted.

For more information on working in the Hierarchy, Expanded, or Occurrence Edit mode, see **Modes in Design Entry HDL** on page 127.

**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 396.

Let's take the following example to see how module ordering impacts hierarchical plotting:
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 13-2 Hierarchy After Module Ordering**

Hierarchy

- [ ] Plot All Occurrences
- [ ] poa (1)
  - [ ] flashcard <page1_i11> (2)
  - [ ] clock <page1_i17> (3)
  - [ ] clock <page1_i30> (4)
  - [ ] exterior io port <page1_i19> (5)
  - [ ] lcd display <page1_i2> (6)
  - [ ] rf transceiver <page1_i20> (7)
  - [ ] analog_io <page1_i23> (9)
  - [ ] power supply <page1_i26> (9)
  - [ ] memory_diag <page1_i27> (10)
  - [ ] keyboard <page1_i3> (25)
Notice that the schematic page numbers have also changed after module ordering. In Figure 13-1 on page 507, 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 416.

**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 424.

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 thelcd display module by performing module ordering.

**Figure 13-3 Hierarchy After Excluding the lcd display Module**

![Hierarchy After Excluding the lcd display Module](image)

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**

Hierarchical plotting is supported in Hierarchy, Expanded, and Occurrence Edit modes. If you want to plot occurrence properties, you must change to the Occurrence Edit mode before plotting the schematic. For more information on working in the Occurrence Edit mode, see Occurrence Edit Mode on page 129.

1. Choose *File – Save All*.

2. Choose *File – Plot*.
The *Plot* dialog box appears if you are performing Windows Plotting. The *HPF Plot* dialog box appears if you are performing HPF Plotting on UNIX.

3. Select *Hierarchy* to extend the *Plot* dialog box.

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 416.

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 416.

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.

In the Hierarchy or Expanded mode, only one occurrence of a block is plotted by default. There are two occurrences of the block clock in the above figure. The second instance of the block clock will be grayed out in the Hierarchy or Expanded mode. In the Occurrence Edit mode, all occurrences of the block clock will be plotted by default. For more information, see Hierarchical Plotting in Hierarchy, Expanded, and Occurrence Edit Modes on page 504.

5. Select the **Plot All Occurrences** check box if you want to plot both the occurrences of the block clock in the design.
**Note:** This check box is displayed only if you are in the Hierarchy or Expanded mode.

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 506.

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 `pca` in a grey background. This indicates that Design Entry HDL will plot `pca` 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.

Customization of Plotting on UNIX Platforms

This section describes the customization of Windows and HPF plotting on UNIX platforms.

Customizing Windows Plotting on UNIX Platform on page 512

Customizing HPF Plotting on UNIX Platform on page 518

Customizing Windows Plotting on UNIX Platform

Till release 14.2, to add a new printer as the destination for your plots, you would add printers in the win.ini file for Windows type plotting in UNIX. This resulted in the printers being displayed as available options from the Printer field of the Plot dialog box in Design Entry HDL. Release 15.0 onwards, Design Entry HDL uses Mainwin features to display the available printers in Windows.

Mainwin no longer reads any printer input from the win.ini file. It allows adding printers only through the Add Printers Wizard.
You use a control script for launching the Add Printers Wizard, which adds the printers that you can use for plotting. You need to use the control script to add printers for each system. The control script is available in the standard installation path.

Content of the control file:

```bash
#!/bin/sh

# CONCEPT_INST_DIR=`cds_root projmgr`;
CDS_TOOLS_PATH=$CONCEPT_INST_DIR/tools
CDS_FET_BIN=$CDS_TOOLS_PATH/fet/bin
PROG=mwcontrol

CONCEPT_INST_DIR=`cds_root projmgr`;
CDS_TOOLS_PATH=$CONCEPT_INST_DIR/tools
CDS_FET_BIN=$CDS_TOOLS_PATH/fet/bin
PROG=mwcontrol

if [ -f $CDS_FET_BIN/fet_env.sh ]; then
  . $CDS_FET_BIN/fet_env.sh "$@
else
  echo "$CDS_FET_BIN/fet_env.sh does not exist."
  echo "Please re-install CDS Project Manager kit properly."
  exit 1
fi

$PROG "$@
STATUS=$?
exit $STATUS"
will allow all users access to the script file. Alternatively, you can place the file at any location and make sure it is executable, and execute it using the appropriate path to the location.

2. In the UNIX shell/terminal/command window, type *control*.

This displays MainWin Control Panel.
3. Double click *Printers*.

![MainWin Printers](image)

4. Double click *Add New Printer*.

The Add Printer Wizard is displayed.

![Welcome to the Add Printer Wizard](image)
5. Click *Next*. 
6. Type printer name in the *Unix Printer* text box and click *Next*. 

---

**Print Command**

Please confirm the print command.

This command is the Unix command used to print. The default command will work for most users. Do not change the default command unless instructed to do so by your system administrator.

Note: Use %s to specify the temporary print file name in the print command.

To confirm the print command, please click Next.

Print Command:

```
/bin/sh -c "cat %s | lpr -P in13s02; rm %s"
```
7. Click Next.

8. Specify a name for the printer in the *Printer Name* text box. This is the name that will appear in the printer drop-down list.

9. Click Next.

10. Click *Finish*.

The new printer will now be displayed as an available option from the *Printer field of the Plot* dialog box in Design Entry HDL.

**Note:** To set up a site for multiple users with MainWin, specify a directory (for example, `/net/server/.mw`) with all the required printer information. All the users can then point to the directory by setting the following environment variable.

```bash
setenv ALLEGRO_MWUSER_DIR /net/server/.mw
```

All the users would require read-write permissions on the `/net/server/.mw` directory

**Customizing HPF Plotting on UNIX Platform**

HPF plotting can be customized on UNIX using the `.cdspplotinit` file.
Note: The .cdsplotinit file is only used by HPF plotting. It is not used by the MainWin control program.

Before you plot a drawing, or create a plot file, you must specify the plotter you want to use and plotter-specific parameters in the .cdsplotinit file. This file can reside in any of the three locations given below. Design Entry HDL looks for the file in the following order:

1. Your home directory
2. The design directory
3. <your_install_dir>/tools/plot

While you can build a site-specific .cdsplotinit file using the files listed below, Cadence recommends using the interactive plotconfig utility located in <your_install_dir>/tools/plot/bin.

- For model file entries: <your_install_dir>/tools/plot/etc.
- For a description of the options used in .cdsplotinit and more example model entries, see <your_install_dir>/tools/plot/samples/cdsplotinit.sample.

The primary purpose for .cdsplotinit is the mapping of the plotter name (also menu name) to the plotter type and to the printcap queue, as shown in the following sample .cdsplotinit entry:

The following sample .cdsplotinit file contains entries for commonly used plotting devices.

vers11|v80: 
   :spool=lp -Pvers11:\
   :query=lpq -Pvers11: \
   :remove=lprm -Pvers11 $3: \
   :manufacturer=Xerox Engineering Systems: \
   :type=intBW: \ 
   :maximumPages#10: \ 
   :resolution#200: \ 
   :compress: \ 
   :residentFonts: \ 
   :outtype=RASTER: \ 
   :instdir=/usr/valid: \ 
   :tmpdir=/usr/tmp: \
   :paperSize="B" 3200 2112:

postscript|Apple Laser Writer II NT/NTX: \
   :spool=lp -Ppostscript: \
   :query=lpq -Ppostscript: \
   :remove=lprm -Ppostscript $3: \
   :manufacturer=Apple: \
   :type=postscript1: \

For more information on the .cdsplotinit file, see the Plotter Configuration User Guide.

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.

Which are the plotters supported in HPF Plotting?

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 I have to set any environment variable to run the hpfhdl command from a UNIX terminal?

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 plot from the command line on UNIX?

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?

Can I plot occurrence properties?

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 SourceLink (http://sourcelink.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:

   
   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 [HPF Plotting on UNIX Platforms](#) on page 481.
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:

```console
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 396.

**Do I have to set any environment variable to run the hpfhdl command from a UNIX terminal?**

Yes. You have to set an environment variable to run the hpfhdl command from a UNIX terminal as below:

**On Sun Solaris**

Set the LD_LIBRARY_PATH environment variable as below:

```bash
setenv LD_LIBRARY_PATH <your_install_dir>/tools/editor/lib $LD_LIBRARY_PATH
```

**On IBM AIX**

Set the LIBPATH environment variable as below:

```bash
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 plot from the command line on UNIX?

On UNIX, use the hpfhdl command or the nconcepthdl command (the command that allows you to run Design Entry HDL scripts in non-graphical mode) to plot from the command line.

To use the nconcepthdl command, do the following:

1. Enter the hardcopy commands for the drawings that you want to plot in a script file, say myplot.scr.
2. Use the following command to plot from the command line:

   nconcepthdl -proj <project_name>.cpm -scr myplot.scr

**Note:** The nconcepthdl and hpfhdl commands do not allow you to plot occurrence properties from the command line. This is because it plots directly from the binary files for the drawing.

For more information on the nconcepthdl command, see *Nongraphical Design Entry HDL* in the Allegro Design Entry HDL Reference Guide.

See also, *Do I have to set any environment variable to run the hpfhdl command from a UNIX terminal?* on page 524.

### 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.

**Caution**

When you print a Design Entry HDL schematic as a PDF to Adobe Acrobat 6.0 (*PDFWriter* or *Distiller*), it crops the page border at the bottom right edge. This happens because Adobe Acrobat 6.0, by default, installs the printer driver with 1200 dots per inch (dpi). Design Entry HDL is not optimized for dpi settings greater than 600. Therefore, you need to change the default dpi setting to 600 dpi before you generate a PDF output file.
**On Unix**

To create a PDF output file of a Design Entry HDL schematic on UNIX, you must have the Adobe Acrobat Distiller software from Adobe Systems Inc., installed on your machine.

**If you are using HPF plotting on UNIX, do the following:**

1. Access the *Plotting (HPF)* tab of the *Design Entry HDL Options* dialog box.
2. Select the *Batch Plot* check box.
   
   The drawing will be plotted to a file `vw.spool` file if this check box is selected.
3. Click *OK*.
4. Choose *File – Plot*.
   
   The *Hard-Copy Plot Facility* dialog box appears.
5. Click *OK*.
   
   The `vw.spool` file is created in the project directory.
6. Open a UNIX terminal.
7. Change to the project directory.

   The project directory is the directory that has the `<project_name>.cpm` file for your project.
8. Enter the following command in the UNIX terminal:
   
   ```
   distill vw.spool
   ```
   
   The `vw.spool` file is processed and a `vw.spool.pdf` file is created in the project directory.

   **Note:** `distill` is the name of the executable for Adobe Acrobat Distiller. For more information on the options for the `distill` command, use the following command:
   
   ```
   distill -help command
   ```

**If you are using Windows plotting on UNIX, do the following:**

1. In Design Entry HDL, choose *File – Plot*.
   
   The *Plot* dialog box appears.
2. Select a postscript plotter from the *Printer Name* drop-down list.
3. Click the *Properties* button next to the *Printer Name* drop-down list.
   The *Options* dialog box appears.

4. Select the *Encapsulated Postscript File or File* check box and enter the name of the file, say `myplot.ps`, in the *Name* field.

5. Click *OK*.
   The *Plot* dialog box appears.

6. Click *Plot*.
   The `myplot.ps` file is created in the project directory.

7. Open a UNIX terminal.

8. Change to the project directory.
   The project directory is the directory that has the `<project_name>.cpm` file for your project.

9. Enter the following command in the UNIX terminal:
   
   ```
   distill myplot.ps
   ```
   The `myplot.ps` file is processed and a `myplot.pdf` file is created in the project directory.

   **Note:** `distill` is the name of the executable for Adobe Acrobat Distiller. For more information on the options for the `distill` command, use the following command:
   ```
   distill -help command
   ```

**How can I plot a schematic for which I have only read-only permissions?**

*Windows Plotting Mode*

On both Windows and UNIX, 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 on both Windows and UNIX and in HPF plotting mode on UNIX. Refer to **Hierarchical Plotting** for more details.

**Can I plot occurrence properties?**

Yes, you can take plots of occurrence properties on a schematic using both Windows plotting and HPF plotting.
To plot occurrence properties, you have to switch to the Occurrence Edit mode in Design Entry HDL.

**Note:** In the Windows plotting mode, you can plot occurrence properties of any drawing in the design. However, in the HPF plotting mode you can plot only the occurrence properties of the currently open drawing.

**Is previewing supported for HPF plotting?**

Previewing is supported only in Windows plotting mode on both Windows and UNIX platforms. 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 invokes a dialog box showing the system level plot setup options.

It is not recommended to use *Properties* dialog box for setup options. This dialog box should only be used on UNIX for setting the *Print to File* option.
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 14-1 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

1. Specify the drawing name with *File – Open* and create page1 of the design.
2. Use *File – Save* to save page 1.

DESIGN.SCH.1.1

LARGE DESIGN.SCH.1.3

LARGE DESIGN.SCH.1.2

LARGE DESIGN.SCH.1.1
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 43.

On UNIX, each designer must set umask settings so that all other designers have write permissions to all the files in the schematic view of the design. For example, if each designer sets umask to 0 before starting Design Entry HDL, other designers will be able to simultaneously work on different schematic pages of the flat design.

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.
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 14-2 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 14-2 Using the SIZE Property To Structure LS374**

Figure 14-3 on page 536 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.
Figure 14-4 Structured and Flat Design Techniques

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 14-5, a single part is driving too many inputs on SIZE-replicated parts.
Figure 14-5 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; Packager-XL would report a loading error.

The TIMES property is used to correct loading violations in structured designs, as illustrated in Figure 14-6.
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 14-7 on page 538.

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 14-8 on page 540 shows the hierarchical levels of the computer.

**Figure 14-8 Levels of Hierarchy**

![Diagram of levels of hierarchy](image_url)

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 14-9 on page 541 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 14-9 Full Adder Logic and Symbol Drawings**

---

**FULL ADDER.SYM.1.1**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>CIN</td>
<td>CI</td>
</tr>
<tr>
<td>SUM</td>
<td>ADDER</td>
</tr>
<tr>
<td>COUT</td>
<td></td>
</tr>
</tbody>
</table>

**FULL ADDER.SCH.1.1**

---

A\I  B\I  CIN\I  2AN  2AN  EXO  EXO  SUM\I  COUT\I
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 for drawing bodies and establishing the relationships between the symbol drawings and the logic drawings 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 (\textbackslash 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 \textit{Tools} – \textit{Generate View} to automatically create a symbol drawing either from an existing schematic drawing or from a list of pins.
- Use \textit{File} – \textit{New} to create the symbol by drawing.

You can also use \textit{Tools} – \textit{Generate View} and then edit the symbol drawing to move the pins or change the shape.

Editing Symbols

To create a symbol drawing by editing it,

1. Use \textit{File} – \textit{Open} to edit a symbol drawing.

   The \textit{View Open} dialog box is displayed.

2. Specify the name of the \textit{Cell}.

3. Select the \textit{View} as \textit{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.

4. 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.

5. 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.

6. 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-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 545 for information about defining low-asserted pins.

7. 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.

8. 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.

9. Use Text – Note to place labels within the symbol drawing.

This makes the purpose of the symbol and each pin clear.

10. Mark the clock signal with a wedge. Choose Wire – Draw and press the middle button to draw diagonal lines.

11. Use File – Save to save the symbol drawing.
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:

   - 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 (*).

![Diagram](image)

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.
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
efficiency for the design engineer.

<table>
<thead>
<tr>
<th>Flat Designs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Best suited for</td>
<td>Small designs</td>
</tr>
<tr>
<td></td>
<td>Designs that do not re-use modules</td>
</tr>
<tr>
<td></td>
<td>Designs that do not use buses</td>
</tr>
<tr>
<td>Benefits</td>
<td>Short learning curve</td>
</tr>
<tr>
<td>Considerations</td>
<td>Long design cycle time</td>
</tr>
<tr>
<td></td>
<td>Cluttered print sets</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structured Designs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Best suited for</td>
<td>Designs that use sophisticated bus structures</td>
</tr>
<tr>
<td>Benefits</td>
<td>Shortened design cycle time</td>
</tr>
<tr>
<td></td>
<td>Fewer errors during data entry</td>
</tr>
<tr>
<td></td>
<td>Less cluttered print sets</td>
</tr>
<tr>
<td></td>
<td>Print sets organized in the logical flow of the design</td>
</tr>
<tr>
<td></td>
<td>Cross-reference listings</td>
</tr>
<tr>
<td>Considerations</td>
<td>Additional training required for design troubleshooters</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hierarchical Designs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Best suited for</td>
<td>Designs that re-use modules</td>
</tr>
<tr>
<td></td>
<td>Large designs that can be organized into separate components</td>
</tr>
<tr>
<td>Benefits</td>
<td>Shortened design cycle</td>
</tr>
<tr>
<td></td>
<td>Fewer data entry errors</td>
</tr>
<tr>
<td></td>
<td>Easy-to-read print sets</td>
</tr>
<tr>
<td></td>
<td>Cross-reference listings</td>
</tr>
<tr>
<td></td>
<td>Effective debugging capability</td>
</tr>
<tr>
<td></td>
<td>Print sets organized in logical (top-down) flow of design</td>
</tr>
<tr>
<td>Considerations</td>
<td>Additional training required for design troubleshooters</td>
</tr>
</tbody>
</table>
Simulating using AMS Simulator

What is AMS?

You can simulate designs created in Design Entry HDL using the PSpice library components with the AMS simulator and the Advanced Analysis add-on program.

AMS is a simulation program that models the behavior of a circuit. AMS simulates analog-only, mixed analog/digital, and digital-only circuits. Used with Design Entry HDL, AMS 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 the Allegro AMS Simulator User’s Guide (Design Entry HDL Version).

Advanced Analysis is an add-on program for AMS. Use these four Advanced Analysis tools to improve circuit performance, reliability, and yield:

- Sensitivity tool allows you to examine how much each component affects circuit behavior by itself and in comparison to the other components. It also varies all tolerances to create worst-case (minimum and maximum) measurement values.

- 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.

- 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.

- 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 the Allegro AMS Simulator Advanced Analysis User’s Guide (Design Entry HDL Version).
Notes for Using Design Entry HDL with AMS

Note the following when using Design Entry HDL with the AMS simulator and the Advanced Analysis add-on program:

- You must use components from the PSpice libraries in your design if you want to simulate the design using AMS. See the Library List for the list of PSpice library components.

- If you are creating a schematic that you want to simulate using AMS, 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 `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 65.

  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 AMS Projects to New Project Format

When you open a AMS project created in earlier versions (14.2 and older) of Design Entry HDL, the Update Analog Project dialog box appears.

The Update Analog Project dialog box allows you to convert your AMS project to the Design Entry HDL 15.5 format. Converting your AMS project to the Design Entry HDL 15.5 format has the following benefits:

- The directory structure for AMS projects makes it easier to manage the AMS files for the project. For more information, see New Directory Structure for AMS Projects on page 552.
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 AMS 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 AMS project created using version Design Entry HDL 14.2 or older versions that you open in Design Entry HDL 15.0 or later will be automatically converted to the new format.

Later on, if you want to disable automatic conversion of old AMS 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).
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 AMS 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 AMS Simulator Online Help.

**New Directory Structure for AMS Projects**

All files related to AMS projects created using Design Entry HDL version 14.2 or older versions were maintained in the cfg_analog view of a design.
Figure 15-1  Directory structure for the files in cfg_analog view of RF_AMPLIFIER design in RFAMP project created using Design Entry HDL 14.2 or older versions

In the above figure, the rfamp project has a design named rf_amplifier. All the AMS related files for the design are stored in the cfg_analog view of the design.

In the directory structure for AMS projects, the design level and simulation profile level AMS files are organized in their respective directories. This makes it easier to manage the files for the project.
**Note:** If you open a AMS 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 AMS Projects to New Project Format on page 550.

**Figure 15-2 New directory structure for AMS files related to RF_AMPLIFIER design in RFAMP project**

In the new directory structure all the AMS related files for the `rf_amplifier` design are maintained in a view named `psp_sim_1`. 
The AMS 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 AMS projects?* on page 555.

The AMS 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 AMS 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 AMS 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 AMS projects?* on page 556.

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 AMS files that existed in the cfg_analog view at the time of conversion. The AMS 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 AMS 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 15-2, 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 AMS 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.
How are files configured at the profile level maintained in the new directory structure for AMS projects?

The AMS 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 15-3, the AMS 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 15-3, the AMS files related to the Trans simulation profile are maintained in the Trans folder under the \rfamp\worklib\rf_amplifier\psp_sim_1\profiles\ directory.
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 15-3, the Trans profile directory contains a `Trans_profile.inc` include file that includes information on the `decoder.lib`
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 AMS project uses FILESTIMn 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 FILESTIMn part, you must specify the path to the stimulus file in the value for the FILENAME property on the FILESTIMn part.
Design Entry HDL Bias Display with AMS

What is a bias point value?

AMS 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.

AMS calculates and saves the bias point values for every simulation. After simulating with AMS, 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.
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 with 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.

If you have an hierarchical design with reusable blocks, the bias point values are visible only in the Occurrence Edit mode. In case the hierarchical block has no reusable blocks, the bias point values are visible in the hierarchy mode.
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 AMS and then display the values. To load the bias point values generated by AMS, 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 AMS, select Bias Points from the AMS 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 AMS may take some time.

- Automatically loading bias point values
  
  a. Select Bias Points from the AMS 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 AMS, 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 *AMS* drop-down menu, see Figure 15-4 on page 562.

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.

To display bias point values on the schematic, complete the following steps:

a. From the *Bias Points* submenu, choose *Enable*.

   Menu options for displaying the bias Voltage, bias Current, and bias Power are enabled.

b. 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 15-5 on page 563.

![Analog toolbar]

Figure 15-5 The Analog toolbar

**Important**

Though 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 AMS 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).
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.

Caution

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 AMS 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.
Important

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 AMS 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.

Important

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 AMS. 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 stating 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 AMS 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.
Updating bias point data by changing profiles

- From the Analog toolbar in Design Entry HDL, click in the Profile Name text box.

- 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 AMS, see *Allegro AMS 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 \texttt{SEC} property from symbol.css has the same value for one or more parts of different instances.

**Displaying PSpice Names**

You can choose AMS Simulator - Display PSpice Names to control the display of the \texttt{$\$PSPICE\_LOCATION} value on the schematic. Selecting the Display PSpice Names option displays the \texttt{$\$PSPICE\_LOCATION} value. If this option is not selected, the \texttt{$\$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/nets, 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.
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.

Component Browser
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.

DOC drawing
   A drawing containing only graphics. DOC drawings are used for documentation purposes; no electrical or logical checks are done on them.

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.

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 three grid types:
- Logic grid for schematic
- Symbol grid for symbol drawings
- Document grid for DOC 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.

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 CheckPlus 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 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.

occurrence properties
Properties defined on an object based on its place in the design hierarchy. Occurrence properties are not stored with the Design Entry HDL drawing but in the Occurrence Property File (OPF) instead. Occurrence properties are manipulated by front-end and back-end designers.

Occurrence Property File (OPF)
A file used by Design Entry HDL and other tools. It contains occurrence properties. Design Entry HDL reads occurrence properties from the OPF for the current page and displays them on the drawing. Occurrence properties are saved back to the OPF when occurrence editing is disabled. Other tools also use the OPF.

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 predefine 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.

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 \.

November 2008 576 Product Version 16.2
Glossary

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.

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.

vectored 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).
Index

Symbols

..cdsplotinit file 519
   example 519
[] in syntax 22
{} in syntax 22
| in syntax 22

A

abstract data types 207
add a library 226
add a pin 230
add library to search stack 226
Add Printers Wizard 512
Adding a printer on UNIX platform 512
Advanced Analysis 549
   Monte Carlo 549
   Optimizer 549
   Sensitivity 549
   Smoke 549
alias 175
alias symbols, rules 177
ALLEGRO_MWUSER_DIR 518
ALLOW_POWER_PINS 289, 304
ALLOW_PROPERTY_LOCKING
   directive 267
AMS Simulator 549
   Advanced Analysis 549
   bias display 559
   directory structure 552
arc command 545
ascend the hierarchy 120
assertion level
   definition 569
   specifying 175
assign power pins 289
attachments 104
   verifying 543
attribute file 569
   creating 273
Attributes 267, 274
   OPTION_SET_ATTRIBUTES 157
   PROPERTY_ATTRIBUTES 157
   PROPERTY_SET_ATTRIBUTES 158
Attributes drop-down list 269
   adding properties 270
Attributes form 136
   automatic page borders 83

B

back annotate
design 395
backannotating 569
baselining a design 433
   metadata creation 434
   metadata folder 434
batch mode in Windows plotting 474
Bias display 559
bias display 559
   bias point values 561
   color 565
   cpm directives 564
   menu 562
   precision 564
   settings 564
bias point value 559
bias point values
   display properties 565, 566
   displaying 562
   loading 561
   relocating 563
   setting precision 564
   updating 566
BIDIRECTIONAL property 234
blank page 414
block
   definition 569
   drawings
      in hierarchical designs 541
      origin 543
      PIN NAMES 544
bounding box 134
braces in syntax 22
brackets in syntax 22
browse a library 226
bubble pins 570
<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allegro Design Entry HDL User Guide</td>
<td></td>
</tr>
<tr>
<td>BUBBLE property</td>
<td>545</td>
</tr>
<tr>
<td>bubbled pins</td>
<td>545</td>
</tr>
<tr>
<td>defining</td>
<td>545</td>
</tr>
<tr>
<td>bus</td>
<td></td>
</tr>
<tr>
<td>notation</td>
<td>535</td>
</tr>
<tr>
<td>bus-through pins</td>
<td>570</td>
</tr>
<tr>
<td>capturing an image</td>
<td>103</td>
</tr>
<tr>
<td>case sensitivity</td>
<td>283</td>
</tr>
<tr>
<td>cds.lib</td>
<td>57</td>
</tr>
<tr>
<td>cds.lib file</td>
<td>570</td>
</tr>
<tr>
<td>cds.prop.paf file</td>
<td>283</td>
</tr>
<tr>
<td>description</td>
<td>69</td>
</tr>
<tr>
<td>example</td>
<td>70</td>
</tr>
<tr>
<td>cds.prop.tmf file</td>
<td>311</td>
</tr>
<tr>
<td>cell</td>
<td></td>
</tr>
<tr>
<td>deleting</td>
<td>249</td>
</tr>
<tr>
<td>chips.prt file</td>
<td>570</td>
</tr>
<tr>
<td>creating</td>
<td>235</td>
</tr>
<tr>
<td>choosing base signal, rules</td>
<td>178</td>
</tr>
<tr>
<td>circle command</td>
<td></td>
</tr>
<tr>
<td>defining bubbled pins</td>
<td>545</td>
</tr>
<tr>
<td>closing a window</td>
<td>121</td>
</tr>
<tr>
<td>color</td>
<td>90</td>
</tr>
<tr>
<td>specify for an object</td>
<td>90</td>
</tr>
<tr>
<td>component</td>
<td></td>
</tr>
<tr>
<td>instance</td>
<td>571</td>
</tr>
<tr>
<td>rotating</td>
<td>135</td>
</tr>
<tr>
<td>Component Browser</td>
<td>144, 570</td>
</tr>
<tr>
<td>adding a part</td>
<td>152</td>
</tr>
<tr>
<td>browsing parts</td>
<td>147</td>
</tr>
<tr>
<td>comparison</td>
<td>154</td>
</tr>
<tr>
<td>launching</td>
<td>144</td>
</tr>
<tr>
<td>multi-attribute search</td>
<td>147</td>
</tr>
<tr>
<td>new</td>
<td>154</td>
</tr>
<tr>
<td>old</td>
<td>154</td>
</tr>
<tr>
<td>refining search results</td>
<td>149</td>
</tr>
<tr>
<td>replacing a part</td>
<td>153</td>
</tr>
<tr>
<td>search parts by classification</td>
<td>145</td>
</tr>
<tr>
<td>searching parts</td>
<td>145</td>
</tr>
<tr>
<td>single-attribute search</td>
<td>145</td>
</tr>
<tr>
<td>standalone</td>
<td>145</td>
</tr>
<tr>
<td>viewing detailed part information</td>
<td>149</td>
</tr>
<tr>
<td>viewing symbols/footprints</td>
<td>150</td>
</tr>
<tr>
<td>component information</td>
<td>104</td>
</tr>
<tr>
<td>components</td>
<td></td>
</tr>
<tr>
<td>modifying</td>
<td>238, 350</td>
</tr>
<tr>
<td>modifying (in a group)</td>
<td>238</td>
</tr>
<tr>
<td>replacing</td>
<td>239</td>
</tr>
<tr>
<td>sectioning</td>
<td>242</td>
</tr>
<tr>
<td>unsectioning</td>
<td>248</td>
</tr>
<tr>
<td>compressing a page</td>
<td>411</td>
</tr>
<tr>
<td>concatenating signals</td>
<td></td>
</tr>
<tr>
<td>example</td>
<td>189</td>
</tr>
<tr>
<td>use</td>
<td>187</td>
</tr>
<tr>
<td>connectivity design data file</td>
<td>571</td>
</tr>
<tr>
<td>console commands</td>
<td></td>
</tr>
<tr>
<td>about</td>
<td>39</td>
</tr>
<tr>
<td>console window, display</td>
<td>109</td>
</tr>
<tr>
<td>constant signal</td>
<td>571</td>
</tr>
<tr>
<td>constraint</td>
<td>571</td>
</tr>
<tr>
<td>continue expansion even if netlisting</td>
<td>453</td>
</tr>
<tr>
<td>errors</td>
<td></td>
</tr>
<tr>
<td>control file</td>
<td>513</td>
</tr>
<tr>
<td>controlling project settings</td>
<td>74</td>
</tr>
<tr>
<td>conventions</td>
<td></td>
</tr>
<tr>
<td>user-defined arguments</td>
<td>22</td>
</tr>
<tr>
<td>user-entered text</td>
<td>22</td>
</tr>
<tr>
<td>copy</td>
<td></td>
</tr>
<tr>
<td>opf property</td>
<td>268</td>
</tr>
<tr>
<td>copy and paste feature</td>
<td>92</td>
</tr>
<tr>
<td>commands</td>
<td>97</td>
</tr>
<tr>
<td>copying a group</td>
<td>96</td>
</tr>
<tr>
<td>copying bitmap</td>
<td>98</td>
</tr>
<tr>
<td>copying blocks</td>
<td>95</td>
</tr>
<tr>
<td>copying components</td>
<td>93</td>
</tr>
<tr>
<td>copying hyperlinks</td>
<td>96</td>
</tr>
<tr>
<td>copying page border</td>
<td>95</td>
</tr>
<tr>
<td>copying special characters</td>
<td>96</td>
</tr>
<tr>
<td>copying tables</td>
<td>96</td>
</tr>
<tr>
<td>copying text</td>
<td>95</td>
</tr>
<tr>
<td>copying wires</td>
<td>93</td>
</tr>
<tr>
<td>Paste Special</td>
<td>93</td>
</tr>
<tr>
<td>cmpr</td>
<td>52, 53, 58, 59</td>
</tr>
<tr>
<td>bias display directives</td>
<td>564</td>
</tr>
<tr>
<td>locking directives</td>
<td>74</td>
</tr>
<tr>
<td>cmpr files</td>
<td>52, 53, 58, 59</td>
</tr>
<tr>
<td>create a symbol</td>
<td>230</td>
</tr>
<tr>
<td>creating a table of contents</td>
<td>438</td>
</tr>
<tr>
<td>creating Design Entry HDL parts</td>
<td>229</td>
</tr>
<tr>
<td>cross probe</td>
<td>571</td>
</tr>
<tr>
<td>overview</td>
<td>394</td>
</tr>
<tr>
<td>cross-referencing the design</td>
<td>46</td>
</tr>
<tr>
<td>custom flow in Project Manager</td>
<td>59</td>
</tr>
<tr>
<td>custom text</td>
<td></td>
</tr>
<tr>
<td>definition</td>
<td>315</td>
</tr>
<tr>
<td>custom variables</td>
<td></td>
</tr>
<tr>
<td>case-sensitivity</td>
<td>323</td>
</tr>
</tbody>
</table>
Allegro Design Entry HDL User Guide

D

default
  body
    drawing grid  543
grid setting  543
default dpi  99
DEFINE symbol  309
defining
  bubbled pins  545
  low-asserted pins  545
defining a default text editor  84
definitions
  assertion level  569
  attribute file  569
  attributes  569
  automatic routing  569
  backannotate  569
  blank page  414
  block  569
  body  570
  constraint  571
  cross probe  571
  entity  571
  filter  572
  flat design  572
  grid  572
  hard property  572
  hierarchical design  572
  injected property  572
  interface signal  573
  key property  573
  library  573
  library properties  573
  netlist  573
  non-constant signal  573
  occurrence property  574
  package  574
  page gap  414
  placeholder property  575
  project  575
  property  575
  scalar signal  576
  schematic  576
  schematic properties  576
  section  577
  signal  577
  soft property  577
  structured design  578
  symbol  570
  symbol properties  578
  system property  578
  user-defined net  578
  vectored signal  578
  view  579
deleting a page  409
design
  cross-referencing  46
Design Entry
  font  477
Design Entry commands
  set
    default_body_grid  543
Design Entry HDL
  about  23
  console command window  33
  getting started  27
  introduction  23
  menu bars  30
  modes
    expanded  128
    hierarchy  128
  overview  23
  panning  40
  starting  27, 140
  toolbars  31
  Unix  27
  user interface  29
  zooming  40
Design Entry HDL commands
  about  39
  arc  545
  circle  545
  conventions  39
  dot  543
  move  543
  note  543
  pinnames  544
  show
    vectors  535
    signame  543
  wire  543
  write  543
Design Entry HDL commands, enter  39
Design Entry HDL Options dialog box  133
Design Navigation tab  404
Design Entry HDL paper sizes 480
Design Entry HDL variables 317, 318, 319
Design Entry HDL, running 27
design file structure 58
design name 56
design techniques
  flat 531
  hierarchical 539
  overview 531
  structured 534
designs
  flat design 572
  hierarchical design 572
  rules-driven design 576
  structured design 578
dialog boxes
  Add Printer Wizard 513
  Import Design 428
  Property Options 155
digital simulation
  available simulators 454
  generate a netlist 455
  specify options for netlist 454
directives
  ALLOW_POWER_PINS 304
  ALLOW_PROPERTY_LOCKING 267
  bias display 564
  RETAIN_HARDLOCATION_ON_COPY 93
  RETAIN_PREVIOUS_HILITE 110
  SYNC_ON_PAGE_EDIT 257, 436
  SYNC_ON_STARTUP 257, 435
disable netlisting 451, 452
display console window 109
display grid 109
display information on schematic 104
display status bar 109
display toolbars 108
dot command
  in body drawings 543
dots per inch 99
drawing
  BODY 541
  names
    multiple page drawings 533
    pages 532
Drawings
  adding PIN NAMES bodies 544
drawings 143, 249
  ascending 120
  deleting cells, views, and files 249
descending 120
  LOGIC 541
  multiple page 533
  panning 118
  placing a border 143
edit text in console window 91
edit text in dialog box 91
edit text in dialog box, Windows 92
enabling metadata creation 436
entity 571
entity declaration
  generating 232
  entity, properties 232
  error 126 457
errors in netlist for digital simulation 456
exiting Design Entry HDL 126
expand design
  how to handle properties 69
  expand the design
    how to 352
      meaning 352
  expanded mode 128
files
  .cdsplotinit 519
  .opf 313
  cds.lib 570
    adding libraries 64
  cdsprop.paf 283
  cdsprop.tmf 311
  chips.prt 235, 570
  connectivity design data file 571
  created for new project 57
  deleting a file 249
  hdldir.log 457
  netlister.mkr 452
  part table file(.ptf) 237
  ppt_optionset.dat 155
  props.opf 129
  script file 577
  simNetlist.mkr 456
  verilog.v 456
  vhdl.vhd 456
  xmodules.dat 400
Filters 383
finding nets and cells 122, 352
flat design 572
flat designs 531
creating 532
flattener schematic 539

Genview 337
global find limitations 354
global navigate
zooming to a region 135
global property display 279
global signals
shorting 179
syntax 183
grid 572
default settings for bodies 543
grid, display 109
group
creating 339, 340
definition 339
operations 339
guidelines for creating symbol 231

handle properties during expansion 69
hard property 572
hdldir.log file 457
hierarchical designs 539
hierarchical design 572
create 38
module ordering 396
hierarchical design creation, steps 38
hierarchical designs 531
creating 540
hierarchical plotting 504
Hierarchy Editor 572
hierarchy mode 127
hierarchy viewer 396, 397
sheet names 401
highlighting 46

I
interface signal property 544
image
capturing 103
inserting 100
scaling 101
image, copying 98
Import Design dialog box 428
importing designs 426
block 429
blocks 430
multiple sheets 429
sheet 426
sheets 430
sheets containing blocks 430
in occurrence 572
inbuilt custom variables 317
injected property 572
INPUT_LOAD property 234
inserting a page 405
inserting an image 100
instance 571
interface signal 573
interface signal property 542
interface signal property (I) 542, 544
introduction to Design Entry HDL 23
italics in syntax 22

K
key properties 266
key property 573
keywords 22

L
lib-cell-view 50
library
add 226
add to search stack 226
browse 226
remove from search stack 227
library clauses 235
library components
deleting 249
library properties 573
LIBRARY property 235
Allegro Design Entry HDL User Guide

limitation 167
limitations of global find 354
literal characters 22
location of netlist file 456
locking key properties 266
locking project file directives 74
log file 70
LOGIC drawings 541
logic type
  Verilog 197, 199
  VHDL 200
low-asserted pins 545

M
Mainwin 512
MainWin environment variable
  ALLEGRO_MWUSER_DIR 518
marker 573
markers 110
maximum length
  property name 266
  property value 266
menus
  changes in windows mode 132
merge symbol
  rules 193
  types 190
  use 190
MERGE_NC_PINS 289
MERGE_POWER_PINS 289
metadata folder 434
mode of a port 234
module ordering 396
  hierarchy viewer 396
move
  grouped objects 342
  move command 543
  moving a page 411
  moving a window 121
multiple
  page drawings
    flat designs 532
    names 533
    signals 533
drawing
  multiple page drawings 533
  signal 533
navigating nets 123
NC 167
NC pin 167
NC_PINS 289
net 573
netlist 573
  disable netlisting 452
  for analog and mixed signal simulation 460
  for digital simulation
    generating 455
    simulators to use 454
    specify options 454
    view errors 456
  for packaging the design 450
  for PSpice simulation
    generating 460
    for simulation 454
    for synthesis 456
    generating 459
  packaging 450
  PSpice 460
  verilog 453
netlist errors
  verilog 451
  VHDL 452
netlist files 456
netlist for digital simulation 455
netlist for hierarchical design 455, 456
netlist for PSpice simulation 460
netlist for synthesis 456, 459
netlist.mkr file 452
nets
  navigating 123
  navigating in a design 354
  unnamed 186
non-constant signal 573
notation
  bus 535
  note 274
  note command
    labelling body drawings 543

N
names

O
occurrence edit 313
  enabling/disabling 313
viewing occurrence properties 313
occurrence edit mode 129
definition 312
occurrence property 313, 574
add 168
viewing 313
occurrence property file
definition 313
Occurrence Property File (OPF) 574
operations on groups 339
opf file
creating 314
definition 313
opf property
copy 268
option set 155
copying 159
removing 159
or-bars in syntax 22
origin
body 543
OUTPUT_LOAD property 234
OUTPUT_TYPE property 234
inserting pages 405
moving 411
a set of pages outside the current range of pages 413
non-contiguous pages to contiguous locations 413
page before an existing page 412
removing page gaps 411
page move 411
page renumbering
about 422
commands
page move 424
using 423
pages
drawing 532
pan schematic 118
paper sizes
supported by Design Entry HDL 480
PARAM symbol 310
PARAMETER symbol 310
Part 381
Part Manager
applying changes 387
displaying canonical path 390
highlighting part instances 389
invoking 379
overview 378
refreshing 388
Resetting values of part instances 387
updating multiple parts 388
updating part instances 385
use model 378
user interface 379
columns 380
Filters 383
left pane 380
part status 381
right pane 380
Save Schematic 384
status of logical parts 382
summary 383
part status 381
part table file 154
about 66
add 67
example 237
creating 237
part table file(ptf)
parts
sizeable 535
package 574
page 574
renumbering 422
page border
adding 143
page borders
automatic 83
types 142
page delete 409
page gap 414
page insert 405
page locking 43
page management 405
deleting 409
a set of pages 410
current page 409
page gaps 411
GUI 405
inserting
a set of pages 406
a single page 405
beyond the end of the schematic 408
page gaps 408
standard library  539
parts summary  383
Physical Part Table (PPT)  575
physical part table file  66
physical property options
  configure  154
  defining  155
pin  575
pin name  231
PIN NAMES body  544
pin_name property  231
pinnames command  544
pins
  bubbled
    defining  545
  changing states  240
  low asserted  545
placeholder property  575
plot
  font  468, 477
plot command  479
plotting
  hierarchical plotting  504
  Windows plotting  465
port association  171
port mode  234
port names
  adding from corresponding symbol  278
ports
  custom  170
  rules  170
  Verilog logic type  170, 197, 199
  VHDL logic type  170, 200
power pin  289
power pins  289
POWER_GROUP  289
subtype  304
POWER_PINS  289
ppt_optionset.dat file  155
  sample  156
pre select mode  83
primitive  575
project  575
project files  51
Project Manager
  create project  54
  custom flow  59
  files  57
Project Manager, HTML browser  54
  Unix  54
project settings
controlling  74
Project Setup
  add library  64
  change root design  62
  create new root design  62, 66
  edit cds.lib file  63
  select library  65
  specify expansion type  71
  specify log file  70
Project Setup, tasks  61
properties
  BUBBLE  545
  case sensitive  283
  hard  572
  injected  572
  key property  573
  library  573
  occurrence property  574
  placeholder property  575
  schematic property  576
  signal
    interface (i)  544
  SIZE  534
  soft property  577
  symbol property  578
  system property  578
  TIMES  536
  viewing occurrence properties  313
  properties for Design Entry HDL  266
  properties for entity  232
  properties window  136
  properties, definition  575
  property  575
    BIDIRECTIONAL  234
    definition  265
    INPUT_LOAD  234
    LIBRARY  235
    locking  266
    OUTPUT_LOAD  234
    OUTPUT_TYPE  234
    PATH  268
    pin name  231
    reattaching  279
    swap  278
    types  265
    vhdl_mode  230, 233, 234
    vlog_mode  230, 233, 234
    property name, maximum length  266
    property options
      using  155
Property Options dialog box  155
assertion level 175
concatenate 187
concatenate graphically 188
concatenate textually 187
global 179
initial value 172
naming 162
properties
  interface (I) 542
unnamed 186
Verilog logic type 197, 199
VHDL logic type 200
name command
  in body drawings 543
simNetlist.mkr file 456
simulators for digital simulation 454
Single File Netlist 455, 456
single-page flat designs
  creating 532
site 52, 53, 58
site.cpm 52, 53, 58, 59
SIZE property 534
size replication 208
sizeable
  parts 535
SKILL 577
SLASH 174
slice 194
soft property 577
specifying range
  ascending 204
  descending 204
  rules 204
  unconstrained 205
  use 204
spin a component 242
Standard library
  overview 225
standard library 539
standard port symbols 169
starting
  Design Entry HDL 27, 140
status bar, display 109
status of logical parts 382
step size 165
strict entity check 458
strokes
  guidelines 127
  strokes, running commands 126
structured design 578
structured designs
  about 531, 534
  creating 534
summary (Total/Defined/Undefined) 383
support for keyboard operations 133
swapping notes and properties 278
symbol 570, 578
  create 230
  symbol properties 578
  symbols
    creating 232
    synonym and alias, difference 176
syntax
  signal name 535
  synthesis
    generate a netlist 459
    specify options for verilog netlist 456
    specify options for VHDL netlist 458
  system properties 578

T

table of contents 438
tap
  bus tap 570
  C-tap body 570
tap body 578
temp directory 68
text
  adding 276
  adding port names from corresponding symbol 278
  changing the editor 278
  editing in dialog boxes and the console window 91
  modifying 276
  resizing 277
text editor
  changing 278
  Unix 69
  Windows 69
text macro
  define in a file 311
  defining 309, 310
  definition 308
  use 308
TIMES property 536
TOC 438
toolbar
  QuickPick 112
toolbars
Allegro Design Entry HDL User Guide

QuickPick 115
toolbars, display 108
tools setup 68
transcribe 539
type conversion functions
  restriction in Design Entry HDL 207
  restriction in Verilog 206
  use 206

U
unbubbled pins
  defining 545
unconnected pins 136
undo 85
unnamed net, specify size 174
unnamed signals or nets 186
unsectioning components 248
URL 274
usability changes 130
use
  clause 235
  property 235
user-defined net 578

V
variables
  CRef variables 319
  drawing specific 317
  global 318
  parent variables 319
vectored signal 578
verbose output 458
verifying attachments 543
Verilog net type 197, 199
verilog netlist 450, 453
  for synthesis
    specify options 456
verilog.v file 456
version 578
vertical bars in syntax 22
VHDL netlist
  for synthesis
    specify options 458
VHDL scalar type 200, 202
VHDL vector type 200, 202
vhdl.vhd file 456
VHDL_IN_CONVERT 206
VHDL_INIT 172
vhdl_mode property 230, 233, 234
VHDL_OUT_CONVERT 206
VHSIC 579
view 579
  deleting 249
view names
  changing 74
view pages of a drawing 108
viewing a design 432
vlog_mode property 230, 233, 234

W
windows
  closing 121
windows mode 130
Windows Plotting
  font 468, 477
  previewing 466, 469
  setup 466
wire 579
wire command
  marking clock signals 543
wiring ports together 169
worklib 57

X
xmodules.dat file 400
x-replication 208

Z
zoom 40, 119
zoom by points 135