

DesignWorks™ Professional 5

Windows® Version
Users Guide

Revised for version 5.0
October 15, 2007

IMPORTANT NOTICE

Capilano Computing Systems Ltd. (“Capilano”) retains all ownership rights to the DesignWorks™ Professional program and all other software and documentation making up the DesignWorks package. Use of the DesignWorks software is governed by the license agreement accompanying the original media.

Your right to copy the DesignWorks software and this publication is limited by copyright law and your end user license agreement. Making copies, adaptations or compilation works (except copies for archival purposes or as an essential step in the utilization of the software) without prior written authorization of Capilano, is prohibited by law and constitutes a punishable violation of the law.

Capilano provides this publication “as is” without warranty of any kind, either express or implied, including but not limited to the implied warranties or conditions of merchantability or fitness for a particular purpose. In no event shall Capilano be liable for any loss of profits, loss of business, loss of use of data, interruption of business, or for indirect, special, incidental or consequential damages of any kind, even if Capilano has been advised of the possibility of such damages arising from any defect or error in this publication or in the DesignWorks software.

Capilano reserves the right to update this publication from time to time without notice. Some of the information in the publication refers to characteristics of third party products over which Capilano has no control. This information is provided for the convenience of DesignWorks users only and no warranty is made as to its correctness or timeliness.

Copyright ©2000,2003,2007 All rights reserved.

DesignWorks is a trademark of Capilano Computing Systems Ltd. Windows is a registered trademark of Microsoft Corporation. Other trademarks used in this publication are property of their respective owners.

Printed in Canada.

Capilano can be contacted at:

Capilano Computing
2631 Viking Way, Unit 218
Richmond, B.C., V6V 3B5
Canada

phone +1-604-522-6200

fax +1-604-273-9397

email info@capilano.com

WWW <http://www.capilano.com>

Table of Contents

Chapter 1—Introduction	1
Where to Start	1
Notes Regarding Copyright and Trademarks	2

Chapter 2—User Interface	3
Mouse Button Usage	3
Dialog Boxes	4
Window Usage	4
Document (Circuit, Part and Text) Windows	4
Tool Panels	5
Closing a Document Window	7
Redisplaying a Circuit Window	7
The Window Menu	7
Keyboard Usage	7
Pop-up Menus	8
Toolbars	8
To Discover the Meaning of a Tool	8
Moving a Toolbar	9
Showing and Hiding a Toolbar	9
Status Display	9
Use of the Pointer or Cursor	9

Chapter 3—Tutorial	11
Manual Format	11
The 5-minute Schematic Diagram	12
Advanced Schematic Editing	25
Device Symbol Editing and Hierarchical Design	37
Using DesignWorks with SPICE-based Simulators	45

Chapter 4—Basic Procedures	49
DesignWorks File Types	49
Design Structure	50
What is a Circuit?	51
Types of Objects in a Circuit	51
Creating a New Design	52
Choosing a Template	52
Opening a Design	54
Compatibility With Older Versions	54
Navigating Around a Schematic Page.	54
Zooming In and Out	54
Opening Circuit Page Windows	56
Locating Circuit Objects with the Find Tool	56
Locating Objects Using the Browser Tool	56
Saving a Design.	57
Reverting to a Saved File	57
Saving a Circuit Page in WMF, DXF or PDF Graphics Formats.	57
Printing	59
Specifying the Page Number Range	59
Setting the Printer Page Setup.	60
Fitting the Diagram to the Available Paper	60
Backup Procedures	61
Enabling Auto-Backup on Save	61
Closing a Design	62
Disposing of a Design.	62
Exiting DesignWorks	63

Chapter 5—Basic Schematic Editing	65
General Editing Operations	65
Undo and Redo	65
The Clipboard Commands	65
Selection	71
Zooming in on Selected Objects	73
Adjusting the Position of All Objects on a Page.	73
Showing Overall Circuit and Design Statistics.	73

Setting Design Attributes	75
Making a Circuit Read Only	75
Adding, Deleting and Titling Circuit Pages	76
Working with Device Symbols	77
Placing a Device From a Library	77
Using the Parts Palette	79
Duplicating an Existing Device	81
Deleting a Device	82
Moving a Device	82
Flipping and Rotating a Device	82
Displaying and Setting Device Information	82
Device Names	84
Selecting the Part and Package Type	87
Selecting the Gate Unit	88
Creating and Editing Signals	89
Interconnecting Signals	89
Naming Signals	94
What Signal Names are Used For	95
Invisible Signal Names	95
Using the Auto-Naming Features	99
Pin Numbering and Information Entry	100
Getting and Setting Pin Information	100
Pin Numbering	102
Using Text and Graphic Objects	104
Creating a Text Block	104
Editing a Text Block	104
Text Style and Display Options	106
Using Text Variables	107
Drawing Graphics	109

Chapter 6—Before Starting a Major Design. 113

The Golden Rule - Try a small design first!!! . . . 113

Design Process Checklist 114

Schematic Creation	114
Printing and Plotting	118
Reports	119
Interfaces to Other Systems	119

Chapter 7—Device Packaging and Naming 121

Packaging vs. Auto-Name Assignment 121

Choosing Options for your Design 122

Enabling Naming and Packaging Options 123

Naming and Packaging Options in Flat Hierarchy Mode 123

Naming and Packaging Options in Physical Hierarchy Mode 124

Naming and Packaging Options in Pure Hierarchy Mode 126

Using Device Packaging 127

Re-enabling Auto-Packaging After Manual Edits 128

Auto-Packaging Limitations 128

Bringing the Design's Package Table Up to Date 129

Getting a Report of Unused Gates 129

Batch Repackaging the Entire Design 129

Performing Manual Packaging 130

Setting the Auto-Generated Name Format 131

Batch Reassigning Device Names 132

Setting the Name Prefix for a Symbol 133

Specifying that a Device Should be Unnamed When Placed 133

Selecting an Alternate Prefix Field 133

Setting Device Packaging Options 134

Overriding Default Name and Unit Visibility 135

Using Packaging in Hierarchical Designs 136

Using Device Libraries Without Packaging Information 137

Using Back Annotation 138

Back Annotation and Packaging 139

Back Annotation in Hierarchical Designs 139

Back Annotation from PADS PCB 140

Attribute Fields Affected By Back Annotation 141

Back Annotation File Formats 142

Creating a Symbol with Multi-gate Packaging .. 144

Setting Packaging Attribute Fields While Creating a Symbol 144

Creating a Symbol for Multiple Gates With Same Symbol - Example 147

Creating Symbols for Multiple Gates With Different Symbols - Example 148

Creating a Symbol for a Discrete SIP Package - Example 149

Specifying PCB Package Type Information. 150

Design Attribute Fields Used By the Packager .. 151

Device Token Values	152
Using Packaging with Connector Symbols.	152
Handling Discrete Components.	154
Device Date Stamping.	154
Disabling Date Stamping	155

Chapter 8—Attributes 157

Attribute Organization	157
Attribute Definition Table.	157
Predefined Fields.	158
User-defined Fields.	158
Primary vs. Secondary Fields	158
Definition vs. Instance Fields	159
Temporary Fields	159
Attribute Limitations.	159
Entering and Editing Attribute Data - Basic Procedure	160
Entering Design Attributes	162
Entering Pin Attributes	163
Controlling Attribute Display Characteristics. . . 163	
Rotating Attribute Text	163
Hiding a Visible Attribute Value	164
Clearing a Visible Attribute Value	164
Displaying an Invisible Attribute Value	165
Setting Attribute Text Style	165
Setting Attribute Justification	167
Displaying an Attribute Value in Multiple Locations.	168
Showing the Field Name with an Attribute Value	168
Other Ways of Viewing and Editing Attributes. . 168	
Editing Attributes on the Schematic	168
Using Value List Sub-menus	169
Probing Attributes on the Schematic	169
Using the Name and InstName Fields 170	
Choosing Whether to Use Name or InstName	170
Using Value List Fields 173	
Creating a Value List Field.	174
Using Default Position Fields 175	
Default Position Data Format	175

Setting Default Values	177
Defining a New Attribute Field	177
Setting Attribute Field Options	178
Using Duplicate, Merge & Delete for Global Editing	182
Globally Duplicating Attribute Data	183
Merging Two Existing Attribute Fields	184
Delete	185
Temporarily Displaying Attributes	185
Permanently Showing Data Throughout a Design	186
Merging Dissimilar Designs	187
Importing Attribute Definitions	187
Pasting from the Clipboard or Placing a Library Device	188
Converting Files from Older Versions	188
Changes in Standard Fields	188

Chapter 9—Making Signal Connections 189

Using Busses	189
Properties of Busses	189
Creating a Bus	191
Getting Bus Information	191
Adding Signals to a Bus	192
Getting Information on Signals Inside a Bus	192
Using Bus Breakouts	193
Using Bus Pins	196
Changing Bus Pin Connections	197
Inter-page Connections	200
Automatic Display of Page References	201
Connecting Busses Across Pages	202
Using Page Connectors on Internal Bus Signals	203
Changing the Page Connector Symbol	204
Tracing Connections Through Page Connectors	205
Power and Ground Connections	205
Power and Ground Naming Convention	207
Power and Ground Connections in Attributes	207
Signal Connector (Power and Ground) Symbols	208
Using Signal Auto-Naming	210
Enabling Auto-Naming	210

Disabling Signal Auto-naming	211
How Names are Generated	211
Using Signal Token Values	212
Signal Connectivity Rules	213

Chapter 10—Hierarchical Design 215

General Concepts 215

What is Hierarchy?	215
A Simple Hierarchy Example	215
Definition vs. Instance	217

Choosing a Hierarchy Mode 218

Flat Hierarchy Mode	218
Physical Hierarchy Mode	218
Pure Hierarchy Mode	218
Setting the Hierarchy Mode	219
Effect of Changing Hierarchy Mode	219

Navigating in Hierarchical Designs 219

Opening (Pushing Into) a Subcircuit	220
Closing (Popping Out of) a Subcircuit	220
Locking and Unlocking Subcircuits	221

Creating a Hierarchical Block - Top Down 221

Creating a Block Symbol	222
Placing the Block Symbol	223
Auto-Creating the Internal Circuit	224

Creating a Hierarchical Block - Bottom Up 224

Creating a Subcircuit	225
Placing a Subcircuit	226

Generating Netlists from Hierarchical Designs. . 227

Generating Hierarchical Netlists	227
Generating Flattened Netlists	227
Using Hierarchical Names	228
Changing the Hierarchical Name Separator	228

Printing Hierarchical Designs 229

Determining Print Page Order	229
Setting Printing Scope	229
Printing Sequential Page Numbers in a Hierarchical Design	230

Associating a Subcircuit with a Device Symbol . 230

Working with Subcircuits	231
Making Connections Across Hierarchy Levels . .	236
Creating and Using Port Connectors	237
Setting the Port Pin Type	237
Using the New Port Connector command to Create a Signal Port .	238
Creating Bus Ports	240
Modifying an Existing Bus Port	242
Making Power and Ground Connections Across Hierarchy Levels	242

Chapter 11—Searching and Browsing Tools 245

Introduction	245
Using the Find Tool	246
Starting Find	246
Using the Browser Tool	248
Opening the Browser	248
Closing the Browser	250
Updating the Browser Window	250
Selecting the Type of Objects Displayed	250
Determining Where to Search for Objects	251
Displaying Attributes	251
Changing Attribute Values	252
Saving and Printing Data in a Browser Window	253
Showing Objects in the Schematic	254
Sorting Displayed Objects	254
Adjusting the Spreadsheet	254
Customizing Search Scripts for the Find Tool . .	255
Find Data File Format	255
Data File Example	256
Generating a Find Data File Using the Export Tool	257
Using the User Text Entry Box	259
Where Search Scripts are Placed	259

Chapter 12—Device Symbols and Libraries 261

Working With Symbol Libraries	262
Creating a New Library	262
Manually Opening a Library	264
Automatically Opening Libraries at Startup	264

Manually Closing a Library	264
Copying Symbols from One Library to Another	265
Copying Symbols from a Design to a Library	265
Deleting Symbols from a Library	266
Duplicating a Symbol Within a Library	266
Renaming a Symbol in a Library	266
Getting Information on a Symbol in a Library	267
Reordering Symbols Within a Library	267
Compacting a Library	268
Using Circuit Elements as Library Items	268
Operations on Symbols in a Schematic	269
Making a Single Device Into a Unique Type	269
Updating a Symbol from a Library	269
Saving a Symbol Definition from a Schematic to a Library	272
Saving All the Symbols in a Design to a Library	274
Editing a Device Symbol in a Schematic	275
Creating a Design With One of Each Symbol in a Library	275
Editing Device Symbols	276
Creating a New Part from Scratch—Basic Procedure	276
Editing an Existing Part in a Library	280
Editing an Existing Part on a Schematic	281
Closing the Device Symbol Editor Window	281
Saving an Edited Part Back to its Original Library	281
Saving the Part Under a New Name	281
Zooming the Symbol Editor Window	282
Adding Sequential Pin Names	283
Deleting Pins	285
Setting Part and Pin Attributes	285
Editing Symbol Graphics	288
Using the Drawing Tools	288
Reordering Graphical Objects Front-To-Back	290
Grouping Graphical Objects	290
Aligning Graphical Objects	290
Rotating and Flipping Graphical Objects	290
Setting Grids	290
Placing Pins on a Symbol	292
Saving Frequently-Used Graphics and Pins	296
Displaying the Symbol Gallery Window	296
Hiding the Symbol Gallery Window	296
Using Elements from the Symbol Gallery	296

Adding Elements to the Symbol Gallery	297
Specifying a Symbol Gallery File	297
Entering Pin Information.	298
Selecting Items in the Pin List	298
Setting the Pin Name	298
Setting the Pin Number	298
Setting the Pin Type	299
Displaying the Pin Name	299
Reordering Pins in the Pin List	299
Creating a Part With a Subcircuit	300
Creating the Port Interface	300
Selecting the Subcircuit	300
Drawing the Graphics and Placing Pins on the Subcircuit Symbol	303
Opening the Subcircuit Associated with a Symbol	303
Automatically Creating Symbols	303
Auto-creating Rectangular Symbols	304
Creating a Breakout	307
Creating a Power and Ground (Signal) Connector	308
Power and Ground Connections with the DesignWorks Simulator	308
Creating a Port Connector	308
Creating a Signal Port Connector	309
Creating a Bus Port Connector	309
Creating a Page Connector	311
Making a Signal Page Connector	311
Making a Bus Page Connector	311
Creating Special-Purpose Symbols	312
Assigning a Primitive Type	312
Creating Primitive Devices for use with the DesignWorks Simulator	312
Symbol Date Stamping	313
Features Requiring Symbol Attributes	313
Gate Packaging	314
Auto-Naming	314
Specifying Part and Package Type Information	315
Using the Standard DesignWorks Libraries.	318
Symbol Format	318
Finding a Library	319
Interpreting Library Part Names	320

The Permutable Attribute	323
Package Codes	325
Function and Category Codes	327

Chapter 13—Design Templates and Customization329

Creating Design Templates..... 329

Contents of a Design Template.....	331
Naming a Design Template.....	332
Working from an Existing Design Template	332
Where Design Templates are Stored.....	332

Setting Sheet Sizes and Borders..... 333

About Sheet and Border Settings	333
Importing Sheet Settings from Another Design, Page or Template.....	334
Setting Custom Sheet Size using Sheet Wizard	335
Setting Border and Background Grid Settings with the Border Wizard.....	337
Creating a Title Block.....	339
Creating Custom Sheet Border Graphics	340
Setting Text Styles	340

Creating Multipage Templates..... 342

Sheet Border Setup for Multi-Page Designs.....	342
--	-----

Creating Symbol Libraries..... 342

Creating Custom Menus..... 342

Defining a Menu in the Main Menu Bar.....	343
Adding a Menu Item to Popup Device, Signal, Pin or Circuit Menus.....	344
Creating Scripts for Use with Custom Menus	345

Creating Scripts..... 345

Creating Netlists and Reports	346
Error Checking	347
Data Entry	348
Back Annotation	348
Invoking Scripts	348

Using Custom Panels..... 350

Creating an HTML Page for the Custom Panel	350
How the Custom Panel is Displayed.....	350

Chapter 14—Report and Netlist Generation..... 353

Introduction to the Export Tool..... 353

General Information on Export	353
Generating Standard Netlist and Report Formats	354
Basic Report Export Procedure	354
Invoking Export Using Custom Menus	356
Device Reporting Options	356
Signal Reporting Options	356
Common Changes to Standard Report Forms . . .	357
General Rules	357
Default File Name	357
Attribute Field Usage	358
Extracting Power and Ground Connections from Attributes	358
Script Errors	359

Appendix A—Predefined Attribute Fields 361

Appendix B—Primitive Device Types 369

Schematic Symbol Primitive Types	370
Pseudo-Device Primitive Types	370
Simulation Primitive Types	371

Appendix C—Device Pin Types 373

What Pin Types are Used For	373
Pin Types Table	374

Appendix D—Ini File Format 375

Specifying File and Folder (Directory) Names . . .	375
Absolute and Relative Path Names	375
Using Environment and Registry Variables	376
Using Common System Locations	377
Section [Drawing]	377
Initial Directory Settings	377
Font Settings	377
Color Settings	378
Specifying the Location of Design Templates	379
Specifying the Location of Example Files	379
Adding Custom Menu Commands to Popup Menus	379
Adding Default Attribute Field Definitions	379
Enabling Auto-Backup and Timed Auto-Save	381

Disabling Device Date Stamping	382
Specifying Standard Sheet Sizes	382
Solid Grid Lines	383
Zoom Factors	383
Pin Spacing	383
Breakout Parameters	383
Disabling "Loose End" Markers on Signal Lines	384
Undo Levels	384
Fine-Tuning Pin Number Text Display	384
Internal Error Checking	385
Section [Libraries]	385
Specifying Libraries to Open at Startup	385
Section [DevEditor]	386
Default Font	386
Grid Settings	386
Default Pin Name	386
Symbol Gallery Location	387
Section [Export].	387
Specifying Predefined Script Variables	387
Creating Custom Menus	387
Specifying the Location of Export Scripts	388
Section [System]	388
Tools Directory	388
Default System Font	388
Printer Line Scaling	388
Printer Color Mapping	389
Clipboard Color Mapping	389
Section [System Font Translations]	389
Section [Find]	390
Specifying Search Script Location	390
Section [TextEdit]	390
Specifying Additional Text Document Types	390

Appendix E—Command Line Arguments 393

Specifying File Names on the Command Line . . .	393
-exp (Export) Option	393
-exit (Exit Immediately) Option	394
-hide (Hide Window) Option	394

-nodoc (No Document) Option	394
-js (JavaScript) Option	395
-bp (Browser Panel) Option	395
Appendix F—Installation	397
Installing on a Write Protected Server	397
Specifying a Root Directory	397
Files in Root Directory	397
File Search Paths	398
Installing and Locating Symbol Library Files . . .	398
Location of Libraries	398
Opening Libraries Manually	399
Opening Libraries Automatically at Startup	399
Appendix G—Technical Support	401
Internal Error Detection	401
Index	403

Welcome to the DesignWorks™ Professional for Windows schematic entry tool from Capilano Computing. DesignWorks is built with features designed to allow it to form the core of your electronics design system.

This chapter will point you to the resources you need to get started using the package as quickly as possible.

NOTE: This manual can also be used in conjunction with DesignWorks™ Lite, although some features, such as hierarchy and multipage support, are not available in the Lite version.

Where to Start

We suggest you ease yourself into the world of schematic editing with DesignWorks by taking the following steps:

STEP 1—Install the package using the procedures outlined in the on-line installation notes provided on the release CD-ROM or downloadable files.

STEP 2—Work first through Chapter 3—Tutorial on page 11. It provides step-by-step instructions for basic schematic editing.

STEP 3—Refer to Chapter 6—Before Starting a Major Design on page 113 for information on choices that you should consider before investing too much work in a schematic.

STEP 4—Refer to Chapter 4—Basic Procedures on page 49 and Chapter 5—Basic Schematic Editing on page 65 for general schematic editing procedures.

As you work with DesignWorks you will have occasions to look for information in this manual. It is organized into a number of parts, sorted more or less

in order of the depth and complexity of the material. Later parts address issues in larger designs, interfacing to other systems, customizing the package, etc.

If a question comes up that the manual doesn't answer, we are available for technical support by phone or on-line. See Appendix G—Technical Support on page 401.

Notes Regarding Copyright and Trademarks

The DesignWorks software and manual are copyrighted products. The software license you have purchased entitles you to use the software on a single machine, with copies being made only for backup purposes. Any unauthorized copying of the program or documentation is subject to prosecution.

A number of product trademarks are referred to in this manual. DesignWorks, LogicWorks, MEDA, and Modular Electronic Design Application are trademarks of Capilano Computing Systems Ltd. All other trademarks used are property of their respective holders.

This chapter provides general information on the use of windows, drawing tools and other user-interface features of DesignWorks.

Mouse Button Usage

Several different mouse button actions are used for various functions in DesignWorks. For clarity we will use the following terminology when referring to these actions in the remainder of the manual:

Click—means press and release the left mouse button without moving the mouse. Example: To select a device, click on it.

Click and drag—means press the left mouse button and hold it pressed while moving the mouse to the appropriate position for the next action. Example: To move a device click and drag it to the desired new position.

Double-click—means press and release the left mouse button twice in quick succession without moving the mouse. Example: To open a device's internal circuit, double-click on the device.

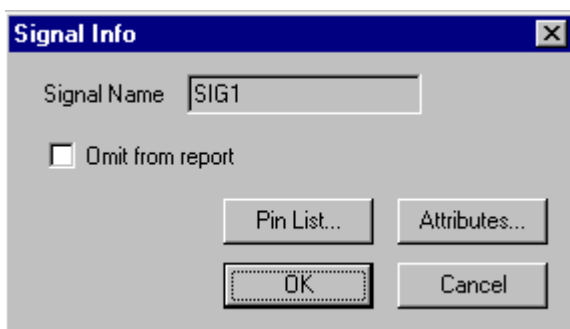
Right-click—means press and release the right mouse button without moving the mouse. In DesignWorks, the right mouse button is only used to display a pop-up menu of shortcut commands which depends on the item clicked on. Example: To display the popup menu for a device, right-click on it.

NOTE: It is possible for users of a Windows system to change the meaning of the mouse buttons on that specific system. We will use the terms "left" and "right" buttons consistent with the standard usage, but you may need to translate these items if your system settings have been changed.

Dialog Boxes

Many DesignWorks functions require information to be displayed or prompted from the user. To do this a special window called a “Dialog Box” is displayed, such as the following one which is used when a Properties command is executed for a signal.

In general, the controls in dialog boxes will behave according to Windows standards.



In dialog boxes requiring text entry, the keyboard equivalents for the clipboard commands Cut (-X), Copy (-C) and Paste (-V) are active and can be used to transfer text to or from a text box.

Window Usage

Document (Circuit, Part and Text) Windows

Each circuit window displays one page of a circuit schematic. The title on a circuit window will be the name of the circuit file, followed by the page number and title, if any. No page number is displayed if the circuit only has one page. Any number of pages in a given circuit and any number of circuits can be displayed simultaneously.

At any given time only one page of one circuit is “current”, that being the one

in the topmost window. Any other window can be made current simply by clicking the mouse anywhere in the window. Many of the menu commands, such as Save As, Export, etc., apply to the "current" circuit.

Document windows can be displayed in three different forms:

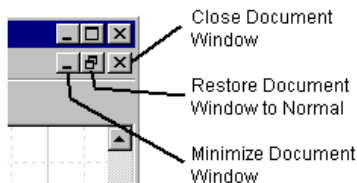
Normal—In this state, windows are stacked on top of one another and can be moved and resized independently.

Maximized—In this state, a single document window takes up all available area within the application frame.

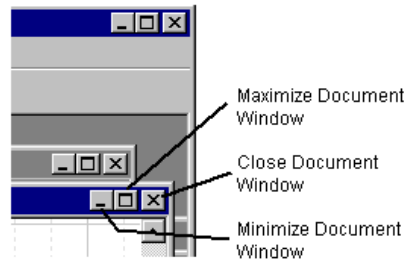
Minimized—In this state, the window is displayed as an icon at the bottom of the application frame. I.e. The associated document is still open, but the contents are not displayed.

The positions of the window controls in the Maximized and Normal state are shown here:

**Window Controls with
Document Window Maximized**



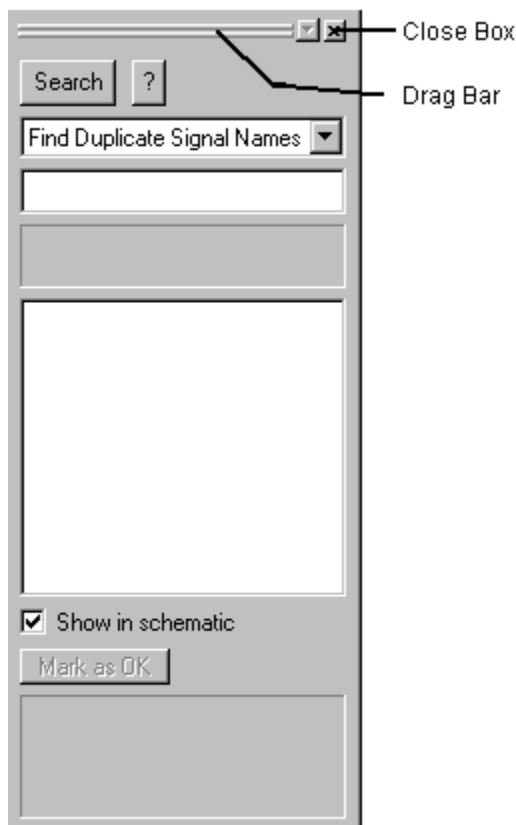
**Window Controls with
Document Window Normal**



Tool Panels

Program modules, or Tools, can create their own windows, toolbars and panels which may be displayed concurrently with other DesignWorks windows.

A typical tool panel looks like this one, created by the Find command:



This type of panel have a number of controls that can be used to move the panel and change how it is displayed.

Click the Close Box—to close the window. To redisplay the window, use the associated command in the menu bar.

Click and Drag in the Drag Bar—to move the panel to a different location. This type of panel can be located on any side of the main application frame.

Right Click in the Drag Bar—to display a small menu with other display options:

Allow Docking—When this item is checked (the default), the panel will try and "stick" to the edges of the main application window. When this item is unchecked, the panel will float as a separate

window that can be moved out of the way of the main application window, if desired.

Hide—This is the same as clicking the Close Box.

Float in Main Window—This command causes the panel to be displayed like a document window.

Closing a Document Window

To close a document window, simply click in the Close Box of that window. The effect of this on the document file depends on the type of information that was displayed in the window.

Circuit Windows—Clicking in the Close Box on a Circuit window has the effect of closing the circuit page. If the page being closed was the only one open on the design then the design file is closed.

Symbol Editor Windows—Clicking the Close Box of a Device Symbol Editor window closes that symbol editor session. If the symbol had been modified, you will be prompted to save the changes back to a library.

Text Windows—Clicking the Close Box of a text window closes the text document. If the document was unsaved, you will be prompted to save any changes.

Redisplaying a Circuit Window

When a circuit page window is closed it is removed from the Window menu to avoid clutter. To reopen a page in the current circuit level, use the Pages command in the Drawing menu. To reopen an sub-circuit window, double-click on the device in question.

The Window Menu

The Window menu provides a means of bringing to the front any window currently open.

Keyboard Usage

The keyboard is only absolutely required when entering names for devices or signals, or for placing random text notations on the drawing. However, the

, , and keys on the keyboard can be used with many editing operations to invoke special features such as auto-alignment, auto-naming and different signal line drawing methods. In addition the arrow “cursor” keys (if available) can be used as a convenient way of "nudging" selected items, or setting symbol rotation while placing devices or pasting circuit groups. These options are described in detail in the relevant chapters.

Pop-up Menus

At any time while editing a diagram you can right-click on a schematic object. A pop-up menu will appear under the cursor allowing you to select from commands appropriate to that object. E.g. the menu for a device contains commands to get device information, edit attributes, open the internal circuit, flip or rotate the symbol, Cut and Copy operations, change gate package assignment, etc.

Separate pop-up menus are available for devices, signals, pins, attributes, and (if clicking in open space on the drawing) the circuit itself.

Toolbars

DesignWorks uses a variety of toolbars to give quick access to program functions..



To Discover the Meaning of a Tool

To find out what a tool does without risking trying it, just move the mouse pointer over the tool and stop. A one-word description of the tool will pop up after a pause and a longer description will appear in the program status bar at the bottom of the application frame.

Moving a Toolbar

A toolbar can be moved to another location in the application window by clicking and dragging in an unused area of the bar (i.e. not on a button).

Showing and Hiding a Toolbar

Toolbars can be displayed or hidden using the commands in the View menu.

Status Display





The lower area of the application window is used to optionally display several items of status information. This display can be showed or hidden The selected orientation will be displayed in the tool palette icon.













Use of the Pointer or Cursor

In subsequent descriptions, we will refer to the small shape which tracks the mouse position on the screen as the “pointer” or “cursor”. In DesignWorks there are a number of different cursor modes used which determine what action will be performed when the mouse button is clicked. Following is a summary of the cursor modes. More detailed descriptions of operations performed in each mode are provided in later reference chapters.

Note that the cursor shape sometimes differs from the tool palette icon for ease of pointing.

Tool Palette Icon	Initial Cursor Shape	Equivalent Menu Command	Description
		Point	Used to select or drag objects, extend signals.
		Draw Bus	Used to create a new bus line or extend an existing bus. Clicking once fixes a corner, double-clicking terminates the line.

		Draw Sig	Used while creating a new signal line or extending an existing signal. Clicking once fixes a corner, double-clicking terminates the line. Note that most signal drawing operations can also be done in Point mode.
		Text	Used to select a signal or device to name, or to place random text on the diagram. Point at the item you want to name and press and hold the mouse button. Move to where you want the name to appear, then release the button.
		Zap	Used to remove single objects. Press the button to remove whatever the tip of the cursor is pointing at. Objects can also be removed in groups by selecting them and using the Clear command or delete key.
		Magnify	Used to zoom in and out. Clicking on a point or dragging down and right zooms in, dragging up and left zooms out.
		Attribute Probe	When the arrow portion of the cursor is clicked on a device, pin or signal, the contents of the primary attribute fields for that object are displayed.

The purpose of this tutorial chapter is to give you a quick overview of the steps involved in schematic creation. You can then refer to the later reference chapters to find more information on specific topics of interest.

TIP: We strongly suggest that you take a look at Chapter 6—Before Starting a Major Design on page 113 before proceeding with any significant design projects. A small amount of time spent on thinking through your design process now can save a lot of headaches later!

To allow you to get a quick overview and then delve into specific topics of interest, this tutorial chapter is divided into these sections:

The 5-Minute Schematic Diagram -- *NOTE: All other sections below assume you have worked through this one!*

Advanced Schematic Editing

Creating Device Symbols and Hierarchical Design

Using DesignWorks with SPICE-based Simulators

NOTE: If you have not yet installed DesignWorks on your computer, follow the instructions in the ReadMe file provided on the installation CD before proceeding.

Manual Format

In all of the following sections, action instructions are shown in **bold face like this**, whereas explanatory text is shown in normal typeface.

The 5-minute Schematic Diagram

In this first tutorial section we will show you how quick and easy it is to put together a complete schematic diagram and generate a netlist out for your board layout package.

TIP: If you are new to Windows or have any questions about the operation of the tools and controls you see on the screen, you may want to refer to Chapter 2—User Interface on page 3 before proceeding.

Starting the Program



If it is not already running, double-click on the **DesignWorks Professional 5** icon, or select the **DesignWorks Professional 5** item in the Start menu to start the program.

After a moment of loading the program and opening libraries, you will see the New Design dialog box.

Creating a New Design

The Welcome box allows you to select a design mode for your application. Each selection enables the group of options most commonly used for the application type. All options can be changed later, if desired.

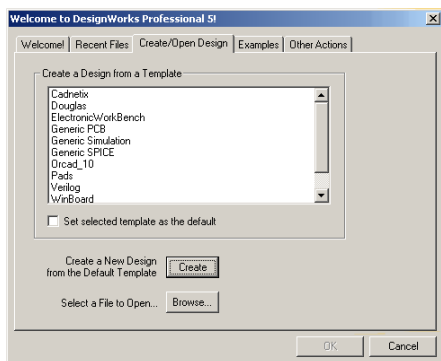
Click on the Create/Open Design tab.

Select “Generic PCB” in the list.

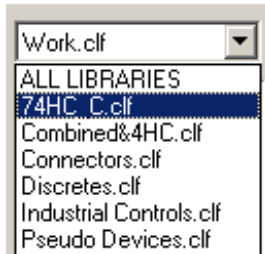
Click the OK button.

We have selected a mode which enables PCB auto-packaging functions and assumes a flat (i.e. non-hierarchical) design.

You will now see an empty schematic drawing window ready for editing.



Choosing a Library

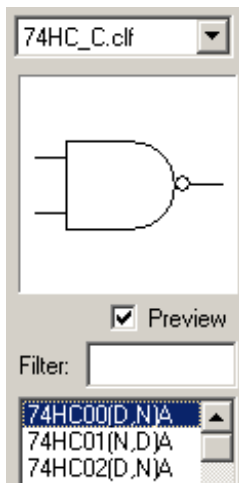


Move the cursor to the library selection pop-up menu and choose the library “74HC_C”.

The contents of this library will now be displayed in the part selection list.

NOTE: The exact parts listing may vary from what is shown here.

Selecting a Part



Move the cursor to the part selection list and click once on the item “74HC00(D,N)A” to select it. You will see a preview image appear in the box above the list.

Double-click on the same item to place it in the drawing. Move the cursor into the schematic drawing area.

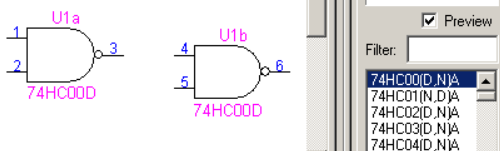
An image of the selected part will follow the cursor movement. This part does not become a permanent part of the schematic until you click the mouse button.

TIP: To place a single device, you can also just drag and drop it from the parts list.

Placing a Device

Move the cursor to an area just left of the parts palette and click the mouse once to place a device in the schematic.

Move and click again to place a second device as shown. Order of placement is not important.

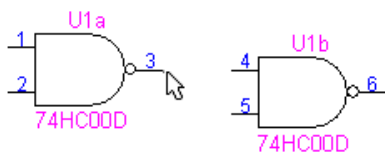


You can continue to place devices of the same type just by clicking in the desired locations. Notice that each device is automatically labeled with its part type and package assignment. This automatic assignment can be disabled if desired.

Return to the normal pointer by pressing the spacebar on your keyboard.

Wiring Pins

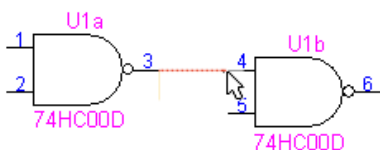
Move the arrow cursor exactly to the end of the output pin on the left-hand device, as shown, then click and hold the mouse button.



Wiring Pins (cont'd)

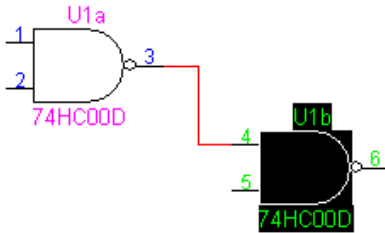
With the mouse button still held down, move to the right away from the output pin.

Move so the arrow cursor is exactly positioned over the end of the upper input pin of the right-hand device, then release the mouse button.



You will see the signal line flash briefly indicating that a connection has been made.

Moving Devices



Click and drag on the right-hand device and move it to a new position, as shown.

Notice that the connected signal line stays attached and stretches to follow the device movements.

Using Undo/Redo

Select the Undo command in the Edit menu.

Select the Undo command again, and repeat until all items you have placed have disappeared.

Now select the Redo command and repeat it until all edits are redone.

Most schematic editing operations can be undone and redone up to 10 levels. Major structural changes (like adding a page or hierarchy level) or any operation involving a dialog box cannot be undone.



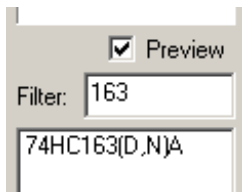
Part Selection by Name

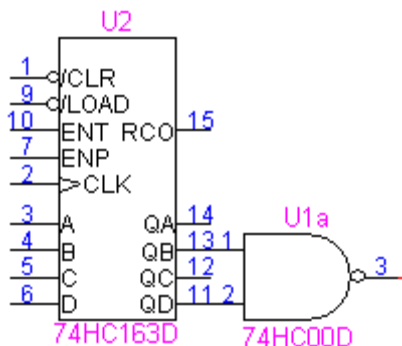
Click the mouse once in the Filter text box in the Parts Palette.

You will notice a text cursor starts to flash in this box.

Type the characters “163” on the keyboard.

Double-click on the part “74HC163(D,N)A” in the list.





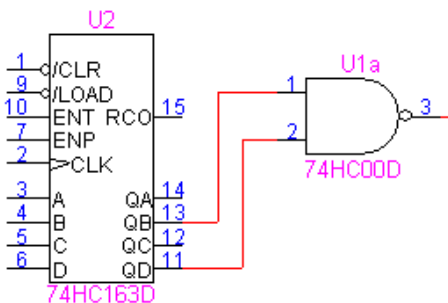
Automatic Pin Connection

Move the cursor so that you can place the 74HC163 device exactly as shown so that the QB and QD pins just touch the two inputs on the NAND gate.

Click the mouse button to place the device at this point. (Depending on the size of your screen, you may need to use the scroll bar at the bottom of the schematic window to make room on the left.) You will notice the two pins flash to indicate a connection.

Press the spacebar to return to the normal pointer.

Automatic Pin Connection (cont'd)

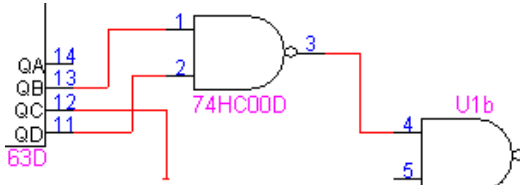


Click and drag the 74HC163 device to the left and notice that right-angle lines maintain the connections between the two devices.

This auto-connection feature can be a convenient way of making connections between large devices, such as 8-bit registers. Just place them so the two sets of pins touch, then drag them apart and all the connections are made automatically.

Wire Editing

Using the pointer tool, click and hold on the end of output pin QC on the 74HC163.



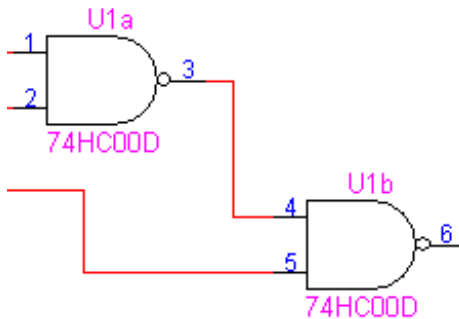
Move to the location shown and release the mouse button. (You may have to modify this procedure slightly depending on the exact positions of your symbols.)

Notice that a small perpendicular mark is placed at the end of the signal line. All unconnected line ends are marked this way automatically to simplify checking for missed connections.

Click and hold on the end of the line just completed and connect it to the lower pin on the NAND gate.

Wire Editing (cont'd)

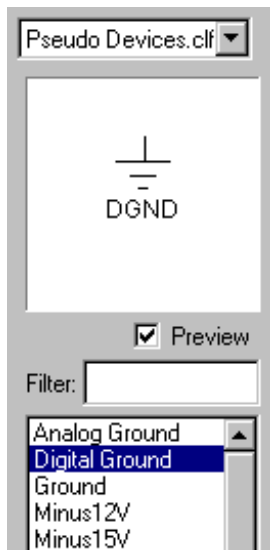
NOTE: If you're using a small screen, you can hide the Parts palette at any time by clicking in the "go away" box at its top left corner. To re-display it, use the "Parts Palette" command in the Window menu.



Using the pointer tool, click and hold at a point midway along the vertical line (or any line) just created. Notice that you can drag this line segment sideways.

With the pointer tool, clicking at the end of a pin or line segment or at an intersection allows you to extend the signal. Clicking in the middle of a segment allows you to move that segment.

The signal drawing tool (+) can be used to draw from any point.



Power and Ground Connections

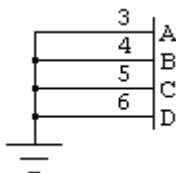
If necessary, use the scroll bar at the bottom of the schematic window to expose some drawing area to the left of the 74HC163 device.

Go to the library selection pop-up menu and select the Pseudo Devices library.

The term “pseudo device” is used in Design-Works to refer to symbols that are edited like devices on the schematic, but are actually symbols used to modify signal connections. Examples of pseudo-devices are power and ground symbols, page connectors and bus breakouts. These items will be discussed in more detail later.

Power and Ground Connections (cont'd)

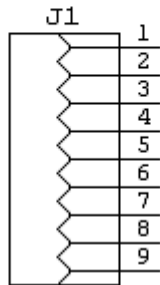
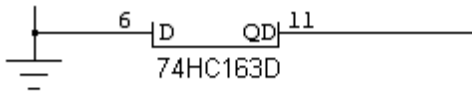
Select the Ground item in the parts palette, then place one as shown.



Click on the signal tool (+) in the toolbar and wire it to the 74HC163 device as shown.

The Ground symbol automatically names the attached net “Ground” and causes it to be logically connected to all other ground nets in the circuit.

Connector Devices



Select the library Connectors and place a DB9F connector symbol below the 74HC163, as shown.

Connectors can be treated as a single unit, as in this case, or broken up into multiple symbols each with 1 or more connector pins on them. In the netlist, these will be treated as a single device. See “Using Packaging with Connector Symbols” on page 152 for more information.

Connecting Signals by Name

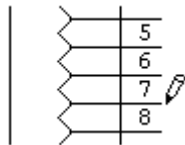


Click on the text (A) tool in the toolbar.

This tool is used to name devices and signals, edit device pin numbers, edit attribute text or create miscellaneous text notations, depending on where it is clicked. The cursor will initially take on a pencil shape (✎), allowing you to point accurately at the item to be named.

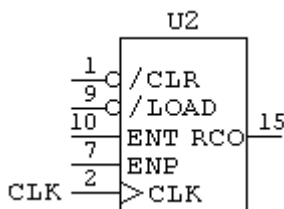
Connecting Signals by Name (cont'd)

Click the pencil tool exactly at the end of connector pin 7, as shown.



When you release the button a default name is displayed. Unless auto-naming is disabled, every signal is assigned a unique name as it is created. This name is normally only displayed on the diagram when explicitly requested.

Type the name “CLK” on the keyboard and hit the Enter key to terminate text entry.



Connecting Signals by Name (cont'd)

Move to the CLK input on the 74HC163 device and repeat the same procedure, naming this pin “CLK” as well.

Notice that when you hit the Enter key this time, the signal flashes to indicate a connection has been made with the other CLK label.

Confirming Connections

Press the spacebar to return to Point mode.

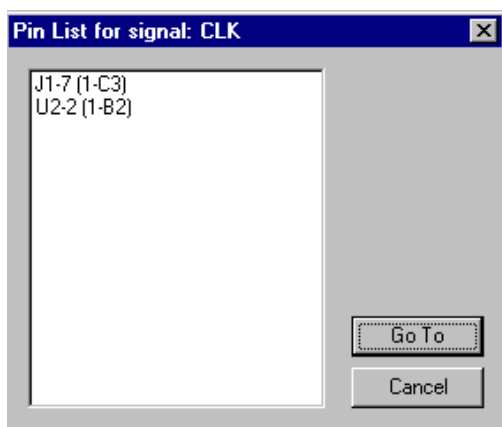
Click once on the name CLK that was just typed so that the name itself and the device pin is highlighted.

Select the Properties command in the Options menu. Click on the Pin List button.

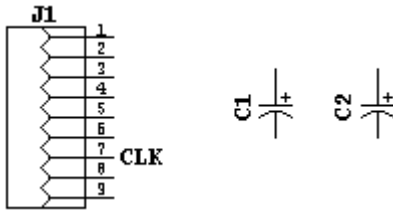
(A shortcut for this procedure is to right-click directly on the signal and select the Pin List command.) The Pin List dialog shows you all device pins that are linked by this signal, along with their page number and grid reference.

Double-click on any item in this list to display it on the screen.

The Pin List box will locate items on any page in the circuit, whether connected by signal line, by name, by off-page connector or by bus.



Discrete Components



Using the library selection pop-up menu, select the library Discretes. Double-click on the item CAP POL a and move the cursor into the schematic area.

Press the arrow keys on the keyboard (or, if you don't have arrow keys, select the Orientation command in the Options menu and select a new orientation) to orient the symbol vertically.

Place two capacitors as shown.

Return to the pointer cursor.

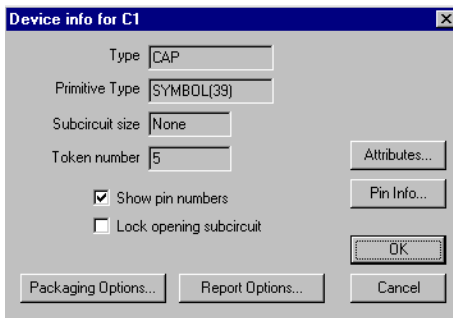
Devices can be rotated to one of 8 orientations (the 4 compass points plus mirrored versions of each). Device text notations can optionally be rotated to match the device.

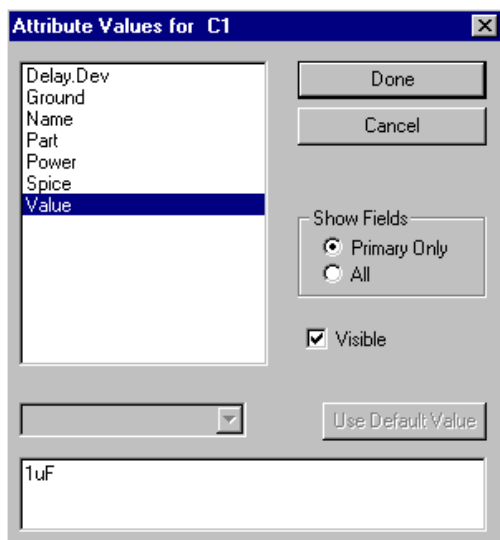
Setting Component Value

Click once on one of the capacitors just placed, so that it is highlighted.

Choose the Properties command in the Options menu, then click the Attributes button.

The box that appears allows you to view and edit text “attributes” of a device. The list at the left shows the available field names. Clicking on one of these will display the associated value in the text box.





Setting Component Value (cont'd)

Click on the item Value in the field list at left. Type the value “1uF” on the keyboard.

Make sure that the Visible box is checked, then click Done, then click the OK button on the info box.

Repeat this procedure to assign the same value to the other capacitor.

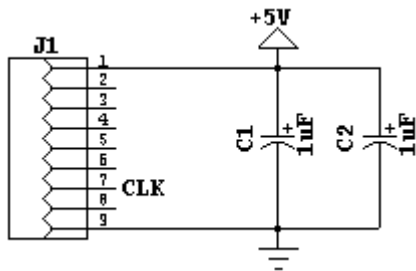
TIP: As a shortcut to get directly to the attributes box, right-click on the device and select Attributes in the pop-up menu.

The component value just entered will now appear adjacent to the device. It can be moved around independently, if desired. We will see in a later tutorial how to edit, rotate, hide and set text style for this text.

More on Power and Ground

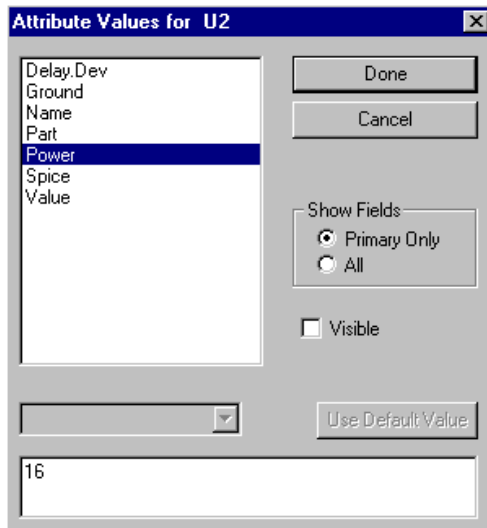
Go back to the Pseudo Devices library and place a Ground and a Plus5V symbol. (You may have to use the arrow keys again to return to normal orientation.)

Wire them to the capacitors and connector, as shown.



The Ground and Plus5V symbols are a special class of pseudo-device known in DesignWorks as a “signal connector”. They cause all like-named nets to be connected together, even across multiple pages. You can customize your own signal connectors for other types of common connections.

More on Power and Ground (cont'd)



Right-click on the 74HC163 device and select the Attributes command.

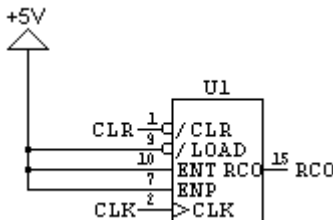
Select the field Power in the attribute field list and note the field value.

Select the field “Ground”.

You will notice that the fields named “Power” and “Ground” contain the numbers of the power supply pins for this device. You can add other pins to this list, if needed, separated by commas. This allows you to create power and ground connections without showing them explicitly on the diagram. The standard power and ground connections are included in all integrated circuit parts in the DesignWorks libraries.

Click the Cancel button on the attributes box.

More on Power and Ground (cont'd)

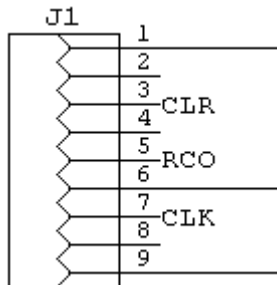


To complete the counter wiring:

Add a Plus5V symbol and wire it as shown.

Apply names to the CLR and RCO counter pins, as shown.

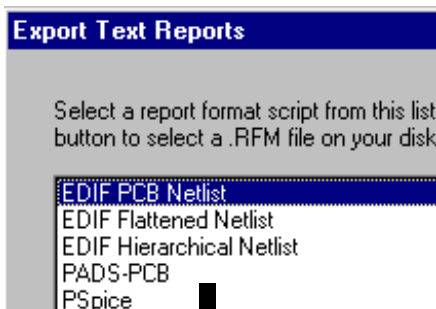
More on Power and Ground (cont'd)



To complete the connector wiring:

Using the text (A) tool, apply names CLR and RCO to the connector pins shown.

Using the signal (+) tool, draw a wire from the unconnected NAND gate output to connector pin 6.



```
(net (rename CLK "CLK")
  (joined
    (portRef P7_7 (instanceRef J1))
    (portRef CLK_2 (instanceRef U2))
  )
)
(net (rename GROUND "GROUND")
  (joined
    (portRef B_2 (instanceRef C1))
    (portRef B_2 (instanceRef C2))
    (portRef P9_9 (instanceRef J1))
    (portRef GROUND_7 (instanceRef U1))
    (portRef A_3 (instanceRef U2))
    (portRef B_4 (instanceRef U2))
    (portRef C_5 (instanceRef U2))
    (portRef D_6 (instanceRef U2))
    (portRef GROUND_8 (instanceRef U2))
  )
)
```

Generating a Netlist

We have now completed our schematic diagram and the last step is to generate a netlist for PCB layout purposes.

Select the Export command from the File menu.

Click on the Text Formats button.

Click Next.

Select the EDIF PCB Netlist item in the list of available formats.

Click the Finish button.

You have now saved a text report file that contains a connection list (netlist) for your circuit. You can use any text editor (such as Simple-Text) or a word processor to view the file. Note that the assignment of default signal names depends on order of placement and may not exactly match the sample shown at left.

Chapter 14—Report and Netlist Generation on page 353 describes more features of the report generation tool.

This completes the tutorial section “The 5-Minute Schematic Diagram”.

You may wish to use the Save As command in the File menu to save the completed example at this point. This file will be referred to in later sections.

Advanced Schematic Editing

In this tutorial section, we will cover advanced schematic editing techniques, including:

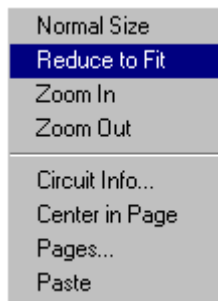
- More editing techniques and shortcuts
- Using pop-up menus
- Busses and breakouts
- Multipage schematics
- Sheet size settings
- Text notations and graphics
- Creating attribute fields
- Using the Browser tool to edit attributes
- Device packaging.

This tutorial will continue using the file created in the last section. Open the file you created, or use the file “5-Minute Schematic” provided in the “Examples” folder.

Navigating Around the Schematic


Right-click in an empty area of the schematic (i.e. not on a device or signal line).

In the pop-up menu that appears, select the command Reduce to Fit.



The screen display will be zoomed out to fit the entire schematic in the window. Holding the \S key while clicking in the schematic displays a pop-up menu containing short-cut editing commands. \S -clicking on a device, signal, pin or attribute field will display a special menu for each type of object.

Navigating Around the Schematic (cont'd)

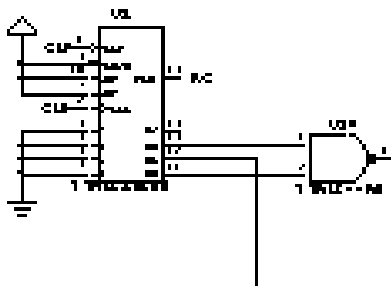
Click on the magnify () tool in the toolbar and then click near the A B C D input pins of the 163 counter.

Clicking the magnifying glass tool zooms in on that part of the schematic. Some less obvious uses of this tool are:


Clicking and dragging down and right causes the display to zoom so that the area swept over just fits in the window.

Clicking and dragging up and to the left a small amount causes the display to zoom out one step.

Clicking and dragging up and to the left a large amount (more than 1/2 the screen) causes the display to do a Reduce to Fit.

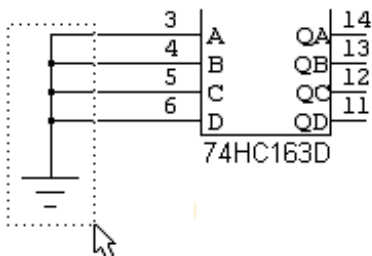


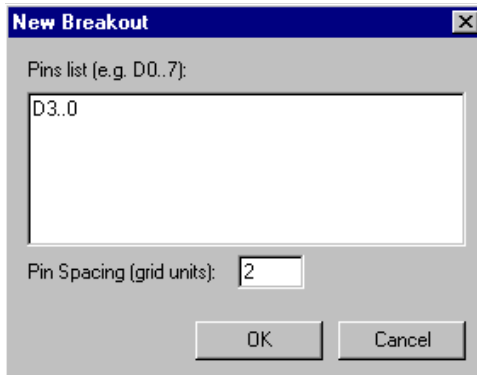
Deleting a Group of Objects

Using the pointer () tool, click and drag from above and to the left of the ground symbol to below and right of it.

Make sure that the ground symbol itself and the attached signal line are highlighted and no other items.

Hit the Delete key on the keyboard to remove these items.





Creating a Bus

Select the New Breakout command in the Options menu.

In the pin list area of this box, type “D3..0”.

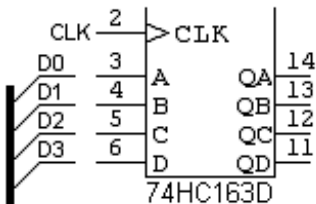
Click the OK button.

We used sequentially-numbered signals in this case, but any collection of names can be used, for example “CLK CLR SIZ0 SIZ1”. “0..3” places the highest numbered signal at the top, “3..0” places it at the bottom.

Creating a Bus (cont'd)

Place the “breakout” symbol adjacent to the 163 device, as shown. Use the arrow keys or Orientation command if necessary to orient the breakout symbol.

If not done already, connect the 4 breakout pins to the inputs of the 163 device.



A “breakout” is a pseudo-device symbol that ties any collection of signal lines into a single bus line. You can extend a bus line away from either end of the “spine” of the breakout symbol.

Creating a Bus (cont'd)

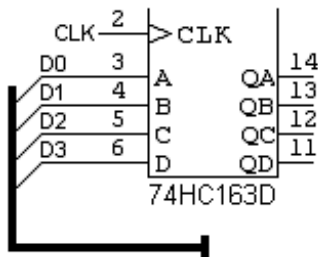
Using the pointer tool, click and hold exactly at the lower end of the “back-bone” of the breakout symbol.

Drag down and right from this point, then release the mouse button. This will have created a bus line as shown.

Right-click on the bus line. In the pop-up menu, select the Properties command.

When the Properties command is selected for a bus, a list of the internal signals is displayed.

Click OK to close the bus info box.

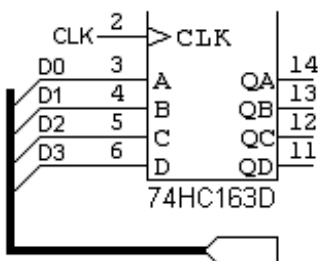


Creating a Bus (cont'd)

In the Parts palette, select a “Page Conn (Bus)” device in the Pseudo Devs library.

Place this symbol as shown so that it connects to the bus that was created above.

The Page Connector symbol indicates that this bus will be connected to other busses with the same name on other pages of the schematic. Without this symbol, connections by name occur only within a single page.



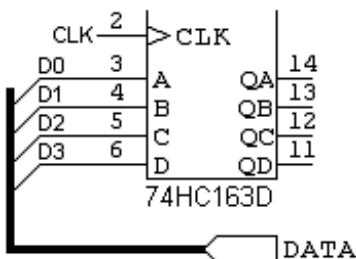
Creating a Bus (cont'd)

Select the text (A) tool in the toolbar.

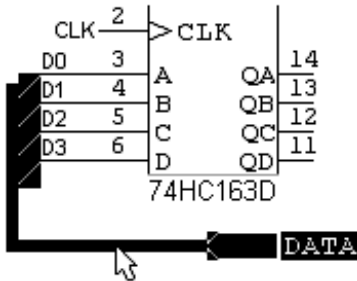
Click on the Page Connector symbol.

Type the name “DATA” on the keyboard, then hit the Enter key.

We have now named the bus “DATA”. This has no effect on the names of the internal signals, which still retain the names that were assigned in the New Breakout box.



Using the Clipboard



Return to the pointer tool and click in an unused area of the schematic to ensure that no items are selected.

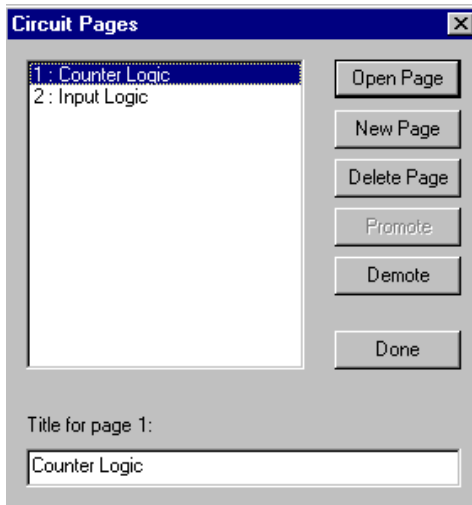
Click on the breakout symbol (i.e. on the diagonal line area in the middle of the symbol), to select it.

Holding the **Shift** key on the keyboard pressed, click on the bus line and then on the Page Connector symbol. All three of these items should now be highlighted.

Using the **Ctrl** key, you can select any collection of items on one page. These items can then be operated on in subsequent menu commands.

Select the **Copy** command in the **Edit** menu.

Adding a Page



Select the **Pages** command in the **Drawing** menu.

Click on the **New Page** button.

Enter the title “Input Logic” on the keyboard.

Click on the item “1” in the page list and enter the title “Counter Logic” on the keyboard.

Click on the **Done** button.

We have now added a second page to the circuit, which can be edited in its own window. The second page is still logically part of the same circuit and will be stored in the same file. A single circuit can have up to 1000 pages.



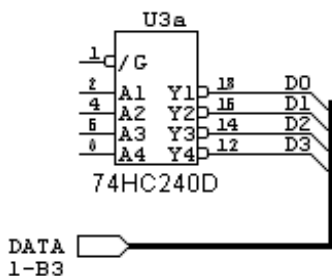
Adding a Page (cont'd)

Select the Paste command in the Edit menu. A flickering image of the bus lines copied from the first page will now appear.

Using the arrow keys, orient this entire group so that the Page Connector is on the left and the breakout is on the right.

Click the mouse button to place the circuit segment in the middle of the page.

You will notice that a new notation appears next to the Page Connector symbol. This is an automatic page reference which indicates the page and grid position of other page connectors attached by name to this one.



Adding a Page (cont'd)

Select a 74HC240 device and place it so that its outputs connect with the breakout. (You may have to use the arrow keys to set the device orientation before placing it.)

Drag this new device away from the breakout to expose some signal lines between the two.

Right-click on one of these lines.

In the pop-up menu, select the Pin List command.

The Pin List box will now indicate that the pin on the 240 device is connected via the bus and page connector to the 163 device on Page 1.

Double-click on a pin on U2 to return to the first page.

Setting Sheet Size

Select the Sheet Size Wizard command from the Drawing menu.

Click on the “Adjust all pages in the current circuit” box, if it isn’t checked already.

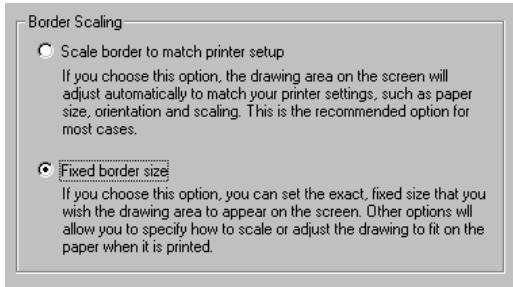
Click on the Next button.

Click on the Fixed border size button, if it isn’t selected already.

Click on the Next button.

There are two general methods for setting sheet sizes in DesignWorks. The “use printer setup” mode automatically adjusts the size of the sheet you see on the screen to match your current printer settings. This is generally the most convenient in that it guarantees that the border you see on the screen will fit exactly on the paper.

The “fixed sheet size” mode is preferable if you will be switching to a variety of printers and want the border size to stay fixed at a specified size.

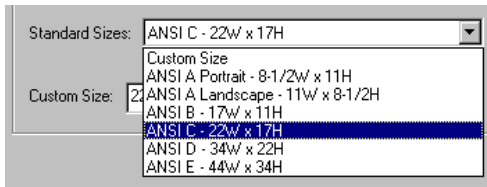


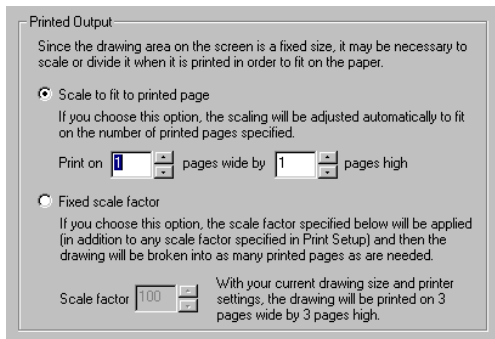
Setting Sheet Size (cont'd)

Select the standard sheet size ANSI C – 22W x 17H from the popup list of standard sizes.

Note that the list of standard sheet sizes can be customized using settings in the initialization file, so the list of settings you see may not match the ones shown here. If the one mentioned above is not in the list, pick any one that suits you at this point.

Click on the Next button.





Setting Sheet Size (cont'd)

Select the “scale to fit printed page” option, if it is not already.

Set the width and height settings to 1 and 1.

These settings will produce a large printed area on the screen and cause the program to scale the page automatically when printed to provide the best fit on the paper available.

Click on the **Finish** button to apply the new settings.

Select the Reduce to Fit command in the Drawing menu to observe the new sheet size settings.

Adding Text Notations

Move to an unused corner of the schematic page.

Select the text (A) tool in the toolbar.

Click in an open area of the schematic (i.e. not on an existing device, signal or text item). You will see a blinking text insertion point appear.

Type “All capacitors 10% tol. unless otherwise noted”, or any other notation that suits you. Carriage returns can be used in text notations, if desired.

Click outside the text entry rectangle to terminate editing.

Text notations created in this fashion are stored with the circuit, but are not associated with any device or signal and will never appear in a netlist.

All capacitors 10% tol
unless otherwise noted.



Adding Text Notations (cont'd)

Right-click on the text item that was just created.

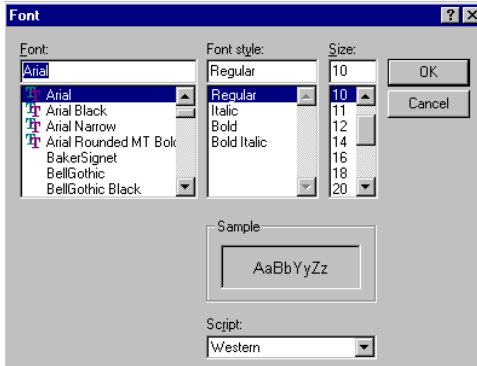
Select the Draw Frame option.

Click on the Text Specs button.

Use the controls in this box to set any desired text style.

Click OK on both boxes to return to the schematic.

Text style can be set individually for random text notations such as this one. Other forms of text such as attribute and pin numbers have global text style settings, but cannot be set individually. See the Design Preferences command in the Drawing menu.



Creating Attribute Fields

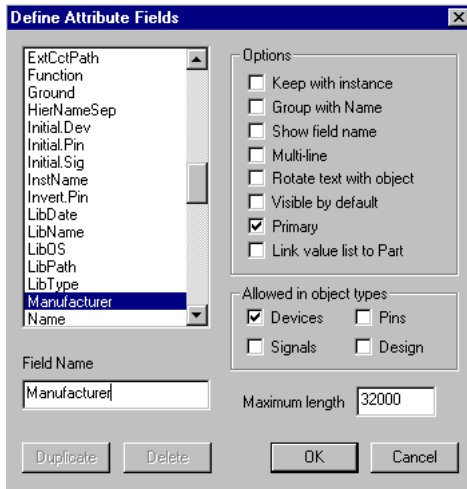
Select the Define Attribute Fields command in the Options menu.

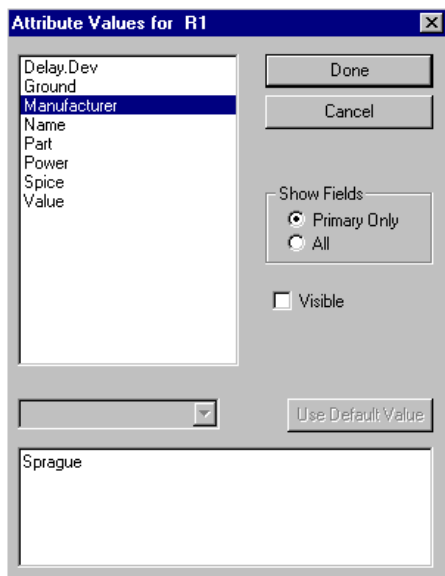
Type the name “Manufacturer” in the Field Name box, but don't hit Return or Enter yet.

Click on the Devices option under “Allowed in object types”.

Click on the OK button.

You have now defined a new attribute field that will be associated with devices. A separate value can be entered for each device in your design. The other options in this box determine how the value is stored and displayed on the diagram.





Creating Attribute Fields (cont'd)

Right-click on any device in the schematic.

In the pop-up menu, select the **Attributes** command.

Type a value for the **Manufacturer** field in the text box.

Click on the **Done** button or hit the **Enter** key.

If desired, you can enter values for this field into all the devices in the schematic. The Browser tool is a quicker method of doing this and is described later.

Attribute values can be entered for devices, signals, pins and for the design itself. These values can be displayed on the schematic and extracted in netlists and bills of materials for external use.

Using the Browser Tool

Click in an unused area of the schematic to ensure that no devices or signals are selected.

Select the **Browser** item in the **View** menu.

The Browser displays the selected type of object in a spreadsheet format. This allows you to easily locate items and edit attributes without having to search for them on the schematic sheet.

Click on any device name in the list and note that the corresponding item in the schematic is selected.

The Browser can also view Signals or Pins (see the Options menu in the Browser panel).

	Name	Ground	Power	Package
1	C1			XX2XX
2	C2			XX2XX
3	J1			XX9XX
4	U1	7	14	TI14DPN
5	U1	7	14	TI14DPN
6	U1	7	14	TI14DPN
7	U2	8	16	TI16DPN

Using the Browser Tool (cont'd)

Click on the attribute type control and select the Primary attributes.

In the field list box, double-click on the Manufacturer field, or any other desired field, to add it to the Browser window.

File	Edit	Options		Name	Ground	Power	Package	Manufacturer
Primary			1	C1			XX2XX	Sprague
			2	C2			XX2XX	Sprague
			3	J1			XX9XX	IDC
Description			4	U1	7	14	TI14DPN	Texas Instruments
✓Ground			5	U1	7	14	TI14DPN	Texas Instruments
InstName			6	U1	7	14	TI14DPN	Texas Instruments
✓Manufacturer			7	U2	8	16	TI16DPN	Texas Instruments
✓Name								

Double-click on any box in the “Manufacturer” column and entered any desired values.

Use the Enter key to move down to other cells in the same column.

Changes made to data in the Browser cause the Schematic to be updated immediately. These changes cannot be canceled or undone!

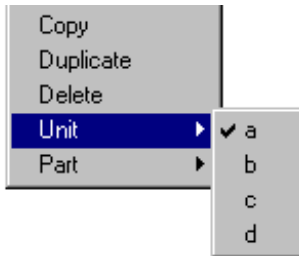
Close the Browser by clicking in the close box (X) at the top right corner.

Device Packaging

Right-click on one of the NAND gate devices on page 1.

Move down to the Unit sub-menu, near the bottom.

Select Unit “c” in the Unit sub-menu.



The Unit sub-menu contains a list of all gate units available for this part type. Selecting a different unit will update the pin numbers to correspond.

NOTE: If you select a Unit that is already in use, the program will allow the change, but Auto-Packaging will be disabled. You can re-enable it using the menu items in the Packaging sub-menu of the Options menu.



Device Packaging (cont'd)

Right-click on the same device as in the previous section and select the **Properties** command.

Click on the **Packaging Options** button.

Select the “**Lock and check package and unit**” option.

Click **OK** on both boxes.

The Packager has now been instructed to keep this device name unchanged but check it for conflicts in future packaging operations.

Device Packaging (cont'd)

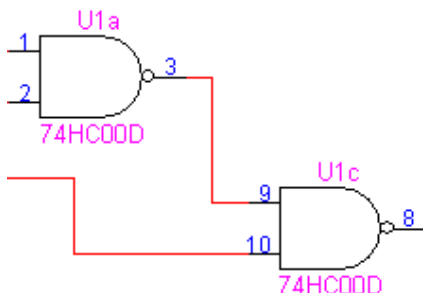
Select the **Device Naming and Packaging Options** command in the **Naming and Packaging** sub-menu of the **Options** menu.

Type the Default name prefix “**XX**” and click the **OK** button.

Select the **Repackage Design** command in the **Naming and Packaging** sub-menu of the **Options** menu.

Click the **OK** button in the **confirmation** box.

Notice that new names have now been assigned to all devices except the one that we marked as “**Lock and Check**” (and anything else in the same package). Names are assigned starting in the “**A1**” grid of the schematic, working toward the opposite corner.



This concludes the tutorial section “Advanced Schematic Editing”. The remaining tutorials deal with specific areas of the program and do not make further reference to this file.

Device Symbol Editing and Hierarchical Design

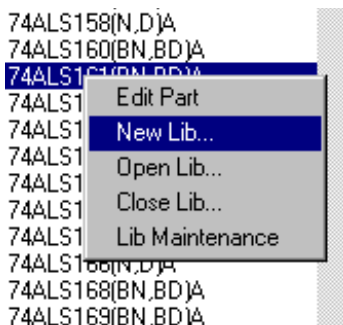
In this tutorial section we will cover some of the DesignWorks features for creating and editing device symbols and associating internal circuits with them.

Creating a New Library

Right-click on the Parts palette and select the New Lib command in the pop-up menu.

Create a new library called My Lib in the Libs folder inside the DesignWorks folder.

Device library files hold collections of part symbols along with associated pin function information, default attribute values and internal circuit definitions. A single library can contain from one to thousands of part definitions, to suit your needs.

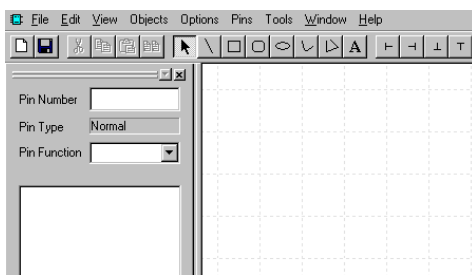


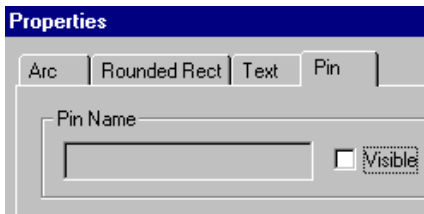
Creating a Device Symbol

Select New command in the File menu.

Select Device Symbol option in the document type list and click OK.

The device symbol editor window contains a drawing area for your symbol, a toolbar and a pin list. The toolbar includes standard drawing tools plus special items for normal, inverted and bus pin placement.





Setting Pin Name Visibility

By default, the symbol editor displays the name of each pin next to the pin stub. For this example, we want to turn this off. To do this:

Make sure no graphic items are selected in the drawing area, then select the Properties command in the Options menu.

Click on the Pin tab and turn OFF the Visible switch in the Pin Name area.

Click OK

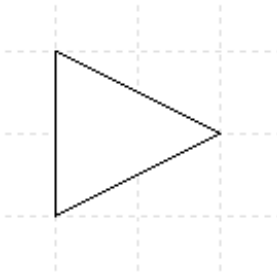
Setting items in the Properties command with no objects selected sets the default properties when future objects are created.

Creating a Device Symbol (cont'd)


Click on the polygon () tool in the toolbar.

Draw a symbol similar to the one shown by clicking once at each corner point and then double-clicking to terminate the polygon.


The position of the symbol in this window is not important.



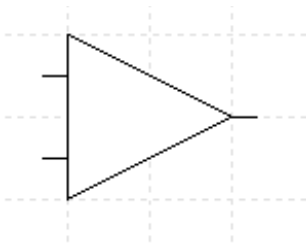
Creating a Device Symbol (cont'd)

Select the () pin tool.

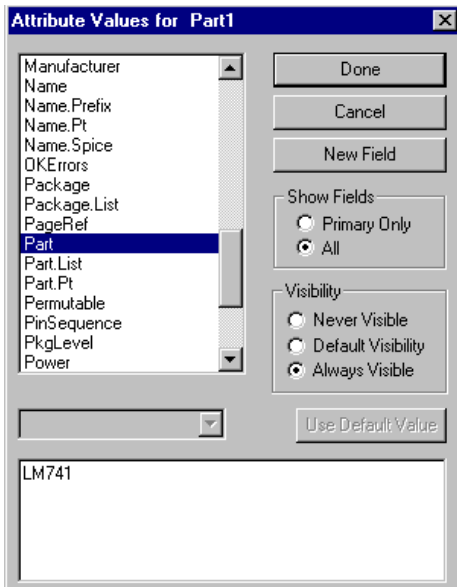
Place input pins on the symbol by clicking at the positions shown.

Select the () pin tool.

Place an output pin as shown.



NOTE: The crossbar portion of the T pin tool only appears during placement and dragging for alignment purposes.



Setting Default Part Attributes

Select the **Part Attributes** command in the **Options** menu.

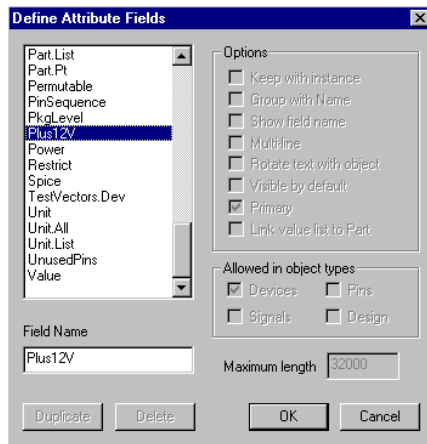
Select the **Part** field in the list.

Enter the value **“LM741”** or any other value.

Select the **Package** field in the list.

Enter the value **“NAT8DPN”** or other package code.

These attribute values will appear as the defaults when this part is used on a schematic. If desired, these values can be overridden for each individual device.



Setting Default Part Attributes (cont'd)

Click on the **New Field** button in the attributes box.

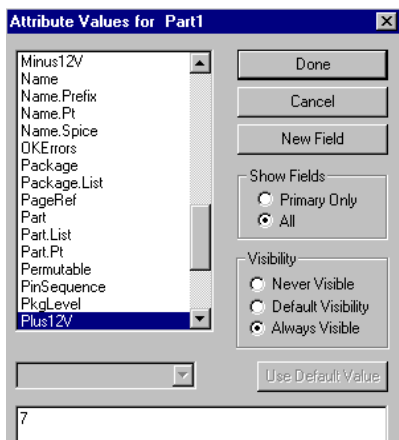
Type the field name **“Plus12V”** then type the **Tab** key to terminate this value.

Select the **“•New Field•”** item at the top of the field list again.

Enter the field name **“Minus12V”**.

Click the **OK** button.

If you create multiple devices in succession, you only need to perform the **“New Field”** operation once since the symbol editor keeps the same field list for all devices.



Setting Default Part Attributes (cont'd)

Enter the value “4” for the new Minus12V field.

Enter the value “7” for the “Plus12V” field.

These entries will determine the default power connections for this device in a netlist.

Click on the Done button to close the attributes box.

Entering Pin Names and Numbers

Double-click on the PIN1 item in the pin list.

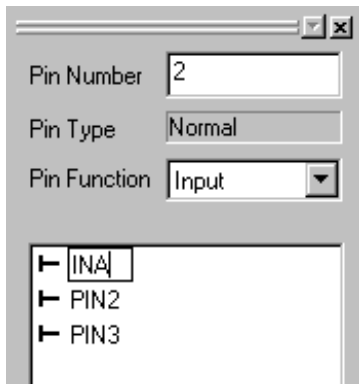
Enter the pin name “INA” then press the Enter key.

Enter the pin number 2 in the Pin Number box.

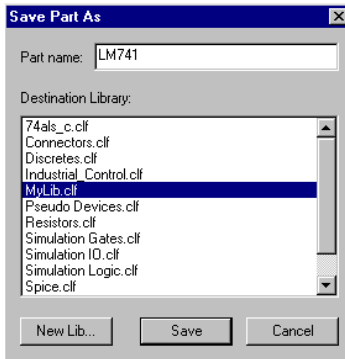
Double-click on PIN2 and enter the pin name “INB” and the pin number 3.

Similarly, change “PIN3” to “OUT” with pin number 6 then select Output from the Pin Function pulldown box.

We have now entered default values for the pin numbers that will appear in a netlist. These can be edited on the schematic for individual pins, if desired.



Saving and Using the Part



Select the Save As command from the File menu.

Enter the part name “LM741” or any other desired name.

Click on the “MyLib.clf” library in the list to select a destination and press “Save”.

Close the symbol editor window.

Saving and Using the Part (cont'd)

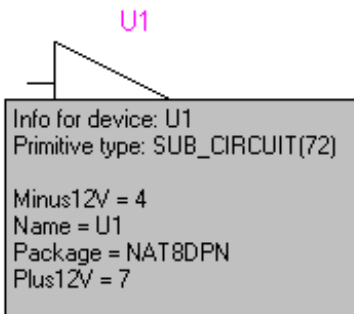
If it is not already selected, select the My Lib library in the pop-up library list in the Parts palette.

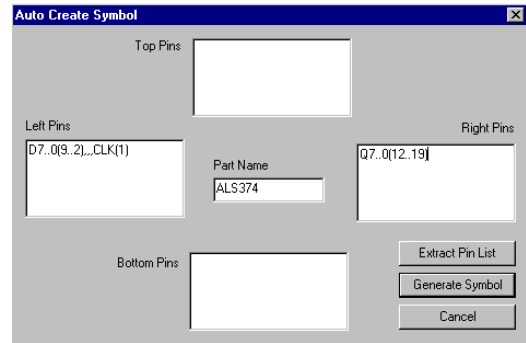
Double-click on the newly-created part and place one in the schematic.

NOTE: A warning box will be displayed indicating that new attribute fields are being defined in the design.

Select the Attribute Probe (?) tool in the schematic toolbar.

Click on the new device to verify that the default attribute values appear.





Auto-Creating a Symbol

For standard types of rectangular symbols, the Auto Create feature will generate a symbol for you in seconds.

Select New in the File menu.

Select the Device Symbol option and click OK.

Select the Auto Create Symbol command from the Options menu.

In the Part Name box, enter “ALS374”, or any other desired symbol name.

In the “Left Pins” box, enter the text “D7..0(9..2),,CLK(1)”.

“D7..0” will generate a set of 8 pins named D7, D6, etc. “(9..2)” are the corresponding pin numbers. The three commas indicate that we want extra space between these pins. “CLK(1)” creates a single pin called CLK with pin number 1. The pin numbers can be omitted, if desired.

In the “Right Pins” box, enter the text “Q7..0(12..19)”.

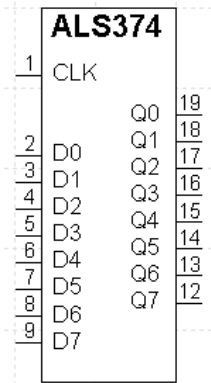
Click on the Generate Symbol button.

Auto-Creating a Symbol (cont'd)

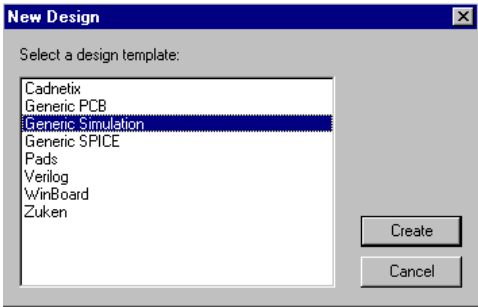
The auto-generated symbol should now display the pins and pin numbers entered above. These items can be edited using the drawing tools and Pin Info box, if desired.

Select the Save As item in the file menu and save the new part to the My Lib library.

Close the symbol editor window.



Creating a Hierarchical Block



Select the New command from the File menu.

Select the Design option and click OK.

Select the Generic Simulation design template.

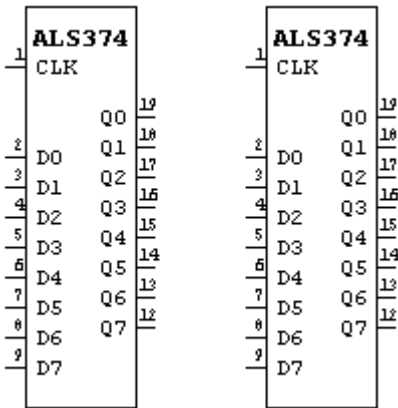
The Generic Simulation template is preconfigured to allow hierarchical design, which will allow us to create nested circuit blocks inside device symbols.

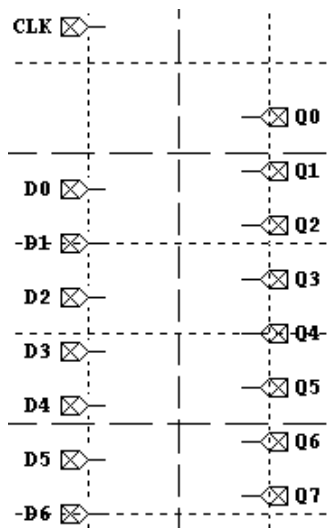
Creating a Hierarchical Block (cont'd)

In the Parts palette, double-click on the ALS374 device that was just created.

Place two of these devices on the schematic, as shown.

When new symbols are made in the symbol editor, they are set by default to be “Sub-Circuit” devices, meaning that an internal circuit can be created in them, even if none is provided initially.



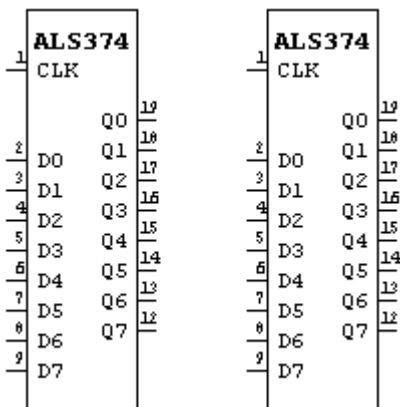


Creating a Hierarchical Block (cont'd)

Double-click on either of the ALS374 devices that were just placed.

Click OK in the warning box to create a subcircuit.

A new window will now appear showing the internal circuit for this block. When a new subcircuit is created, “port connector” symbols are automatically laid out to match the pins on the parent symbol. Any signal connection made to a port connector becomes logically connected to the pin with the same name on the parent symbol.



Creating a Hierarchical Block (cont'd)

Close the subcircuit window.

This will cause the definition of all devices of the same type to be updated throughout the design. Double-clicking on any other device of the same type will now reveal the same subcircuit.

NOTE: Unless you are an experienced user of hierarchical design concepts, please refer to Chapter 6—Before Starting a Major Design on page 113 before using these features.

This concludes the tutorial section on Device Symbol Editing and Hierarchical Design. More information on hierarchical design features is found in the Appendix.

Using DesignWorks with SPICE-based Simulators

DesignWorks makes an ideal schematic entry tool for SPICE-based simulators and others with similar formats, such as PSpice™, SMASH™ and standard SPICE. The same principles also apply to exporting data for use with any external simulation, layout or analysis system.

In this tutorial section, we will create a simple SPICE circuit and write out the complete netlist. You can then try this netlist out in your SPICE package.

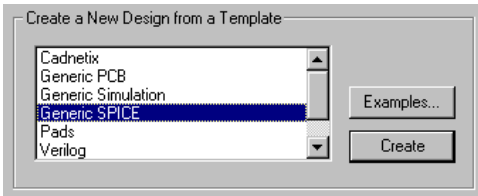
Creating a SPICE Design

Start the DesignWorks program or, if it's already running, select the New command in the File menu and select the Design document type.

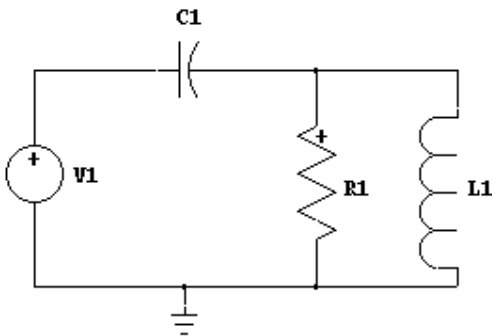
Select Generic SPICE template.

Click on the Create button to create an empty design window.

This template enables an option that auto-names devices using a SPICE prefix supplied in the library. In the SPICE Devices library provided with DesignWorks, this field contains the standard prefixes for SPICE models.



Placing SPICE Devices



Select Spice in the library selection pop-up menu in the Parts palette.

The SPICE library contains symbols that have the standard SPICE prefix and pin order pre-defined so that appropriate names are assigned and netlist pin order is correct.

You can use any symbol from any library in a SPICE design, but you will need to ensure that an appropriate name is assigned and that the pin order in the netlist matches the requirements of your particular SPICE-based simulator.

Using the devices Ind Volt. Src, Spice Ground, Spice Res, Spice Cap and Spice Ind, create the circuit shown at left, or anything similar.

Some of the passive components in this library, such as the Spice resistor, as marked to ensure consistent polarity for current-measuring purposes.

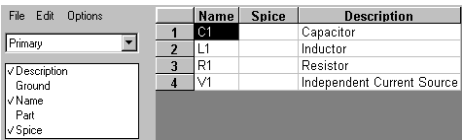
Entering SPICE Parameters

Click in an unused area of the schematic to ensure that no devices or signals are selected.

Select Browser from the View menu.

If the “Spice” attribute field is not already displayed in the spreadsheet area, click on the Attr Type control and select the Primary attributes, then, in the Attr List Box, double-click on the Spice field to add it.

All text entered in the Spice attribute will be included in the netlist for with the associated device.



Entering SPICE Parameters (cont'd)

	Name	Spice
1	C1	1000pF
2	L1	100nH
3	R1	1K
4	V1	PULSE(0V 10V 50NS 10NS 10NS

Click in the box under Spice and adjacent to device C1 in the list.

Enter the value “1000pF”.

Enter values for the remaining devices as follows:

L1 100nH

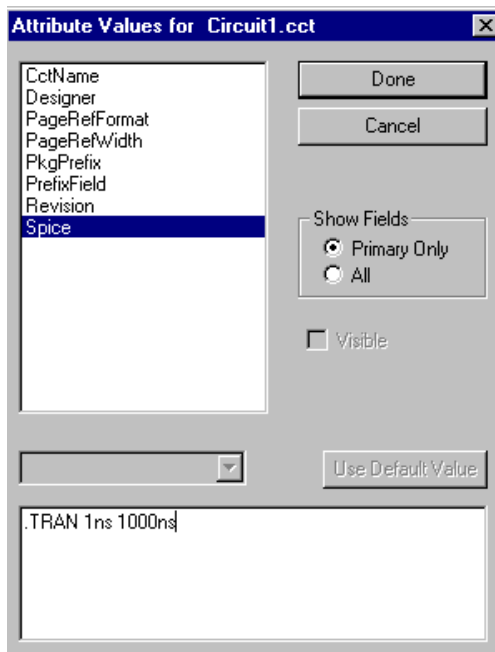
R1 1K

V1 PULSE(0V 10V 50NS 10NS 10NS 100NS 1US)

Hit Tab after entering the last value to be sure the schematic is updated.

Close the Browser by clicking in the window's close box.

Entering SPICE Parameters (cont'd)



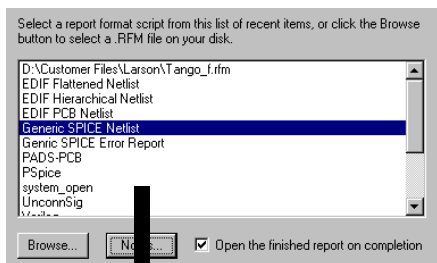
Select the command Design Attributes in the Options menu.

Select the Spice field in the list.

Enter the text “.TRAN 1ns 1000ns”. If you are using PSpice, you may wish to hit the Return key and enter a “.PROBE”. line to generate output for the Probe display program.

Any text entered in the design's Spice field will appear at the beginning of the SPICE netlist. This can be used to enter SPICE commands, model definitions, etc. Multiple lines can be entered if desired using the Return key. You can also use the -V (paste) keyboard combination to paste larger numbers of lines into this box.

Click on the Done button to close the attributes box.



```
Design2 Monday, October 23, 2000 11:52 AM
.TRAN 1ns 1000ns
C1 SIG3 SIG7 1000pF
L1 SIG7 SIG8 100nH
R1 SIG7 SIG8 1K
V1 SIG3 SIG8 PULSE(0V 10V 50NS 10NS 10NS 100NS
1US)
.END
```

Creating a SPICE Netlist

Select Export from the File menu.

Select the Text Formats option and click Next.

Select the script Generic SPICE Netlist in the format list.

Click on the Finish button.

Select an appropriate output file name and click on the Save button.

The netlist file just created can now be fed directly to your simulator.

This concludes the tutorial section Using DesignWorks with SPICE-based Simulators.

DesignWorks can also directly create hierarchical SPICE netlists with .SUBCKT definitions. Hierarchical block symbols can be nested to any desired depth.

This chapter provides a quick overview of a DesignWorks schematic and how it is stored. It will then provide simple procedures for creating, saving, opening, navigating around and printing documents, and look at the important issue of backup procedures.

DesignWorks File Types

DesignWorks deals with a number of different types of files. Far and away the most important type is the design file, which contains all the stored data of your schematics. Operations on design files are discussed in this chapter. More information on the other types of data files that you may encounter can be found in these parts of the manual:

This table summarizes the types of files used in DesignWorks:

Description	File Extension	More Information
Design Files	CCT	This chapter
Symbol Library Files	CLF	Chapter 12—Device Symbols and Libraries on page 261
Report Script Files	RFM	DesignWorks Export Script Language Reference (separate manual on disk)
JavaScript Files	DWJ	DesignWorks JavaScript User's Guide (separate manual on disk)
Design and Sheet Template Files	CCT	Chapter 13—Design Templates and Customization on page 329
Initialization Files	INI	Appendix D—Ini File Format on page 375
Text Files	TXT	Supported for report generation and utility purposes.

Design Structure

In DesignWorks the term “design” refers to a complete, independent logical entity which is saved in a single file. The following rules outline how a design is stored:

A single design is stored in a single file and no logical connections are made between designs. All information required to display and edit a design is stored in the design file.

A design never makes reference to external library files. When a library device is used, all information needed is read from the library and stored with the design. Changing the original library definition will not automatically update the design. More information on the relationship between symbol libraries and designs can be found in Chapter 12—Device Symbols and Libraries on page 261.

When a design file is opened, the entire contents of the design are read into memory. This means that design sizes are limited by the available memory in your computer and increasing the memory allocated to the program will increase the size of the designs you can work with.

A number of user-settable parameters are stored with the design and affect the entire design when changed. These include:

- attribute and pin number text style settings
- the attribute definition table
- signal and device auto-naming settings
- display options, such as crosshairs and printed page breaks
- page reference format
- hierarchy mode

In a hierarchical design (that is, one in which a symbol can represent a nested circuit block), all hierarchy levels are stored in a single file. More information on hierarchical design is in Chapter 10—Hierarchical Design on page 215.

For the balance of this chapter, we’ll assume we’re dealing with a simple, “flat” design (i.e. not hierarchical) consisting of a single circuit level.

What is a Circuit?

A circuit in DesignWorks has these characteristics:

A circuit can be drawn on one or more pages (up to 1000 in this version). You can elect to draw the entire circuit on a single page or divide it up functionally onto a number of pages.

Each page is viewed in a separate circuit window and any or all pages in a circuit can be displayed on the screen simultaneously.

A page is drawn on the screen as if it was a single piece of paper, although it may have to be broken up into a number of individual sheets of paper for printing or plotting.

Logical connections can be made between pages using the Page Connector device allowing for full simulation and netlisting across pages.

Types of Objects in a Circuit

A DesignWorks circuit is made up of four types of entities: devices, signals, text objects and picture objects.

A device is an object having a symbol, signal connection points called “pins”, and optional text attributes, internal circuit and simulation information. A device in DesignWorks can correspond to a physical device in a circuit, or it can represent a sub-circuit block or a *pseudo-device*, such as a page connector or bus breakout.

A pin is a connection point on a device. A pin is not an independent entity since it only exists as part of a device and cannot be created or deleted separately. However, pins can have attributes, pin numbers and other parameters that may be different from pin to pin on the same device. A *bus pin* is a special type of pin that represents an arbitrary number of internal pins. The internal pins are not visible on the schematic but can still have the same logical properties as other pins.

A signal is a conductive path between devices. Signal connections can be made visually by drawing lines between device pins, or logically by name or bus connection.

A text object is used to place a title block or other notation on the diagram. Text can be typed and edited directly within DesignWorks, or can be created externally and pasted onto the diagram from the clipboard. Text objects are not associated with any other object and are not

accessible through net or component lists. The attribute facilities should be used to associate text with specific devices or signals.

A picture object is used to place any graphics item imported from another program or the symbol editor tool via the clipboard. The Schematic tool cannot edit picture objects directly, other than moving, deleting and copying them whole. They can, however, be pasted into a symbol editor window, edited using the device symbol tools, then copied and pasted back into the schematic.

Creating a New Design

A new design is created by selecting the New command in the File menu, then selecting Design, and then choosing from one of the available templates. The new design will consist of an empty circuit with a single page. This page will appear in a circuit window entitled “Design1:1” (the “:1” portion indicates that this is page 1). The New command does not create a disk file. The design exists only in memory until you do a Save As.

Your circuit diagram is created by first placing one or more devices in the circuit window (as described below), and then interconnecting the device pins with signal connections.

Choosing a Template

DesignWorks does not have a fixed set of templates, but rather displays the ones that are available in the designated Templates folder. See the accompanying sidebar for more information on how this is done. A template may contain a variety of initial settings for options and items such as a sheet border and title block, that make it easier for you to get started on a design. We encourage you to create your own templates that suit your application, once you are comfortable with the basic operation of DesignWorks.

See Chapter 13—Design Templates and Customization on page 329 for more information on creating design templates.

NOTE: A template is only a convenient way of starting out with the appropriate settings for a certain type of design. There is nothing in a template that is carved in stone and all settings in the design can be changed later if your requirements

change.

To get started:

If you have just installed DesignWorks and want to start a basic design, pick the template that is closest to your application. The Generic PCB template is a reasonable default choice for board-level designs. Before you start any significant design entry, you should refer to Chapter 6—Before Starting a Major Design on page 113.

If you are an experienced user of earlier versions of DesignWorks: You should refer to “Creating Design Templates” on page 329 for information on setting up templates with your preferred initial settings.

If you are a new user of DesignWorks, but you have walked up to an already-installed package, the list of templates you see will depend on which design kits are installed. If there is any doubt about the usage of the templates that you see, you should consult the person who installed them, or refer to Chapter 13—Design Templates and Customization on page 329 for more information on the types of settings that should be in a template.

How Design Templates Work

A design template is simply a normal design file that has its sheet size, attribute fields, hierarchy mode and other settings pre-defined for the application at hand. The simplest way to create a template may be to take an existing design that is set up the way you like it, delete all the circuit elements and extra pages out of it and save it in the appropriate template directory.

When you create a new design and select one of the template files listed, DesignWorks just reads the file in the normal way, then renames it “Design1” (or the next available number), and disassociates it from the original file so that it cannot accidentally be Saved on top of the template. In all other respects, creating a new design using the New command is the same as doing an Open on the template design. All the settings and contents of the design template file become part of the new design.

Opening a Design

A design file can be opened in one of two ways:

Select the Open command in the File menu. This command allows a design to be opened from a disk file using the standard Windows file open box.

Locate the file using the Windows Explorer and double-click on the file's icon. If DesignWorks is not already running, it will be started up. If it is already running, it will come to the front and the selected file will be opened.

When you open a file, the circuit data is read into memory in its entirety and no more access to the disk file is required.

There is no fixed limit on the number of designs that can be open at once, although the complete contents of all open designs must fit into memory.

Compatibility With Older Versions

DesignWorks Professional for Windows 5.0 can directly read files created by all DesignWorks Professional 3.x versions on Windows as well as all versions of DesignWorks Lite and LogicWorks versions 3.0 and above. The files created by DesignWorks 5.0 cannot be read by older versions.

Navigating Around a Schematic Page

In addition to the standard scroll bars, DesignWorks has these convenient features for moving around a diagram:

Zooming In and Out


Controlling Zoom by Menu Commands

The Drawing menu contains four menu items for controlling the zoom factor of a window: These commands control screen display only and have no effect

on the stored circuit information, printed output, or graphics files. Due to the integer calculations that are done by DesignWorks and by the Macintosh system, device symbols and text may be displayed rather crudely at scale factors other than 100%. It is best to do most editing at normal size to ensure that everything lines up as you would expect.

Normal Size	Sets the screen scale to 100%.
Reduce to Fit	Sets the scale factor and centers the display so that the entire diagram fits on the screen. If the size of the diagram and the size of the window is such that this would require a scale factor of less than 20%, then the scale is set to 20% and the diagram is centered. If the diagram fits completely in the window at 100%, then the scale is set to 100% and the diagram is centered.
Enlarge	Increases the scale factor by about 20%, causing the diagram to appear larger on the screen. If this causes the setting to exceed the maximum of 200%, it is set to the maximum.
Reduce	Decreases the scale factor by about 20%, causing the diagram to appear smaller on the screen. If this causes the setting to go below the minimum of 20%, it is set to 20%

Controlling Zoom with the Magnifying Glass Tool

The  item in the toolbar is a powerful tool for moving around in a schematic diagram. It can be used to zoom both in and out and can control the exact area displayed on the screen.

Clicking and releasing the mouse button on a point on the diagram will zoom in to that point by one step.

Clicking and dragging the mouse down and to the right zooms in on the selected area. The point at which you press the mouse button will become the top left corner of the new viewing area. The point at which you release the button will become approximately the lower right corner of the displayed area. The circuit position and scaling will be adjusted to display the indicated area.

Clicking and dragging the mouse upward and to the left zooms out to view more of the schematic in the window. The degree of change in the scale factor is determined by how far the mouse is moved. Moving a small distance zooms out by one step (equivalent to using the Reduce command). Moving most of the way across the window is equivalent to

doing a Reduce to Fit.

Opening Circuit Page Windows

In DesignWorks, each circuit page is displayed in a separate window that can be opened, closed, scrolled and zoomed independently. The Pages command can be used to display any existing page, as well as adding new pages and setting page order and title. The Pages command can be found in the Drawing menu in the main menu bar, or in the circuit popup menu which is displayed by right-clicking in an empty location in the diagram.

See more information on the Pages command in “Creating a New Page” on page 76.

Locating Circuit Objects with the Find Tool

The find tool allows you to locate any device, signal or pin by name or other parameter. It will automatically open closed pages and adjust zoom and scrolling to highlight the located objects. To use it:

Select Find in the Drawing menu.

Select the type of object you wish to search for, device, signal or pin.

Enter the name of the object to search for, or other search criteria.

Click the Search button.

The Find tool automatically displays the first item found. The list of found items can be used to look at any other objects matching the search criteria.

TIP: The Find tool is scriptable and can be customized to find objects or scan the design for many different purposes.

More detailed information on the Find tool is provided in “Using the Find Tool” on page 246.

Locating Objects Using the Browser Tool

The Browser tool is a convenient way of viewing all objects in the design in tabular format. Clicking on any item in the Browser’s spreadsheet area automatically displays the corresponding item in the schematic. To display the Browser, select the Browser command in the View menu.

See “Using the Browser Tool” on page 248 for more information.

Saving a Design

An open design can be saved to disk using the standard **Save** and **Save As** commands in the File menu. **Save** saves the circuit back into the file that was most-recently opened. It will be disabled if no file has been opened or if you've created a new design that has never been saved. If you select **Save As**, a standard file save box will be displayed requesting the name the new file. The default name will be the current title of the circuit window, i.e. the name of the most recently opened or saved file.

Reverting to a Saved File

The Revert command in the File menu rereads the current design from the disk file it was last saved to or read from. If any changes have been made since the last Save, you will be prompted to confirm the choice before they are discarded.

Saving a Circuit Page in WMF, DXF or PDF Graphics Formats

DesignWorks can save diagrams in the standard WMF (Windows Metafile), DXF (AutoCAD) and PDF (Portable Document Format, also known as Acrobat format) graphics formats. This capability allows you to pass graphics to other programs for plotting, enhancement, or incorporation into other documentation. Here is some more information on these formats:

WMF is a standard format under Windows for representing graphics objects such as lines, circles, text, etc. This format can be read by many Windows-based graphics and word processing packages. Note that DesignWorks cannot read these files, it can only create them. A WMF file is strictly a graphics format and does not contain information about signal connectivity, pin functions, attributes, etc.

DXF is a format developed by AutoDesk for use with their AutoCAD series of general CAD packages. In addition to AutoCAD, most general drafting packages can read and write DXF. Like WMF, a DXF file is strictly a graphical representation of a diagram and does not contain all the information necessary to construct a logical schematic. DesignWorks has no ability to read DXF files. Note that Enhanced Metafile and bitmap

pictures that have been pasted onto a diagram (e.g. for a logo in a title block) are not exported to DXF.

PDF is a standard file format used for displaying print-quality documents in portable format that can be sent over the internet and displayed on any platform. A PDF format is strictly a graphics format and does not contain any information about signal connectivity, etc. For this reason, it cannot be re-imported back into DesignWorks.

Generating a WMF File

- ➔ Open the design file you wish to export. If your design has multiple pages and you only want to export one of them, make that the current page (i.e. the topmost window).
- ➔ Select the Export command in the File menu.
- ➔ Select the WMF option and click Next
- ➔ Select either “Export all Pages” or “Export Current Page Only”, as appropriate.
- ➔ If desired, click the Browse button to enter a different filename. The default will be the same name as the design with the extension changed to “WMF”.
- ➔ Click the Finish button to complete the operation.

Generating a DXF File

- ➔ Open the design file you wish to export. If your design has multiple pages and you only want to export one of them, make that the current page (i.e. the topmost window).
- ➔ Select the Export command in the File menu.
- ➔ Select the DXF option and click Next
- ➔ Select the appropriate color option. If you choose the “Export Color Information” option, DesignWorks will choose the closest standard DXF color for each object on the schematic.
- ➔ Choose whether you want to use Release 12 or Release 13 format. Release 13 provides better representation of curves and ellipses, but is not compatible with all readers. You should try a test run with the desired target application (making sure your test includes a variety of symbols) before doing any major conversions.
- ➔ Click the Next button.

- ➔ Select either “Export all Pages” or “Export Current Page Only”, as appropriate.
- ➔ If desired, click the Browse button to enter a different filename. The default will be the same name as the design with the extension changed to “WMF”.
- ➔ Click the Finish button to complete the operation.

Generating a PDF File

- ➔ Open the design file you wish to export. If your design has multiple pages and you only want to export one of them, make that the current page (i.e. the topmost window).
- ➔ Select the Export command in the File menu.
- ➔ Select the PDF option and click Next
- ➔ Select either “Export all Pages” or “Export Current Page Only”, as appropriate.
- ➔ If desired, click the Browse button to enter a different filename. The default will be the same name as the design with the extension changed to “PDF”.
- ➔ Click the Finish button to complete the operation.

NOTE: Please note that DesignWorks makes use of a third-party PDF-export utility called NovaPDF that is included with the DesignWorks Professional package. If this package was not installed when DesignWorks was installed or has been removed from your system, the PDF export function will not be available.

Printing

The Print command allows you to print all or part of your circuit diagram to any printer or other output device that is selectable as a printing device.

Specifying the Page Number Range

For purposes of specifying a range to print, printed sheets are numbered from top to bottom, then left to right, starting on circuit page 1. The page range settings in the standard Print dialog box correspond to printed sheet numbers, not

to circuit pages.

In multi-page circuits the print pages are numbered sequentially from the top left corner of circuit page 1 to the bottom right corner of the last circuit page. Page numbers do not appear in the printed output unless they are explicitly placed there using a text variable. See “Using Text Variables” on page 107.

In hierarchical designs, pages are numbered separately for each circuit level. The Design Preferences command is used to determine printing extent. See more information on printing in hierarchical designs in “Printing Hierarchical Designs” on page 229.

Setting the Printer Page Setup

The Print Setup command in the File menu allows you to choose the size and orientation of printer paper you wish to use. Once chosen, this information will be stored with your design file and affect the page outlines shown in the command and the Show Printed Page Breaks option in the Design Preferences command. Depending on the settings in the Sheet Size Wizard, the page setup may affect the displayed page border in your schematic

Fitting the Diagram to the Available Paper

DesignWorks has a very flexible ability to set the size of a circuit page, either independently of the current printer, or derived from the printer settings. The Sheet Size Wizard command in the Drawing menu is used to select from a number of size options. If, as a result of these settings, the diagram will not fit on a single page, it will be broken into as many parts as are needed, based upon paper size specified in Print Setup. You can preview the page breaks by using the Show Printed Page Breaks option in the Design Preferences command.

Fitting a Printout to a Single Sheet

If the current sheet size settings are such that your printout does not fit to a single sheet on the current printer, you can force it to be scaled by taking these steps:

- ➔ Select the Sheet Size Wizard command in the Drawing menu.
- ➔ If your design has multiple pages or hierarchy levels, you will be asked which of these you want to modify, so make the appropriate selection, and click Next.

- ➔ Select the “Fixed Border Size” option and click Next
- ➔ The Wizard will have filled in the sheet size setting with the current sheet size, which you should not change, so just click Next.
- ➔ Select the “Scale to Fit Printed Page” option and adjust the controls to print on 1 wide by 1 high.
- ➔ Click Finish.

When the design is printed, each page will be scaled as needed to fit on a single sheet of paper according to the current printer settings.

More information on sheet size settings is given in “Setting Sheet Sizes and Borders” on page 333.

Backup Procedures

DesignWorks 5.0 has several backup features to allow you to fine-tune your backup strategy to suit your requirements. These features are controlled by inserting keywords in the INI file, as described in the following sections.

Enabling Auto-Backup on Save

When auto-backup is enabled, the following sequence of actions is performed every time you select the Save command in the File menu when a circuit file is current:

The existing design file is renamed to the same name with “.bak” appended. If a file with the “.bak” extension already exists, it is discarded.

The design is saved to a new file with the original design name in the same directory.

Note that the “created date” of the new file is set to be the same as the old file so it will retain its historical created date.

The auto-backup feature is enabled by placing the following line in the INI file:

Backup = On

IMPORTANT: If a disk failure occurs (such as disk full) while the file is being saved, the

existing file will already have been renamed to “design.bak” and will not be automatically switched back to its original name by the program.

Enabling Timed Auto-Save

Timed auto-save allows you to request a prompt to save the design every time a specified number of minutes has elapsed. At the appointed time, a box will give you a choice of:

Saving the file. This will also create an auto-backup file, as described above, if enabled.

Postponing until another interval elapses.

Disabling timed auto-save completely.

Note that if you choose the last option, timed auto-save will be disabled until you quit and restart the program.

Timed auto-save is an extension of the auto-backup feature and is enabled by placing the following line in the setup file:

```
AutoBackup = 30
```

where “30” can be replaced by any desired number of minutes.

See “*Enabling Auto-Backup on Save*” on page 61 for more information.

Closing a Design

The **Close** command in the File menu closes all the circuit windows associated with the current design and removes all data from memory. If the design is hierarchical, all hierarchy levels will be closed. If any changes have been made to your design since the last Open or Save, then you will be asked if you wish to save those changes. The same effect is achieved by clicking the “Go Away” box in the upper left corner of the last circuit window.

Disposing of a Design

DesignWorks has no built-in command to dispose of a design file. All information about a design is stored in a single file. Deleting the file using the

Windows Explorer (or any similar system command) has the effect of erasing all data concerning that design.

Exiting DesignWorks

The Exit command in the file menu closes the DesignWorks application. If any designs are open and have been modified, DesignWorks will ask if you wish to save them.

The purpose of this chapter is to give you enough basic schematic editing procedures to allow you to create and edit a complete schematic diagram in DesignWorks. Before you start working on a major design, though, we do suggest that you review Chapter 6—Before Starting a Major Design on page 113. That chapter will look at interfacing with PCB and simulation systems, generating hardcopy output and other issues. Giving those points a bit of thought before you start can save you a lot of headaches later. This chapter will also point you to later chapters for more in-depth information on specific topics.

General Editing Operations

Undo and Redo

The Undo command undoes the last editing operation that was performed. The text of this menu item will change based on the type of operation. Generally, only schematic editing operations can be undone. Major structural changes, such as Define Attribute Fields, adding pages, or any menu commands involving a dialog box are usually not undoable. DesignWorks stores up to 10 levels of Undo information.

Undo never changes the contents of the Clipboard. E.g. after a Cut operation, Undo will restore the schematic, but leave the Cut objects on the Clipboard.

The Redo command redoes the last Undo command. It will only be enabled immediately after an Undo operation. Any other editing operation will disable this item.

The Clipboard Commands

The standard clipboard commands Cut, Copy, Paste and Paste Special can be

used to move or copy circuit fragments, graphical and text information within a single circuit window, between multiple windows, and between different programs (e.g. word processing or drafting).

Using Clipboard Data From Other Programs

When you enter DesignWorks, the clipboard may contain graphical or text information Cut or Copied from a document in another program.

Picture (Graphical) Clipboard Objects:

A picture can be Pasted directly onto your diagram in any desired position. Once placed, it can be dragged, Zapped, Cut, Copied, Duplicated, etc. using the various editing commands available for other circuit elements. Here are some notes on pasting pictures onto a schematic.

When a picture item is selected, the Properties command can be used to select an optional border and to indicate that the picture should be considered to be part of the sheet background or border.

A picture object is considered to be a monolithic item on a schematic and can only be moved or deleted in its entirety.

DesignWorks accepts pictures from the clipboard in either Bitmap or Enhanced Metafile format. The Bitmap format is typically produced by “paint” programs, such as Windows Paint, Corel Photo-Paint, Paint Shop Pro, and many others. Bitmap format is not generally recommended as it does not scale evenly and gives relatively poor printing results. Enhanced Metafile Format (EMF) is produced by object-oriented drawing programs, such as the drawing tools in Microsoft Word, Corel Draw, etc. EMF is much better suited for creating mechanical drawings or logos since it will reproduce much better on a variety of output devices. In some cases, programs may place both types of images on the clipboard. You can use the Paste Special command (described on page 68) to select which type of data is used.

Text Information on the Clipboard

Text information from a word processor or text editor can be pasted into a text block. See “Creating a Text Block” on page 104 for more information..

Using Clipboard Data From DesignWorks

When a Cut or Copy is done, two types of data are placed on the clipboard:

A picture of the selected items, which can be pasted into a graphics document using most drawing programs. Both bitmap and Enhanced Metafile Format versions of the picture are placed on the clipboard.

The DesignWorks circuit info for the selected items. This data is in a format that only DesignWorks can understand and is discarded when you exit the program.

WARNING: Circuit structural information on the clipboard is discarded when you quit the program. Only picture and text data is retained. You cannot Copy and Paste circuit data between DesignWorks sessions.

Cut and Copy work on the currently selected group of circuit objects and will be disabled if no objects are selected. When items are copied onto the clipboard, their names are copied with them, which may result in duplicate names. If duplicate signal names are pasted back into the circuit page they were copied from, then logical connections will be made between the like-named segments.

Cut

Cut removes the currently selected objects from the circuit and transfers them to the clipboard. It is equivalent to doing a Copy and then a Clear. Cut will be disabled if no objects are selected

TIP: As well as the standard location in the Edit menu, there are also Cut, Copy, Paste, Clear and Duplicate commands in the Device and Signal pop-up menus. These operate on the single object that was clicked on to display the menu.

Copy

Copies the currently selected objects onto the clipboard without removing them. This can be used to duplicate a circuit group, copy it from one file to another, or to copy a picture of the circuit group to a drawing program. See the notes on clipboard data above. Copy will be disabled if no objects are currently selected.

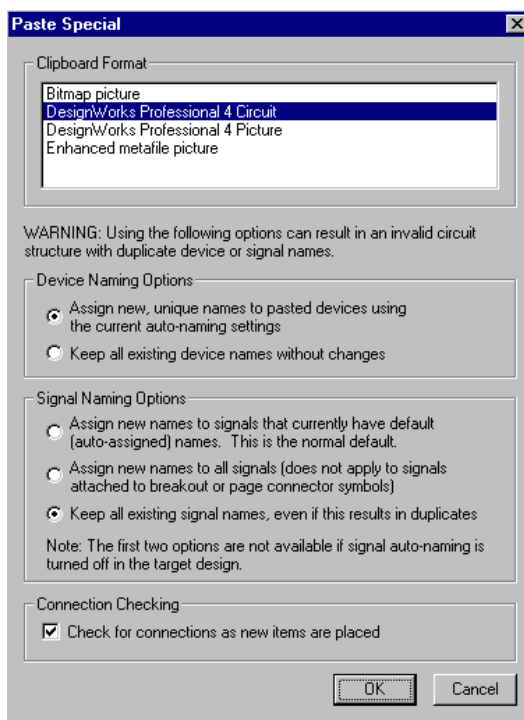
Paste

Paste replaces the cursor with an image of the contents of the Clipboard in the circuit window. The data being pasted may be a circuit group copied from

within DesignWorks, or it may be text or picture information created by another program. The image of the clipboard data can be dragged around and positioned as desired. The item will be made a permanent part of your diagram when the mouse button is pressed. Paste will be disabled if there is no information on the clipboard of a recognized type.

Paste Special

The Paste Special command performs the same operation as the Paste command described above, but it allows you to select a number of options for how the data is used. Here is a typical box that will be displayed in response to this command:



Clipboard Format

The clipboard format selection provides a list of the data formats that are currently on the clipboard and are acceptable to DesignWorks. The following

table summarizes the common formats that you will see in this list.

Bitmap Picture	This is the most common graphics format created and used among Windows applications. In general, this format is not recommended as it does not print and scale well.
Enhanced Metafile Picture	This is the Windows “object graphics” format, i.e. it describes images as collections of lines, circles, etc. For this reason, it is better suited as a format for printed logos, mechanical drawings, etc.
DesignWorks Picture	This is the internal graphics format for picture data (i.e. without circuit information) used in DesignWorks. This is the best format for copying images within the program as it preserves the image information exactly as it appears on a drawing. WARNING: This is strictly a graphics image format and contains no circuit connection data. If you copy a collection of circuit elements, then paste it as a picture you are losing all the device and signal connection information.
DesignWorks Circuit	This is the format used internally for normal clipboard operations involving schematic data. Unless you have a good reason for choosing another format, this should be the default choice for all Paste operations on a schematic.
Unformatted Text	This is a plain text format, accepted by most text processing and drawing programs. Pasting text will place a “miscellaneous text block” on the schematic.

Obviously, the exact contents of this list will depend on what Cut or Copy operations have preceded the selection of the Paste Special command.

Device Naming Options

This selection allows you to choose how names are applied to any devices contained in the circuit data on the clipboard. You can choose to apply a new name based on the current settings for naming and packaging options (the default operation), or to keep all names as they are.

IMPORTANT: Pasting with the “keep existing names” option may create duplicate device or package names that will render the name table invalid. It will then be your responsibility to make any corrections needed and use the Rescan Design command to bring the package table up to date.

See “Using Device Packaging” on page 127 for more information.

Signal Naming Options

The Signal Naming Options give you more control over the process of assigning new names to signals when circuit scraps are pasted back into a design.

The available options are:

Assign new names to signals that currently have default names—This is the normal operation performed by Paste. This option also renames signals that have a non-default name, but whose name is no longer visible because of the particular combination of elements that was copied.

Assign new names to all signals—All signals will have a new name assigned except those that have a hard connection to a pseudo-device such as a bus breakout or ground connector that forces a name. This guarantees that no connections by name are made with the new elements.

Keep all existing signal names—This option guarantees that all names are pasted exactly as they appear in the original.

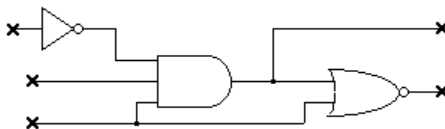
Connection Checking

This checkbox allows you to turn off the normal connection checking that occurs when circuit elements are pasted. This is described in more detail below.

IMPORTANT: This option can result in diagrams that look connected, but are not. It is the user's responsibility to make sure the diagram is cleaned up to produce appropriate connection information for netlist purposes.

Signal Connection Checking on Paste

DesignWorks checks for signal connections only at “loose ends” in the signal lines being pasted, i.e. ends of line segments which do not touch devices or other line segments. E.g. if the following circuit scrap was pasted, the points marked X would be checked for connection to the existing circuit.



NOTE: Connection “hit testing” can be disabled by holding the key on the keyboard depressed while clicking the mouse button (this also applies to single device placing), or by using the Paste Special command, described above. In this

case, the circuit scrap is placed, but no connections will be made to adjacent items. This allows the group to be selected again (by ☐-double-clicking on any device in the group) and moved without interactions with other objects in the circuit.

Duplicating One or More Objects

The Duplicate command makes a copy of the selected circuit group which can be dragged and positioned as desired. This is equivalent to selecting Copy and then Paste, except that the selected circuit scrap is not placed on the clipboard for future use. See the notes under Paste, above, on how connections are made when a group is placed in the circuit. Note that the duplicated objects can be rotated using the arrow keys on the keyboard.

Rotation on Paste and Duplicate

Any group of objects being Pasted or Duplicated can be rotated using the same controls as when placing a device:

The Orientation command in the Options menu.

The arrow keys on the keyboard.

Note that these controls are only effective while actually moving the flickering image of the object being pasted. Each Paste or Duplicate always starts in the same orientation as the source.

NOTE: The Orientation command cannot be used for Paste or Duplicate since selecting this menu command will abort the paste operation.

Selection

Many DesignWorks commands, such as Properties, Cut, Copy, etc., operate on the currently selected objects. In general, any object can be selected by clicking on it, but a few exceptions exist and numerous other convenience features are available. For a detailed list of the techniques available for selecting objects, see the following procedure sections in this chapter.

Selecting Groups of Objects

Several methods are available for selecting multiple objects:

1) Any group of adjacent items can be selected by clicking and dragging across the group. A flickering rectangle will follow the mouse movement.

Any object that intersects this rectangle when the button is released will be selected (except background objects).

2) A group of interconnected devices and signals is selected by double-clicking on any device in the group while holding the ☐ key pressed. If a circuit is completely interconnected, this will select the entire circuit.

3) The Select All command in the Edit menu selects all items on the current page except background items.

See “Selecting a Background Object” on page 109 for more information on background objects and how to select them.

4) The ☐ key can be used in combination with any of the above methods to select multiple items. When the ☐ key is held, the previously selected items are not deselected when a new item is clicked on. Thus you can add to the selected group until the desired collection of items is selected.

Deselecting a Selected Object

All currently selected objects are deselected by clicking in an empty area of the schematic window. A single item can be deselected by holding the ☐ key while clicking on it.

Multi-page Selection

The DesignWorks commands that operate on selected objects normally only affect objects on the current page. However, some external tools, such as the Browser and Find, operate across pages and can select objects on non-current pages. This is done to allow global operations to be done without having to go to each page and perform separate editing operations. For example, Find can be used to locate all devices of a particular type and select them, then Browser can be used to display and edit their attributes.

Note the following rules for multi-page selection:

Clicking in an empty part of a schematic window deselects all objects everywhere in the design.

Switching from one schematic window to another does not change the status of any objects that were selected on either page.

You can select objects on one page, switch to another page, and (using the normal ☐ key technique) add items on the new page to the selected group.

Editing commands like Cut, Copy and Clear will only affect selected items on the current page, even if objects are selected on other pages.

Zooming in on Selected Objects

The Go To Selection command causes the circuit position and scaling to be adjusted so that the currently selected items are centered and just fit in the circuit window. The scaling will be set to 100% maximum. This command works even if the selected objects are on a page other than the current one.

Adjusting the Position of All Objects on a Page

If you wish to adjust all the objects on a page to make room for editing or to re-center an edited drawing, you can:

Use the Select All command in the Edit menu to select all the objects on the page, then click and drag inside any device symbol to move the entire group.

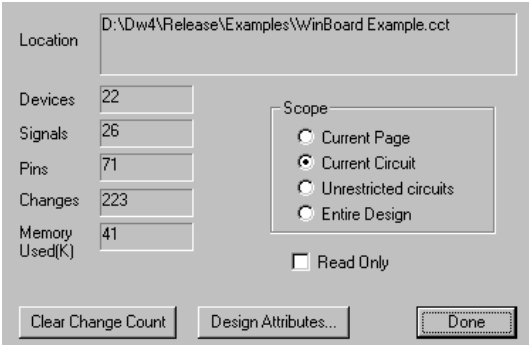
Use the Center in Page command in the Drawing menu. This command moves all items on the current page (except text and picture objects marked as “border” items) so that the circuit objects (taken as a group) are centered in the page. If the “Auto-Expand” option is turned on in the “Custom Sheet Info” box, then this command may result in the number of pages required being reduced.

More information on the Custom Sheet Info command can be found in “Setting Sheet Sizes and Borders” on page 333.

Showing Overall Circuit and Design Statistics

If no objects are selected in the circuit (i.e. if you have clicked in an empty portion of the diagram) then Properties will display this general design infor-

mation box:



The buttons on the right hand side of this box allow you to chose the scope of the displayed counts:

- Current Page

This selection shows information only on objects displayed in the circuit page displayed in the frontmost window.
- Current Circuit

This selection shows information on objects on all pages of the current circuit level.
- Unrestricted Circuits

This selection shows information on objects in the master circuit of the design and all subcircuits not marked as “Locked” in their respective Properties boxes.
- Entire Design

This selection shows information on all circuits in the design.

The following items of information are shown for the selected scope:

- nnn devices

This is a count of devices in the selected scope. Pseudo-devices, such as page connectors and breakouts are not included. In “Current Page” or “Current Circuit” mode, this count includes devices that have subcircuits. In “Unlocked Circuits” and “Entire Design” mode, this count includes only “bottom-level” devices, i.e. those without a subcircuit or whose subcircuit is not being listed.
- nnn signals

This is a count of signal nets in the design, including unconnected pins. Signals that pass through more than one hierarchy level are included once for each level that they exist in.

nnn pins	This is a count of device pins, not including pseudo-devices.
nnn changes made	This is a count of editing changes made since the design was created. This is intended to allow comparisons of different versions of the same file. This value can be cleared using the Clear Change Count button and can be extracted in reports using the \$CHANGECOUNT script keyword.
nnnK memory used	This is a count of the amount of main memory occupied by the selected part of the design, in Kbytes.

Setting Design Attributes

To edit text attributes associated with the design, you need to display the general design attribute editing box. This can be done in either of these ways:

Click in an empty area of the schematic to ensure that no objects are selected, then choose the Properties command in the Options menu. This will display the circuit info box already described. In this box, click on the Design Attributes button.

Select the Set Design Attributes command in the Options menu.

For more information on the usage of attributes, see “Entering and Editing Attribute Data - Basic Procedure” on page 160.

Making a Circuit Read Only

You can make a circuit read-only to prevent accidental modification to it. This is done by following these steps:

Click in an empty area of the schematic to ensure that no objects are selected.

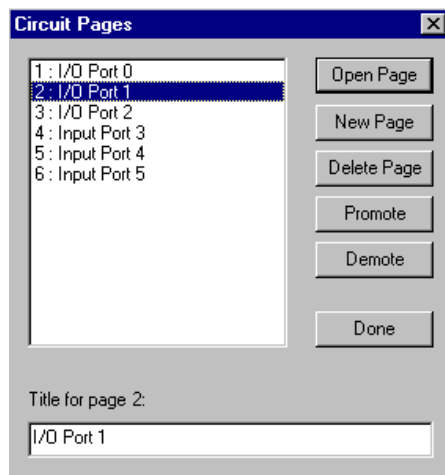
Choose the Properties command in the Options menu. This will display the circuit info box already described.

In this box, turn on the Read Only switch. Note that this switch is enabled only when the “Current Circuit” button is selected since it only applies to one circuit level.

In a hierarchical design, the “Read Only” status applies only to the current circuit level in a hierarchical design and to all levels below it. Note that “Read Only” is not the same as “Locked” in that it allows you to view the circuit but not change it.

Adding, Deleting and Titling Circuit Pages

The Pages command is used to create a new page, display an existing page, change page order, or set a page title. Executing this command will produce the following box:



Operations on pages are performed by selecting the pages in the scrolling list and clicking on one of the operation buttons. More than one page can be selected at a time (for all operations except setting the page title) by holding the **Ctrl** or **Shift** keys pressed while clicking in the list.

TIP: You can right-click in an empty area of the circuit (i.e. not on a device or signal) and select the Pages command as a shortcut to display this box.

WARNING: Operations in this box cannot be undone!!!

Opening a Page

The Open Page button causes the pages selected in the list to be displayed in a window. If a window is already open with the selected page, then it is brought to the front.

Creating a New Page

The New Page button causes a new, empty page to be added to the circuit and opens it in a new window.

Deleting a Page

The Delete Page button causes the page selected in the list to be deleted from the circuit. Note that the page must already be empty, i.e. all objects must have been removed from the page (for example using the Select All and Clear) commands before this will be allowed. Higher numbered pages will be renumbered when this page is removed.

Changing Page Order

The Promote and Demote buttons allow you to change the number order of the pages. Promote causes all pages selected in the page list to be moved one step toward the front of the list, i.e. their page number will be reduced by one. All non-selected pages will be renumbered as needed. Demote similarly moves all selected pages toward the end of the list. Promote will be disabled if the first page is selected and Demote will be disabled if the last page is selected.

Setting a Page Title

The page title text box allows a title of up to 63 characters to be entered for the selected page.

TIP: This title can be displayed in text notations and title blocks using the \$PAGETITLE variable. See “Using Text Variables” on page 107.

Working with Device Symbols

Placing a Device From a Library

To select a device from a library for placement in the schematic:

Select the desired library using the pop-up menu at the top of the Parts palette. The palette displays only the contents of only one library at a time. If the library file you need is not open, you can open it using the Open Lib command in either the Libraries sub-menu of the File menu, or in the Parts palette pop-up menu.

If necessary, use the scroll bar to scroll the list until the desired part name is in view.

Double-click on the part name in the list.

Move the cursor to the current schematic window.

The cursor will be replaced by an image of the selected device. While moving this image around, you can use the arrow keys on the keyboard or the Orientation menu command to rotate the symbol.

Clicking anywhere in the circuit diagram will make a permanent copy of the device at that point.

TIP: Holding the key while clicking will inhibit checking for pin connections. This allows you to select the device again and drag it to a new position without affecting any existing connections.

Setting Device Orientation

Device symbols can be placed in any of 8 different orientations, that is, the 4 major compass points, plus a 180 degree flip around each of those axes. The

About Device Libraries

The symbols and related parameters for DesignWorks devices are stored in data files called symbol libraries. Libraries can be opened and closed using the Open Lib and Close Lib commands, or using entries in the initialization file.

For each device type in a library the following data is stored:

- general information on the type, such as number of pins, number inputs, number of outputs, type name, default delay, default attributes, position, orientation and type of each pin, etc.
- a picture representing the symbol for this type.
- a polygon outlining the symbol, used for highlighting and erasing the symbol.
- an optional internal circuit definition.

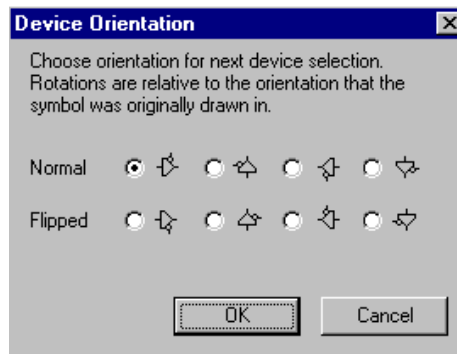
How Device Symbols are Created and Stored

Libraries are created and modified using the device symbol editor tool, which is described elsewhere in this manual.

See Chapter 12—Device Symbols and Libraries on page 261 for more information.

orientation that a symbol will be placed in is determined by the current orientation setting. This can be controlled in any of the following ways:

The Orientation command sets the orientation (up, down, left, right, mirrored) that will be used next time a device is created. When this command is selected, the following box is displayed:



The orientation can be set by clicking directly on the Orientation command in the Options menu. This displays a pop-up list of orientation icons that you can choose from.

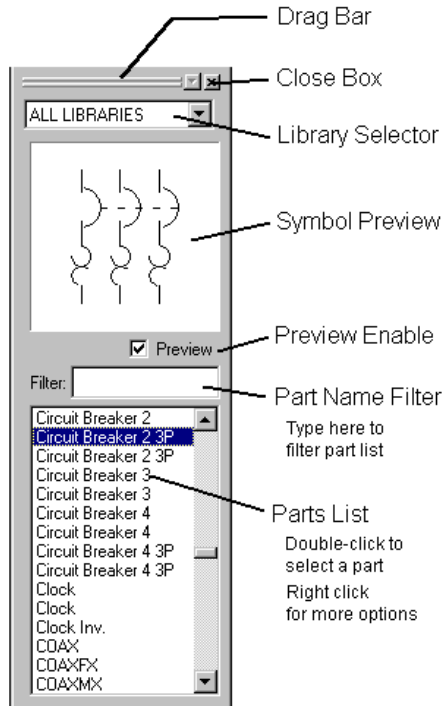
The orientation can be set using the arrow keys on the keyboard. Press any arrow key once to select that direction. Press the same key again to flip the symbol around that axis.

You can also flip and rotate a device symbol after it has been placed in the schematic. See “Flipping and Rotating a Device” on page 82.

Using the Parts Palette

Parts library contents are displayed in a floating palette window which looks

like this:



The Parts palette displays the contents of the selected library file and allows any item to be selected for placement in the schematic.

Using the Part Name Filter

The Filter text box allows you to type characters which will reduce the size of the list and make it easier to locate the desired part. Simply click in the text box and type the desired characters. After a brief pause, the displayed part list will be reduced to only those parts that contain the typed string of characters. To return to the full selection, delete all characters from this box.

The name filter also accepts several simple “wild card” characters which help

in locating items:

Character	Meaning	Example
*	Matches any string of zero or more characters	1*7 matches 17, 1775, DD123-7A
#	Matches any string of 1 or more numeric characters	#P matches XX123P5, 1P, #5PART
?	Matches any single character	#?# matches 1234, 12X12, XX1X1, but not 1XX1

Creating or Editing a Part Symbol

To create or edit a part symbol, right-click in the parts list area and select the New Part or Edit Part commands.

See “Working With Symbol Libraries” on page 262.

Moving the Parts Palette

The Parts palette can be moved to any desired location on the screen by clicking and dragging in its drag bar. Right-clicking in the drag bar also displays several display option for this window.

See “Tool Panels” on page 5 for more information.

Hiding the Parts Palette

The Parts palette can be removed from the screen by clicking in its close box. To re-display the palette, select the Parts Palette command in the Window menu.

Duplicating an Existing Device

To duplicate an existing device, either:


- Select a similar device anywhere on the current circuit page and use the Duplicate command (either in the Edit menu or in the device pop-up menu).
- Select a similar device in any other page of any open design and use the Copy command. Return to the destination circuit and select the Paste command.


After either of these operations, the cursor will be replaced by an image of the

selected device which can be placed by clicking in the schematic, as discussed above.

Deleting a Device

Devices can be removed by either of these two methods:

Select the device by clicking on it (holding the  key if it is a switch or other input device) and then hit the “delete” or “delete” key on the keyboard, or select the Clear command from the Edit menu.

Enter Zap mode by selecting the Zap command or clicking on the  in the toolbar, then click on the device in question.

Moving a Device

Devices can be moved by either:

Clicking and dragging to the desired new position, or,

Using the arrow cursor keys on the keyboard to “nudge” the selected items one grid step at a time.

If more than one device is selected, all the devices, and all signals connecting between them (whether or not selected) will be moved. Signal lines will be adjusted to maintain right angles at points where moving signal lines intersect with non-moving ones.

Flipping and Rotating a Device

A set of four commands for flipping and rotating an existing device symbol can be found in the device pop-up menu, i.e. by right-clicking on the device. The Rotate Left/Right and Flip Vertical/Horizontal commands are equivalent to deleting the selected device and replacing it in the selected new orientation.

When placing a device symbol, you can determine its orientation in advance using the Orientation command or the arrow keys on the keyboard. See “Setting Device Orientation” on page 78.

Displaying and Setting Device Information

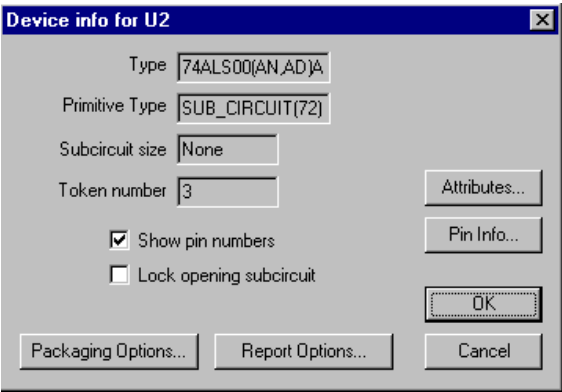
The device information box allows you to view and set a lot of vital information about the selected device. To display the device information box, either:

Right-click on the device, then select the Properties command in the pop-up menu, OR,

Select the device by clicking on it, then choose the Properties command in the Options menu.

Quite a few options are available from this box, as summarized in this table:

When a normal device symbol is selected on the schematic (i.e. not a pseudo-device), then the following information box is displayed:



The following table lists the information and options available in this box.

Part Type	This is the library type name of the device symbol, i.e. the name as it appears in the Parts palette. This is <u>not</u> the same as the Part attribute field, which is normally used as the part name in netlists.
Primitive Type	This is the primitive type of the symbol. For standard types, the name is shown, otherwise the name “Reserved” is shown. The ordinal number of the primitive type value is shown in parentheses. This is normally only of interest in specialized applications.
Token Number	This is a permanent number that is assigned to this device instance for use in internal DesignWorks operations and some back annotation and other interface operations. More information on tokens can be found in “Device Token Values” on page 152.
Subcircuit Size	If the selected device has a subcircuit, its memory size is shown in Kilobytes.
Show Pin Numbers	This switch allows you to disable the display of pin numbers for the entire device. This is intended for discrete components or others where pin numbers are not normally shown on the diagram.

Lock Opening Subcircuit

This switch allows you to prevent the subcircuit (if any) of this device block from being opened for editing. See “Locking and Unlocking Subcircuits” on page 221 for more information.

Report Options

This button allows you to select whether this device or its internal circuit will be listed in a netlist. See “Device Reporting Options” on page 356 for more information.

Pin Properties

This button displays the Pin Properties box for the first pin on the device. Buttons on this box allow you to sequence through the other pins on the same device. See “Getting and Setting Pin Information” on page 100 for more on the Pin Properties box.

Packaging Options

This button displays the current packaging level for this device. The meanings of these settings are described in “Setting Device Packaging Options” on page 134.

Attributes

This button displays the standard attribute edit box for the device. See “Entering and Editing Attribute Data - Basic Procedure” on page 160 for more information.

WARNING: Clicking Cancel on the Properties box does not cancel changes that were made in other boxes displayed using option buttons.

Entering Device Attributes

To enter device attribute, either:

Right-click on the device, then select the Attributes command in the pop-up menu, OR,

Select the device by clicking on it, then choose the Properties command in the Options menu, then click the Attributes button.

See more information on attributes in Chapter 8—Attributes on page 157.

Device Names

NOTE: In this manual, we use the term “device name” to refer to the character string that identifies a unique device in the circuit. Typical device names might be “U23”, “C4”, “IC12A”, etc. This is distinct from the “type name” or “part name” that is used to distinguish the type definition that is read from a device library. Typical part names are “74ALS138”, “MC68HC11L8”, etc.

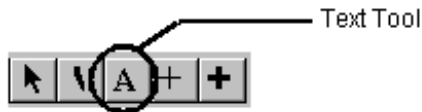
Device names may contain any letters, numbers or special characters that you can type on the keyboard, but are restricted in length to at most 15 characters.

The name associated with an object can be placed anywhere on the diagram and will be removed if the object is removed.

NOTE: Although DesignWorks does not enforce restrictions, we recommend not using blanks or special characters in device or signal names. This can result in problems exporting data to external systems such as PCB or simulation tools. See Chapter 6—Before Starting a Major Design on page 113 for more information on naming conventions.

Adding a Device Name by Typing on the Schematic

Enter “Text” mode either by selecting the Text command in the Edit menu, or by clicking on the text tool in the toolbar:




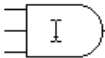
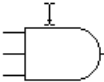
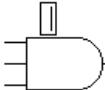
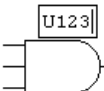
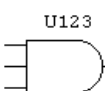
Note that once “Text” is selected, the cursor changes to a pencil icon. Two techniques are available to determine the position of the name:

If you click the pencil cursor on a device and release it immediately, the flashing text cursor will jump to the default name location for that symbol. You can then type the name as desired on the keyboard, ending with the Enter key.

NOTE: A default position can be specified for any attribute use the methods described in “Using Default Position Fields” on page 175. If no position is specified, the program will calculate one.

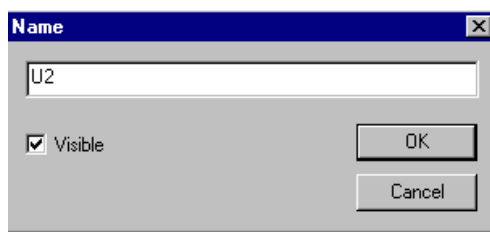
If you wish to determine the starting position of the text before you type it, press and hold the mouse button with the tip of the pencil positioned inside a device symbol. As long as you hold the mouse button down an I-beam cursor will track the mouse movements. The device name text will start at the position where you release the button. Type the desired name and press Enter, or click the mouse button anywhere. This sequence of

steps is illustrated below.

- 1  Position the pencil cursor anywhere inside the device symbol.
- 2  Click and hold the mouse button. The cursor changes to an "I-beam".
- 3  Still holding the mouse button pressed, position the cursor where you want the name to start.
- 4  Release the mouse button. A blinking insertion marker and entry rect will appear.
- 5  Type the desired name, up to 16 characters.
- 6  Click the mouse button outside the entry rect to make the name permanent.

Adding a Device Name Using a Menu Command

The name can be set by holding the **ſ** key while clicking on the device, and selecting the Name command out of the device pop-up menu. This command displays a simple edit box allowing you to enter or edit the device name. The “Visible” option in this box allows you to select whether the name is made visible on the schematic or simply kept as an invisible text attribute of the device.




Repositioning or Removing a Name

Once a name is placed, it can be repositioned by dragging it using the Point cursor or removed using the Zap cursor. The device name will be removed automatically if the device is removed. Holding the **ſ** key depressed while moving a name (or any attribute value) disables the grid snap, allowing you to micro-position the name for alignment with other graphic items on the schematic.

NOTE: The “Name” is actually an attribute field, so all attribute editing techniques can be used on it. In particular, you can hold the $\$$ key while clicking on the name and the attribute pop-up menu will appear, giving you a variety of editing commands, described “Other Ways of Viewing and Editing Attributes” on page 168.

Selecting a Device

To select a device, the cursor must be in the normal pointer () mode. A single device is selected by clicking the mouse button with the pointer positioned anywhere inside the device symbol or in any displayed attribute value associated with the symbol.

If the DesignWorks Simulator option is installed, simulated input devices such as switches and keyboards can only be selected by holding the $\$$ key while clicking. This is necessary because a normal click is used to change the state of these devices when a simulator tool is active.

Selecting the Part and Package Type

If you place a symbol that has the standard Part and Package attribute fields predefined in it, these text annotations will appear next to the symbol. Many of the standard symbols provided with DesignWorks have several possible Part names for one symbol, corresponding to various package types. To select one of the values for Part, you can either:

Use the device pop-up menu. This is done by right-clicking on the device. Move to the bottom of this menu where a sub-menu with the allowable values for the Part attribute are shown. A new value is set by selecting a unit from the Part menu item. This sub-menu will only appear if the symbol has multiple values specified in the Part.List attribute field.

Use the device attributes box to enter a new value for the Part attribute. One way to do this is to select the device by clicking on it, select the Properties command in the Options menu, then click on the Attributes button. Select Part item in the field list and enter the desired new value.

In the standard DesignWorks libraries, the Package code is automatically linked to the Part type. In other words, when you select one of the alternate Part values from the list, the Package code is automatically updated to match.

For more information on the Part, Package and other attribute fields, see “Selecting the Part and Package Type” on page 87.

Selecting the Gate Unit

If you place a gate-type symbol (i.e. one that has multiple logical units per package) from the standard DesignWorks PCB libraries, a gate unit will be assigned automatically and the unit value will be displayed after the name, e.g. U23b. When you see such a value, there are actually two attribute fields displayed side by side. In this case, the “U23” portion is the Name field and the “a” portion is the Unit. The Unit will only be shown for symbols that have gate packaging information in them.

NOTE: If Auto Packaging is enabled (it normally is for PCB designs) then you should not manually change Unit settings. See “Using Device Packaging” on page 127 for more information.

To change the values for the Unit you can either:

Use the device pop-up menu. This is done by holding the key down and clicking and holding the mouse button with the cursor over the device. Move to the bottom of this menu where a sub-menu with allowable values for the Unit attribute are shown. A new value is set by selecting a unit from the Unit menu item.

Use the device attributes box to enter a new value for the Unit attribute. One way to do this is to select the device by clicking on it, select the Properties command in the Options menu, then click on the Attributes button. Select the Unit item in the field list and enter the desired new value.

Once you select a new Unit, the pin numbers on the device will be updated to match. If you select a Unit value that is already in use, you will be given a choice of swapping with the other unit, forcing the new value, thus creating an error, or canceling the operation.

NOTE: This procedure works only for symbols that have the correct packaging attributes set up in advance. For more information on these settings, see “Creating a Symbol with Multi-gate Packaging” on page 144.

Creating and Editing Signals

In DesignWorks, a signal represents the electrical connection between any number of device pins. A signal can simply be represented on a schematic by a single line or a number of connected line segments, or more complex structures such as connection by name, busses, cross-page connectors and hierarchy blocks can be used to simplify the representation of large designs. In this chapter, we will look at simple signals and connection by name within a single circuit page. Busses and multipage signal interconnection schemes are covered in Chapter 9—Making Signal Connections on page 189.


Interconnecting Signals

If you draw a signal line such that the end of the line contacts a second signal line, then those two signals will be interconnected. Also, if you place a new device such that one of its pins contacts an existing signal line, that pin will be connected to the signal. If both of the two signals being connected were named, then you will be prompted to choose the name of the resulting signal. Whenever three or more line segments belonging to the same signal meet at a given point, an intersection dot will be placed at that point automatically.

NOTE: For efficiency, signals are only checked for connections at their endpoints and only signals actively being edited are checked. It is possible to create overlapping lines that do not connect by unusual combinations of editing operations. This situation is usually visually apparent at the time the editing is done since the intersection dot will be missing and the entire signal will not highlight when clicked on. You may wish to also use the Find tool to locate these situations.

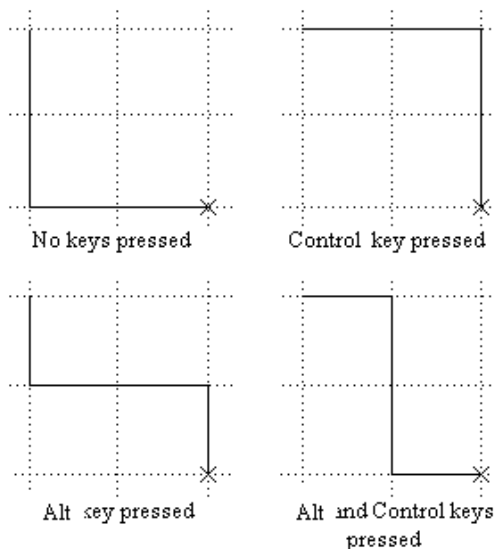
See more information on connection checking in “Signal Connection Checking on Paste” on page 70.

Drawing From an Existing Line or a Device Pin

A line can be extended from the end of an existing line or device pin using the normal () cursor. Click and hold on the end of the pin and drag away from the pin. A pair of right-angle lines will follow the cursor away from the pin and long as the mouse button is pressed. Releasing the mouse button makes these lines permanent. If the end of the line (i.e. the point where the

mouse button was released) touches another signal line, a connection will be made at that point.

Alternate line routing methods can be activated by pressing the **Control** and **Alt** keys, as follows:



The **Control** key constrains the movement to a single vertical or horizontal line. The **Alt** key inverts the order of line drawing, and the **§** key switches to three line segments with a center break.

NOTE: Holding the **Control** key while clicking will inhibit checking for pin connections. This allows you to select the signal again and drag it to a new position without affecting any existing connections.

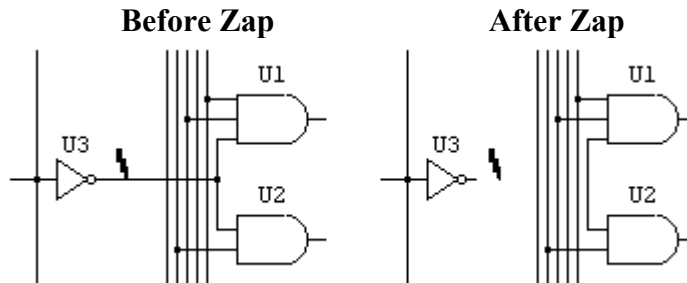
Creating an Unconnected Signal Line

The Draw Sig (**+**) cursor can be used to create an unattached signal line, or can be used to extend an existing signal. Simply click anywhere in the schematic and drag away in the desired direction. Unlike the Point mode drawing method, above, the mouse button does not have to be held while creating signals in this mode. Double-clicking terminates the signal line.

Editing a Signal Line

The following features are available to edit signal lines:

Zap mode (entered by selecting the Zap command in the Edit menu or the Zap item in the toolbar) allows you to remove any single line segment from a signal connection. Zapping on a signal line removes only the line segment being pointed at, up to the nearest intersection, device pin or segment join point.



Selecting a signal line (by clicking anywhere along its length) and hitting delete or selecting the Clear command removes an entire signal trace.

Drawing backwards along the length of an existing line causes the line to be shortened to end at the point where you let the button go.

Clicking and dragging the middle of a signal line segment with the pointer cursor allows you to reposition the line. Vertical lines can be moved horizontally and vice versa.

The pointer cursor can be used to start drawing from the ends or corners of an existing signal.

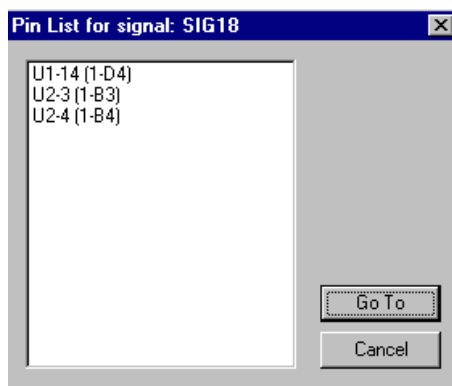
The Draw Sig (+) cursor can be used to start drawing from anywhere along an existing signal line. Double-clicking terminates drawing.

Checking Signal Interconnection

Double-clicking anywhere along a signal line will cause that signal segment and all logically connected segments on the current page to be selected.

You can also use the Pin List command in the signal pop-up menu (by right-clicking on the signal) to view the list of pins attached to the signal and go to any of them. This command displays a list of all device pin connections comprising the selected signal. Only pins in the current circuit are listed; pin con-

nections in other hierarchy levels are not shown.



Selecting any item in the pin list and clicking on the Go To button will cause the appropriate page to be opened (if necessary) and that pin to be displayed in the center of the circuit window. If the selected pin is invisible (e.g. a bus internal pin) the window will be centered on the parent bus pin.

For convenience in navigating a schematic, connections to pseudo-devices such as bus breakouts and ground symbols are also shown in this list. In this sense, the list is not a logical “netlist”, but rather a method for checking connectivity and navigating the diagram.

Each item in the pin list is formatted as follows:

For normal devices: *device-pin (page ref)*

For pseudo-devices: *device type (page ref)*

The elements of this format are as follows:

- | | |
|-----------------|---|
| device | If the device is named, this will be its name, otherwise it will be a “#” mark plus the type name. |
| pin | If the pin has a pin number, this will be the pin number, otherwise a “#” and the pin name. |
| page ref | This will be a reference to the page number and grid position, formatted as specified using the Design Preferences command. |

For more information on page references, see “Automatic Display of Page References” on page 201.

Setting Signal Color

To select an individual color for a single signal line, right-click on the line and select the Color command in the popup menu. This setting will override the default signal color, which is determined by settings in the initialization file, described in “Color Settings” on page 378. To return a signal to the default color, select the Color command again and click the Default button.

Selecting a Signal

A single signal is selected by clicking anywhere along the signal line. This selects only the part of the signal directly attached to the clicked line. Double-clicking the signal selects all parts of the signal on the given page including logical connections by name or bus.

Selecting a Pin

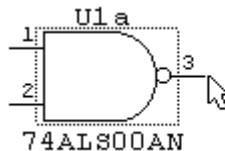
A pin is selected by clicking on the pin line close to the device.

NOTE: Since an unconnected device pin is both a pin and a signal, you determine whether you get the pin or signal pop-up menu as follows:

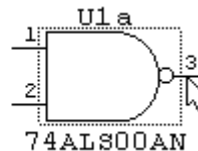
clicking on the pin in the last 1/4 of the pin length away from the device will display the signal menu.

clicking on the pin close to the device symbol will display the pin menu.

Selecting the Signal



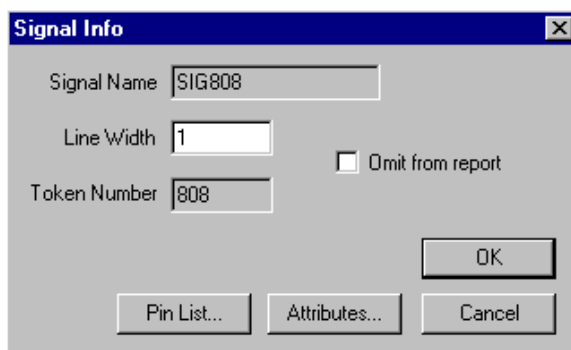
Selecting the Pin



Getting and Setting Signal Information

Selecting the Properties command with a signal selected causes the following

box to be displayed:



This box can also be displayed by holding the **Alt** key pressed while clicking and holding on a signal line. In the pop-up menu, select the Properties command.

Here are the options presented in this box:

- | | |
|-------------------------|---|
| Omit From Report | This switch controls whether the selected signal is included in any netlist output. See “Signal Reporting Options” on page 356 for more information. |
| Line Width | The number typed into this box determines the displayed width of the signal. Any number > 1 displays the signal that number of times wider than the normal value. The maximum value is 255. |
| Pin List | This button displays a list of the real device and pseudo-device pins attached to this signal. Double-clicking on any item in this list will display the selected pin. More information on this box is given in “Checking Signal Interconnection” on page 91. |
| Attributes | This button displays the general attribute data entry box for the selected signal. More information on the functions available in this box are given in “Entering and Editing Attribute Data - Basic Procedure” on page 160. |

Naming Signals

Names may contain any letters, numbers or special characters that you can type on the keyboard, but are restricted in length to at most 15 characters. The name associated with an object can be placed anywhere on the diagram and

will be removed if the object is removed.

What Signal Names are Used For

The signal name is referenced by the following DesignWorks functions:

The signal name is used in report output, such as netlists and error checking reports.

The signal can be located by name using the Find tool.

Signals can be logically interconnected by name.

Invisible Signal Names

DesignWorks allows you to assign and edit signal names in a circuit without making them visible on the diagram. This can be used to assign names that cannot be conveniently placed on the diagram, or that are needed for report generation purposes only. Signals with invisible names are not connected by name, except for invisible names created by a Signal Connector device, as described above.

Connecting Signals by Name

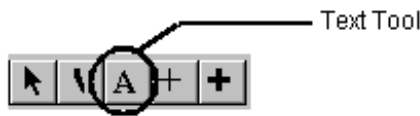
Signal names can be used to make logical connections between wires that are not visually connected on the schematic. Like-named signal traces on a single page are logically connected for simulation and netlisting purposes. Whenever a signal name is added or changed, the circuit is checked for a change in connectivity. If the name is now the same as another signal on this page, the two signals are merged into one. If this signal segment was previously connected by name to others and the name is changed, then the logical connection is broken. Whenever a name change causes two signals to be connected, both parts will flash on the screen to confirm the connection.

More information on connecting across pages, using power and ground connectors, using busses, and other signal connection issues are covered in detail in Chapter 9—Making Signal Connections on page 189. In particular, you may wish to refer to “Signal Connectivity Rules” on page 213.

Adding a Signal Name

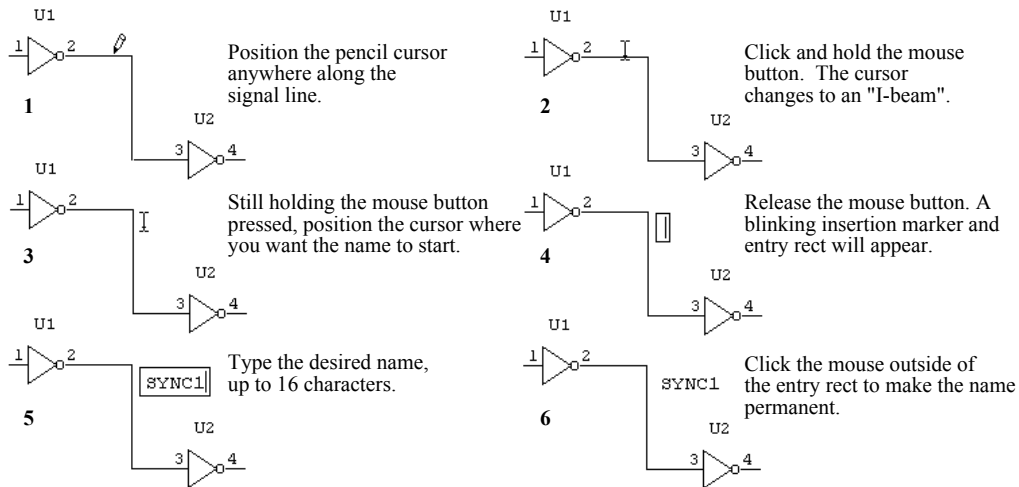
Enter “Text” mode either by selecting the Text command in the Edit menu, or

by clicking on the text tool in the toolbar:



Note that once this command is selected, the cursor changes to a pencil icon.

Press and hold the mouse button with the tip of the pencil positioned anywhere along a signal line except within 5 screen dots of a device. As long as you hold the mouse button down an I-beam cursor will track the mouse movements. The signal name text will start at the position where you release the button. Type the desired name and press Enter or click the mouse button anywhere. This is illustrated below.



Adding an Invisible Name

An invisible name for a signal can be created by either:

- 1) Clicking the right mouse button on the signal, then selecting the Name command from the pop-up menu.

OR

- 2) Selecting the desired signal, then selecting the Properties command in the Options menu, then clicking on the Attributes button in this box, then selecting the Name field in the attributes box.

In either case, if the name is already visible on the diagram, changing it here will change all displayed occurrences of it.

IMPORTANT: If two signals are given the same invisible name, they will not make a logical connection in the DesignWorks schematic, since they must be visible for the connection to be made. However, some PCB or simulation systems consider any two signals with the same name to be connected. In order to minimize the chance of creating an accidental short between two signals with the same name, the DesignWorks auto-naming feature will always assign a new, unique name to any signal that has an invisible name that is copied, duplicated or otherwise edited on the schematic.

Making an Invisible Name Visible

An invisible name can be made visible by simply clicking the Text cursor anywhere on the signal or device. When the mouse button is released, the name will be positioned at that point, as shown in the general naming instructions above. Alternatively, you can use the Name command in the signal pop-up menu to change the Visible status of the name.

Removing a Name

A signal name can be removed by using the Zap cursor. If the signal has been named in multiple locations then Zap removes the name only at the location zapped, but the signal still retains the logical name. Alternatively, you can hold the key while clicking on the name text on the schematic and use one of these commands in the attribute pop-up menu:

The Hide command can be used to hide the name without changing it.

The Clear command can be used to remove the name completely.

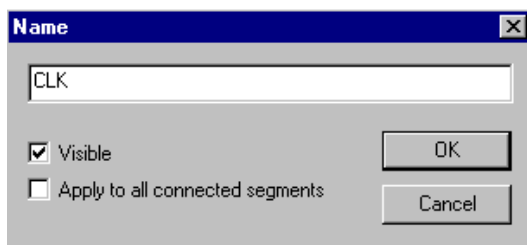
IMPORTANT: See “Adding an Invisible Name” on page 96 for an important note about invisible signal names and auto-naming.

Editing a Name

The name can be changed by simply clicking the Name cursor on the signal name and editing using the keyboard as desired. Alternatively, a name can be edited by using the Name command in the signal pop-up menu.

This command displays a name edit box for the selected signal. The following box is displayed offering you several options on how the name changed is to

be applied:



Apply to All Connected Signals—This option allows you to choose whether the name change applies only to the selected signal segment (i.e. thereby breaking its connection with other like-named signals), or to all interconnected signal segments.

Visible—This option allows you to choose whether the name entered should be displayed on the schematic. If the name was already visible and you uncheck this box, it will be removed from the schematic. In this case, the name will still be associated with the signal as an invisible attribute. If the name was not previously visible and you check this box, it will be displayed somewhere adjacent to one of the signal line segments.

Moving a Signal Name

A signal name can be moved by selecting the Point (normal) cursor, clicking and holding the mouse button on the name, and dragging it to the desired new position. Holding the Alt key depressed while moving a name (or any attribute value) disables the grid snap, allowing you to micro-position the name for alignment with other graphic items on the schematic.

NOTE: The “Name” is actually an attribute field, so all attribute editing techniques can be used on it. In particular, you can hold the Alt key while clicking on the name and the attribute pop-up menu will appear, giving you a variety of editing commands, described in “Other Ways of Viewing and Editing Attributes” on page 168.

Multiple Naming of Signals

A signal name can appear in up to 100 positions along the length of a signal line. To add a new position, simply use the normal naming procedure given in the section on signal naming i.e.:

Select Name mode

Click and drag anywhere along the signal line

With the mouse button pressed, move to the desired position for the name

Release the mouse button

A new copy of the signal's name will appear at this point followed by a flashing cursor. To accept the name, simply click the mouse button once or press the Enter key. If you edit any occurrence of a name along a signal segment, all other occurrences will be updated to reflect the new name.

Any occurrence of a signal name can be removed using the Zap cursor. If you remove the last visible name from a signal segment then the logical connectivity to other like-named signals is removed.

Using the Auto-Naming Features

Two features are available to simplify the naming of groups of related signals, devices and pins. These features are activated by holding down the **Alt** or **Ctrl** keys while selecting the signal to be named with the Text cursor.

Auto-alignment - If the **Alt** key is held down while the signal is selected, the text insertion point will be positioned horizontally aligned with the last signal name that was entered. The vertical position is determined by the vertical position of the line that was clicked on. This feature works only with signal names, not with devices or pin numbers.

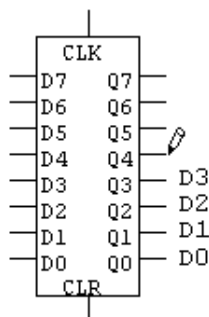
Auto name generation - If the **Ctrl** key is held down while a signal, device or pin is selected, a new name is generated automatically for this item. The new name will be the same as the last one entered, except that the numeric part of the name will have been incremented. If the previously-entered name did not have a numeric part, then a "1" digit will be appended to it. If the **Alt** key is pressed at the same time, the number will be decremented instead of incremented.

Sequential Naming

The above two features can be used in combination to perform easy naming of sequential signals. The normal symbol standard in DesignWorks is to position the highest numbers at the top, so you can either:

number the bottom-most line in the group (e.g. D0) using the normal

naming technique, described above, then hold **Alt** and **Tab** while clicking on successive higher-numbered lines.



number the topmost line in the group (e.g. D7) using the normal naming technique, described above, then hold **Alt**, **Tab**, and **Tab** while clicking on successive lower-numbered lines.

Note that when you select each successive line, the new name appears, then it is necessary to click again or press Enter to make the name permanent.

Pin Numbering and Information Entry

Pins are circuit objects that provide a link between devices and signals. A pin cannot exist independently of a device, but it can have a variety of information associated with it that can be set separately for each pin. This information includes the visible pin number on the schematic and any number of attribute fields.

Getting and Setting Pin Information

You can display a general pin information box by either selecting the pin and choosing the Properties command in the Options menu, or, right-clicking on the pin, then selecting the Properties command in the pin pop-up menu.

See “Selecting a Pin” on page 93 for the technique used to select a pin.

In either case, this box will be displayed:

The following information and options are available:

- | | |
|-----------------------------------|---|
| Pin Number | This is the physical pin number corresponding to this device pin. This can be empty if desired. See “Uses of Pin Numbers” on page 102 for more information. |
| Visible | This check box determines whether the pin number is displayed on the schematic. For some devices, such as discrete components, it may be desirable to have a pin number associated with the pin for netlist purposes without displaying it on the diagram. |
| Pin Type | This information item gives the function and visible status of the pin. |
| Pin Ordinal Number | This number is the pins ordinal position in the device's pin list, i.e. as viewed in the device symbol editor. This can be important in some netlist formats where pin order is critical. See “Reordering Pins in the Pin List” on page 299 for more information. |
| Associated Internal Signal | For subcircuit devices, this item shows the name of the signal attached to the associated port connector in the internal circuit. For other devices, this will be “None”. See “Making Connections Across Hierarchy Levels” on page 236 for more information. |
| Attributes | This button displays the general attribute data entry box for the selected pin. See “Entering and Editing Attribute Data - Basic Procedure” on page 160. |
| Bus Pin Options | This option will be enabled if the selected pin is a bus pin. This will display the Bus Pin Options box which provides a number of information and editing operations for working with bus pins. See “Using Bus Pins” on page 196 for more information. |

Prev Pin/ Next Pin These buttons allow you to move to the next or previous pin (by ordinal number) on the same device, without having to return to the schematic and select the pin.

Pin Numbering

Pin numbers may contain 0 to 8 characters (not necessarily just numeric) and are always positioned adjacent to the associated pin.

Uses of Pin Numbers

Pin numbers are used only for labeling purposes and have no particular connectivity significance to DesignWorks. Pin numbers are not automatically checked for duplicates or other invalid usage although you can use the Find tool to run such error checks manually. Pin numbers placed on a diagram can be used in creating a netlist, and will appear when the circuit is printed. If a pin is unnumbered, it will appear in a netlist with a “?” unless the \$AUTONUMBER option is used in the report script.

For information on extracting pin numbers in netlists, see the ReadMe file provided with the design kit you are using and refer to the \$AUTONUMBER and \$PINNUM script commands in the DesignWorks Script Language Reference (separate manual on disk).

Default Pin Numbers

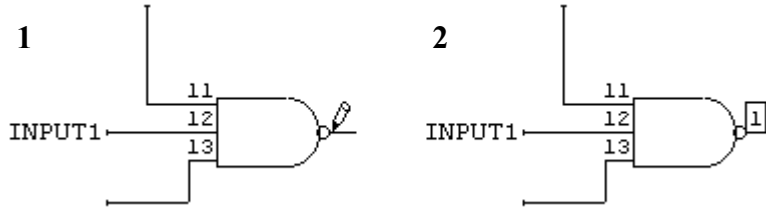
A device type may have default pin numbers associated with it which will appear when the device is first created. These pin numbers are not permanent and can be edited or removed by the techniques discussed in this section. Devices in a library can have default pin numbers assigned to them using the device symbol editor tool.

See “Setting the Pin Number” on page 298 for information on setting default pin numbers while editing a symbol.

Editing Pin Numbers On the Schematic

If the mouse button is pressed with the tip of the pencil positioned on a signal line within 5 screen dots of a device, a blinking insertion bar will appear immediately where the signal joins the device. You cannot set the text position for pin numbers. Type the desired 1 to 8 character number and press Enter, or click the mouse button anywhere, to make the number permanent. See the sequential numbering option discussed below under “Auto-naming

Features”. Pin numbers cannot be repositioned.



Quickly Applying Sequential Pin Numbers

If the **Increment** key is held down while a pin is clicked on with the pencil cursor, a new pin number is generated automatically for that pin. The new number will be the same as the last one entered, except that the numeric part of the string will have been incremented. If the previously-entered item did not have a numeric part, then a “1” digit will be appended to it. If the **Decrement** key is pressed at the same time, the number will be decremented instead of incremented.

Editing Pin Numbers Using the Properties Command

To edit pin numbers using the Properties dialog box:

- ➔ Click on the device in question to select it.
- ➔ Select the Properties command in the Options menu.
- ➔ Click on the Pin Properties button. This will display the pin information for the first pin.
- ➔ Edit the pin number as desired.
- ➔ Click the Next Pin button to see the next pin in the list.

Editing Pin Numbers Using the Browser

To edit pin numbers using the Browser tool:

- ➔ Select the Browser command in the View menu.
- ➔ Select the Pins object type.
- ➔ Select the Number item in the Pseudo pop-up menu.

Pin numbers can now be edited in the “Number” column of the spreadsheet.

Setting Pin Number Text Characteristics

The text font used for displaying pin numbers is setting by selecting the Design Preferences command in the Drawing menu, then selecting the Text tab, then clicking the Pin Text Style... button. This displays the text specification box, allowing you to select a font family, face and point size. Any changes made to this text style will be applied to all pin numbers displayed throughout the design. There is no ability to select different fonts for individual devices or pins. The text changes are applied when the OK button in the Design Preferences box is clicked.

Allowing Rotated Pin Numbers


You can optionally show pin numbers on all north- and south-facing pins rotated 90° to run along the length of the pin. This is set by selecting the Design Preferences command in the Drawing menu, selecting the Text tab, then checking the Rotate Pin Numbers item. If this item is unchecked, all pin numbers will be displayed horizontally adjacent to the pin.

Using Text and Graphic Objects

Text and graphic objects are used only to enhance the graphical appearance of a schematic diagram. They have no logical significance in the design.

Text objects are not associated with any particular device or signal on the screen and should not be used to set a name or attributes for devices or signals. The text in these boxes is not accessible in net or component lists. Use the naming and attribute features to attach text to devices and signals.

Creating a Text Block

If you click the  cursor on the diagram a blinking cursor will appear at that point and you will be able to type any desired text on the diagram. The Return key can be used to enter multiple lines in a single text block. Text entry is terminated by the Enter key.

Editing a Text Block

If you double-click on or click the text cursor inside an existing text item the

insertion point will be positioned at the click point. You can then use normal Windows text editing techniques to modify the text. Note that text on the clipboard can be pasted into an existing text box using the $\text{Ctrl}+\text{V}$ key equivalent for the Paste function. The Paste menu command will cause the current text entry to be terminated and a new text box to be created. Similarly, the $\text{Ctrl}+\text{X}$ key equivalents for Cut (X) and Copy (C) can also be used while editing a text box.

Text boxes can be Zapped, Duplicated, Cut, Copied, Pasted and dragged just like any other item on the screen. See the descriptions of these commands for more information.

Background and Border Objects

Either text or picture blocks can be set to be “border” or “background” objects. These two options will normally be used together, but can be enabled separately if desired. These two options are enabled by selecting the object in question and using the Properties command.

Border Objects

If a text or picture is marked as a “border” item, then it is considered to be part of the sheet border information for the page it is on. This has the following effects:

- if a new page is added to the circuit using the Pages command, this item will automatically be copied to the same position on the new page.

- if the page containing this item is used as the source for an Import Sheet Info operation, this item will be copied to the destination.

- if a Center in Page command is used, this object will not be repositioned since it is assumed to be part of the border.

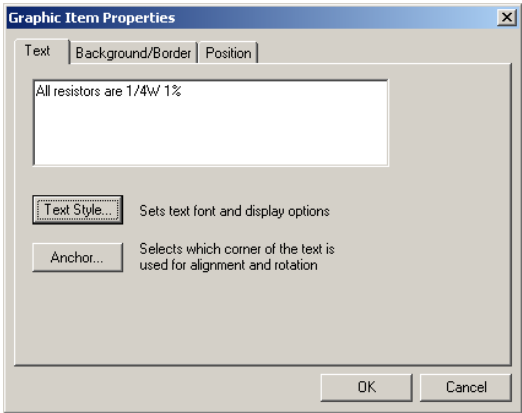
Background Objects

Text or picture items marked as “background” objects cannot be selected, modified or deleted using normal circuit editing techniques. This is used to prevent custom sheet backgrounds or border graphics from interfering with circuit editing operations. This is normally used in combination with the “border” status discussed above.

To select a background object, hold the Ctrl and Alt keys pressed while clicking on it, or use the Background Layer Only command in the View menu. The Properties command can then be used to disable the background status.

Text Style and Display Options

To set text display options and text style, select the text block by clicking on it with the Point cursor, then select the Properties command in the Options menu. This will display the following box:



This table summarizes the options available.

Text Style

This button displays the text style options box which sets the text font, colour and frame options.

Anchor

This button displays a box allowing you to set the text anchor point. This is the point that remains fixed when the text box is resized due to content changes. This is relevant if the text is to be aligned with some other objects on the drawing.

Background/Border

This tab contains options for making the selected text block a background or border. Background items cannot be affected by normal editing operations. In order to select it for editing, hold the **Ctrl** and **Alt** keys pressed while clicking on it or select the Background Layer Only command in the View menu.

Border objects will be treated as part of the sheet border. That is, it will be updated by the Import Sheet Info command, copied to new pages as they are created, and unaffected by the Center in Page command. See "Setting Sheet Sizes and Borders" on page 333 for more information.

Position

These options allow you to fix the position of the item relative to one edge of the border. This will cause the item to be repositioned automatically if the border size is modified.

See more information on using text notations in title blocks and sheet borders in “Setting Sheet Sizes and Borders” on page 333.

Using Text Variables

The text facility allows special entries called variables to be placed in text boxes. When the text is displayed or printed, the appropriate system quantity or attribute field value is substituted for the variable name. Any variable name that is not recognized will be displayed verbatim. This facility is intended to assist in creating title blocks and similar additions to the schematic with information such as the date, page number and title, revision level, engineer's name, etc. Using variables will allow this information to be entered in one place so that changes are reflected automatically on all schematic pages.

System Variables

System variables start with a “\$” and are enumerated here:

\$DATENOW	This will be replaced by the system date written mm/dd/yy. This is the current date as maintained by the system clock.
\$TIMENOW	This will be replaced by the system time written hh:mm. This is the current time as maintained by the system clock.
\$DATECREATED	This will be replaced by the date the file was created, as maintained by the file system.
\$TIMECREATED	This will be replaced by the time the file was created, as maintained by the file system.
\$DATEMODIFIED	This will be replaced by the date the file was last modified, as maintained by the file system.
\$TIMEMODIFIED	This will be replaced by the time the file was last modified, as maintained by the file system.
\$PAGENUM	This will be replaced by the number of the page (within the current circuit level) the text is on.
\$NUMPAGES	This will be replaced by the total number of pages in the circuit level.
\$PAGETITLE	This will be replaced by the title of the page the text is on, as entered using the Pages command.

\$PRINTPAGENUM

This will be replaced by the page number in the printed sequence within the current Print command. This is used to number pages within a hierarchical design. NOTE: This is only valid for printed output. When it is drawn on the screen, it will show the same value as \$PAGENUM.

\$PRINTNUMPAGES

This will be replaced by the number of pages that will be printed by the current Print command. This is used to number pages within a hierarchical design. NOTE: This is only valid for printed output. When it is drawn on the screen, it will show the same value as \$NUMPAGES.

\$FILENAME

This will be replaced by the name of the design file, not including its directory path.

\$FILEPATH

This will be replaced by the name of the design file, including its directory path. If the file hasn't been saved, this will be empty.

\$CIRCUITNAME

This will be replaced by the name of the circuit being printed. In the topmost circuit of a hierarchical design (or in any flat design), this is the same as \$FILENAME. In a subcircuit, this will be the hierarchical name of the circuit.

NOTE: The format used to display date and time values is determined by the Regional Settings control panel in Windows.

Attribute Variables

Attribute variables start with a “&” mark and are used to refer to fields stored in the attributes for the design. These can be used to place information at multiple points on a diagram which will be updated automatically when the design attributes are changed. For example, if the design attribute field Revision was defined for the design with the following contents:

2.1A Mar 18, 2008

then the variable &Revision would appear as “2.1A Mar 18, 2008” on the diagram.

See Chapter 8—Attributes on page 157.

Editing Text with Variables

Text items on the schematic will normally be displayed with variables replaced by their values. When a text item is selected or clicked on with the

Name cursor it is redisplayed in its raw format with the variable names shown without interpretation. This allows the items to be edited with the text in its actual stored position.

Text Frame Size with Variables

The framing rectangle for a text item is calculated after the variable substitution is done. This may cause the item to be highlighted or deleted incorrectly if the variable values are shorter than the names. These errors are not serious or permanent and will disappear when the screen is updated. The text box can be expanded as necessary by adding blanks at the end of any one line in the item.

Selecting a Graphic or Text Object

A single, non-background graphical or text item is selected by clicking the mouse button with the pointer positioned anywhere inside the item.

Selecting a Background Object

To select a graphical or text item marked as “background” while editing a schematic, you must hold the **Alt** and **Ctrl** keys pressed while clicking on it. Alternatively, you can select the Background Layer Only command in the View menu, which hides all normal schematic objects and enables all background items for editing.

Drawing Graphics

Graphic items such as lines, boxes, ovals, etc. can be drawn directly on the schematic diagram using the drawing tools.



NOTE: Items drawn with these tools have no circuit significance and will not be included in any netlists or reports.

Setting Graphic Object Properties

All graphic items have various visible properties that can be set, such as line width, colour, text font, etc. To set the properties of any object, either:

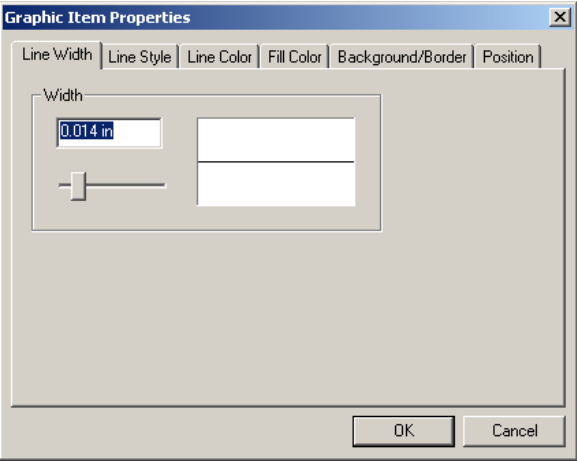
Right-click on the object and select the Properties command in the pop-up

menu, or,

Select the object and select the Properties command in the Options menu.

You can select multiple objects at the same time for this operation. Any changes will be applied to all selected objects.

The exact contents of the box that is displayed will depend on the type of object(s) selected. A typical box such as the following will be presented for a rectangle:



This table summarizes the features available in this box:

Line Width	Allows you to specify the width of the line portion of the object.
Line Style	Allows you to specify dotted or dashed lines. Note that there are limitations in line width when these options are used.
Line Color	Specify the color of the line portion of all selected objects.
Fill Color	Specify the fill color of any fillable selected objects.

Background/Border

This tab contains options for making the selected text block a background or border. Background items cannot be affected by normal editing operations. In order to select it for editing, hold the **Alt** and **Ctrl** keys pressed while clicking on it or select the Background Layer Only command in the View menu.

Border objects will be treated as part of the sheet border. That is, it will be updated by the Import Sheet Info command, copied to new pages as they are created, and unaffected by the Center in Page command. See “Setting Sheet Sizes and Borders” on page 333 for more information.

Position

These options allow you to specify whether the graphic object should “stick” to one side of the border, i.e. should it be moved automatically to retain its position relative to the border if the sheet is resized. See “Setting Sheet Sizes and Borders” on page 333 for more information.

Grouping Graphic Objects

Any collection of graphic objects, text and pictures can be made into a “group”, which allows the items to be moved and edited as a single item.

Creating a Group

Select any collection of items by dragging the mouse across them and/or by holding the SHIFT key and clicking on the individual items. Now right-click on any item and select the Group command. Clicking on any item in the group will now select the entire collection as if it was one item.

Setting Properties on Grouped Items

To set properties on a group, right-click on any item in the group and select the Properties command. Any changes made will be applied to all items in the group.

Ungrouping Grouped Items

To revert a group back to a collection of separate items, right-click on any item in the group and select the Ungroup command.

Rotating Graphical Objects

To rotate any graphic objects, right-click on the object (or on any one of a collection of objects) and select one of the four rotation commands: Flip Vertical, Flip Horizontal, Rotate Left or Rotate Right.

Setting Front-to-Back Ordering

Front-to-back ordering of graphic objects determines which items are drawn first and therefore appear "behind" other objects. To make an object appear behind all others, select the Send to Back command. To make an object appear in front of all others, select the Bring to Front command.

Aligning Graphical Objects

To align multiple graphical objects, simply select all the objects in question, right-click on any one of them and select one of the command in the Align submenu.

See "Setting Sheet Sizes and Borders" on page 333 for more information on creating title blocks and sheet borders.

This chapter provides information that should be considered before starting any major design. We will take a look at the overall design process, focussing on the schematic diagram. Correct choices made in advance concerning usage of hierarchy mode, attribute format, sheet layout etc. will save time and frustration later.

What uses will the design data be put to?

Are you going to pass data to a PCB design, synthesis, simulation or other design package?

Does your organization have a parts database you need to work from?

How are hard copies to be printed or plotted?

This chapter is not intended as a specific reference for any DesignWorks features. It focuses on your desired end-product and how best to make use of this package to achieve that goal. Its primary aim is to raise issues that you should consider and alternatives that you may wish to test before proceeding.

This chapter makes general reference to third-party products and specific issues on interfacing with them. Refer to Read Me files provided with the design kits included with DesignWorks for more information on specific systems.

The Golden Rule - Try a small design first!!!

Regardless of how simple your design process may seem, a small test of the entire flow will pay large dividends in reduced future headaches. If your design process involves moving netlist or other data files between software systems, file compatibility is an important issue. Refer to the design checklist given in the next section and create a small design that will exercise the areas of each system that may cause problems.

Design Process Checklist

Every design process has its own unique elements:

- Company standards for printed documents,
- Software and hardware compatibility,
- Device symbol requirements

and so on. Even if all you want to do is draw a small schematic and print it on your local laser printer, there are issues of fonts, sheet sizes, logos, etc. that will affect your final result.

In the following sections we present a checklist of issues that may affect your design process. Many of the items in the list will probably not apply to your situation. Make your own list of issues that need a second look and produce a small test design to check them out before proceeding with a large design.

The points given here are intended primarily to prompt you to look at issues in your design process, rather than to provide any specific information on DesignWorks features. In many cases, references are given to other areas in this manual where further information can be found.

Schematic Creation

In this section, we'll list the issues that affect the creation and printing of the schematic diagram itself.

Attribute Usage

The DesignWorks *attribute* facilities provide a powerful mechanism for storing arbitrary text information with a schematic. This information can be shown on the printed sheet and/or passed to external systems in netlists and reports. Defining a clear list of the attribute data required in your design will help reduce the effort required to get the right text displayed on the printed sheets and the right data passed out to your PCB layout or simulation package.

Complete details on the attribute facilities in DesignWorks are given in Chapter 8—Attributes on page 157.

Here are some ideas to consider regarding using attributes in your designs:

Define fields for design before creating libraries—The set of attribute fields used in symbol definitions in libraries is independent of any particular design. While this creates a great deal of flexibility, it can also lead to problems if you have variations in attribute field naming and value format. It pays to give some thought in advance to how you will name fields (e.g. upper case first letter, underscores between words, etc.) and how you will format part names, component values, etc.

Applying attributes on the schematic—DesignWorks has a number of options for fonts, text rotation and positioning of attribute fields. You may wish to create and print a small design first to verify that things will appear the way you want in the final hardcopy.

Back annotating attributes—Some applications require bringing text information back into the schematic from other systems. The most common application is PCB package names, but other simulation and layout data may be candidates as well. You should determine if this data is available in some machine-readable format and if it can be automatically read back in by DesignWorks.

Design attributes for Designer, Revision, etc., PCB Info, Simulation Info—You may wish to create a number of design attribute fields that can be displayed on each sheet using the text variable facility described in “Using Text Variables” on page 107.

Device attributes—part names, package codes, in-house part number, etc., simulation info, PCB info, gate packaging—Device attributes have a wide range of applications in interfacing to external systems. You should confirm what part information is needed by the target system before proceeding with creating a design or its symbol libraries.

Pin attributes: gate packaging, simulation info—Some layout and simulation systems require specific names or other information on device pins.

Signal attributes—Signal attributes are less commonly used, but may be needed to store PCB track width, cabling information or other data. Especially in cabling applications that require a sequence of cable and connector information to be stored with a net, you should confirm that the data can be stored and output in the format you need.

Schematic Portability

In some applications, the portability of schematic data between systems may

be an issue. DesignWorks runs on both MacOS and Microsoft Windows-based machines and there are some issues of text font usage and background graphics that may affect the movement of data between machines.

In addition, some companies or projects may require that schematic data be stored in some standard form for archival purposes when a project is finished.

Power and Ground Nets

There are a variety of methods that are commonly used to show power and ground connections on a schematic and to represent them in netlists and reports. For each design, you should decide what power and ground nets are needed and whether you wish to show them explicitly as lines on the schematic or use attributes or signal connector symbols. More information on this topic is provided in “Power and Ground Connections” on page 205.

Symbol Libraries

Device symbols are an important resource in your design creation process. Whether you primarily use the symbols provided with DesignWorks, or you create special libraries for your own use, the completeness and accuracy of this data has a major effect on your design flow. Library files generally outlast any one design and are used for many years across many projects, so they become an important asset for your company.

Here are some symbol-related issues that you may want to consider:

Symbol standards—Does your company or customer require any specific symbol standards?

Attributes—Do you have a clearly defined set of attribute fields that you want to include in each symbol?.

Pin ordering for reports—Some netlist formats, notably SPICE, require that pins appear in a very specific order. This is determined by the order in which you place the pins when creating a symbol.

Library organization—Library organization becomes an issue especially if you are working with a team of designers. Are project libraries shared in a central location? Who is in charge of maintenance? Does each designer have their own custom libraries?

Hierarchical Design

Hierarchy allows you to present a system design at different levels of abstrac-

tion, from a simple block diagram at the top to specific physical details at the bottom. While it may seem that this would always be beneficial, in practice, hierarchy does not lend itself to all applications. DesignWorks offers three hierarchy modes to suit a variety of applications. Here are some points to consider before settling on which to use.

Hierarchical design is a powerful technique, but does add some complexity to the design process. If you are one of the following:

- a new user of schematic capture tools
- working with small-to-medium designs (say, less than 200 ICs)
- interfacing to a third-party Printed Circuit Board layout package

...then we recommend sticking with “Flat” mode and not using hierarchical blocks in your designs. Once you have gained some experience with creating schematics in DesignWorks and moving design data between systems, you can review the hierarchy issues discussed in this manual “Choosing a Hierarchy Mode” on page 218.

If you are working as part of a team where another individual will be responsible for the manufacturing of the final product, and another person again for the long-term maintenance of the design, then hierarchy may not be appropriate. Hierarchy is a good concept for thinking about and presenting system design concepts, but is not good for tracing physical connections from part to part. Many companies do not allow the use of hierarchical design for PCB-based projects.

Once you have decided to use hierarchy, you need to decide which of the DesignWorks hierarchy modes to use, whether to use. Pure hierarchy mode is intended for large designs that:

- lend themselves to a very structured hierarchical definition.
- do not require device instance data.
- do not require flat netlist output.

Unless you are sure your design meets these requirements, do not use Pure mode. Pure mode is not recommended for printed circuit board-level designs.

Physical hierarchy mode is intended for designs that require data to be associated with each physical device instance. This will be the case for most PCB designs and any design using the DesignWorks Simulator option.

See Chapter 10—Hierarchical Design on page 215 for more information.

Printing and Plotting

The size and format of the schematic sheets you use will likely be determined by your organization's drawing standards and by what form of hardcopy device you have available. It is best to do some test sheets in your chosen format before proceeding with a large design. Reformatting a design to a different sheet size is an unpleasant task.

Sheet Layout

You will need to choose a sheet size that suits your project and the available hardcopy output devices. DesignWorks allows each page of a circuit to be different, if necessary, so you may wish to have a different layout for a title page, an “unused gates” page, or other parts of the design. If you want a company logo in your title block, it is best to test this on your output device and make sure it prints or plots correctly.

Border Size and Scaling

Most laser or inkjet printers have a number of options for print area size and scaling. For this reason, it is best to either:

Use the “Use Page Setup” option in the Sheet Wizard command. This way, the schematic drawing area will automatically adjust to changes in printer page setup.

OR

Choose a format and scale factor that looks good on your printer and create a fixed sheet template for it. Use the “Fit to single sheet” option in the Sheet Wizard box to force the printer to scale the output to the available printing area.

For more information on sheet settings and scaling, see “Fitting the Diagram to the Available Paper” on page 60.

Plotting

Following are a number of issues to consider before creating a design that will output to a plotter.

If you create any graphics such as a company logo using an external

drawing package, be sure not to use any “paint” or “bitmap” features. Any bitmapped graphic objects will plot poorly, if at all, on a pen plotter.

Most pen plotters come with printer driver software that makes the plotter act like a printer to the DesignWorks software. However, pen plotters have different physical limitations and may not produce the same results as a laser printer. It is best to follow the golden rule and plot some small test designs to ensure that your schematics will be reproduced as desired.

Reports

DesignWorks has a powerful report generation and scripting capability that can be used to produce text reports in a variety of formats.

Bills of Materials—Does your company or purchasing department have a standard for machine-readable or hardcopy reports?

Error Reports—DesignWorks includes error checking scripts for some common types of errors. Are there other errors that could be caught before sending your design out to the PCB house?

Exporting to Spreadsheets and Databases—DesignWorks has the ability to create tab-delimited text data formats that can be read directly most spreadsheet or database applications. This may allow you to simplify the entry of data for project costing and documentation.

See “Report and Netlist Generation” on page 353 for more information.

Interfaces to Other Systems

No design process is complete without considering the final destination of your schematic data. In many cases, the simulation or layout package that you are interfacing to will impose rules on device and symbol naming and attribute formats.

Netlist format—The single most important issue in exporting data to a third-party system is the format of netlist required. DesignWorks includes a large collection of netlist generating scripts and also allows you to create your own. Is the one you need supported? Can you get the format information needed to create your own? What information does the format require for each device or signal in a design? As with other issues presented here, it is best to test a small design first and make sure these issues are resolved before investing a large effort in a design

project.

Attributes required—Most PCB or simulation packages will require some extra information to be entered into your libraries or schematics, such as package codes, simulation parameters, etc. In some cases it may be possible to use a translation table in a netlist script to generate package codes for a target system from the ones provided in the DesignWorks libraries. Alternatively, you can consider back-annotating the desired values into the design from an external database. In any case, you should confirm that your chosen method is going to work before proceeding.

Naming restrictions—Most external packages have some restrictions on name length or special character usage in names. This is especially an issue if you are using hierarchical design to create the schematic, since this can result in long names being automatically generated. In addition, you should be clear on the case sensitivity of names in the target system. Are signals “CLOCK” and “Clock” going to be treated as the same thing?

Back annotation—If your design process requires back annotating information from a PCB layout system into the schematic, make sure the format you have available is supported in DesignWorks.

Documentation

Most projects have some requirement for written documentation in addition to the schematics. If you plan to include any graphics extracted from schematics, you should verify that you can cut and paste these successfully from the schematic into your word processor.

DesignWorks can also generate report outputs in a variety of formats that may be incorporated into written documentation.

This chapter provides information on the automatic gate packaging and naming features for devices in DesignWorks. Device name assignment is an important issue because the device name frequently has special significance when data is exported to another design system. For example, if you are using a SPICE-based simulator, the first letter of the name determines its model type. If you are working with a PCB-layout package, the name is used to hold the package assignment.

The auto-packaging feature is intended for PCB-related applications, where multiple logical symbols on a schematic may be grouped together in a single physical device package on a board. DesignWorks handles this by allowing you to store information in attributes that indicate the number and type of logical devices in a package.

Auto-naming is a simpler feature that ensures that every symbol that is placed in a circuit has a unique name assigned. Auto-naming does not take packaging information into account.

This chapter also discusses device date stamping. Date stamping is a mechanism that automatically marks devices with a time value when they are created. This can be used in conjunction with some PCB systems for forward- and back-annotation purposes.

Packaging vs. Auto-Name Assignment

The Packaging and Auto-Name Assignment features can both be used to assign names to devices. Both of these features are controlled using the Device Naming and Packaging Options menu command in the Naming and Packaging sub-menu of the Options menu. These are the key distinguishing features of these two options:

Packaging assigns a name, unit and pin numbers to each device based on physical packaging considerations using information stored in attributes with the symbol. Note that packaging does not take any layout considerations into account, it simply assigns the next available package to each symbol as it is placed.

Auto-naming assigns a name to each device symbol in a circuit without looking at any attribute information except the default name prefix. This ensures that each device symbol has a unique name.

Packaging operates across an entire design whereas auto-naming applies separately to each circuit level in a hierarchical design. That is, in a hierarchical design, there can be two or more devices with the same name, e.g. “R23”. See “Using Hierarchical Names” on page 228 for more information on using hierarchical names to distinguish objects in a hierarchical design.

In flat designs, the only difference between auto-packaging and auto-naming is in the use of the packaging information during name assignment. If the device libraries you are using have no packaging information, these two mechanisms will have exactly the same effect.

Choosing Options for your Design

Whether you choose to enable the auto-packaging or auto-naming options depends upon your particular application, the hierarchy mode being used and on your personal work style. For more information on choosing a hierarchy mode, see “Choosing a Hierarchy Mode” on page 218. Once that issue is settled, here are some issues that will affect the naming and packaging options that you use:

It almost always makes sense to have either or both auto-naming and auto-packaging turned on. Even if you will be changing the names later based on some other layout considerations, using one or the other of these features ensures that every symbol can be easily distinguished on the schematic or in a netlist or report.

If your design is destined for a PCB layout, then it is best to have auto-packaging enabled, for the same reasons given above. In addition, auto-

packaging will give you a running idea of how many packages have been used, even if the actual package assignments will be changed later.

If your design is not intended for PCB layout, e.g. a SPICE simulation or an FPGA design, then it is best to use only auto-naming. With auto-packaging, there is a possibility of assigning the same name to multiple devices, which may confuse the target layout system.

In designs using flat hierarchy mode (the default), you can only have one or the other of these options enabled. That is, if auto-packaging is enabled, auto-name assignment will have no effect. This is because both these mechanisms assign values to the Name attribute field in flat designs.

In physical hierarchy mode, both features can be used together, since the name is placed in the Name field, while the package assignment is placed in a separate attribute field, InstName. See “Using the Name and InstName Fields” on page 170 for more information on the use of the InstName field.

In pure hierarchy mode, packaging is not available since it is not possible to store the physical assignments of devices inside subcircuits that have been used multiple times. Auto-naming is available and is on by default.

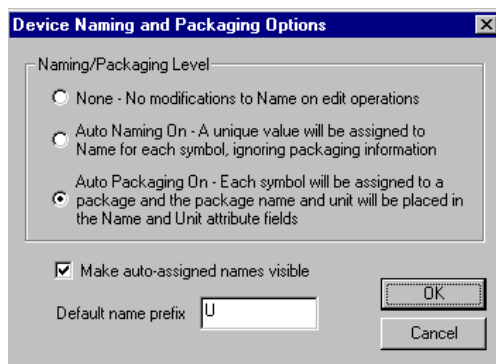
Enabling Naming and Packaging Options

The naming and packaging options that are available to you depend upon which hierarchy mode you are using.

Naming and Packaging Options in Flat Hierarchy Mode

If the current design is in flat hierarchy mode, selecting the Device Naming

and Packaging Options menu command will display this box:



The following options are available:

None Both options are off. The Name attribute field will not be modified when a device symbol is placed.

Auto Naming On A unique (within the circuit) name will be generated in the Name field when a symbol is placed.

Make auto-assigned names visible If this box is checked, the auto-assigned name will be displayed on the schematic next to the symbol. If this box is not checked, the name will be stored with the device, but not displayed on the schematic.

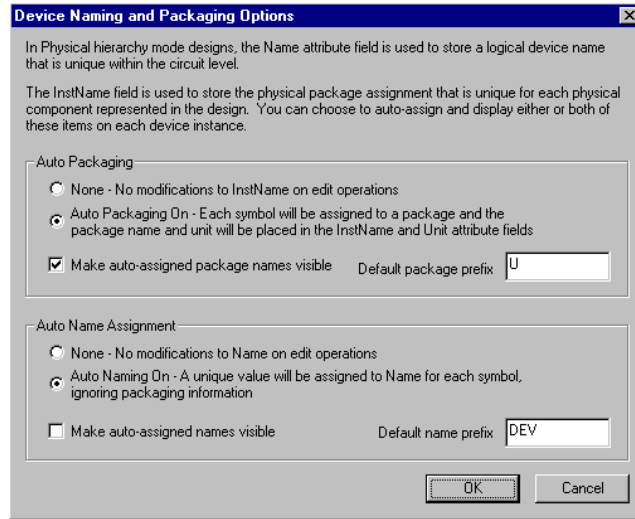
Auto-Packaging On A package assignment will be made for each symbol placed on the schematic and the resulting information stored in the Name and Unit attribute fields. The Unit assignment will cause the pin numbers on the symbol to be updated, if that information is provided with the symbol.

Default name/package prefix The text in this box will be the default prefix used to generate a device name, if no prefix is provided in the symbol attributes. See “Setting the Auto-Generated Name Format” on page 131 for more information. (Note that the title on this box changes depending on which option is enabled.)

Naming and Packaging Options in Physical Hierarchy Mode

If the current design is in physical hierarchy mode, selecting the Device Nam-

ing and Packaging Options menu command will display this box:



In physical hierarchy mode, the auto-packaging and auto-naming options can be enabled separately. The following naming options are available:

None

Auto-naming is off. The Name attribute field will not be modified when a device symbol is placed.

Auto Naming On

A unique (within the circuit) name will be generated in the Name field when a symbol is placed.

Make auto-assigned names visible

If this box is checked, the auto-assigned name will be displayed on the schematic next to the symbol. If this box is not checked, the name will be stored with the device, but not displayed on the schematic. If auto-packaging is also enabled, you will probably want this to be off so that you don't get a confusing display of two different names.

Default name prefix

The text in this box will be the default prefix used to generate a device name, if no prefix is provided in the symbol attributes. See "Setting the Auto-Generated Name Format" on page 131 for more information.

The following packaging options are available:

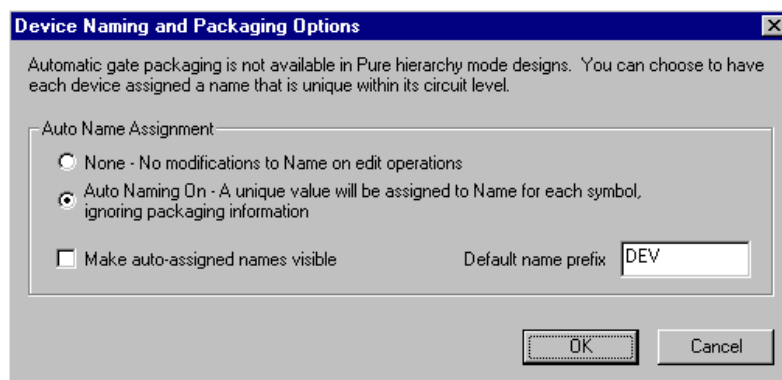
None

Auto-naming is off. The InstName attribute field will not be modified when a device symbol is placed.

Auto Packaging On	A package assignment will be generated in the InstName field when a symbol is placed.
Default package prefix	The text in this box will be the default prefix used to generate a package name, if no prefix is provided in the symbol attributes. See “Setting the Auto-Generated Name Format” on page 131 for more information.

Naming and Packaging Options in Pure Hierarchy Mode

If the current design is in pure hierarchy mode, selecting the Device Naming and Packaging Options command will display this box:



Note that auto-packaging is not available in pure hierarchy mode. The following naming options are available:

None	Auto-naming is off. The Name attribute field will not be modified when a device symbol is placed.
Auto Naming On	A unique (within the circuit) name will be generated in the Name field when a symbol is placed.
Make auto-assigned names visible	If this box is checked, the auto-assigned name will be displayed on the schematic next to the symbol. If this box is not checked, the name will be stored with the device, but not displayed on the schematic.
Default name prefix	The text in this box will be the default prefix used to generate a device name, if no prefix is provided in the symbol attributes. See “Setting the Auto-Generated Name Format” on page 131 for more information.

Using Device Packaging

DesignWorks provides features for automatic or manual assignment of devices to physical packages. This mechanism is intended for use when creating designs that are destined for production in printed circuit board (PCB) or other discrete package format.

The Packager tool has the following features:

- Automatically generates a unique package assignment for each device placed on the schematic.

- Uses a name prefix stored with the device symbol, or a default value stored with the design. This allows appropriate names to be generated for discrete parts, e.g. “D1”, “D2”, “D3”, etc. for diodes, and “R1”, “R2”, “R3”, etc. for resistors.

- Automatically handles device types having multiple units per package, e.g. gates. Special attributes are stored with the symbol to tell the packager how many units are available and the pin assignments for each.

When Auto Packaging is enabled (i.e. the Auto Packaging On item is checked in the Device Naming and Packaging Options command), the Packager assigns each device to a package as soon as it is placed on the diagram. When a device is deleted, its package assignment becomes available for the next device of the same type that is created. In this mode, package assignments will be in chronological order of placement on the diagram.

It is normally convenient to leave Auto Packaging turned on while initially creating a design. This way, all devices get reasonable initial name assignments. The design can be repackaged later once the physical layout of the design becomes more clear. Repackaging may be done by any of these methods:

- Using the Repackage Design command in the Naming and Packaging Options menu. This will assign symbols to packages sequentially based on their position in the schematic.

- Manually, by changing the Name (or InstName) and Unit attributes.

- Using the Back Annotate command or other automatic back annotation

method to read package assignments from a PCB layout system.

Re-enabling Auto-Packaging After Manual Edits

If Auto-Packaging has been disabled while manual editing was done, then the design's package table will be out of date. It may be desirable to select the Rescan Design command from time to time to recheck the validity of the current package assignments. This command brings the package table up to date based on name and unit assignments currently in the design, without changing any values. In any case, Auto-Packaging cannot be re-enabled until a successful Rescan is completed.

Auto-Packaging Limitations

The Packager can be confused by some conditions when package assignments are changed manually. Such conditions include:

- Two devices given the same packaging assignment.

- A name or unit attribute field set to an invalid value.

- Packager attributes changed.

In these cases, the Packager will display a warning box showing the error location. After the warning has appeared, the Packager will attempt to continue assigning symbols to packages, although its accuracy cannot be guaranteed.

In order to re-synchronize the Packager with the design, you should take these steps as soon as possible:

- Correct the erroneous assignment, or mark the offending device with the “Ignore” option described in “Setting Device Packaging Options” on page 134.

- Select the Rescan Design command in the Naming and Packaging submenu.

NOTE: The Packager looks only at the name and unit attribute fields, not at the pin numbers themselves. If you manually change any pin numbers on devices without making corresponding changes to the Unit field, invalid package assignments will result.

Bringing the Design's Package Table Up to Date

The Rescan Design command forces the Packager to rebuild the design's package table from the current state of all device attribute fields. This can be used in the following circumstances:

To check for errors or obtain a free unit report after manual packaging changes have been made.

After back annotation or any other external process that may have changed package assignments.

After changing any of the attribute fields that may affect packaging.

If any packaging errors are encountered (e.g. duplicate gate assignment or invalid attribute fields), they will be reported at the end of this process and the design's package table will be considered invalid. No device attribute fields are affected by this command.

Getting a Report of Unused Gates

The Export tool is capable of generating a report of the packages in the design that have not been completely used. All PCB-related design kits supplied with DesignWorks include a script for generating such a report.

For information on creating your own report format that includes a free unit report, refer to the entry for \$UNUSEDUNITS in the DesignWorks Script Language Reference (separate manual on disk).

Batch Repackaging the Entire Design

The Repackage Design command in the Options menu allows you to request a complete reassignment of device packages. This command performs the following steps:

Clear the design's package table.

Scan all devices in the design, placing all devices marked as “Lock and Check” or “Lock and Don't Check” in the package table. This ensures that these names will not be used for automatic unit assignment. Devices marked “Ignore” are not placed in the table, meaning that their names could be used in automatic unit assignment. Any errors detected in the package assignment of devices marked “Lock and Check” are reported.

Sort all devices in the design by hierarchy level, then by page, then by horizontal grid position, then by vertical grid position.

In the sorted order defined above, assign a new package name and unit to all devices not marked “Lock” or “Ignore”.

For packaging, devices are sorted by page number, then by position on the page. Names are assigned in vertical strips corresponding to the lettered grid references on the page. I.e. all devices in column A are assigned, starting at the top of the page and working down. Next, all devices in column B are assigned, etc.

Performing Manual Packaging

All aspects of device packaging can be controlled manually, if desired. This can be done on two levels.

Manual/Automatic Packaging

The higher-level approach takes advantage of the packaging information in the device libraries and the error checking capabilities of the Packager. The following packaging features can be used:

Each library entry contains information about available gate package assignments. The list of available units appears in the Unit sub-menu of the device pop-up menu. Selecting one of these items automatically updates the Unit field and the pin number assignments on the device.

The Browser tool can be used to view and change all name and unit assignments for the design. Changing the Unit value from the Browser will automatically update the pin numbers (assuming the pin numbers are correctly set up in the packaging attribute fields).

The Rescan Design command can be used to check for packaging errors without affecting any existing settings. Any duplicate assignments or invalid unit settings will be announced.

The following cautions should also be noted:

The Packager does not check pin numbers when checking package assignments. It uses only the contents of the Unit attribute field. If you manually change any pin numbers on devices without making corresponding changes to the Unit field, invalid package assignments will result.

It is best to set the Packaging Level to Lock (using the Properties command) for devices that you wish to have a fixed package assignment,

even if not using auto-packaging. This will avoid loss of data if auto-packaging is inadvertently enabled or a Repackage Design is done.

Fully Manual Packaging

The lower-level approach completely ignores all packaging features. This can be used if unusual pin assignments are needed, or libraries without packaging information are being used. Note the following points when doing manual packaging:

Pin numbers can be edited directly on the diagram or assigned using the pin Properties box or the Browser tool.

If you do not wish to display a unit assignment (e.g. “a”, “b”, “c”, etc.), you can use the Define Attribute Fields command to turn off the Visible by Default option for the Unit field.

If you wish to manually assign a unit value, it is best to create a user-defined attribute field (e.g. “GateUnit”) and make all entries in that field. This is necessary to circumvent the automatic updating of pin numbers when the Unit field is changed.

When a netlist or Bill of Materials is created using the Export tool, all devices with the same name are normally treated as a single device. This means that they are, in effect, assigned to the same package. The Export tool does not check for duplicate pin numbers or name assignments, although it is possible to customize a report format that will assist in locating these problems.

Setting the Auto-Generated Name Format

An auto-generated device name consists of two parts, the fixed prefix and the numeric suffix. The prefix portion is derived from one of two sources:

The device's prefix attribute field, or if that is empty,

The value set using the Device Naming and Packaging Options menu command.

The numeric suffix is assigned sequentially for each different prefix found in the design. E.g. the packager will assign names “U1”, “U2”, “U3”, etc. to devices with the value “U” in their prefix field, and names “R1”, “R2”, “R3”, etc. for devices having prefix “R”.

NOTE: 1) The default prefix field is Name.Prefix. This can be changed by entering

the name of any other field in the design's PrefixField field.

2) Most library parts provided with DesignWorks that represent logic devices or integrated circuits have no Name.Prefix value. They will therefore use the value of the design's PkgPrefix field, "U" by default. Discrete parts or others that normally have specific standard prefixes are set to the common values.

TIP: There are no options in this version of DesignWorks for setting the number origin for name assignment, for example to start at R100 instead of R1. However, you can force this behavior by creating a separate page in your circuit and placing a number of devices with the names you wish to have unused. If needed, you can mark these devices with the "Omit from Report" option to ensure they do not appear in any netlist output. Alternatively, you can use a Export script to assign names in any desired format as a batch process after placing all your parts.

Batch Reassigning Device Names

The Reassign Device Names command is a more restricted version of the Repackage Design command and is intended for use when packaging is not being used, or for reassigning names in Physical hierarchy mode.

The Reassign Device Names command assigns a new name to all devices in the current circuit that are either unnamed or have a default name. It is intended as a quick means of tidying up automatically-assigned names when a circuit is created or edited.

NOTE: If the design is in "Flat" hierarchy mode (the default) then this command will override the gate package assignments made by the Packager. In other hierarchy modes, the Packager stores the package assignments in the "InstName" field, which is unaffected by this command.

Note the following rules:

- Only the current circuit is affected. Other circuit levels higher or lower in the hierarchy are unaffected.

- Gate packaging information is ignored, i.e. every device symbol is assigned a unique name regardless of package availability or "locked" status.

- The "Unit" attribute field is unaffected.

- Only the "Name" field is changed, regardless of hierarchy mode.

Names are assigned in order sorted first down the columns of the references grid, then across the page, then through subsequent pages.

Names are assigned using the device's prefix field (i.e. the one named in the design's "PrefixField" field) or the design's "DevPrefix" field.

Names that have been edited by the user are no longer considered "default" names and will not be changed.

Setting the Name Prefix for a Symbol

The name prefix is set by filling in the appropriate attribute field while editing the part in the device symbol editor. For example, for a resistor, you would place the value "R" in the Name.Prefix field using the Part Attributes command.

See "Setting Part and Pin Attributes" on page 285 for more information on using this command.

Specifying that a Device Should be Unnamed When Placed

In some cases it may be desirable to have a device remain unnamed when it is placed on a schematic. An example might be a symbol that is used as a place holder for some global schematic information or design parameters, a test point, I/O pad, etc. but does not represent a real device.

The keyword "\$NONAME" can be placed in the Name.Prefix field in a device symbol to indicate that no name should be assigned when the symbol is placed.

Selecting an Alternate Prefix Field

The prefix used in generating device names is derived from another device field known as the "prefix field". The default prefix field is "Name.Prefix". DesignWorks allows the prefix field to be selected for a given design so that multiple prefixes for different naming conventions can be stored with the same part. For example, the standard Discretes library included with the package has two prefix fields:

"Name.Prefix" is the normal one used by default for generating names.

"Name.Spice" contains prefix values specific to SPICE-based analog simulators.

In addition, the standard field “Function” contains a short function code for the part that can also be used as a descriptive prefix.

To set the prefix field:

- 1) Select the Design Attributes command in the Options menu.
- 2) Select the PrefixField item in the field list.
- 3) Type the name of the desired prefix field in the data box.

IMPORTANT: The prefix field name must match an existing field name exactly, including case, or it will be ignored.

Setting Device Packaging Options

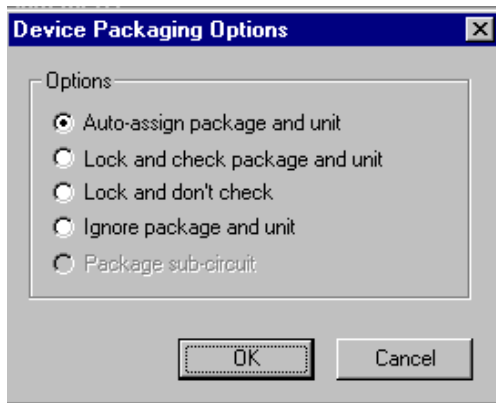
DesignWorks allows you to mark individual devices for special handling during packaging operations. To set device packaging options, bring up the Properties box by either one of the following methods:

Select the device by clicking on it, then select the Properties command in the Options menu.

OR

Right-click on the device, then select the Properties command in the pop-up menu.

Next, click the Packaging Options button in the Properties box: This displays the following box:



These options are described in the following table.

Auto-assign package and unit (the default value)	This device will get a package name and unit number assigned automatically by the Packager. The assignment can be changed by future repackaging operations if necessary.
Lock and Check	This device's package assignment will not be changed, but it is recorded in the package table so no other device will be assigned the same name. The device will be checked for packaging errors.
Lock and Don't Check	The same as Lock and Check except that the package assignment is not checked for errors. This could be used in special cases where the same name is purposely assigned to two different symbols.
Ignore	This device is completely ignored by packaging operations. I.e. its name and unit will not be changed, no error checking is done, and some other device might be given the same name.

The packaging level is actually stored in the PkgLevel attribute field of each device. You can set a default value for a symbol by placing the appropriate text in this field when creating a symbol. See “Setting Packaging Attribute Fields While Creating a Symbol” on page 144 for more information on this process.

WARNING: The Rescan Design command does not guarantee the correctness of your design. Pin numbers are not checked. Check your design carefully after any Back Annotation process.

Overriding Default Name and Unit Visibility

Normally, when a device is placed on the schematic, its name (Name or Inst-Name attribute field) and unit (Unit attribute field) will be made visible when the values are automatically assigned. This can be turned off globally using the Device Naming and Packaging Options command, but in some cases, it may be desirable to override this for specific symbols. For example, you may wish to use gate packaging to assign sequential pin numbers to segments of a connector symbol without displaying a Unit attribute next to each one.

To override the default visibility for a symbol:

Open the symbol in the symbol editor.

Select the Part Attributes command in the Options menu.

Select the attribute field in question (Name, InstName or Unit).

Enter a dummy value in the value text box. (This value will be overwritten by the packager in any case, but a non-empty value must be placed here to force the attribute field settings to be stored with the symbol.)

Select the Always Visible or Never Visible option, as desired.

Click OK.

Resave the part.

Using Packaging in Hierarchical Designs

Hierarchy Mode

The usage of attribute fields by the Packager is different between flat and physical hierarchy modes. See the discussion under “Attribute Fields Set By the Packager” on page 145. Packaging is not available in pure hierarchy mode.

Name vs. InstName

In hierarchical designs, a device symbol in a subcircuit can represent multiple physical devices. This happens if the parent device has been instantiated multiple times in the design. For this reason, a single device symbol has two different values that might both be considered a “name” in a hierarchical design. These two values are stored in the following two attribute fields:

The “Name” attribute field is associated with the definition of the subcircuit and will be the same for all instances of the subcircuit. For this reason, it can be thought of as the “logical name” or “definition name” of the device and cannot be used to hold the package assignment.

The “InstName” attribute field is associated with each instance of the subcircuit. It can take on a different value for each physical device it represents and is used to store the package assignment in hierarchical designs. You can choose to display either or both of these names on the schematic.

IMPORTANT: In Flat mode designs, you *should not* use the InstName field unless you have a very specific reason for doing so. InstName is used only for package assignments in Physical mode designs and will not be correctly output in netlists and reports in Flat mode.

See “Using the Name and InstName Fields” on page 170 for more information.

Restricting Packaging Depth

In a hierarchical design, a device symbol that represents a physical device may have a subcircuit for simulation or analysis purposes. Normally, the parent device rather than the contents of this subcircuit should be packaged. The packaging options in the device Properties command can be used to control packaging of internal circuits. Standard library devices provided with DesignWorks will default to packaging the parent device even if an internal circuit is added.

NOTE: The depth restriction on a device is stored in its Restrict attribute field. You can set a default value for this field when creating a device symbol. See “Setting Packaging Attribute Fields While Creating a Symbol” on page 144.

Name Assignment Order

In hierarchical designs, all packagable devices in a given circuit level are assigned before the internals of any subcircuit devices in that circuit. This ensures that a given circuit level contains sequential device names and units.

Using Device Libraries Without Packaging Information

If the Packager encounters a device without any packaging attribute fields define, it takes the following minimum action:

- The device is assumed to have only one unit per package, i.e. each device will be given a unique name.

- The default package name prefix for the design will be used to assign a name.

- No unit assignment will be displayed on the schematic.

- No Unit selection menu will appear.

- Default pin numbering defined in the library will be used.

In summary, the device will be named as if it has one unit per package.

Using Back Annotation

The Back Annotate command in the File menu is used to update device PCB package assignments from an external file. It can read files in several common industry file formats and uses this information to modify the Name (or InstName) and Unit attributes.

TIP: The Back Annotate command is intended specifically to update PCB package assignments and only operates on specific attributes using specific industry file formats. It is also possible to use scripting to read line-oriented text files and update arbitrary attributes in a design. See “Creating Scripts” on page 345 for more information.

To back annotate your design:

NOTE: PADS PCB users, please see the section below on considerations specific to PADS back annotation.

Save a backup copy of the design. Back annotation can make arbitrary changes to name and unit assignments in your design and it is important to be able to revert back to a known good copy in case of any file format or version problems.

Select the Back Annotate command in the File menu.

Enter the full path name of the back annotation file, or use the Browse button to locate it with a standard file box.

Select the file format from the Input File Format list.

Select which attribute field is being used for the physical device name. This will normally be the Name field in flat designs and InstName in physical hierarchy mode designs.

Select the desired Update Screen options. These options do not affect the back annotation process itself, but provide different methods of giving you feedback on the changes that have been made.

Click the Proceed button.

Click OK on the final confirmation.

Note these points about the back annotation process:

Back Annotation never adds or deletes devices or signals or changes connections on the schematic. The changes made are strictly attribute and pin number changes.

It is quite possible for the back annotation file to contain changes that produce subtle inconsistencies in the resulting design. E.g. Pin numbers may be mixed up to create invalid gate assignments. Back Annotation checks for major format errors and unresolvable names, but it assumes that the new values being assigned are correct and they are not extensively checked.

Back Annotation and Packaging

The Back Annotate command can make arbitrary name, unit and pin numbering changes to a design file. Obviously, any such a process can potentially bypass the Packager and create invalid package assignments. It is recommended that the Rescan Design command be used after any such process to ensure that the name and unit assignments are still valid.

Back Annotation in Hierarchical Designs

The Back Annotation module can be used with flat or hierarchical designs. Following are a number of points that should be considered, depending on the hierarchy mode of the design.

Flat Mode—Back annotation in flat designs is straight-forward. All device name references in the annotation file are assumed to refer to the Name attribute field in the design (the InstName field is not normally used in flat designs).

Physical Mode—In designs created using Physical hierarchy mode, device package names are normally stored in the InstName attribute field. Because InstName is an "instance" field, it can take on different values inside different copies of the same hierarchical block. Hierarchical names (i.e. names consisting of the lowest level device name prefixed with all parent symbol names) cannot be used by Back Annotation.

Pure Mode—Back annotation should not normally be done in Pure mode since the same device name may exist in various levels of the design. This should only be done with a clear understanding of the Pure hierarchy mechanism.

File Formats

It is important to remember that the InstName attribute field or the Name

attribute field should be used for both generating a netlist and for back annotations.

Back Annotation from PADS PCB

To back annotate from PADS to the DesignWorks schematic requires an additional file if the back annotation includes any "gate swap" commands. This is because PADS Perform back annotates gate swaps only by their unit name. For example,

```
*PADS-ECO*
*SWPGATES*
U1.B U1.C
*END*
```

DesignWorks also has the concept of unit names, but since PADS has no way to synchronize this information then we can't rely on the unit names representing the same pins. To resolve the problem a part type file (*.asc) from PADS Perform is used along with the PADS ECO file (*.eco). DesignWorks then uses the part type file to translate PADS unit names to real pin numbers. A part type file needs to be produced only once for a PCB design. In general it should be produced immediately after importing a netlist into PADS Perform. To produce the part type file select the following commands in PADS Perform, In/Out/Ascii Out. From the dialog select only the option Part Types. The default output file pperform.asc is saved in the directory ...\\PADS\\files.

In addition to the *.asc file, part number information must exist in each symbol in the DesignWorks so that the back annotator can look up the corresponding gate information in the *.asc file. The Back Annotation box has a control that allows you to select which attribute field is used for this purpose. The *.asc file is not needed if the *.eco file does not contain any SWPGATES commands.

Importing Back Annotation into DesignWorks

Back Annotation in DesignWorks is performed by opening the schematic to be back annotated and following these steps:

Select the Back Annotate command in the File menu.

In the dialog select PADS as the back annotation format

Press the Select button and locate the *.eco file which was produced by PADS Perform. If the file open dialog is not prompting for *.eco files then the PADS file format wasn't selected earlier.

Select the attribute field that will be used as the source for the part number used to look up gate swap information in the PADS *.asc file.

Press the Proceed button. Another file open dialog will appear, this time prompting for the location of the part type file *.asc. If you are sure that the *.eco file does not contain any SWPGATES commands, you can press Cancel at this step.

After selecting the part type file back annotation will begin.

IMPORTANT:

Doing a back annotation can make major changes to your schematic! Make sure you have saved your design and made a backup before proceeding.

Care must be taken to delete the contents of the ECO file after it has been back annotated to the DesignWorks schematic. Failing to do so will result in the same changes being back annotated again at a later date as part of a series of future ECO changes.

Only once you have confirmed that all changes were made correctly should you use the back annotated design as your working design.

Attribute Fields Affected By Back Annotation

The Back Annotation module uses the information in the back annotation file to update pin number and attribute field data in the current design. The following items of information will be affected:

Pin numbers will be modified or swapped as specified in the file. Note that changing the pin numbers on a device may require a corresponding change in the Unit attribute field. See the Unit attribute field below.

The normal selection for back annotating Flat or Pure hierarchy designs is to use the Name attribute field as the source and destination of device name data. The normal selection for back annotating Physical hierarchy designs is to use the InstName attribute field as the source and destination of device name data. These attributes are used to identify devices. Signals are always identified by their Name attribute field

The Unit attribute field is normally not directly referred to in a back annotation file, but will be set as a consequence of other operations. In file formats that contain explicit device swapping commands (Douglas, RINF,

Cadnetix), the Unit attribute value will be swapped with the name and pin numbers associated with a gate. The XCAD format contains only a sequence of pin number changes that could result in invalid unit assignments. Back Annotation attempts to set an appropriate value given the final pin number settings for each gate, but will set Unit to "?" if no matching assignment can be found.

The VisPin.List attribute field is associated with a specific pin and normally contains a list of the pin numbers that could be assigned to that pin. When a pin swap is executed, the contents of this field are swapped as well to maintain a valid correspondence with the actual pin number. In formats that directly set a pin number without indicating where it is being swapped to (XCAD), this field is left unchanged. This may result in surprising pin assignment changes if the DesignWorks packaging features are used after Back Annotation.

Back Annotation File Formats

The information in this section does not need to be read for normal Back Annotation operation. It is provided for cases where file format differences exist or for those who wish to create back annotation files by other means. In each of the supported file formats, Back Annotation recognizes only the keywords necessary to implement back annotation of an existing schematic. In all of these formats, when a package is specified by name, all devices in the DesignWorks design having the given name will be affected. E.g. if you have 6 inverters all named U23 and the back annotation specifies that U23 is renamed to U16, all 6 devices in the schematic will be renamed.

NOTE: We recommend the RINF format for users who wish to generate their own files by automatic or manual means. This format is more human-readable and more explicit than the others.

Douglas Format

The Douglas Professional Layout system uses the RINF back annotation format described below.

Cadnetix (SCICARDS) Format

In the Cadnetix format the following general format is recognized:

```
PACKAGES
pkgName11
```

```

pkgName22
pkgName33
.
.
SWAP pkg# pin# pin# ... WITH pkg# pin# pin# ...
.
.
.   RENAME pkg# TO newName
.
.
.

```

The PACKAGES keyword must appear first, followed by all package number definitions. The SWAP and RENAME lines can then follow in any order.

Racal-Redac (RINF) Format

The RINF specification includes numerous keywords for creating and modifying schematic and PCB data. The only ones recognized by Back Annotation are the REN_COM (rename component), SWA_PIN (swap pin) and SWA_GAT (swap gate) keywords, in the following general form:

```

.REN_COM
    oldPkgName  newPkgName
.SWA_PIN pkgName
    pin# pin#
.SWA_GAT
pkgName1 pkgName2
pkg1Pin  pkg2Pin
pkg1Pin  pkg2Pin
pkg1Pin  pkg2Pin

```

Operations can be specified in any order and are applied sequentially, i.e. changes are made immediately and subsequent references to the same package should be by the new name. Unrecognized keywords are skipped.

XCAD Format

The XCAD format uses no keywords but simply specifies changes, one change per line. All names and pin numbers on the left hand side are those existing in the schematic before any changes were made. Two possible formats are available, as follows:

```

oldName    ,oldPin#  newName,newPin#
oldName    newName

```

In the first format, all devices having the name "oldName" is renamed to "newName" and the pin "oldPin#" is renumbered to "newPin#". Note that if

one pin number on a device is changed to one in a different package or gate, all the pin numbers on that device must be changed. Back Annotation does not enforce this!

In the second format, all devices with the name "oldName" are renamed to "newName".

NOTE: Both of these formats are strictly name changes, not swaps. As a side-effect of the gate-swapping process, an XCAD file can contain names that no longer exist in the schematic. For this reason, unrecognized names are ignored in this format and not flagged as an error.

Creating a Symbol with Multi-gate Packaging

A multiple gate package is single physical component that may be represented by a number of separate symbols placed in a schematic.

NOTE: Even though this mechanism can be applied to types of components other than strictly gates, we will refer to each logic component as a “gate” for this discussion. If this is being used for connectors, for example, then each connector pin (or group of pins) is equivalent to a gate as far as the Packager is concerned.

All the gates in a package may have the same symbol and attributes or they may be different. If every gate in the package has the same symbol and attributes then the package may be stored in a DesignWorks library as one part. If any of the units have different symbols or attributes then they must have an entry in a library for each different type of gate.

The gate is automatically selected by the packager when a device is placed if Auto-Packaging is enabled. The gate may also be manually selected via the “Unit” sub-menu in the device popup menu.

There are a number of design attribute fields that also affect packaging operation. See “Design Attribute Fields Used By the Packager” on page 151 for more details.

Setting Packaging Attribute Fields While Creating a Symbol

A number of attribute fields are used by the Packager when it assigns a name

and unit to a device. If you intend to create your own library symbols that require gate packaging, you will need to be familiar with these fields. If you are only using the libraries provided with DesignWorks, or if you have no requirement for gate packaging, you will not need this information.

Required Packaging Attributes

This table describes the attribute fields must be correctly set for packaging to operate.

See “Setting Part and Pin Attributes” on page 285 for specific instructions on setting attributes while editing a symbol.

Field Name	Set In	Description
Unit.All	Part	Contains a list of all units in the package, e.g. “a,b,c,d”, <i>even if they are not all represented by this symbol</i> . Unit names are normally a single letter, but can be any combination of letters and numbers up to 16 characters, e.g. “Tom,Dick,Harry,Fred”.

Attribute Fields Set By the Packager

The Packager stores the package name and unit assignments in attribute fields associated with each device symbol. The fields that are used depend upon hierarchy mode:

Flat Mode	The package name is stored in the “Name” field and the gate unit (e.g. “a”, “b”, “c”, etc.) is stored in the “Unit” field. This reflects the common usage that the device name used on the schematic is the name of the package on the final PCB.
Physical Mode	The package name is stored in the InstName field and the gate unit is stored in the Unit field. The Name field cannot be used because a single device in a subcircuit may actually represent multiple devices on the physical PCB. Each one of these devices requires a separate package assignment. The “InstName” field can take on a different value for each instance.
Pure Mode	Packaging is not supported in Pure mode. This is because there is no instance data and therefore no place to store different package assignments for multiple instances of the same device.

Unit.List	Part	Contains a list of all the units in the package that can be represented by this particular symbol, e.g. “a,b”. In this case (and in most standard gate devices), all the devices in the package use the same symbol, so Unit.List and Unit.All are the same.
VisPin.List	Pin	Contains the list of visible pin numbers, one for each unit in Unit.List. The number of items and order of <i>VisPin.List</i> must match the order of <i>Unit.List</i> such that the first number in <i>VisPin.List</i> corresponds to the number of the pin for the first Unit in the package.

Optional Packaging Attribute Fields

This table lists fields that are optional, but that you may wish to set them for most packaging cases. These attributes are normally set in the part definition using the Set Part Attributes and Set Pin Attributes commands in the device symbol editor tool.

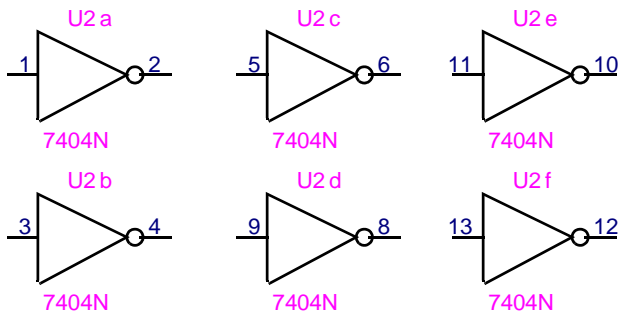
Part	Contains the part name, e.g. the name that will appear in a bill of materials. The Part field is also used to determine which devices can share a package. This can be any desired name, but must be exactly the same in all symbols that can be combined into a single package. The Packager will never combine two symbols with different Part values into the same package. In devices having several different types of gates in one package, a different symbol is used for each different type of gate, but each must have the same Part value. If there is no value in the Part attribute field, the library type name will be used.
Name.Prefix	This attribute field contains the prefix used to generate the device name. If it is empty, then the design's PkgPrefix field is used. NOTE: Name.Prefix is the default field used for this purpose, but this can be overridden. See the PrefixField description in “Design Attribute Fields Used By the Packager” on page 151.

PkgLevel This field contains the package level setting for this device instance. If the field is empty, or contains the character “0”, Normal mode is assumed. “1” means Lock and Check, “2” means Lock and Don’t Check, “3” means ignore. Any other value will be interpreted as Lock and Check. This value is not normally defined in the library entry for the device. The package level is usually set using the buttons in the Device Properties box, but the attribute field can be set directly if desired.

There are a number of other attribute fields that you may wish to set while creating a new device symbol for automatic power and ground connections, etc. See “Setting Part and Pin Attributes” on page 285 for more information on setting attributes in a symbol.

Creating a Symbol for Multiple Gates With Same Symbol - Example

A Hex Inverter (e.g. 7404) has six symbols in its package. Each of these symbols is identical and has identical attributes, so they can be represented by one entry for a 7404 in the library.



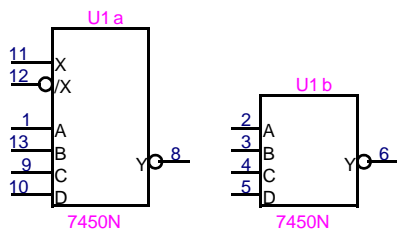
In the above example the packaging attributes are set as follows:

Field	Value
Unit.List	a,b,c,d,e,f
Unit.All	a,b,c,d,e,f
VisPin.List on the first pin	1,3,4,9,11,13
VisPin.List on the second pin	2,4,6,8,10,12

Creating Symbols for Multiple Gates With Different Symbols - Example

A 7450 Dual 2-Wide 2-Input AND-OR-INVERT Gate has two units. Each unit has a different symbol and thus two entries in the library, a “7450(N)A.a” and a “7450(N)A.b”.

If a package has more than one entry in a library, we annotate this by appending the first unit number of the package that shares the same symbol. If all units are the same then the library name does not include the unit.



The packaging of the And/Or/Invert 7450 is accomplished via these attribute fields:

In the “a” unit:

Field	Value
Part	7450(N)A
Unit.List	a
Unit.All	a,b
VisPin.List on the first pin*	11
VisPin.List on the second pin*	12
etc....	

In the “b” unit:

Field	Value
Part	7450(N)A
Unit.List	b
Unit.All	a,b
VisPin.List on the first pin*	2
VisPin.List on the second pin*	3
etc....	

NOTE: Since there is only one of each type of unit, it is not strictly necessary to set any value in the VisPin.List field. The pin numbers can be placed on the symbol using the device symbol editor in the usual way.

Creating a Symbol for a Discrete SIP Package - Example

A resistor “SIP” package is unusual because it has a common pin shared by all the symbols that are combined into a package. In this example, we will create a 9 pin device containing 8 resistors all with a common pin on one side. Pin 1 will be the common pin and pins 2 through 9 will be the 8 resistors.

To create this we will use the device symbol editor tool and edit the RES symbol provided in the Discretes library (being careful not to save it back under the same name!).

From the symbol editor’s menu, select the Part Attributes command, This brings up the attributes dialog. Scroll to the “Unit.List” attribute and set its value to “a,b,c,d,e,f,g,h”.

NOTE: The quotation marks "" are shown for clarity only and are not entered in the value field.

Next select “Unit.All” and make it the same as Unit.List.

Exit the Part Attributes dialog and then select the Pin Attributes command in the Options menu. The attributes for pin A should be visible in the dialog. Select the “VisPin.List” attribute and set its value to “2,3,4,5,6,7,8,9”.

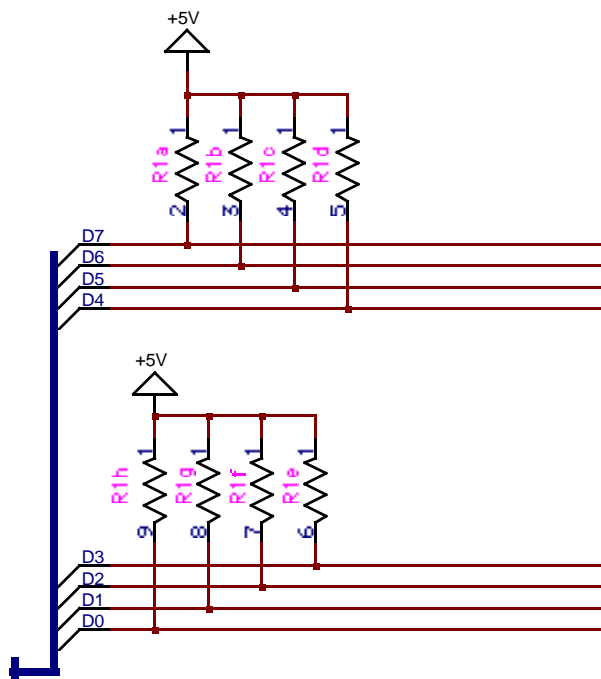
Now press the Next Pin” button at the bottom of the pin attributes dialog. The dialog should show the attributes for pin B. Set the VisPin.List attribute to “1,1,1,1,1,1,1,1”.

Giving pin B all the same pin numbers for each unit will give the other side of the SIP for each resistor a pin number of 1 so that when individual resistors are placed around a schematic the report generator will merge all the pins together.

NOTE: It is not strictly necessary to place the value “1,1,1,1,1,1,1,1” in the VisPin.List attribute for pin B. Since there is only one possible value for this pin number, you can also just place the pin number “1” on pin B on the symbol in the usual way and leave VisPin.List empty for this pin.

Exit the Pin Attributes dialog and use “Save As...” to save the new part to one of your custom libraries. We don't want to change the original RES definition.

Now the new SIP type may be placed around a schematic with each resistor automatically packaged by the packager. It is still up to you to make sure that the common pin 1 is always connected to the same signal.



To change the above example into a resistor DIP package without a common pin, you only need change the VisPin.List for pin A to “1,2,3,4,5,6,7,8” and VisPin.List for pin B to “16,15,14,13,12,11,10,9”.

Specifying PCB Package Type Information

See “Selecting the Part and Package Type” on page 87 for more information on specifying the package code for use by an external PCB package using the Package attribute field.

NOTE: The package code entered in the Package attribute field is used only to pass information to an external PCB layout package. It is not used internally by DesignWorks for gate packaging or any other operation.

Design Attribute Fields Used By the Packager

The following information is used by the Packager in determining what units are available in each package and pin assignments belong with each unit. Devices with only one unit per package do not need any special attribute fields.

PkgPrefix	This design attribute field contains the default package name prefix used if no prefix field appears in the device.
PrefixField	This design attribute field contains the name of the field to use for a device name prefix. The default is Name.Prefix, but this can be overridden to use specialized prefix values.

Packager Error Codes

A number of error messages may be displayed when an attempt is made to package a device with incorrect package attribute data. These are summarized in the following table.

Error Message	Meaning
An error occurred trying to get a new package name...code 1	The package name generated by this operation was too long. I.e. prefix is too long or package numbers are getting very large.
An error occurred trying to get a new package name...code 2	Some package names in Unit.List did not appear in Unit.All.
An error occurred trying to get a new package name...code 3	Unit.List was empty when Unit.All was non-empty, i.e. the list of allowable gates was empty.
An error occurred trying to get a new package name...code 4	Internal error - shouldn't occur.
An error occurred trying to get a new package name...code 5	Internal error - memory allocation problem.
Error in package attributes or internal packager error	Unit.List or Unit.All are empty, prefix too long, package number too large, or memory error. Occurs during Rescan Design.

Device Token Values

Every time a device is created in a DesignWorks circuit, it is assigned an integer value known as its “token”. The token number stays with the device for its lifetime and numbers are not re-used. This ensures that a given device can always be recognized despite duplicate names or name changes. The token is used for a number of internal operations in DesignWorks, but can also be seen by the user in the following circumstances:

The token number is displayed in the Properties box.

The token number can be written out in netlists or bills of materials whenever a guaranteed-unique identifier is needed.

Note the following characteristics of tokens:

Tokens are assigned independently for each circuit in a hierarchical design and are thus only unique within a circuit, not across the entire design.

Each logical symbol on the diagram (including pseudo-devices) has its own token. In a netlist, several symbols may be combined into a single package, so there is not necessarily a one-to-one correspondence between tokens and physical packages.

Using Packaging with Connector Symbols

Schematic symbols that represent connectors can generally be handled like any other device. However, for large connectors, you may wish to take advantage of gate packaging to allow you to use a number of smaller symbols instead of one large one.

Like any other large component, connectors can be handled in one of these ways:

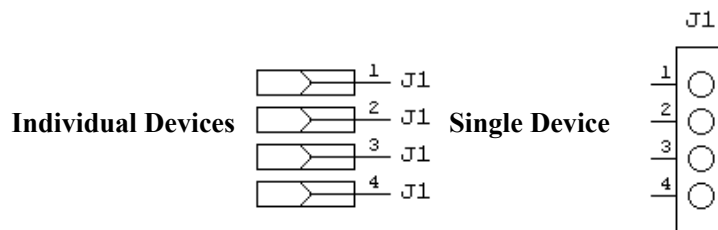
A special symbol can be created for the connector with the appropriate number of pins and pin numbering specified for each pin. This can be

done using the device symbol editor tool to create a device symbol using your own picture.

Each connector pin can be created as a separate, 1-pin device using the symbols from the Connectors library or a custom symbol. This option is preferable only if you need to spread the connector pins over different parts of the diagram. In this case, each “device” must be given the name of the connector and the pin number associated with that pin. We are relying on the fact that the report generator will normally merge all devices with the same name into a single component entry. Attributes (if any) are taken from the first device encountered with the given name.

Connector pins can be grouped with any number of pins in each group. This grouping may be done functionally, e.g. all data pins on one symbol, or simply by number of pins, e.g. 8 pins on each symbol. In the first case, the connector can be treated like a multi-gate package with different symbols being combined into a single package. See “Creating Symbols for Multiple Gates With Different Symbols - Example” on page 148 for information on creating a symbol for this type of usage. If you are using fixed-size groups, the same symbol can be re-used for each group, with only the pin numbers changing. See “Creating a Symbol for Multiple Gates With Same Symbol - Example” on page 147 for an example of creating a symbol with the appropriate packaging attributes for this method.

Here is an example showing these two methods:



NOTE: When the single-pin devices are used, every device must carry exactly the same name, although the names can be invisible if desired.

TIP: If you don’t want the Name and Unit values displayed on each individual segment, you can disable them using the steps described in “Overriding Default Name and Unit Visibility” on page 135.

Handling Discrete Components

From a packaging and pin number point of view, discrete components such as capacitors, transistors, etc. can be handled just like any other device, although you may want to note the following special considerations.

Pin Numbering on Discrete Components

Pin numbers are not normally placed on discrete component pins on a diagram. If pin numbers are omitted from a device, DesignWorks will normally put a question mark in the netlist item for that device. Two methods are available to provide pin numbers for netlisting purposes:

To provide automatic numbering of discrete devices pins, the Export tool provides the \$AUTONUMBER option, which can be specified in the form file. This option causes any device with less than or equal to the given number of pins to be numbered automatically if no pin numbers are present on the diagram. In the standard form file, this value is set to 3 to cover the common discrete components.

WARNING: This option assumes that the pin number order of the discrete components is not significant. If a specific order is important, then you should specify pin numbers explicitly and make them invisible.

Pin numbers can be assigned but left invisible. This is done using the Properties command for either the pin or the device.

Device Date Stamping

Date stamping is a mechanism that automatically marks devices with a time value when they are created. This can be used in conjunction with some PCB systems for forward- and back annotation purposes.

The date stamp is stored using the system's internal integer date format, that is, an unsigned integer representing the number of seconds since January 1, 1970. When a device is created or undergoes any major editing operation, the

current time value is converted to a decimal string and assigned to the DateStamp.Dev attribute field. For PCB packages that simply require a unique identifier for devices, this can be used directly with the knowledge that it can be sorted in time order. To create a more human-readable date value, the Export tool has date conversion functions available.

See more information on date conversions in report scripts, see the section Date and Time References in the DesignWorks Script Language Reference (separate manual on disk).

Disabling Date Stamping

The date stamping process incurs a small overhead in processing time and memory space. If you are not using this feature and wish to disable all internal processing related to it, you can place the following line in the initialization file dw.ini:

```
NODATESTAMP
```

This will take effect the next time you start the DesignWorks program.

See “Disabling Device Date Stamping” on page 382 for more information.

This chapter describes the methods in DesignWorks for entering, editing, displaying and using attributes. Attributes are arbitrary blocks of text that can be associated with any device, signal or pin in a design, or with the design itself. Attributes have a wide variety of uses, including:

- Displaying device name, package assignment, part type, value, etc. on the schematic.

- Storing manufacturing information such as catalog numbers, manufacturer's name, price, part description, etc.

- Storing data for use by external systems such as simulators, PCB layout, analysis tools, etc.

Attributes can be used in conjunction with the scripting and report generation features of DesignWorks to implement powerful interfacing, customizing and error checking features. You can define as many attribute fields as you need to store the text data required for your application.

In addition, many DesignWorks features use attributes as temporary or permanent storage locations for data relating to a device or signal. For example, each device in a design has the name and location of the library it was read from stored in an attribute field. This allows the library to be located easily for later updating.

Attribute Organization

Attribute Definition Table

Attributes are divided into named *fields*. The list of allowable field names is stored with each design in the *attribute definition table*. For each field, the table contains:

the field name (up to 32 characters)

what types of objects that field can be used in (devices, signals, pins or design)

data entry constraints (maximum length, carriage returns, etc.)

default settings, such as whether the field should be made visible on the diagram.

Once a field is defined in the table, its name is never typed again. When attribute data is entered for a selected object, the field name is picked from a list of allowable fields. This eliminates any chance of accidentally misspelling the field name in one object.

Predefined Fields

When a new design is created, a default attribute definition table is created with the standard, predefined fields. Predefined fields cannot be deleted from the design since many of them have specific internal functions in DesignWorks.

See Appendix A—Predefined Attribute Fields on page 361 for a complete listing of predefined fields.

User-defined Fields

You can add more fields to a design at any time using the Define Attribute Fields command in the Options menu.

Primary vs. Secondary Fields

DesignWorks has 50 or more predefined fields that are used for various program features and more can be added by the user. Most of these fields never need to be viewed or set directly by the user. To reduce clutter, attribute fields can be defined as being *primary* or *secondary*. When entering attribute data, you can choose whether you wish to see “primary fields only” or “all fields”. The Attribute Probe tool (🔍) in the toolbar shows primary-only or all fields based on the last setting used in an attribute entry box.

The primary/secondary setting is for viewing and editing convenience only and has no effect on the meaning or internal usage of the field. This setting for individual fields can be changed at any time using the Define Attribute Fields command in the Options menu.

Definition vs. Instance Fields

In a hierarchical design, a single device, pin or signal in a sub-circuit may actually represent several physical objects. This happens if the parent device containing the sub-circuit is used more than once in the design.

For this reason a distinction is made between fields that are associated with the definition of a sub-circuit and fields associated with the instance.

Fields associated with the definition will be the same in all instances of the sub-circuit. Changing the value in one instance will affect all instances equally.

Fields associated with the instance can take on a different value in each physical instance of the device, pin or signal. Changing the value in one instance has no effect on the others.

When defining a new attribute field for use in a hierarchical design, careful consideration should be given to whether to consider it a definition or instance field. If the value of the field has a significant impact on the logical functioning of the sub-circuit, then the field should probably be a definition field. Thus, if two sub-circuits require different values in this field, then separate definition circuits will have to be created. This makes it clearer to other users of the design that a distinction exists.

See Chapter 10—Hierarchical Design on page 215 for more information.

Temporary Fields

Attribute fields can be marked as “temporary”, meaning that the data is not saved with the design file. These fields can be useful for holding calculated or transient values that are to be displayed on the schematic, or for holding intermediate values that need to be stored with objects during report generation. To create a temporary field, check the appropriate box in the Define Attribute Fields dialog when defining the field.

Attribute Limitations

Attribute fields have the following specific limitations:

Maximum length of field name: 32 characters

Maximum length of field data item: 2,000,000,000 characters. Note: In principle, binary data could be pasted into an attribute field although this is not recommended since DesignWorks provides no mechanism for displaying or editing this data.

Maximum number of user-defined fields: 900

Maximum number of displayed positions of a single attribute item: 100

Like all other circuit data, the amount of attribute data that can be associated with a design is limited by available memory.

Entering and Editing Attribute Data - Basic Procedure

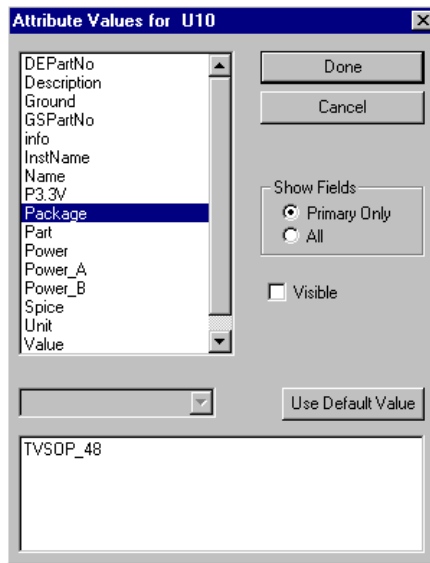
To edit the attributes associated with a device, pin or signal, you can either:

Select the object of interest on the schematic (you can only enter data for one object at a time using this method), then select the Properties command in the Options menu. In the info box that is displayed, click on the Attributes button.

Right-click on the object and select the Attributes command in the menu.

To edit design attributes, select the Set Design Attributes command in the Options menu.

Any of these methods will display the general attribute editing box which is used to enter or edit attribute data on a single object:



By default the field list at the left side of this box shows only the primary fields available for the selected kind of object. You can click on the All fields button to view all available fields. Custom fields can be added to this list using the procedure outlined in “Defining a New Attribute Field” on page 177.

Entering Data Values

To edit the contents of a field, simply select the field name in the list. The current contents of the field will be displayed in the editable text box. Edit this value using the usual Macintosh text editing techniques. Press the Done button or select another field when done. When you select another field, data entered for the previous field is saved. If the data you typed exceeded the maximum length for the field, or if it contained invalid characters for the field, then you will be asked to correct the data.

You can view or edit as many fields as desired while in this box. No changes are made to the actual design data until you click the Done button. Clicking Cancel will abandon all changes made while in this box.

Restoring the Default Value

Clicking the Use Default Value button sets the value for the selected field to

the default value stored with the symbol. This is valid only for device and pin attributes for which a default value was specified. If this button is inactive (grayed out) then the value is already the default value, or no default value is present. Only devices and pins can have default values.

See “Setting Default Values” on page 177 for more information on entering default attribute values while editing a symbol.

Using the Value List Pop-up Menu

If the selected field has an associated value list field, then a pop-up value menu will appear just below the “Use Default Value” button. An item selected out of this menu will cause the value text box to be updated accordingly. This does not prevent the text data from being edited directly. See more information on value list fields in “Using Value List Fields” on page 173.

Displaying an Attribute on the Schematic

To display device, signal or pin attribute text on the schematic:

Edit the attribute value as desired.

Turn on the Visible switch.

When you click Done, the attribute text will now be displayed in a default position near the device, pin or signal. It can be dragged to the desired location using the pointer cursor.

Design attributes can be displayed on the schematic using text variables. See “Using Text Variables” on page 107.

For information on setting the default position for an attribute on a symbol, see “Using Default Position Fields” on page 175.

Entering Design Attributes

To view or set attributes associated with the design itself, select the Design Attributes command in the Options menu. This command displays the standard attribute data entry box allowing you to enter attribute data for the design itself. Design attributes can be used to create variables in text blocks on the schematic and to insert global data in netlist output.

NOTE: The Visible option is not available for design attributes, since there is no single object on the schematic to associate them with. Design attributes can be

made visible by using text variables, described in “Using Text Variables” on page 107.

See the entry Attribute Field References in the DesignWorks Script Language Reference (separate manual on disk) for more information on design attributes in netlists.

Entering Pin Attributes

To enter or edit attributes for a pin, you can use any of the following methods to display the attribute editing box:

Select the pin of interest on the schematic (make sure you are selecting the pin and not the signal, see “Selecting a Pin” on page 93), then select the Properties command in the Options menu. In the pin info box that is displayed, click on the Attributes button.

Right-click on the pin and select the Attributes command in the menu.

Select the device that the pin is attached to, select the Properties command to display the device properties box, then click the Pin Properties button in this box, then the Attributes button in the pin info box.

The attribute editing box for pins is identical to the one for devices, signals and the design, except that Next and Previous buttons appear at the bottom. These can be used to view and edit other pins on the same device without having to return to the schematic and select them individually.

Controlling Attribute Display Characteristics

Rotating Attribute Text

To rotate an attribute text item that is already displayed on the schematic, follow these steps:

Right-click on the text item. This will display the attribute pop-up menu. Select the Rotate Right or Rotate Left command.


Hiding a Visible Attribute Value

If you wish to hide a device, pin or signal attribute value that is displayed on the schematic (that is, remove the displayed text from the schematic, but keep the value associated with the object), you can use any of these methods:

Right-click on the text and select the Hide command.

Right-click on the text and select the Edit command. In the text edit box that appears, turn off the Visible switch.

Display the general attribute editing box for the object, using the methods described earlier in this chapter. Select the field in question and turn off the Visible switch.

Click on the  tool in the tool palette, or select the Zap command in the Edit menu and then click on the attribute text you wish to hide.

Any of these methods removes the displayed text from the schematic without modifying the value that is stored with the object. The value can still be used for report generation, etc.

NOTE: If the value you are hiding is a signal name, this operation may have side effects on the connectivity represented by the signal. See “Signal Connectivity Rules” on page 213 for more information.

Design attributes can only be displayed on the schematic using text variables. These are described in *“Using Text Variables” on page 107*.

Clearing a Visible Attribute Value

If you wish to remove the displayed attribute text associated with a device, pin or signal and remove its value completely, you can:

Right-click on the text and select the Clear command.

Right-click on the text and select the Edit command. In the text edit box that appears, remove the text value.

Display the general attribute editing box for the object, using the methods described earlier in this chapter. Select the field in question and remove the text value.

NOTE: The first method above (the Clear command) actually performs a slightly different function to the other methods. It removes the visible text on the schematic and sets the attribute to its default value. For devices and pins, this

means any value that was specified with the device symbol when it was created.

Displaying an Invisible Attribute Value

To display an attribute value that is already associated with a device, pin or signal, display the general attribute editing box using the methods described in “Entering and Editing Attribute Data - Basic Procedure” on page 160, then turn on the Visible switch. The value will be displayed in a default location near the object and can then be relocated as desired on the schematic.

Design attributes can only be displayed on the schematic using text variables. These are described in “Using Text Variables” on page 107.

NOTE: The Name attribute is a special case in that it can be displayed by simply clicking the text cursor on the object in question, in addition to the method described here.

Setting Attribute Text Style

The text style used to display attribute values on the diagram is determined by two settings:

The text style set using the Design Preferences command is the default display style for all attribute fields.

A text style can be set for a specific field using the Define Attribute Fields command. If set, this style overrides the default style for the design.

There is no way to set text style for an individual attribute item on the diagram.

WARNING: Changing the attribute text style affects all visible attributes throughout the design. This may alter text alignment and position to accommodate a different text size.

The attribute text style setting affects the following types of items:

Device and signal names.

Bus breakout and bus pin labels.

All other attributes displayed on the diagram.

but does not affect:

Border text (see “Setting Border Text Style” on page 341).

Pin numbers (see “Setting Pin Number Text Characteristics” on page 104).

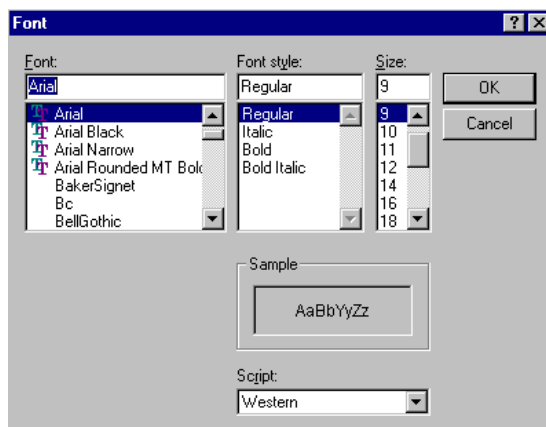
The text changes are applied when the OK button in the Design Preferences box is clicked. For larger designs, there may be a substantial delay while new positions of all displayed attribute items are calculated.

Setting Global Text Style

To set the global text style:

Choose the Design Preferences command in the Drawing menu.

Select the Text tab and click on the “Set Font...” button in the Attribute Text Font area. This will display the following box:



This box allows you to select the text font, size and style used for all attributes displayed in the design.

Select the desired font, size and style, then click OK.

Click OK on the Design Preferences box.

Depending on the size of the design, there may be some delay at this point.

The program must check all visible attribute items to see if there position and framing is affected by the text change.

Setting Text Style for a Single Field

To set the text style for a chosen field:

Select the Define Attribute Fields command in the Options menu.

Click on the desired field in the field list.

Click on the Text Style button.

Choose the desired settings and click OK.

Click OK in the Define Attribute Fields box.

Setting Attribute Justification

Attribute justification determines how text is aligned relative to the device it is associated with. Generally, the program chooses a justification that is appropriate for the initial placement of the text and there will be no need to change it. However, if text is being modified frequently and does not reposition itself as desired, the justification can be modified individually for text items.

To set the justification, right-click on the text item in question and select the Justification command. This command allows you to change the vertical and horizontal justification used in the positioning the attribute text on the diagram. When this command is selected, the following box will be displayed:



The selected point on the text is considered to be the reference point for the given attribute block. This point will be kept fixed if any field value or text style changes cause the box to be resized.

NOTE: The Justification command affects only the position of the text relative to the associated object. It does not affect the alignment of text in multi-line text items on the schematic. DesignWorks does not provide any method of choosing center or right justification of multi-line text items.

Displaying an Attribute Value in Multiple Locations

You can display a single attribute value in multiple locations on a schematic. This is intended for use particularly with signals, which can be spread over large areas of a page or multiple pages. If a value is displayed more than once, editing any one of the displayed values will affect all appearances of it.

To create multiple displayed attribute items, right-click on the text and select the Duplicate command. This command creates another visible occurrence of the same attribute field. This text can then be dragged or rotated to any desired position on the schematic.

Showing the Field Name with an Attribute Value

The Show Field Name command allows you to display the field name with the value on the schematic. When this item is checked, the display will be in the form “fieldName=value”. Selecting this command again will cause the display to revert to the normal value display. This command applies only to the selected field on the selected object.

TIP: You can elect globally to show the field name for a certain field whenever it is used. See “Defining a New Attribute Field” on page 177 for more information.

Other Ways of Viewing and Editing Attributes

Editing Attributes on the Schematic

Any attribute value associated with a device, pin or signal that is displayed on the schematic can be edited right in place using the text tool. To use the text tool, click on the **A** item in the tool palette, or select the Text item in the Edit menu. When you click the pencil cursor on any displayed attribute value, a flashing cursor will appear in the text to indicate that it is editable.

NOTE: DesignWorks does not support in-place editing of attributes that appear rotated on the schematic. If you click in such a value, a simple value editing box will be displayed instead of the edit cursor on the schematic.

An alternative method of editing a single displayed attribute field is to right-click on the text and select the Edit command. This command opens a text box allowing you to edit the contents of the selected field. All locations where this field is displayed on the schematic will be updated when the OK button is clicked.

Using Value List Sub-menus


A device or pin pop-up menu may contain one or more sub-menus as its last items. These sub-menus allow you to select from a list of possible values for an attribute field. Selecting an item in one of these lists changes the value in the associated attribute field.

Most of the standard libraries provided with DesignWorks contain fields defining the part code, the package type and the gate unit (e.g. a, b, c, etc.). When one of these fields has more than one possible value, the list is displayed in one of these menus. For example, if the sub-menu “Unit” appears, this means there are multiple gate units available in this device type. Selecting one of the units in the Unit sub-menu causes the Unit attribute field to be changed for that device and the pin numbers to updated accordingly.

See “Using Value List Fields” on page 173 for information on creating your own value list fields.

Probing Attributes on the Schematic

The contents of all fields associated with an object can be viewed with a single click using the Attribute Probe tool.

Select the Attribute Probe command in the Edit menu or click on the  tool in the tool palette.

Click and hold on the device, signal or pin in question. The dot at the base of the question mark is the “hot spot” that determines which item is selected.

DesignWorks follows these rules in displaying fields in the attribute probe pop-up window:

- Fields are shown sorted by field name.

- Only fields with non-empty values are shown and only the first 32 characters of the field value will appear.

- By default, only Primary fields are shown, but this depends on the last

setting of the “Primary Fields/All Fields” buttons in the general attribute editing box. To change this setting, using the Properties or Attributes command on any object using the general attribute editing box, then select the Primary Fields or All Fields setting as desired. This will affect the attribute probe next time it is used.

Using the Name and InstName Fields

In DesignWorks, the single most important attribute field is the “Name” field. It can be associated with devices or signals and is used to uniquely identify an object within its circuit. In PCB designs, the “name” is also the “package assignment” in that it denotes the physical device that a symbol on the schematic is assigned to.

Names may contain any letters, numbers, “_” (underscore) or “.” (period) characters. They are restricted in length to at most 16 characters. The name associated with an object can be placed anywhere on the diagram and will be removed if the object is removed.

NOTE: DesignWorks does not actually prevent you from using any combination of characters in a name, including blanks or special codes. However, many of the error checking and netlisting scripts will flag unusual characters as an error, depending on the characteristics of the external layout system or simulator being used.

The Name field can be entered and edited just like any other attribute, but because it is so widely used, there are some extra features to facilitate working with it. These are described in detail in the following sections.

NOTE: If you plan to create hierarchical designs, you should understand the distinction between the Name and InstName fields. This is described in detail in the next section. For flat designs, the InstName field is not required and all device naming can be done with the Name field.

Choosing Whether to Use Name or InstName

The distinction between the Name and InstName fields becomes important if you plan to use the physical hierarchy mode in DesignWorks to stored physi-

cal package assignments or other data. Here are some notes to help you choose the appropriate name field to use in hierarchical designs:

In non-hierarchical designs (that is, any design in which you have not explicitly selected a hierarchy mode) you should use only the Name field and leave InstName unused.

The Name is a *definition* field, meaning that it will have the same value inside all instances of the same type of sub-circuit. For this reason, Name cannot be used to store physical data that may be different in separate instances of a sub-circuit. Specifically, for designs intended for PCB layout, the Name field cannot be used as the package assignment.

The InstName field is used in hierarchical designs for PCB package assignments and other physical location data. InstName is an *instance* field, meaning that it can have a different value for every physical device in a hierarchical design. See the section “Names in Hierarchical Designs” on page 172. The InstName field is not normally used in Flat mode designs.

You can find more information on hierarchical design modes in Chapter 10—Hierarchical Design on page 215.

Invisible Names

Like other attribute fields, the Name field can be associated with a device or signal without displaying it on the diagram. The visibility of the name can be set using the Properties or Attributes command associated with the device or signal in question.

NOTE: When signals are connected by name, only visible names affect connectivity. Two signals with the same invisible name will not be connected. See “Signal Connectivity Rules” on page 213 for more details.

Device Names

Typical device names might be “U23”, “C4”, “XTAL1”, etc. The Name field for devices is equivalent to the “reference designator” used in other systems.

The Name is distinct from the “type name” that is used to distinguish the type definition that is read from a device library. The “type name” is not an attribute, but a fixed name associated with a device symbol when it is saved in a library. Typical type names are “74LS138”, “MC68000L8”, “SPDT Switch”, etc.

As well as appearing as a text notation on the diagram, the device Name field is used by the following DesignWorks functions:

if the design is in flat hierarchy mode, then the Packager tool places the assigned package name in the Name field. In other modes, the “InstName” field is used.

the device name is used in report output, such as netlists and bill of materials reports.

the device can be located by name using the Find tool.

the device name associated with a Page or Port Connector is used to make logical connections.

Signal Names

Typical signal names are “CLOCK”, “ADDR12”, etc. Signal name is referenced by the following DesignWorks functions:

the signal name is used in report output, such as netlists.

the signal can be located by name using the Find tool.

signals can be logically interconnected by name.

See more information on signal names and logical connections in Chapter IV - Schematic Editing.

Names in Hierarchical Designs

In a hierarchical design, a single device in a sub-circuit may actually represent several physical devices. This happens if the parent device containing the sub-circuit is used more than once in the design.

See “Definition vs. Instance” on page 217 for an example of this situation.

For this reason another attribute field is needed that will be unique for each and every physical instance of a device in the design. The “InstName” field serves this purpose.

If the design is in “physical” hierarchy mode, the InstName field is used by the Packager to store the package assignment. If the Packager is not being used, the InstName field can be used to store notations about the physical placement of the device.

NOTE: In a hierarchical design, the Name field can be thought of as a “logical” name and the InstName field as a “physical” name. The “logical” name should be assigned by the designer based on the device's function in the circuit, not on its physical location.

The visibility of the Name and InstName fields on the diagram can be set separately. It may be desirable to only show the Name field on the diagram during the early stages of design development. The physical package assignments can be shown later when needed for manufacturing.

Using Value List Fields

A value list field allows you to specify a list of possible values that can be used in a selected field. This list of values is then displayed to the user at various locations, allowing easy selection of one of the available items. For example, this mechanism is used to create the “Unit” and “Part” fields used in all standard DesignWorks libraries. These standard fields can be used as an example.

NOTE: DesignWorks does not restrict the user from entering values different from the ones in the list using the usual attribute editing techniques. If you want to restrict the values available in a given field, you may wish to create an error checking script that verifies field values. See “Locating Found Objects on the Schematic” on page 247 for more information.

A value list field has the same name as the associated field but with “.List” appended. When a field named “xxx.List” is seen by the program, it takes these steps:

When the attribute edit box is displayed for a given object and the field “xxx” is selected, a pop-up menu is displayed containing all the possible values for the field. The user may then select the appropriate value from the menu.

When a pop-up menu is displayed by right-clicking on an object, a sub-menu with the field name “xxx” as its title will appear. The list of possible field values will be displayed in the sub-menu. The user can select one of these items to directly change the value of field “xxx”.

Value List Data Format

The value list field must contain a sequence of textual values following these rules:

Items must be separated by commas. A comma will always be taken as a separator and therefore cannot be part of the value.

The value string can contain any characters other than commas, but leading and trailing blanks will be removed.

The length of a value string is limited to 16 characters.

The maximum number of values is limited to 256.

Normally, a default value for the “xxx.List” field should be defined with the symbol in the library using the device symbol editor tool. In this way, the value list will always be available whenever this device type is selected from the library. If the field is defined in only one device on the diagram, the value list will only be available for that device.

Creating a Value List Field

The creation of value list fields will be illustrated by way of an example.

Suppose you wish to create a symbol for a device which will be simulated using an external simulator. This symbol can be associated with one of three models which have the same pinouts, but different timing parameters. The model name is to be included in the netlist. To create a list of the available model values:

Using the Define Attribute Fields command, create two new fields, called “Model” and “Model.List”. For both fields, check the “Devices” box indicating that these will be associated with devices.

Using the device symbol editor tool, create the desired device symbol, or open an existing one for editing. Using the symbol editor's Part Attributes command, create a field called “Model.List” containing the desired list of values. For example:

```
MODEL1, MODEL2, MODEL3
```

Save the edited symbol to a library. Select the part from the library in the usual way and place it on a circuit diagram.

Right-click on the device. You will now see at the bottom of the pop-up menu a new item called “Model”. Moving down to this item will display

a new sub-menu containing the values entered in the “Model.List” field.

Using Default Position Fields

For device or pin attribute fields that are to be displayed on the schematic, an associated field called a “default position field” can be created. For example, this mechanism is used to provide default positions for the “Name” and “Part” fields used in all standard DesignWorks libraries. These standard fields can be used as an example.

A default position field has the same name as the associated field but with “.Pt” (for Point) appended. When a field named “xxx.Pt” is seen by the program, it uses the values in this field to generate a default position for the “xxx” field.

Default Position Data Format

The default position field must contain a sequence of textual values following these rules:

Two items, an X and a Y position, must be specified, separated by a comma. Leading and trailing non-numeric characters are ignored.

The X and Y values are each decimal numbers representing an offset from the top-left corner of the device measured in 1/1000". Negative values are allowed (negative is up and left).

The X coordinate can be optionally followed by an upper case letter indicating horizontal justification, as follows: “L” for left, “M” for middle, “R” for right. I.e. the given number specifies the offset to the left, middle or right position of the displayed text. If no letter appears, middle is assumed, i.e. the text will be centered at the given position.

The X coordinate can be optionally followed by a lower case letter indicating which point on the symbol’s bounding box to use as the reference point, as follows: “l” for the left edge, “m” for the midpoint, “r” for the right edge. I.e. the given number specifies the offset from the left, middle or right position of the symbol. If no letter appears, left is assumed.

The Y coordinate can be optionally followed by an upper case letter indicating vertical justification, as follows: “T” for top, “M” for middle, “B” for bottom. I.e. the given number specifies the offset to the top, middle or bottom position of the displayed text. If no letter appears, middle is assumed, i.e. the text will be centered at the given position.

The Y coordinate can be optionally followed by a lower case letter indicating where on the symbol the position is to be measured from, as follows: “t” for top, “m” for middle, “b” for bottom. I.e. the given number specifies the offset from the top, middle or bottom of the symbol’s bounding box. If no letter appears, top is assumed.

Some examples of value points are:

OR,OT	The value will extend right and down from the top-left corner of the device.
OMm,OBt	The value will be centered above the symbol with the bottom of the text at the top of the symbol.
150,-75	The value will be centered at a position above and to the right of the top-left corner.
(400L,100)	The value will extend left from a point to the right of the top-left corner of the device. (Leading and trailing characters are ignored.)

Normally, a default value for the “xxx.Pt” field should be defined with the symbol in the library using the device symbol editor. In this way, the value will be available for setting the default position of the associated field when the device is placed on the schematic.

TIP: You can place a symbol on a schematic, position all the displayed attributes in the locations you want them, and then use the Save to Lib command to store the symbol back into a library. The Save to Lib command automatically creates .Pt fields for any displayed attributes and saves them with the new definition. See “Saving a Symbol Definition from a Schematic to a Library” on page 272.

Setting Default Values

A device symbol can have associated with it predefined default values for any number of fields. Values can be specified for the device itself, and independently for each pin on the device.

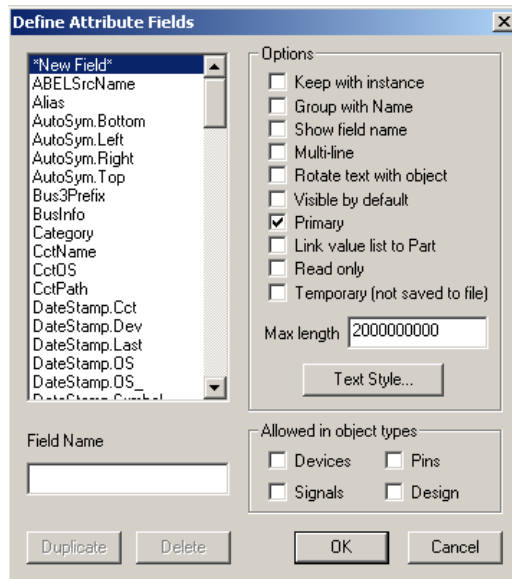
When the standard attribute data box is displayed for a device, you will see a button labeled “Use Default Value”. If this button is grayed out (disabled), then there is no default value or the value shown is already the default.

See “Setting Part and Pin Attributes” on page 285 for more information on creating default attribute values.

Defining a New Attribute Field

All changes in field definitions are done using the Define Attribute Fields command in the Options menu. Selecting this command displays the follow-

ing dialog box:



To create a new field:

Select the •New Field• item in the list, if it is not already.

Type the name of the field in the Field Name box.

Select one or more of the boxes under “Allowed in object types”.

Optionally enter a maximum length.

Optionally enable any of the other field options.

Click OK. The new field will now appear in the attribute data entry box for the selected object types.

Setting Attribute Field Options

When a new field is created using the Define Attribute Fields command a number of options are available which determine how that field will behave in your design. These options are described in the following sections.

NOTE: Changing any of the following options in a field that has already been used in the design has no effect on items already displayed on the schematic.

TIP: To make a temporary change (e.g. to display a field that is not normally displayed) you can use the Duplicate button in the Define

Attribute Fields box to create a new field containing the same data but with new display options. This field can be deleted again when no longer needed.

Note the following characteristics of these settings:

Changing the values of any of the option switches for an existing field do not affect any values already in the design. Only new data will be affected. See “Using Duplicate, Merge & Delete for Global Editing” on page 182 for tips on using the Duplicate function to update the way existing data is displayed.

The name and option settings for most predefined fields are fixed and will be disabled in this box.

Regardless of changes made in this box, no design data is updated until you click the OK button.

Field Name

The Field Name box allows you to enter the name of a new field or rename an existing user-defined one. If you rename a field so that it has the same name as another existing field, this becomes a Merge operation described in “Merging Two Existing Attribute Fields” on page 184. You will be prompted to confirm a merge before proceeding.

Keep with Instance

This option applies only in Physical Hierarchy mode. When this switch is checked, the selected attribute field may take on a different value for each instance of a device, pin or signal.

See “Definition vs. Instance” on page 217 for more information on instance data.

Group with Name

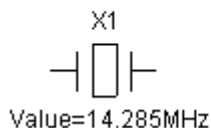
When this switch is checked, any value entered for this field will be displayed beneath the Name field by default. When the Name is moved, this field will also be moved.

This option can be used for fields that are typically shown with the name on a schematic. E.g. the Unit field (gate unit) has this setting enabled by default.

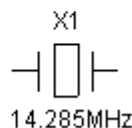
Show Field Name

When this switch is checked, the field name will be shown with the value on the schematic.

With Show Field Name



Without Show Field Name



The Show Field Name option can also be controlled for each individual item on the schematic using the Show Field Name command in the attribute pop-up menu.

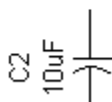
Allow Carriage Returns

If this switch is checked, then the Return key may be used to enter Carriage Return characters into this field.

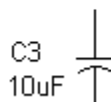
Rotate Text With Object

If this box is checked, displayed values will be rotated if the associated object is rotated on the schematic. Otherwise, when the object is rotated, the text will move to an appropriate new position but be drawn in the normal orientation.

Rotate With Object On



Rotate With Object Off



Visible by Default

If this box is checked, whenever a value is entered for this field it will be displayed on the schematic in a default position. If the field is found in the default attributes for a device, the field will be displayed automatically on the diagram when the device is placed.

NOTE: Changing this setting has no effect on values already in the circuit. If you want to globally show all the values of a field on the schematic, see “Temporarily Displaying Attributes” on page 185.

Primary

If this box is checked, the field will be considered a “primary” field and will be displayed in all attribute entry boxes when “Primary Only” is selected. It will also appear in the window that is displayed by the Attribute Probe (?) cursor.

Link Value List to Part

If this box is checked, any change in the Part attribute field will cause a corresponding value for this field to be selected from a list of available values.

See “Using Value List Sub-menus” on page 169 for more information.

Read Only

This setting indicates that the field value cannot be changed directly by the user. The value can still be changed by scripts or internal processes.

Temporary

This setting indicates that the field value is not saved to the design file, i.e. it is only valid from when it is set until the file is closed.

Allowed in Object Types

This group of check boxes allows you to select which types of schematic objects may have this field associated with them. You can check any one or more boxes, although in general we suggest using different names for fields used with different object types.

NOTE: If you plan to place pin attributes in a design for later extraction in a report or netlist, you may wish to refer to the entry “Precedence of Field References in Pin Listings” in the DesignWorks Script Language Reference (separate manual on disk) for information on how pin attribute references are located.

At least one of the “Allowed in Object Types:” switches must be selected for a new field.

Maximum Length

You can specify a maximum length (in characters) for the field data. This value does not determine how much storage is allocated internally for the data, it only affects checks that are done when the user enters data. All attribute data stored with the design occupies only the space required for its current value.

Duplicate

This button allows you to create a new field with the same settings and (optionally) data contents as an existing one. You will be prompted to confirm the data duplication option before proceeding. More information on this operation is given in the following section.

Delete

This button allows you to permanently remove a field definition and all associated data from the design.

WARNING: THIS CANNOT BE UNDONE!!!

More information on the delete operation is given in the following section.

Using Duplicate, Merge & Delete for Global Editing

The Duplicate, Merge and Delete Attribute functions do more than just add or delete a name in the attribute list. They will actually scan the entire design and update the data stored with each object. This makes them very powerful tools for updating your design. First we will review how the functions are invoked, then mention some possible uses for them.

These three functions are all invoked from the Define Attribute Fields command in the Options menu. No design data is modified until the OK button is clicked in the Define Attribute Fields box.

NOTE: When the design is updated, all Duplicates are done first, then all Merges and Deletes. This means that a field can be duplicated, then the original source deleted, all in one invocation of the Define Attributes box.

IMPORTANT: There is no guarantee of the order of execution within the Duplicates or within the Deletes and Merges. Therefore, chains of operations (e.g. making a copy of a copy) are not guaranteed to work as expected. If there is any doubt, perform one operation, then exit the Define Attributes box using the OK button, then perform the next operation.

WARNING: Each of these operations can result in data being updated throughout a design, and they cannot be undone! For large designs there may be a considerable delay after the OK button is clicked in the Define Attribute Fields box.

Globally Duplicating Attribute Data

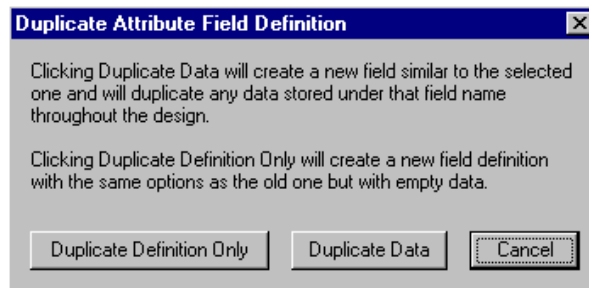
The Duplicate operation adds a new field name to the list with characteristics the same as an existing one. It will also optionally duplicate the data stored under that name throughout the design.

NOTE: The pre-defined fields in DesignWorks cannot be renamed or removed. However, you can duplicate them, and the copy will be marked as “user-defined”, meaning that it can be deleted again or modified in any way.

A field is duplicated by selecting its name in the field list and then clicking on the Duplicate button in the Define Attributes dialog:



At this point a box will be displayed allowing you to choose one of the following options:



Duplicate Definition Only

A new field will be defined with characteristics just like the one being duplicated, but it will have an empty value in each of the objects in the design.

Duplicate Data

A new field will be defined and the associated value in each object in the design will also be duplicated and associated with the new field name. Changing the value of the new field in a given object will have no effect on the original field.

One important feature of this Duplicate operation is that any attribute display options, such as “Visible by Default” or “Show with Field Name”, that are turned on before leaving the Define Attributes box will be applied to the duplicated data. This means that a field that was formerly hidden can be duplicated and displayed throughout the entire design with just this one operation.

Merging Two Existing Attribute Fields

The Merge operation takes two existing attribute field definitions and merges them into one. If a given object (device, signal, pin or design) has values in both of the original fields, the one being renamed takes precedence and the other is lost.

NOTE: Pre-defined fields cannot be renamed, so they cannot be merged into other fields. You can however merge a user-defined field into a pre-defined one.

There is no explicit button for the Merge operation. Merge is invoked whenever a field is renamed so that its name matches another field's. The procedure is as follows:

Click on the field to be renamed in the field list. The data in this field will take precedence when the design is updated.

Type the new name (i.e. the name of the field you wish to merge to) into the Field Name text edit box.

Type the Tab key or click on any of the option switches to indicate that you have finished typing.

A box will be displayed asking you to confirm the merge. Click the OK

button:

The actual merging of design data does not take place until you click on the OK button in the Define Attributes box.

NOTE: You can only merge fields that have similar “Allowed in Object Types” settings. E.g. you cannot merge a signal field into a device field.

Delete

The Delete operation removes the selected name from the list of available fields and removes all data values associated with the field throughout the design. Pre-defined fields cannot be deleted.

WARNING: This operation cannot be undone and should be used with care.

To delete a field, simply select its name in the field list and click on the Delete button:

A rectangular button with a light gray background and a thin black border. The word "Delete" is centered on the button in a black, sans-serif font.

Delete

A box will be displayed to confirm the Delete operation: Click OK to proceed.

In any case, no design data is updated until the OK button is pressed on the Define Attributes box.

Temporarily Displaying Attributes

The Duplicate function is a convenient method of displaying data temporarily on the schematic. For example, you may wish to display some simulation parameters on the schematic while tracking a particular design problem, even though you don't want them there for the final printout.

To do this:

Select the Define Attribute Fields command in the Options menu.

Choose the field you wish to display.

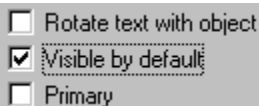
Click on the Duplicate button.

A rectangular button with a light gray background and a thin black border. The word "Duplicate" is centered in a black, sans-serif font.

Click on the Duplicate Data option in the confirmation box.

A rectangular button with a light gray background and a thin black border. The words "Duplicate Data" are centered in a black, sans-serif font.

Enable the “Visible by Default” and (if desired) the “Show Field Name” and “Primary” options.



Click the OK button in the Define Attributes box to complete the operation.

The duplicate field will now be displayed on each object in the design. When the duplicate field is no longer needed, it can be removed with the Delete function described above.

Permanently Showing Data Throughout a Design

The “Visible By Default” option in the Define Attributes box normally has no effect on values that already exist in the design. Enabling the option for a field that is already in use affects only future entries. However, the Duplicate and Merge operations can be used to update the display of existing data throughout the design. This is done as follows:

Select the Define Attribute Fields command in the Options menu.

Select the field to be displayed in the field list.

Click on the Duplicate button:

A rectangular button with a light gray background and a thin black border. The word "Duplicate" is centered in a black, sans-serif font.

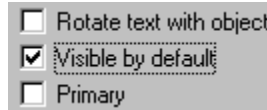
In the confirmation box, click on the Duplicate Data button:

A rectangular button with a light gray background and a thin black border. The words "Duplicate Data" are centered in a black, sans-serif font.

You will now have a copy of the original field: Make sure the newly-created field is selected in the list, then rename it back to the name of the original field. E.g. If the original field was “Value”, the duplicate will be “Value1” and we now select “Value1” and rename it back to “Value”. Press the Tab key when you are finished typing.

The Merge confirmation box will appear. Click the OK button.

Click on the “Visible By Default” option and any other display options desired.



Click the OK button to complete the operation

This operation in effect re-enters all the data into the selected field with the new display options.

Merging Dissimilar Designs

The Merge feature is a useful tool when bringing together designs created with different attribute definitions, or when importing designs from other systems or older versions of DesignWorks.

We strongly recommend reviewing attribute field usage under these circumstances and using the Merge feature to convert all non-standard fields to predefined ones where possible. This will make the design more compatible with the standard report formats provided with DesignWorks and will reduce confusion in future design revisions.

Importing Attribute Definitions

Whenever circuit data is brought into a design, either from a design file, a device library or the clipboard, it is possible that the incoming data contains attribute fields that are not defined in the destination design. This section discusses how DesignWorks deals with this situation.

In general, DesignWorks attempts to ensure that an entry exists in the attribute

definition table for every field used in the design. When new data is imported, field usage in the imported data is compared to the existing table. Normally, no discrepancies are found and this process is invisible. If any mismatch is found, you will be warned and given a chance to add the new definitions to the table automatically.

Pasting from the Clipboard or Placing a Library Device

In these two cases, DesignWorks does the following:

All fields used in the incoming device or circuit scrap are looked up in the design's attribute table.

Any field that doesn't exist in the design is put on a list to be added.

Any field that does exist in the design but is incompatible (i.e. different “object type” or “keep with instance” settings) is renamed by adding an underscore “_” to the name.

A list of the new fields that will be added is displayed in a warning box. If any of these fields are unexpected, or can be converted to standard fields, the names should be noted at this point. Changes in field usage can be made using the Merge and Delete operations mentioned elsewhere in this chapter.

Converting Files from Older Versions

DesignWorks versions prior to 3.0 did not have a standard list of pre-defined fields. For this reason, files converted from older versions should be reviewed to ensure that field names are consistent (i.e. spelling and capitalization variations are eliminated) and that standard names are used wherever possible.

Changes in Standard Fields

In rare cases, variations in the list of pre-defined fields will occur between versions of DesignWorks. This will normally only come up if you have used a pre-release or custom modified version of the package. DesignWorks checks for conflicts as it reads a design file into memory. If a field defined in the file is marked as pre-defined, but does not exist in the internal table, or vice versa, you will be warned when the file reading process is completed. The attribute table is automatically updated to resolve the conflict and resaving the file will update it to the new format.

This chapter provides more information on the advanced features available for making signal connections in DesignWorks. “Creating and Editing Signals” on page 89 covered the basic procedures for connecting pins together by signal line and joining signals by name. This chapter will cover these additional methods of describing signal connections:

- Cross-page Connections and Automatic Page References

- Busses

- Bus Pins on Hierarchical Circuit Blocks

- Using Power and Ground Connector Symbols

- Specifying Power and Ground Connections in Attributes

- Signal Auto-naming

- Signal Token Numbers

- Signal Connectivity Rules

Using Busses

The bussing facility in DesignWorks allows any combination of named signals to be represented by a single line and any subset of these to be brought out through a “breakout” at any point along the bus line.

Properties of Busses

A bus is treated by DesignWorks as a signal with special properties. Thus, bus lines can be drawn and modified on the screen using all the same editing features available for signals. Note these properties of busses:

- Only bus pins on devices can be connected directly to a bus. All other

connections must be made by using a breakout to access the desired internal signals. A breakout is created using the New Breakout command in the Options menu.

You do not need to specify in advance what signals will be contained in a given bus. Any signals that are present in a breakout or bus pin attached to a bus will become part of that bus and can be brought out through another breakout anywhere along the bus.

Any two busses can be joined together, regardless of their internal signals. When two different busses are merged, any signal in either bus becomes available anywhere along the combined bus.

If you select a bus line, then do a Properties command from the Options menu the displayed info box will show a list of the signals currently contained in the bus.

A given signal can be present only in one bus. If you attempt to connect together two signals in different busses, a warning box will be displayed and the connection will be canceled.

A bus can be created by drawing the bus lines first, then creating the breakouts to attach, or by creating a breakout and extending the bus line starting at the bus pin. Bus lines are drawn or extended using exactly the same techniques as for signals, except that the Draw Bus command or cursor is used instead of Draw Signal.

Who Makes the Rules Anyway?

DesignWorks has a set of rules that it applies to determine if a logical connection is made between two signals with the same name in different parts of a design. These rules are outlined in “Signal Connectivity Rules” on page 213. However, these may not be the only rules that apply, depending on where you are going with the schematic data. If you are exporting data via a netlist to another package for layout, simulation or analysis, that software will apply its own rules about connections between nets. Some systems (for example, SPICE-based simulators) may assume that two signals with the same name are connected together, whereas others base connectivity strictly off the structure of the netlist and make no assumptions about names.

The safest course is to ensure that you never use the same signal name twice and to check this using the duplicate signal name checking scripts provided with the package. In any case, it is wise to know the assumptions being made by all the packages that will be receiving your design data.

Creating a Bus

A bus can be created by any one of these methods:

Select the Draw Bus tool (**+**) in the toolbar. Draw any desired contiguous set of lines on the diagram using the usual signal drawing techniques. This bus will have no internal signals initially. Signals will be added implicitly when it is connected to any breakout or bus pin.

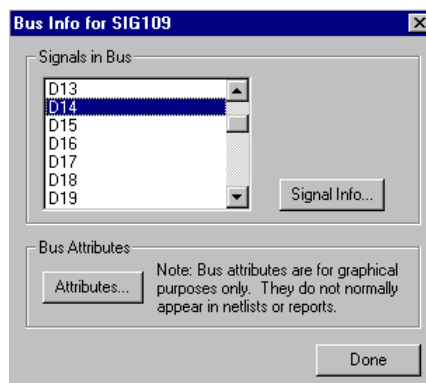
Create a breakout symbol using the New Breakout command discussed on page “Using Bus Breakouts” on page 193. The bus pin (backbone) of the breakout can now be extended using the normal pointer or the Draw Bus cursor. The bus will contain all signals specified in the breakout.



Extend a line out from an existing bus pin on a device using the normal pointer or the Draw Bus cursor. The bus will contain all signals specified in the bus pin on the device. Connections between bus internal pins and bus internal signals can be changed using the Properties command on the bus pin's pop-up menu. See “Using Bus Pins” on page 196 for more information on bus pins.

Getting Bus Information

A list of internal signals can be seen by selecting a bus and using the Properties command, or by right-clicking on the bus and selecting the Bus Info command. This action will display the following box:



Here is a summary of the options presented in the bus information box:

- Signals in Bus** This is a list of the signals contained in the bus. This list is determined by the breakouts and bus pins attached to the bus. You cannot directly change this list. Doubling-clicking on any item in this list is equivalent to clicking once and then clicking the Signal Properties button.
- Bus Attributes** This button displays the general attribute data entry box for the selected bus. NOTE: Except for the Name field, bus attributes are generally not included in any netlist output. More information on the functions available in this box are given in “Entering and Editing Attribute Data - Basic Procedure” on page 160.
- Signal Properties** This button displays the signal properties box for the signal selected in the list. This allows access to signals that may not be visible on the schematic. More information on this box is provided in “Getting and Setting Signal Information” on page 93.

Adding Signals to a Bus

There is no explicit command to add signals to a bus. Signals are added to a bus each time a breakout or device bus pin is connected to the bus. Any signals in the breakout or bus pin are implicitly added to the bus if they don't exist already.

Getting Information on Signals Inside a Bus

Information on signals that are contained in a bus can be accessed in several ways:

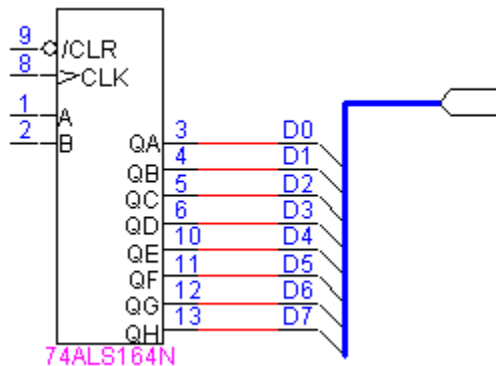
Select the parent bus and select the Properties command in the Options menu, or right-click on the bus and select the Properties command, then use the signal list and Signal Properties button to display the signal info box, as described in “Getting Bus Information” on page 191.

Right click on any bus pin attached to the bus, then choose the signal in the signal list and click the Signal Properties button, as described in “Getting Information on Signals in the Bus” on page 200.

Using Bus Breakouts

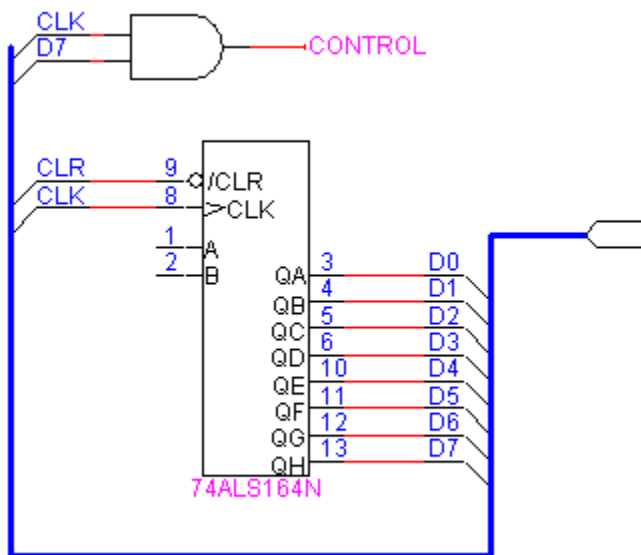
Signals are attached to a bus via a special type of pseudo-device symbol called a “breakout”. It is not legal to attach a signal line directly to a bus line and any attempt to do so will elicit a warning box. In DesignWorks, a breakout is treated as a device with certain special properties. This means that it can be placed in any desired orientation, moved, duplicated, etc. using any of the device editing features available.

A typical breakout is used like this:



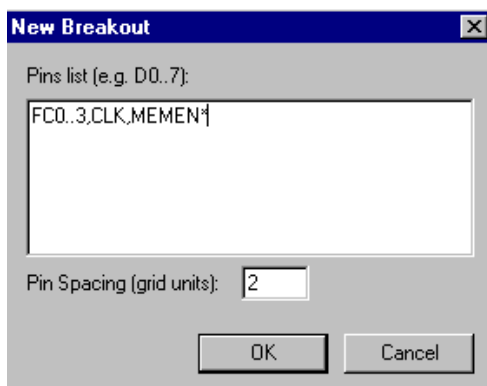
Any breakout can always be attached to any bus. When a breakout is attached that contains signals unknown in that bus, the signals are implicitly added to the bus. For example, if a second breakout was attached to the same bus containing only the signals CLK and CLR, all signals in all attached breakouts would be considered part of that bus. In this case, the list would be D0, D1, D2, D3, D4, D5, D6, D7, CLK and CLR.

Any combination of the internal signals can now be brought out of the bus at any point, as in this addition to the above circuit:



Creating a Breakout

To create a breakout, select New Breakout command in the Options menu, or right-click on the bus that the breakout is to connect to and select the Breakout command. If the new breakout is to be similar to an existing one, first select the similar breakout or the bus to which the new breakout is to be connected. This dialog box will appear:



If a bus or breakout was selected on the circuit diagram then the breakout info box will display a list of the signals in that bus or breakout, otherwise it will be empty. If this list already matches the signals you want in the new breakout, then just click the “OK” button or hit Enter on the keyboard. Otherwise, edit the signal list, noting these options:

blanks or commas can be used to separate individual names in this list, therefore bussed signals cannot have names containing a blank or comma.

a range of numbered signals can be specified using these formats:

`D0..7` or `D0..D7`

is equivalent to

`D0 D1 D2 D3 D4 D5 D6 D7`

`D15..0`

is equivalent to

`D15 D14 D13 D12 D11 D10 D9 D8 ... D0`

`D15..D00`

is equivalent to

`D15 D14 D13 D12 D11 D10 D09 D08 D07 ... D00`

Note that the “..” format implies that bussed signal names cannot contain periods.

the signals specified will always appear in the order given in this list from top to bottom in standard orientation. We recommend always specifying numbered signals from lowest numbered to highest, as in the first example above, since this matches the standard library symbols.

there is no fixed limit on the number of signals in a bus, but we recommend dividing busses up by function (i.e. address, data, control, etc.) for ease of editing.

any combination of randomly-named signals can be included in the list, as in these examples:

`D0..15 AS* UDS* LDS*`

`CLK FC0..3 MEMOP BRQ0..2`

Once the list has been entered, click on the OK button or hit the Enter key. An image of the breakout will now follow your mouse movements and can be placed and connected just like any other type of device.

Setting Breakout Pin Spacing

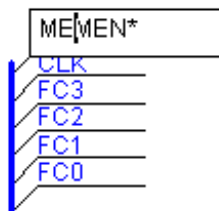
The number in the Pin Spacing box will be the spacing between signal pins on the breakout symbol, in grid units. The default value is 2 to match the standard DesignWorks libraries, but any number from 1 to 100 can be entered.

Editing Breakout Pins

The signal name notation that appears on a breakout pin is actually a pin attribute. It can therefore be edited by the usual attribute editing mechanisms, i.e. either:

Select the pin and choose the Properties command in the Options menu, then click the Attributes button, OR,

Click the text cursor directly in the text on the schematic, as follows:



Type the desired new name.

Press the Enter key. The breakout pin and the attached signal will be renamed as entered.

IMPORTANT: The notation on the breakout pin is always the same as the name of the attached signal name. Changing the breakout pin renames the attached signal and will detach it from any like-named signals already in the bus.

Using Bus Pins

DesignWorks supports user-created bus pins on devices. A bus pin can be defined to have any collection of named internal pins. Note these properties of bus pins:

The bus pin itself does not represent a physical device pin. It is only a graphical place-holder on the schematic representing a group of internal

pins. The bus pin itself never appears in a netlist.

The internal pins represent physical device pins. Even though they do not appear on the schematic, they can have all the same parameters as normal devices pins, including pin numbers and attributes. These parameters can be accessed using the Bus Pin Options command in the pin pop-up menu.

When a device with a bus pin is placed, it has a pre-created bus attached to it by default. This bus will contain one signal for each internal pin, with the initial name of the signal being the same as the name as the pin.

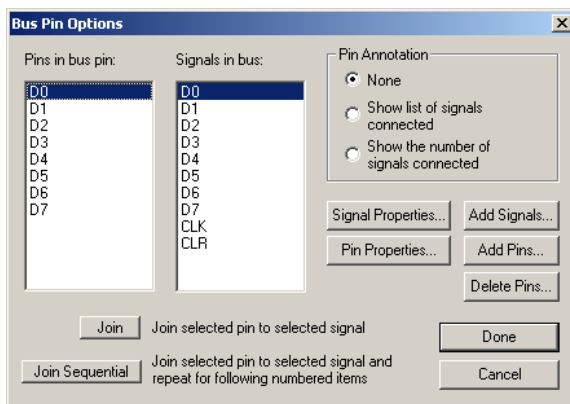
A “splicing” box can be displayed using the Bus Pin Options command in the pin pop-up menu. This box allows any internal pin to be connected to any signal in the attached bus.

For more information on creating device symbols with bus pins, see “Placing a Bus Pin” on page 295.

Changing Bus Pin Connections

When a bus is connected to a bus pin on a device or hierarchical block, the bus internal pins will by default connect to signals with the same name in the bus. To change these default connections, use the Bus Pin Options command in the pin pop-up menu.

This menu item will be enabled only when a bus pin on a device is selected. It allows the association between the bus internal pins on the device and the signals in the bus to be changed. This box will be displayed:



The left-hand list shows the names of the pins contained in the selected bus pin. The right-hand list shows all the signals in the attached bus. For each pin in the pin list, the signal in the same row in the signal list is the one attached to

it. Signals in the signal list beyond the end of the pin list are not connected in this bus pin. The following sections describe the operations available in this box.

Changing Signal Connections in a Bus Pin

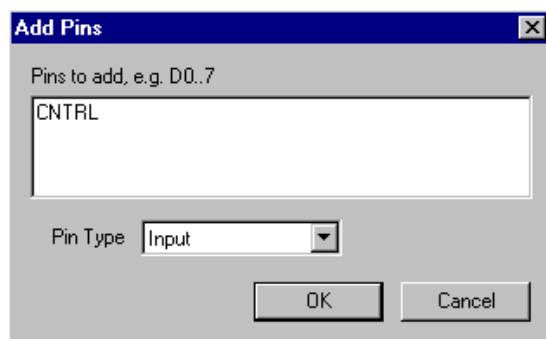
Two buttons are provided to change the association between pins and signals. The Join button causes the selected pin in the pin list to be joined to the selected signal in the signal list. If the selected signal is already attached to another pin in the list, then the signals will be swapped (i.e. a signal can only connect to one pin and vice versa). The signal list will be updated to show the new relationship.

The Join Sequential button provides a quick method of joining multiple numbered pins and signals. The selected pin is joined to the selected signal, as with Join, above. If the signal and pin names both have a numeric part, both numbers are incremented and the corresponding signal and pin are joined. This process is repeated until either the signal or pin name is not found in the list.

For example, given the lists appearing in the above picture, if pin D0 and signal D4 are selected, then Join Selected will join D0-D4, D1-D5, D2-D6 and D3-D7. Since there are no more numbered pins, the process would stop. Note that although the signal and pin names are the same in this example, this is not a requirement.

Adding Pins to the Bus Pin

The Add Pins button allows you to add internal pins to the selected bus pin. When this option is clicked, the following box will be displayed:



NOTE: This operation effectively modifies the definition of the symbol and any subcircuit it may contain, making it different from its original library definition. The pin changes are not permanent until the Bus Pin Options box is closed.

A list of pins can be typed into this box using the same format as the New Breakout command described in “Creating a Breakout” on page 194. A pin type can be selected and will apply to all pins created using this operation. If no pin type is needed in your application, select the default Input type.

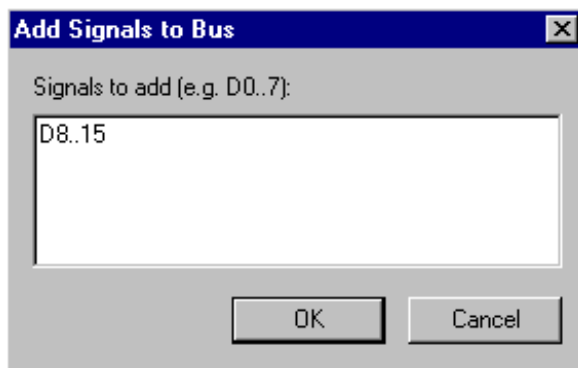
Deleting Pins from the Bus Pin

Clicking the Delete Pins button will remove all internal pins currently selected in the list from the bus pin. This operation does not become permanent until you close the Bus Pin Options box.

NOTE: This operation effectively modifies the definition of the symbol and any subcircuit it may contain, making it different from its original library definition.

Adding Signals to the Bus

The Add Bus Sigs button allows you to add signals to the signal list so that they can then be joined to device pins. Clicking this button displays this box:



A list of signals can be typed into this box using the same format as the New Breakout command described in “Creating a Breakout” on page 194. Following are examples of allowable formats:

```
D0..7
D0..15 AS* UDS* LDS*
CLK FC0..3 MEMOP BRQ0..2
```

The order of entry will affect the order that the signals appear in the list, but is

otherwise not significant.

NOTE: These signals are only added temporarily. When you close the Bus Pin Options box, all signals that are not connected to any pin are removed from the bus.

Getting Pin Information on Internal Pins

The Pin Properties button brings up the standard Pin Properties box for the pin selected in the pin list. See the Properties command for more information.

Getting Information on Signals in the Bus

The Signal Properties button brings up the standard signal info box for the signal selected in the signal list. This box is described in “Getting and Setting Signal Information” on page 93.

Displaying a Bus Pin Annotation

If this option is enabled, a list of the connections made in the bus pin will be displayed adjacent to the pin. This is done by creating a value in the BusInfo attribute field for the selected bus pin. This value can be edited manually, if desired, but will be updated automatically each time this box is displayed. The format of the signal list is the format used by the Add Bus Sigs option, above.

Inter-page Connections

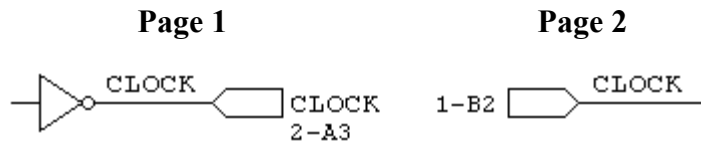
Signals drawn on different pages will initially have no logical connection, even if the names are the same (see the rules “Signal Connectivity Rules” on page 213). Inter-page connections are made using the Page Connector pseudo-device which is in the Pseudo Devs library. When attached to a signal line, the Page Connector makes the name of that signal known across all pages, i.e. any like-named signal on another page which also has a page connector will be logically connected to this one. The Page Connector can be placed anywhere along the signal line although it is normally convenient to place it at the edge of the page. The Page Connector itself can be named (in order to take advantage of the automatic page references mechanism described below), but it must have the same name as the attached signal.

Changing the name of the Page Connector (using the naming procedure for devices) will also change the name of the attached signal. Likewise, changing the name of the signal will rename any attached Page Connectors.

Automatic Display of Page References

DesignWorks has an Automatic Page Reference facility which displays the page number and position of any Page Connectors associated with a given signal. These page references are automatically displayed immediately adjacent to any Page Connector as soon as multiple Page Connectors are attached to the same signal.

For example, in this next case both pages have signals named CLOCK with Page Connectors attached, so the two CLOCK signals are logically connected. Note that both Page Connectors are notated with the page number and grid reference:



When a name is applied to the Page Connector itself, the page reference will appear under the name.

As long as automatic updating is enabled, these references will be updated automatically when any Page Connectors or attached signals are added, deleted, moved or renamed.

Enabling and Disabling Automatic Page References

Automatic page reference updating is controlled by the Design Preferences command in the Drawing menu. Automatic references are enabled whenever the Automatic Page References box is checked. This means that page reference text will automatically be placed next to any named Page Connector that is placed in the design, and will be updated whenever any page connector is moved. If they are disabled, the current page reference settings will be left untouched when any schematic editing is done.

Manually Updating Page References

If automatic page references are disabled, page references will be updated

only when the Update Now button in the Design Preferences box is clicked. This can be used for large designs where page reference updating may cause delays while editing.

Setting Page Reference Format

To change the page reference format, select the Design Preferences command in the Drawing menu. Two aspects of the page reference format can be controlled:

The “Format” text item controls the appearance of each page reference in the reference list.

The “Max. Width” item controls the number of references that will appear on each line before creating a new line. For signals with many connections, this prevents unwieldy page reference strings.

Three characters are special in the format string:

- P** will be replaced by the page number
- X** will be replaced by the X grid position
- Y** will be replaced by the Y grid position

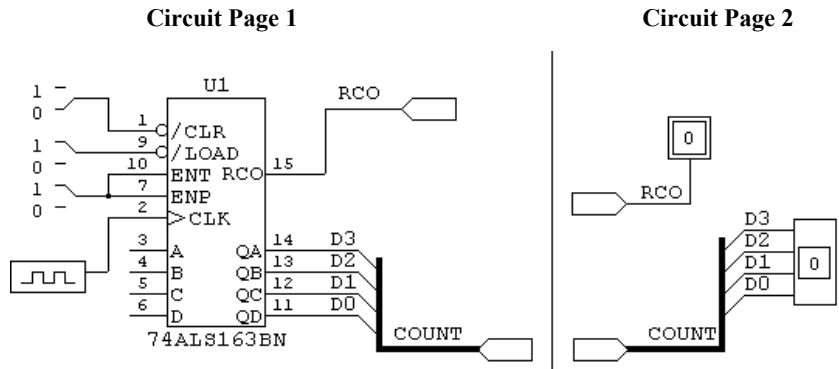
The default format is “P-XY” which generates pages references like “2-C4” (i.e. page 2, grid position C4). All characters other than the special ones above will be placed in the page reference list verbatim.

The maximum length of a format string is 16 characters.

Connecting Busses Across Pages

Busses can also be connected between pages by the same method, except that the Bus Page Connector must be used in place of the Page Connector. This is

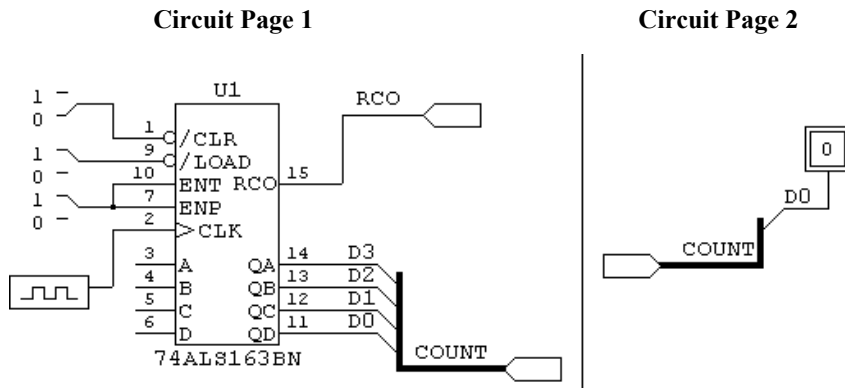
illustrated in this simple circuit:



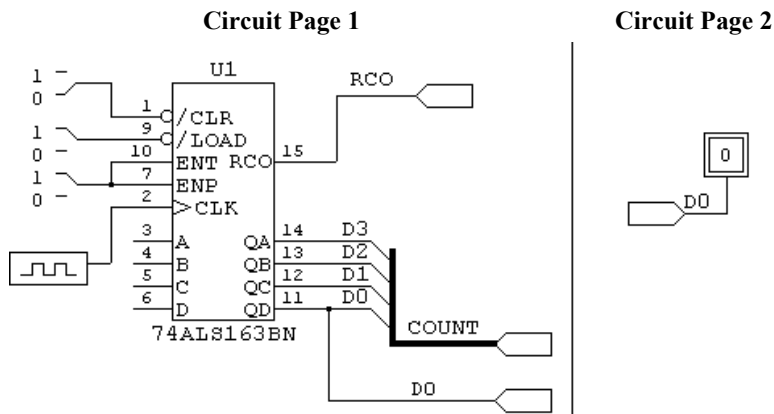
Using Page Connectors on Internal Bus Signals

Note that a Bus Page Connector does not make the names of the internal signals known globally, only the bus itself. To bring an individual internal bus signal across to another page, one of the following methods can be used:

1) Make the entire bus global using a bus page connector, then use a breakout to access the desired signal, as in this example:



2) Place signal page connectors on the individual signal on both pages, as shown here:



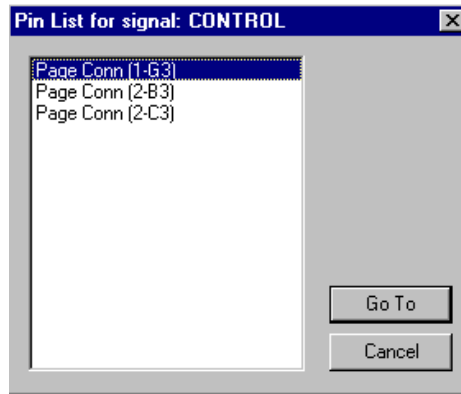
Changing the Page Connector Symbol

Note that the symbol for the Page Connector device can be changed using the device symbol editor tool just as with any other device. In order that the resulting symbol be recognized by the program as a Page Connector, you must either start with an existing Page Connector symbol, or set the primitive type setting appropriately.

See “Creating a Page Connector” on page 311 for specific information on creating Page Connector symbols.

Tracing Connections Through Page Connectors

If a Page Connector device is selected in the schematic, the Properties command displays this box:



This is essentially the same box as is displayed for the signal Pin List command, except that only Page Connectors are listed. Following on each item is the page number and grid reference of the item.

To display the selected page connector, either:

select the item in the list and click the Go To button,

OR

double-click on the item in the list.

Power and Ground Connections

When a schematic diagram is created, the power and ground connections for devices are normally not drawn as signal connections on the diagram. These connections would clutter the diagram and are not necessary for an understanding of the logical function of the circuit. Obviously, though, these connections must be included at some level in order to form a complete netlist. Three methods are available to do this:

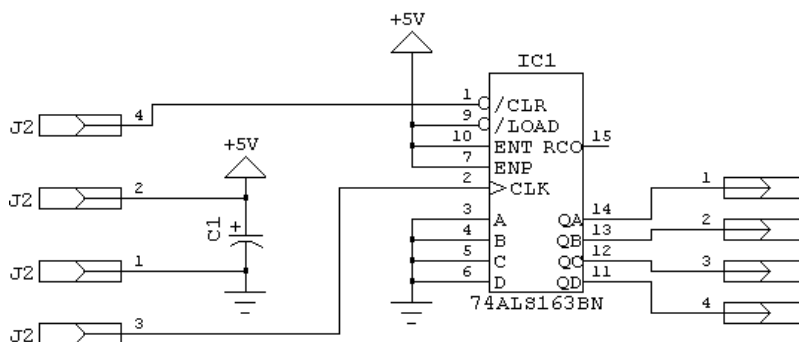
Power and ground pin connections can be specified as attributes for each

device. The Schematic tool itself attaches no significance to the attribute fields, but the report generator reads them and adds the specified pins to the netlist. The libraries of standard types supplied with DesignWorks include this information for each type. It is only necessary for you to enter this data if you are making non-standard connections or using a device that has no default power connections. The pre-defined attribute fields “Power” and “Ground” are intended for this purpose.

Power and ground pins can be added to each device symbol (using the device symbol editor tool) and the connections shown as normal signal lines on the diagram, or connected using Signal Connector (power and ground) symbols. The standard DesignWorks libraries are not set up in this fashion. This method is normally used only for unusual design situations such as an isolated supply for a specific part in a design.

A separate symbol can be created for each device showing only power and ground connections. This allows all power wiring to be shown in one part of the diagram or on a separate page.

In fact, these methods are not mutually exclusive and can be used in combination as needed. This will be illustrated by way of a simple example:



The attributes for the device IC1 were specified in the library entry for type “163” and are as follows:

Power	14
Ground	8

In the above diagram, the signal named “+5V”, the symbols labeled “+5V” and the attribute entries “Power” will all be merged into a single list of connections in the netlist, as follows:

```

S00005IC1-14J1-1
S00007IC1-2J2-3
S00008IC1-1J2-4
S00009IC1-11J1-4
S00010IC1-12J1-3
S00011IC1-13J1-2
GroundC1-2IC1-3,4,5,6,8J2-1
+5V C1-1IC1-10,7,9,16J2-2

```

NOTE: The standard report form files add the entries in the Power field to a net named “+5V”.

The demonstration files provided with DesignWorks provides more complex examples of this usage.

Power and Ground Naming Convention

The standard libraries provided with DesignWorks include the fields Power and Ground for the standard power and ground pins. Since field names cannot contain special characters, we recommend using the words “Plus” and “Minus” to create attribute fields for other supplies. For example, pins to be attached to a -12V supply would be listed in field “Minus12V”. The Report Generator allows the name of the signal net to be different from the field name, if desired.

The supply and ground signal connectors can be found in the Connectors library. The standard device libraries include entries for Power and Ground and other standard power pins. As in the above example, other pins can be attached to these lines using symbols placed on the diagram.

See “Extracting Power and Ground Connections from Attributes” on page 358 for instructions on adding other power nets.

Power and Ground Connections in Attributes

DesignWorks allows any number of special attribute field names to be specified as “signal sources”. When a report is generated, the Export tool searches the attributes attached to each device for fields with these names. Any pin numbers specified in the field will be attached to a list for the signal of the same name. E.g. the value “14” in the Power field will cause pin 14 on this device to be attached to the signal named as the power signal. Multiple pins can be specified using commas, e.g. “1,2,14”.

NOTE: Power and ground connections made through attributes do not appear as

logical connections during schematic editing operations. The merging of these pins into nets occurs only when generating netlist output and exists only in the netlist. These connections will not appear in the Pin List command and will not be simulated by the DesignWorks Simulator option.

IMPORTANT: The use of signal sources in netlist output is completely dependent on the netlist generating script. DesignWorks does not perform this function automatically if it is not specified in the script. Whenever appropriate, the settings for the standard power and ground fields described here have been included with the standard report scripts provided with DesignWorks. However, this can be modified by users and depends on details of the destination system.

You can find more information on the use of power and ground fields in the netlist format you are using in these locations:

The ReadMe file provided with the Design Kit you are using.

The Format Notes built in to the netlist script. For more information on how to see these, see “Viewing Format Notes” on page 355.

“Extracting Power and Ground Connections from Attributes” on page 358 provides more information on specifying signal sources in a report script.

Signal Connector (Power and Ground) Symbols

DesignWorks uses a type of pseudo-device symbol called a “signal connector” to maintain connectivity between like-named power and ground symbols that are used on circuit diagrams.

As soon as a Ground symbol is placed on the diagram, the attached signal will be named “Ground” (the name will be invisible initially). This will cause it to be connected by name to any other signals having Ground symbols, or explicitly named “Ground”.

Connectivity can be checked at any time by double-clicking on any ground or power line. This will highlight all other like-named lines on the diagram.

IMPORTANT: Signal connectors do not cause a logical connection to be made between hierarchy levels in a hierarchical design. These signals can be merged into a single net for netlisting purposes by specifying the net in question as a signal source. However, this processing is done as the report is generated and does not create a logical connection in the design, e.g. for interactive simulation.

See “Making Connections Across Hierarchy Levels” on page 236 for more information on signal connections in hierarchical designs.

Using Signal Connector Devices

Signal Connector Devices are placed on the diagram just like any other DesignWorks device. A set of standard power-supply symbols are included with DesignWorks in the Connectors library. If you connect two different signal connector devices together, you will be prompted to provide a name for the resulting signal.

Creating Signal Connectors in a Library

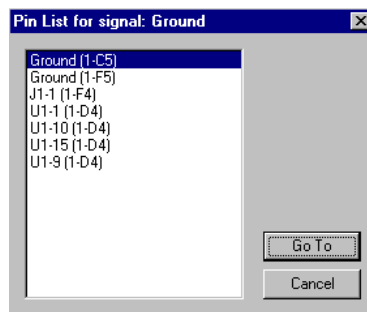
Signal Connectors (like Page Connectors) are special primitive “pseudo-devices” in DesignWorks and can be created using the Subcircuit and Part Type command in the symbol editor to select the SIGCONN primitive type.

NOTE: The signal attached to a signal connector device is actually named to match the pin name of the signal connector pin specified in device symbol editor, not the type name. In most of the power and ground symbols provided with DesignWorks, these two names are the same. However it is possible to create a symbol called, for example, “Ground” in the library which actually names the attached signal “GND”. The Ground symbol in the SPICE Devices library provided is an example of this in that it names the attached signal “0” to match the SPICE ground naming convention.

See “Creating a Power and Ground (Signal) Connector” on page 308 for more detailed information on this procedure.

Tracing Connections Made by a Signal Connector

If a Signal Connector device is selected in the schematic, the Properties command displays this box:



This is the same box as is displayed for the signal Pin List command. Following on each item is the page number and grid reference of the item.

To display the selected pin, either:

Select the item in the list and click the Go To button,

OR

Double-click on the item in the list.

Using Signal Auto-Naming

DesignWorks has an automatic name assignment feature that ensures that every signal object that is created has a distinct name. This ensures that it is always possible to track items in a netlist or error report back to a signal on the schematic. Signal auto-naming is enabled by default when a new design is created, unless the selected template has specifically disabled it.

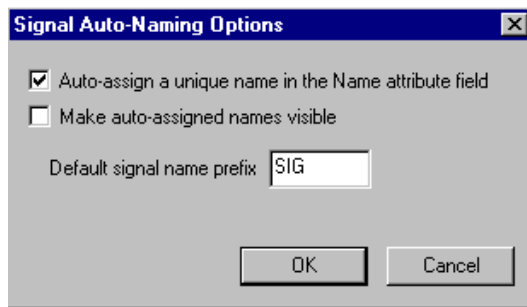
When Auto-Naming is enabled, all signals will be given a default name when they are created. In addition, whenever an editing operation causes a single signal to be broken into two, or part of a signal to be copied without a visible name label, a new, unique name will be auto-assigned. This ensures that every object in a design has a unique name and can be identified in a netlist.

NOTE: The program generates new names using a specified prefix and the signal's token number, which will produce a unique name in most cases. However, the program does not check for uniqueness. If you have manually assigned a name to another signal that matches the auto-assigned format, there is the possibility that an auto-assigned name may already exist. See "How Names are Generated" on page 211. To guarantee name uniqueness, you should run a Duplicate Signal Name error checking script. See "Locating Found Objects on the Schematic" on page 247 for more information on error checking.

Enabling Auto-Naming

Auto-Naming is enabled by selecting the Signal Naming Options command in the Naming and Packaging Options sub-menu in the Options menu. This will

display the following options box:



Checking the “auto-assign” box causes all devices placed in the design subsequently to be given a default name if they are not already named.

If the “Visible” option is selected, the default name will be displayed on the diagram adjacent to the device or signal when it is created.

The “default signal name prefix” text box allows you to specify the characters to be used as a basis for auto-generated names. Changing this value does not affect existing names.

NOTE: Enabling signal auto-naming does not assign names to existing signals that do not already have them. This must be done manually or using a script.

Disabling Signal Auto-naming

Signal auto-naming can be disabled by taking these steps:

Select the Signal Naming Options command in the Naming and Packaging Options sub-menu in the Options menu.

Turn off the Auto-assign box.

WARNING: We do not recommend disabling auto-naming. When signal auto-naming is disabled, no modifications are made by the program to existing signal names. For example, if you split an existing signal into two parts, both parts will retain the existing name until you explicitly change it. This may result in an apparent short in a netlist.

How Names are Generated

The auto-generated name consists of two parts, the fixed prefix and the numeric suffix. The prefix portion is derived from the signal name prefix set

using the Design Preferences command (which is stored in the design's Sig-Prefix attribute field).

The numeric portion of the name is generated using a table of signal names used in the circuit. This value is guaranteed to be unique within a circuit level. This table is not updated immediately after every editing operation and so doesn't guarantee to fill in all unused numeric values. Thus, in a design that has been edited, sequential numbering is not guaranteed.

Using Signal Token Values

Every time a device or signal is created in a DesignWorks circuit, it is assigned an integer value known as its “token”. The token number stays with the device or signal for its lifetime and numbers are not re-used. This ensures that a given device or signal can always be recognized despite duplicate names or name changes. The token is used for a number of internal operations in DesignWorks, but can also be seen by the user in the following circumstances:

The token number is used to generate default names for devices and signals, as described elsewhere in this chapter.

The token number can be written out in netlists or bills of materials whenever a guaranteed-unique identifier is needed.

Note these characteristics of tokens:

Tokens are assigned independently for each circuit in a hierarchical design and are thus only unique within a circuit, not across the entire design.

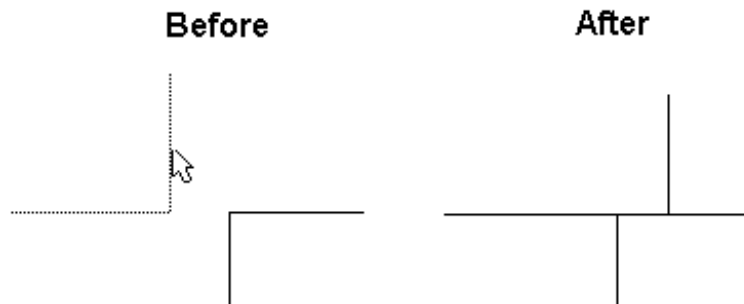
When two signals are joined, one of the two tokens is discarded at random.

Each logical symbol on the diagram (including pseudo-devices) has its own token. In a netlist, several symbols may be combined into a single package, so there is not necessarily a one-to-one correspondence between tokens and physical packages.

Signal Connectivity Rules

This section provides a complete reference for all the methods used to inter-connect signals. Each of these items is covered in more detail elsewhere in this manual.

DesignWorks automatically checks for a connection whenever the endpoint of a signal segment is moved. In most circumstances, any two lines that appear graphically connected are also logically connected. However, there are some editing operations that can cause two lines to overlap without creating a logical connection, for example:



In this case, the corner of one signal overlaps another, but the free ends never touch. Since the program only checks free ends for connections, no connection is made between these items. This can be seen visually in this case by the lack of a connection dot at the intersection. In addition, clicking on either of the signals would highlight only the lines belonging to that signal.

Signal names must be visible to be checked for connections, unless a Signal Connector device (e.g. Ground) is attached. If you use an attribute editing command to make a formerly visible name invisible, any connections caused by that name will be broken.

Signal names are known throughout a single page. Like-named signal traces on a single page are thus logically connected for simulation and netlisting purposes. Whenever a signal name is added or changed, the circuit is checked for a change in connectivity. If the name is now the same as another signal on this page, the two signals are merged into one. If this signal segment was previously connected by name to others and the

name is changed, then the logical connection is broken. Whenever a name change causes two signals to be connected, both parts will flash on the screen to confirm the connection.

Signal names are global across all pages of the circuit when a page connector symbol is added to the signal line. Thus if the name is changed on a signal line having a page connector then all circuit pages are checked for like-named signals having page connectors. If any such signals are found then a logical connection is made between them.

Signals which are contained in busses are a special case. All signals contained in busses have a name, even if this is not displayed on the diagram. However, the names of bussed signals will not be used to make logical connections unless an explicit name label has been added to the signal line.

For example, if you have a bus containing a signal named CLK and a separate signal line also named CLK, there will be no logical connection between these two signals. The name appearing on the bus breakout is part of the breakout symbol and is not considered to be a name label. If an explicit label is added to the bussed CLK signal (using the Name cursor) then the two CLKs will be logically connected.

The same rules discussed above for signals also apply to busses. Whenever two busses are logically connected, all like-named internal signals also become logically connected. Note that connecting busses across pages requires a “Bus Page Connector” device rather than a “Page Connector” to avoid compatibility problems between bus and signal connections.

This chapter provides background and detailed procedures for the hierarchical design features of DesignWorks.

General Concepts

What is Hierarchy?

“Hierarchy” refers to the ability to have a “device” symbol in a schematic actually represent an arbitrary circuit block. The “pins” on the device symbol represent connections to specific input-output points on the internal circuit. For clarity, a device symbol that represents an internal circuit will be called a “hierarchical block”.

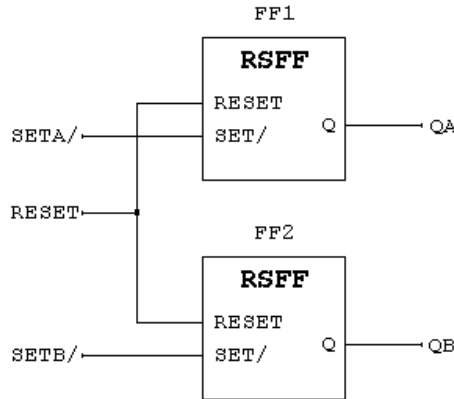
Hierarchical design provides a powerful way of representing complex designs in compact and readable form. A top-level diagram of your system can show only major functional blocks. These blocks can then be opened to show more and more design detail.

Hierarchical design in effect adds a “third dimension” to a schematic diagram. It also raises some complex issues that should be understood before embarking on a major design. Please review this chapter carefully before making extensive use of the hierarchical features of DesignWorks.

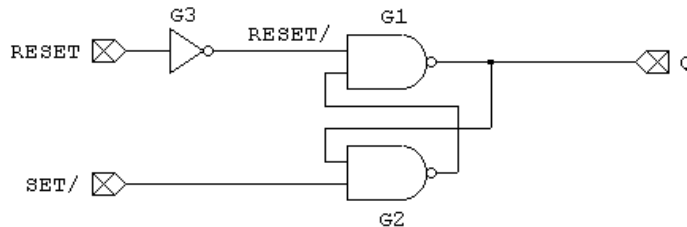
A Simple Hierarchy Example

The following diagram is the master circuit, or top level, of our design exam-

ple:



Note that it contains two symbols, both representing hierarchical blocks. Both symbols are of the same type, “RSFF”, and therefore share the same internal circuit definition. The two blocks are named “FF1” and “FF2”. Opening either one of these blocks reveals the following internal circuit:



This circuit consists of three device symbols, “G1”, “G2”, and “G3”, representing physical devices, and a number of “port connector” symbols. The port connectors define the interface between the internal circuit and the pins on the symbol representing it.

Note the following characteristics of this simple design:

The block “RSFF” has been used twice, so there are actually two G1s, one inside FF1 and one inside FF2. We say that there are two *instances* of G1. Similarly for G2 and G3.

The signals “SET/”, “RESET” and “Q” in the internal circuit will actually get absorbed into the attached signals in the parent circuit because they are attached to port connectors. They do not exist independently in the physical circuit.

Signal “RESET/” does not connect to a port connector, so it represents a separate signal in the internal circuit. Like the devices G1, etc., each signal in RSFF actually represents two physical signals.

These characteristics of a hierarchical design raise the following issues:

Device and signal names inside hierarchical blocks are not unique if the block has been used more than once. Therefore, a mechanism is needed to create a unique name for use in a netlist or bill of materials.

Signals attached to port connectors get absorbed into the signal attached to the parent symbol's pin. If this too is attached to a port connector, the process repeats upward until a unique signal is found. The name that will appear in the netlist will be the name of the highest-level signal.

A single symbol in the definition of an internal circuit can actually represent two or more physical devices. A mechanism must be provided to store separate information for each of these physical devices, e.g. for PCB placement or layout information.

When transferring data to PCB or simulation systems, it is usually necessary to produce a “flattened” netlist. I.e. a netlist representing the same circuit as if it had been expanded out on a single circuit level.

These issues are addressed in the following sections.

Definition vs. Instance

In DesignWorks, circuit data for hierarchical blocks is separated into two groups:

Definition data is all information about the structure and connectivity of a circuit, plus the contents of all attributes not marked as “instance” fields. Definition data is associated with the parent symbol and will therefore be identical for all usages of the parent symbol.

Instance data refers to data kept with each physical instance of a device. For example, in the simple design used in the previous section, gate G1 will have two sets of instance data associated with it, one for each physical device that it represents. Instance data consists of pin numbers, interactive simulation data and any attribute fields marked as “instance” fields.

In a hierarchical design, only one set of definition data is kept for each hierarchical block type, whereas separate instance data is kept for each instance.

More information on defining and using instance attribute fields is given in “Definition vs. Instance Fields” on page 159.

Choosing a Hierarchy Mode

DesignWorks implements a true hierarchical design structure. This means that “device” symbols appearing in a circuit can actually represent another, nested circuit. This subcircuit can be opened and edited at any time in separate circuit window. The term “design” is used in this manual to refer to a complete logical system, including the top level circuit and all subcircuits it contains.

DesignWorks has three hierarchy modes to address different design situations. These are described in the following sections.

Flat Hierarchy Mode

“Flat” Mode is the simplest mode and is intended for most PCB-related designs or any smaller designs. Despite the name, it is still possible to create and edit internal circuits, but they are assumed to be for simulation or analysis only. Devices in subcircuits will not be assigned to physical packages by the Packager tool and will not appear in any netlist output.

Physical Hierarchy Mode

Physical mode allows full use of hierarchical design, while still allowing data associated with individual device instances to be stored at any point in a design. It is intended for larger board-level or FPGA designs where interactive simulation is used or PCB layout information is to be kept with the design.

Pure Hierarchy Mode

Pure mode stores only definition data for each device in a design. The “Keep With Instance” setting in attribute fields will be ignored. This allows very large designs to be represented in a small amount of memory, but does not allow interactive simulation or physical layout data to be associated with the design. It is intended for VLSI or other large-scale design applications.

Setting the Hierarchy Mode

The choice of hierarchy mode is not carved in stone once made, since you can switch modes at any time. “Flat” and “Physical” modes are very similar and switching between them will normally only require that gate packaging be redone. “Pure” mode is substantially different, however, and should only be used after reading the discussion later in this manual.

WARNING: Switching to Pure hierarchy mode from any other mode will cause all instance data to be lost. THIS CANNOT BE UNDONE!!! See the following section.

To set the hierarchy mode for a design, select the Hierarchy Wizard command in the Drawing menu.

Effect of Changing Hierarchy Mode

The following table summarizes the effects of switching between hierarchy modes.

Flat -> Physical	<ul style="list-style-type: none">• No data is lost• Attribute field usage for device packaging changes. Packaging will have to be redone
Flat -> Pure	<ul style="list-style-type: none">• Any instance data associated with internal circuits will be lost
Physical -> Flat	<ul style="list-style-type: none">• No data is lost• Attribute field usage for device packaging changes. Packaging will have to be redone
Physical -> Pure	<ul style="list-style-type: none">• Any instance data associated with internal circuits will be lost
Pure -> Flat	<ul style="list-style-type: none">• No data is lost
Pure -> Physical	<ul style="list-style-type: none">• No data is lost



Navigating in Hierarchical Designs

Hierarchy adds a third dimension to a design that requires some additional commands to allow you to move between levels. Here are some of the techniques available in DesignWorks for navigating around a hierarchical design:

The Push Into command (or simply double-clicking on a subcircuit block)

opens the subcircuit in a new circuit window. The Pop Up command performs the converse operation, closing the current subcircuit window and displaying its parent device symbol. These commands are described in the following sections.

Any utility tool (like Browser or Find) that can display a found object will open a subcircuit containing the object. These tools are described elsewhere in this manual.

Opening (Pushing Into) a Subcircuit

The Push Into command opens the internal circuit of the given device in a separate window.

NOTE: The Push Into command is available in the Options menu as well as in the device pop-up menu, i.e. by right-clicking on the device.

In Physical Hierarchy mode, if you have used the same device type in multiple places in the design, the Push Into command creates a temporary type which is distinct from all other usages. When the subcircuit is closed you will be asked if you wish to update the other devices of the same type.

This menu item will be disabled (grayed out) under any of the following conditions:

- The device is not a SUBCCT (subcircuit) primitive type

- The device has its “restrict open” switch set in the Properties box

If the selected device has no internal circuit, you will be asked whether you wish to create one. Clicking OK will create a circuit containing only the default port connectors matching the parent device.

Simply double-clicking on a device is a short-cut for the Push Into command.

Closing (Popping Out of) a Subcircuit

The Pop Up command closes the current subcircuit and displays the circuit page containing the parent device.

TIP: The Pop Up command is also available in the circuit pop-up menu. That is, right click anywhere in a circuit window not near any circuit objects.

NOTE: Clicking in the close box at the upper left corner of the window is equivalent

to the Pop Up command, except that no attempt is made to display the parent device's window.

Locking and Unlocking Subcircuits

A subcircuit can be locked to prevent accidental opening. This is intended for cases where the subcircuit is derived from a library or is used only to implement a simulation model which should not be edited as part of the parent design.

To lock and unlock a subcircuit from the schematic:

Locate the subcircuit's parent device symbol in the schematic and select it.

Select the Properties command in the Options menu.

Check or uncheck the Lock Opening Subcircuit box, as desired.

Click OK.

TIP: The default locked status can be set when creating a hierarchical block symbol by setting the value of the Restrict attribute field while editing the symbol using the device symbol editor tool. The allowable values for this field are given in Appendix A—Predefined Attribute Fields on page 361.

Creating a Hierarchical Block - Top Down

In concept, a hierarchical design can be created from the top down or the bottom up. In practice, most design processes probably use a mixture of the two methods, depending on the order in which system components were designed.

Working top down means creating the highest level of the design first and then working downward to more detailed levels. In practice, this means that you create and place the symbol for a hierarchy block first, before you necessarily even know what the internals of the block will look like. Once you have completely created the higher-level circuit including the symbols for all the hierarchy blocks it uses, you then proceed to define the internal circuits of the blocks.

input/output pin definition

To create a hierarchical block from the top down, the following steps must be followed:


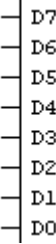
- 1) Create the block symbol using the Create Subcircuit Block command or the symbol editor.
- 2) Place the symbol on the schematic.
- 3) Create the internal circuit by pushing into the new symbol.

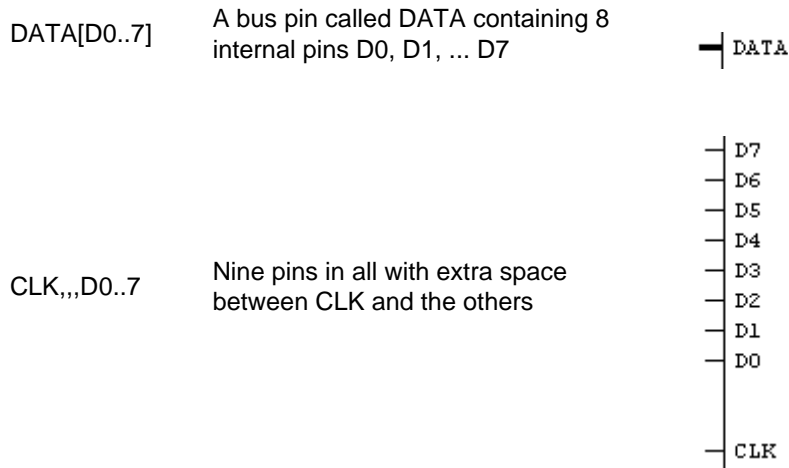
These steps are described in the following sections.

Creating a Block Symbol

Symbols for hierarchical blocks can be created using the Create Subcircuit Block command in the Options menu, or using symbol editor tool. The device symbol editor provides the most flexibility in creating whatever type of symbol you want, but is more involved and requires familiarity with that tool. For this example we will use the Create Subcircuit Block command.

- ➔ Select the Create Subcircuit Block command in the Subcircuit submenu of the Options menu.
- ➔ Select the last option - “None Create an Empty Block Symbol” and click Next
- ➔ In the “Left Pin Names” text box enter a list of pins to appear on the left-hand side of the symbol. Following are some examples of pin lists:

Pin List	Pin Description	Symbol
A B C	Three signal pins A, B and C	
D0..7	Eight signal pins D0, D1, D2, ... D7	



See “Adding Sequential Pin Names” on page 283 for more information.

- ➔ In the “Right Pin Names” text box enter a list of pins to appear on the right-hand side of the symbol, using the same format as above.
- ➔ Click the “Next” button.
- ➔ Select the first name option “Display the name using the Part attribute”
- ➔ Enter a suitable name in the name box and click Next.
- ➔ If you do not already have a suitable temporary library file open, create or open one now using the New Lib or Open Lib buttons.
- ➔ Save the new part into the temporary library by clicking Finish.

Placing the Block Symbol

Hierarchical block symbols are placed just like any other device symbol:

- ➔ In the Parts palette, locate the item you just saved.

Hierarchical Block Primitive Type

Hierarchical block symbols are simply device symbols which have the primitive type “SUBCCT”, or subcircuit. Device symbols with any other primitive type cannot be used as hierarchical blocks. SUBCCT is the default primitive type when creating symbols with device symbol editor, so it is normally not necessary to change this setting.

- ➔ Drag and drop the symbol where desired on the schematic.

Auto-Creating the Internal Circuit

In this section we will take advantage of the fact that DesignWorks creates a default internal circuit template when you open an empty block symbol. It is also possible to create your own internal circuit from scratch. This is discussed below in the section on “bottom-up” block creation.

- ➔ Right-click on the hierarchical block symbol.
- ➔ Select the Push Into command.
- ➔ A box will appear asking you to confirm that you wish to create a new internal circuit. Click the OK button.

A new circuit window will now open with the default port connectors laid out according to these rules:

A default port connector symbol is placed for each pin on the parent block, placed according to their position on the symbol.

Each pin on each port connector has its type set according to the rules outlined in “Setting the Port Pin Type” on page 237.

Creating a Hierarchical Block - Bottom Up

Any open design can be made into the subcircuit of a hierarchical block symbol. To create the internal circuit first and work upwards, follow these steps:

Create or open the design file that is to become the internal circuit and define its port interface.

Create and place the hierarchical block symbol in the target design using the Create Subcircuit Block command.

For this simple procedure we will make use of the Create Subcircuit Block command which performs a number of automatic operations for you. It is also possible to do some of these steps manually to get more control over the symbol graphics and port interface. These two extra steps are also described in the following sections:

Creating the hierarchical block symbol using the device symbol editor tool is described in “Creating a Symbol for an Existing Subcircuit” on page 234.

Linking the subcircuit to the symbol using the Attach Subcircuit command is described in “Attaching a Subcircuit” on page 232.

NOTE: This procedure links the subcircuit to the symbol only for this design. You can also store the subcircuit permanently with the symbol in the library. This allows it to be easily used in multiple designs without having to use the Attach Internal command each time. See “Creating a Part With a Subcircuit” on page 300 for details on how this is done.

Creating a Subcircuit

Creating a circuit that is to be attached to a parent symbol is exactly the same as creating an independent design, except for the necessity of defining the port interface. Note the following issues when creating the subcircuit:

Each signal that is to be an input/output point for the subcircuit (that is, a pin on the parent symbol) must have an attached port connector.

The name of the port connector device must exactly match the name of the associated pin on the parent symbol.

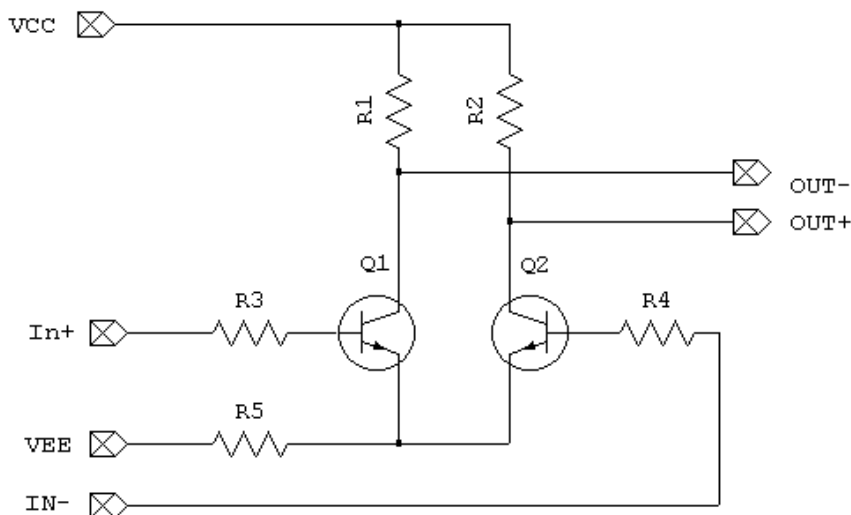
For each port, choose a port connector type (input, output or bidirectional) appropriate to the signal flow in your circuit. This allows for more efficient simulation, improved error checking and design clarity.

Bus ports require special care in creation and naming of port connector symbols. Review the rules given earlier in this chapter carefully.

See “Creating and Using Port Connectors” on page 237 for complete details.

The following sample shows how port connectors are attached to the connec-

tion points in a simple circuit:



NOTE: You may wish to save this circuit as a design file for safekeeping, but this file will not be associated with the target hierarchical design in any way. The Place Subcircuit command described next will completely incorporate the subcircuit data into the target design with no reference to any external files. More information on alternate ways of storing subcircuits is given in “Associating a Subcircuit with a Device Symbol” on page 230.

Placing a Subcircuit

In this section we will use the Create Subcircuit Block command to auto-generate a symbol corresponding to the subcircuit created above, and then place it in the target design. To do this:

Open or create both the subcircuit and the target circuit.

Bring to the front the target circuit window into which you wish to place the hierarchical block.

Select the Create Subcircuit Block command in the Subcircuit sub-menu of the Options menu. This will display a list of the open designs, except the topmost one, which is assumed to be the target.

Choose the circuit that is to become the subcircuit.

Click the OK button. Depending on the size of the subcircuit and the complexity of the symbol required, this operation may take a while.

When the processing is complete, the cursor will be replaced by an image of the new hierarchical block.

Place the new symbol in the desired position in the target circuit.

Generating Netlists from Hierarchical Designs

When producing a netlist for use by an external system, it is necessary to determine whether a “hierarchical” or “flattened” netlist is required.

Generating Hierarchical Netlists

A hierarchical netlist retains the hierarchical structure of the original design. It contains only a single description of each type of hierarchical block used in the design. Normally, the lowest-level blocks are written out first, followed by higher blocks that refer to them, followed finally by the master circuit for the design. By its nature, a hierarchical netlist can contain only definition data.

This format is used by many simulators and FPGA packages, for example, SPICE.

Generally, any package that accepts a hierarchical netlist can also accept a flattened one. A flattened netlist may be useful in cases where unique instance data was stored with devices or signals in internal circuits.

Generating Flattened Netlists

A flattened netlist has had all information about hierarchical structure removed from it. Every device and signal instance is listed separately with its instance data. Flattened netlists are normally required for PCB formats since name, unit and pin number assignments must be different for each device instance.

NOTE: Flattened netlists can only be generated from Physical hierarchy mode designs.

See more information on hierarchical netlists in the entry Script Hierarchy Issues in the DesignWorks Script Language Reference (separate manual on disk).

Using Hierarchical Names

When producing a flattened netlist from a hierarchical design, it is usually necessary to generate a unique name for each device and signal in the design. This can be done in one of two ways:

Manually or automatically assign a unique identifier to each item and store it in an “instance” attribute field. For example, in physical hierarchy mode, the Packager assigns a unique name to each device instance and stores it in the InstName field.

Generate a unique identifier consisting of the definition name (i.e. Name attribute field) of the device or signal, prefixed by the names of all parent devices. For example, in the simple design example given at earlier in this chapter, the six physical devices would be called:

```
FF1/G1  
FF1/G2  
FF1/G3  
FF2/G1  
FF2/G2  
FF2/G3
```

Assuming the Name field is unique within each circuit level, this is guaranteed to produce a unique identifier for each instance.

The “/” slash character is the standard name separator used in DesignWorks but can be changed by the user if needed using the procedure described below.

Changing the Hierarchical Name Separator

The separator character used to generate hierarchical names is stored in the design attribute field HierNameSep. To change it:

- 1) Select the Design Attributes command in the Options menu.
- 2) Select the HierNameSep field in the field list.
- 3) Enter a new value in the data box.
- 4) Click OK.

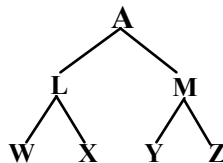
NOTE: The usage of the hierarchy name separator and its appearance in netlist output is completely under the control of the netlist generation script. The script can override the format specified here. See the entry Script Hierarchy Issues in the DesignWorks Script Language Reference (separate manual on disk) for more information.

Printing Hierarchical Designs

A hierarchical design has an extra dimension which must be taken into account in determining the order of page printing.

Determining Print Page Order

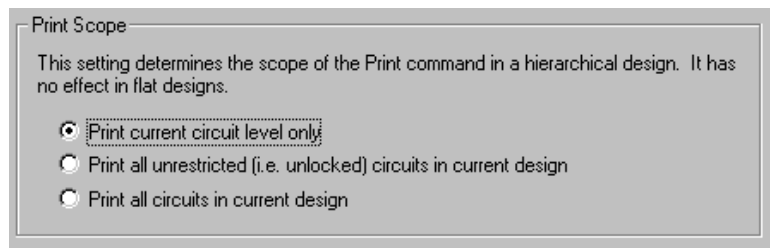
When a design is printed, the master circuit of the design is printed first in its entirety. Then, each hierarchical block is printed, followed by all hierarchical blocks it contains. For example, given the following structure of nested hierarchical blocks:



The print order would be: A, L, W, X, M, Y, Z. The order of printing of hierarchical blocks inside a single circuit is determined only by internal storage order and cannot be controlled by the user. E.g. W and X could be reversed above.

Setting Printing Scope

To set the scope of printing in a hierarchical design, choose the Design Preferences command in the Drawing menu and select the Printing tab.



Three options are available:

Print current circuit only: With this setting, the Print command will print only pages belonging to the circuit level in the current window. With this setting, the page numbers shown by the “Show Printed Page Breaks”

option can be used to select a subset of pages to print.

Print all unrestricted circuits: With this setting, the design's master circuit and all unlocked internal circuits will be printed. (The “locked” setting is controlled by a check box in the Properties command for the parent device.) With this setting, the page numbers shown by the “Show Printed Page Breaks” option will not be relevant since they are numbered separately for each internal circuit.

Print entire design: Print the design's master circuit and all internal circuits, regardless of “locked” status.

Printing Sequential Page Numbers in a Hierarchical Design

When a hierarchical design is printed, the \$PRINTPAGENUMBER text variable can be used to apply sequential page numbers to the printed sheets.

NOTE: This text variable only applies during a Print operation. When it is drawn on the screen, it is interpreted the same way as the PAGENUMBER variable, i.e. it gives the page number within the circuit.

See more information on text variables in “Using Text Variables” on page 107.

Associating a Subcircuit with a Device Symbol

There are a number of ways that a subcircuit can become associated with its parent device in a hierarchical design:

The subcircuit can be stored with the symbol in the library. This is referred to in DesignWorks as an “internal subcircuit”. See “Creating a Part With a Subcircuit” on page 300 for information on using the device symbol editor to attach a subcircuit to a symbol. Also see “Copying Symbols from a Design to a Library” on page 265 for information on using the Save to Lib command to store a symbol from a schematic to a library, including its subcircuit.

The symbol can be stored in a library with no associated subcircuit and one can be created or attached to it interactively after it has been placed in

the schematic. This procedure is described in “Creating a Hierarchical Block - Top Down” on page 221.

WARNING: When you attach or detach device subcircuits, you are in effect changing the definition of the device. This always affects all other devices of the same type in the same design. This applies to all the subcircuit commands following and any other subcircuit editing operation in DesignWorks. If you want to affect only a single device, you must first use the Make Unique Type command in the Part Type sub-menu to isolate it from others of the same type. See a more complete description of this command in “Making a Single Device Into a Unique Type” on page 269.

Working with Subcircuits

The commands described in this section are intended to assist in assembling separate circuits into a complete, hierarchical design, changing the organization of a design, and moving or copying subcircuits between designs. See “Creating a Part With a Subcircuit” on page 300 for more information on storing a subcircuit with a symbol in a library.

Placing a Subcircuit Block in a Parent Circuit

The Place Subcircuit command is a shortcut for creating a symbol to represent an internal subcircuit and placing it in a higher-level circuit. This command performs these steps:

- Displays a selection box allowing you select any other open design to become a new subcircuit block.

- Invokes the device symbol editor’s Auto Create Symbol function to generate a rectangular symbol with pins representing the ports defined in the subcircuit.

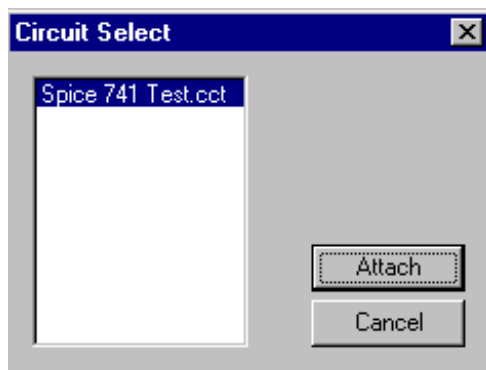
- Attaches the selected design as a subcircuit of the new symbol.

- Enters device placement mode, allowing you to place the new symbol in the current circuit.

Note that the generated symbol is not saved to a library, so if you cancel the device placement mode, you can only recover the symbol by going through the same process. If you wish to modify the auto-generated symbol after it is used, this can be done using the Edit Symbol command described in “Editing a Device Symbol in a Schematic” on page 275.

Attaching a Subcircuit

The Attach Subcircuit command in the Subcircuit menu allows you to select an open design to attach as a subcircuit to the selected device. When this command is selected, the following box will appear:



WARNING: This operation cannot be Undone!!!

Clicking Attach on the Attach Internal box will cause the following actions to be taken:

If the current design (i.e. the one containing the parent device) contains other devices of the same type as the selected device, then a separate, temporary type will be created for the selected device. This allows the definition of this type to be temporarily modified by the addition of the subcircuit. The other devices of the same type will be updated when you close the attached subcircuit.

The attribute definition table in the selected internal circuit is compared with the table in the master design. If the Attach operation would result in new fields being defined in the master design's table, then you will be prompted for confirmation. If you click OK, then new fields are merged into the master design's table.

Attribute values associated with the internal circuit are merged into the master design. If a given field already has a value in the master design, the value in the internal circuit is discarded.

The logical linkage between the selected device and the new internal circuit is completed. If any mismatch is detected between the port connectors defined in the internal circuit and the pins on the parent

device, you will be warned.

The title of the internal circuit is updated to reflect its hierarchical position in the master design.

Auto-packaging, if enabled, is disabled since this operation potentially invalidates the design's package table.

The newly-attached internal circuit's window is brought to the front. It is now considered to be an internal circuit that has been opened for editing and modified. When you close the internal circuit, you will be asked if you wish to update other devices of the same type.

Detaching a Subcircuit

The Detach Subcircuit command makes the currently displayed subcircuit into a separate design and redefines the parent device as having no internal circuit.

WARNING: This operation permanently removes the subcircuit from the selected device and all other devices of the same type in the selected design. If you do not wish to update other devices in the design, use the Make Unique command to isolate the selected device first.

The Detach operation cannot be Undone!!!

In particular, Detach Internal performs the following operations on the subcircuit displayed in the frontmost window:

The circuit is unlinked from its parent device, making it into a separate design.

The title of the subcircuit is set to a default “Designxxx” name.

The internal circuits of all other devices of the same type in the design are removed.

Auto-packaging, if enabled, is disabled since this operation potentially invalidates the design's package table.

Discarding a Subcircuit

To discard the subcircuit associated with a parent device:

Locate and select the parent device in its circuit.

Select the Discard Subcircuit command in the Subcircuit menu.

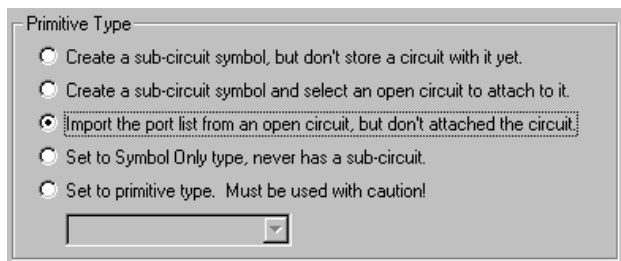
WARNING: All data associated with the subcircuit is destroyed by this command. This cannot be undone!

The Discard Subcircuit command redefines the selected device (and all others of the same type) as having no internal circuit.

Creating a Symbol for an Existing Subcircuit

To create a symbol for a subcircuit that has already been created and is open in a DesignWorks window:

- 1) Select the New command in the File menu and choose the Device Symbol document type.
- 2) Select the Subcircuit & Part Type command in the Options menu.
- 3) Click on the Imported Port Definition Only button. This will display a list of all currently open designs.



NOTE: Choosing the Internal Subcircuit option instead will cause the circuit definition to be saved in the library with the symbol.

- 4) Select the desired internal circuit in the list and click the Internal button.

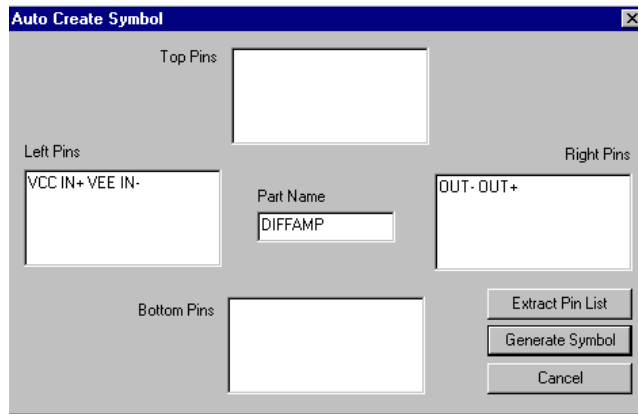
The device symbol editor now examines the selected circuit and creates a list of pins matching the port connectors in the circuit.

- 5) Select the Auto Create Symbol command in the Options menu.
- 6) Click on the Extract Pin List button.

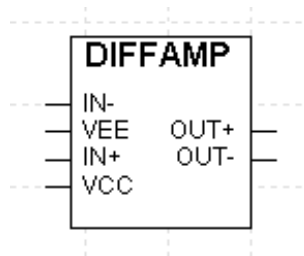
Extract Pin List

This will place input and output pins in default positions in the Left and Right Pin Names boxes. You may move names between these boxes as desired to

determine placement of pins. Be careful not to add or delete any names. You can also change the part name that will appear in the symbol, if desired.



7) Click on the Generate button. A default symbol will now appear in the device symbol editor window.



This symbol can be edited manually using the drawing tools.

8) Create a temporary library to save the symbol in. This is done using the New Lib command in the Parts palette pop-up menu.

9) Select the Save As command in the File menu. Select a suitable part name and double-click on the new library that was created in the last step.

10) Close the device symbol editor window.

Saving a Symbol with a Subcircuit to a Library

If you have created a hierarchy block symbol and its associated definition in a design and wish to save it in a library for use in other designs, follow these

steps:

Create or open the desired destination library.

Locate and select the parent device symbol in your design.

Select the Save to Lib command in the Part Type sub-menu.

In the Save to Lib box, turn on the “Internal circuit” checkbox to indicate that the internal circuit should be saved in the library.

Click the Save button.

Making Connections Across Hierarchy Levels

Signal connections between the levels of a hierarchical design are made using one of these three techniques:

Port connections - This is the normal method of making connections across hierarchy levels. A pin on the parent device symbol matches with a port connector in the internal circuit

Signal connectors - This method makes use of “signal connector” pseudo-device symbols on the schematic and is intended for making power and ground connections. See “Making Power and Ground Connections Across Hierarchy Levels” on page 242 for more information on using signal connectors in hierarchical designs.

Signal sources - This method uses device attributes to specify global connections and is also intended for making power and ground connections. See “Making Power and Ground Connections Across Hierarchy Levels” on page 242 for more information on this method.

IMPORTANT: The port connector method is the only one of these three that makes an logical connection between hierarchy levels as soon as the object is placed in the schematic. The signal connector and signal source methods rely on the Export tool to make name associations when a netlist is generated. This has an important implication for users of the DesignWorks Simulator option: The Simulator does not recognize these types of connections for simulation purposes. To make a signal connection between hierarchy levels for use with the interactive simulator you must use port connections.

Creating and Using Port Connectors

Port connectors are special pseudo-device symbols that associate a pin on a hierarchical block symbol with a signal in an internal circuit. Pre-defined port connector symbols for the three most commonly used pin functions are provided in the Pseudo Devs library with DesignWorks: Port In (Input), Port Out (Output) and Port Bidir (Bidirectional). For cases involving bus pins or any requirement for special symbol graphics, this section describes how to create custom port connector symbols.

NOTE: If you are working top down in a hierarchical design, you should rarely have to make your own port connector symbols. See “Creating a Hierarchical Block - Top Down” on page 221 for more information. If you are working bottom up, then you will need to define the port interface for a circuit using port connectors before making the parent symbol. See “Creating a Hierarchical Block - Bottom Up” on page 224 for a higher level view of this procedure.

The procedure for creating port connector symbol graphics is described in “Creating a Port Connector” on page 308.

Setting the Port Pin Type

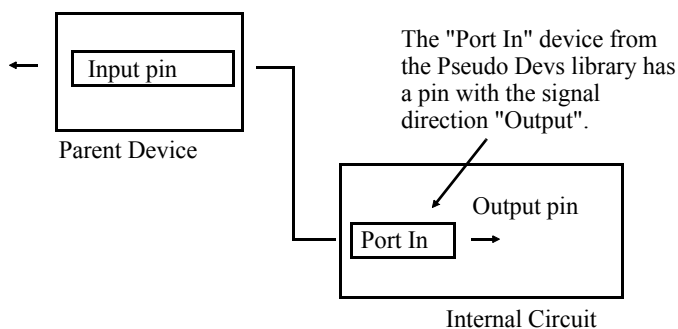
If pin type (input, output, bidirectional, etc.) is significant for your design, then the following rules should be noted when creating and using Port Connectors:

The pin on the Port Connector symbol is normally of the opposite type to the corresponding pin on the parent device symbol. E.g. A signal coming in to the hierarchical block is actually an output from the port connector pin.

DesignWorks does not check correctness of pin types on port connectors.

When an internal circuit is auto-generated, port connectors of a type opposite the parent pin are placed automatically.

This figure illustrates these points:



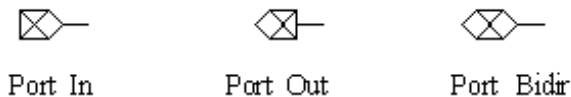
If the parent device pin is an output then it must be driven by the internal circuit, the internal circuit's port connector should have an input pin. To connect a signal in an internal circuit with an input pin on a parent device use the “Port Out” connector in the “Pseudo Devs” library.

If the parent device pin is Bidirectional then the internal circuit's port connector should have a bidirectional pin as well.

NOTE: When the *pin function* on a port connector is examined from the schematic module via the *Properties* dialog, the function shown is that of the parent type pin. This means that if the parent type pin is an input then the matching internal circuit port connector's pin will show as an input pin in the pin properties dialog. If you examine the port connector in the device symbol editor's pin palette, the same pin will show as an output.

For normal signals (i.e. not buses) you can either:

Use one of the Port devices from Pseudo Devs library: Port Bidir, Port In or Port Out. These need to be named to match the pin names of the parent device.



Use the New Port Connector command to auto-generate a port connector symbol. This procedure is described below.

Using the New Port Connector command to Create a

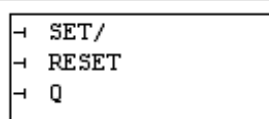
Signal Port

To use this command:

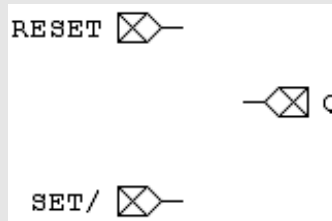
- ➔ Select the New Port Connector command in the Subcircuit submenu of the Options menu.

Port/Pin Naming

The relationship between the port connector in the subcircuit and the pin on the parent device symbol is established by matching the pin name on the parent device with the Name of the port connector. For example, if we were to open the RSFF device used in the example earlier in this chapter using the device symbol editor tool, we would see the following pins listed:



For a complete port interface, a port connector must exist in the internal circuit named to match each one of these pins. In this case, the following port connectors would be required (ignoring all other internal circuitry):



The port interface is rechecked whenever any change is made. Thus, as soon as a port connector is added or removed, or its name changed, the port interface has been updated to reflect the new logical connections. However, to avoid excessive warning messages, error checking is performed only when an internal circuit is opened or closed. A warning box will be displayed if any error is found. This checking cannot be disabled.

NOTE: The name of the port connector's pin and the name of the signal attached to the port connector are not significant in making the port association. Only the contents of the port connector's Name field is used. Note the different rules for bus ports below.

- ➔ Select the desired port type: input, output or bidirectional. Note that the type chosen here is the port type as seen from the parent symbol's point of view. E.g. if this is to be an input into the subcircuit, add an Input port type. An Input port type actually adds an output pin to the port connector as this is an output driving the subcircuit.

IMPORTANT: For bus ports, you have to define the exact list of internal signals to be included in the port. See the section below for more detailed instructions.

- ➔ Enter the name of the port in the Port Name box. This name must match exactly the name of the pin on the parent symbol. This setting actually sets the Name attribute of the resulting port connector and can easily be edited later if needed.
- ➔ Click the Place button to place the new symbol in the circuit.

Creating Bus Ports

Connections can be made between busses across hierarchy levels using Bus Port Connectors. Bus pins on a parent device symbol must be matched with a Bus Port Connector having identical internal pins. For this reason, Bus Port Connectors must always be custom made, with any, or a combination, of these methods:

Using the auto-generate function that creates a subcircuit “shell” for you automatically when you create a new subcircuit. This is described in “Creating a Hierarchical Block - Top Down” on page 221.

Use the New Port Connector command to auto-generate a port connector symbol. This procedure is described below.

Using the device symbol editor tool to create a device symbol with the appropriate pins, either from scratch, or using an existing bus port connector as a guide. This method is described in “Creating Bus Ports” on page 240.

IMPORTANT: Unlike Bus Page Connectors, Bus Port Connectors do not export all the signals in the attached bus, only the ones for which it has explicit Bus Internal pins.

Creating a Bus Port Connector Using the New Port Connector Command

The New Port Connector command is the easiest way to create a bus port connector when working “bottom up”, i.e. you have not yet created the parent symbol. To use this command:

- ➔ If the bus to which you plan to connect the port connector already exists, select it first. This way, all signals already in the bus will automatically be added to the list without any further action.
- ➔ Select the New Port Connector command in the Subcircuit submenu of the Options menu.
- ➔ Select the Bus option.
- ➔ Check the “Internal Pin List”. If you had already selected a bus before entering this command, the list may already have the items you need. If not they will need to be added.
- ➔ To add more pins, enter the pin name (or a list or range of pin names) into the Pin Name box, then click the desired type of port pin. The name format used here is the same as is used in the New Breakout command and other similar commands. See “Creating a Breakout” on page 194. for details on this format.

NOTE: For many applications, the port type is not significant. See “What Pin Types are Used For” on page 373. for a discussion of the uses of this setting.

If you do need to select the port type, note that the type chosen here is the port type as seen from the parent symbol’s point of view. E.g. if this is to be an input

Bus Pin Name Matching

Note the following rules for name matching in bus ports:

As with other Port Connectors, a Bus Port Connector must be given a Name exactly matching the pin name of the bus pin on the parent device.

The internal pins in the parent bus pin must exactly match the internal pins on the Bus Port Connectors bus pin.

The pin name of the bus pin itself on the Bus Port Connector is not significant.

As with normal ports, the names of the signals attached to the Bus Port Connector’s pin are not significant.

into the subcircuit, add an Input port type. An Input port type actually adds an output pin to the port connector as this is an output driving the subcircuit.

- ➔ When you have added all the required internal pins, enter the port name into the Port Name box. This name must match exactly the name of the pin on the parent symbol. This setting actually sets the Name attribute of the resulting port connector and can easily be edited later if needed.
- ➔ Click the Place button to go to device placement mode with the generated port connector symbol.

IMPORTANT: If you cancel device placement, the symbol is not saved anywhere and you will have to repeat the above procedure.

Modifying an Existing Bus Port

An existing bus pin and its corresponding bus port connector can be modified in any of the following ways:

To add or delete internal pins from a bus pin, right-click on the bus pin and select the Bus Pin Options command. See “Adding Pins to the Bus Pin” on page 198 for more information.

IMPORTANT: If you select a bus pin on a parent symbol that has a subcircuit, the Bus Pin Options command will automatically modify the corresponding bus port connector in the subcircuit so that it matches. **THE REVERSE IS NOT TRUE!** Using the Bus Pin Options command on a bus port connector in a subcircuit *does not* automatically update the corresponding parent pin. This must be done manually!

To make any graphical modifications to either the parent symbol or a port connector in the subcircuit, you must use the device symbol editor tool. See “Editing an Existing Part on a Schematic” on page 281 and “Creating a Port Connector” on page 308 for more details. It is the user’s responsibility to ensure that matching modifications are made to both the parent symbol and the port connectors.

Making Power and Ground Connections Across Hierarchy Levels

Power and ground symbols (i.e. Signal Connector devices) do not make an immediate logical connection across hierarchy levels. For this reason, signal connectors should not be used to make active signal connections for interac-

tive simulation purposes.

However, it is possible to merge power and ground nets across levels in netlist output using the Export tool's \$SIGSOURCE function.

Using \$SIGSOURCE and Device Attributes

Any number of special names can be specified as “signal sources” using the \$SIGSOURCE keyword in a report form file. The Export tool searches the attributes attached to each device in a design for fields with these names. The pin number(s) specified in the value field of the attribute will be attached to a list for the signal of the same name.

For example, most of our library parts have a “Power” and “Ground” attribute.

In a circuit with two components U1, a 7404, and U2, a 74133:

U1 has the attribute Power with a value of “14”.

U2 has the attribute Power with a value of “16”.

In a report form the statement \$SIGSOURCE(Power) will cause the Export tool to extract the pin number “14” from the device, say U1, and place it in the pin list for a signal called “Power”.

PowerU1-14, U2-16

This ability is intended to allow power and ground nets to be created without the necessity of having explicit power and ground pins and signals on every device. This should not be used for general signal connection.

Using Signal Connectors

There are several pre-defined signal connectors in the Pseudo Devs library, Plus5V, Plus12V, Minus15V, etc...

You may create new ones using the device symbol editor module. There are only two tricky parts to making a signal connector:

Signal connectors can have only one pin, the name of the pin must be the name of the signal you wish to connect. For example to create a signal AGND the pin of the signal connector must be named AGND.

You must set the primitive type of the new signal connector to “SIGNAL CONNECTOR” using the “Subcircuit & Part Type...” dialog found in the

Options menu.

Information on creating signal connectors with the device symbol editor can be found in “Creating a Power and Ground (Signal) Connector” on page 308. Also see general information on using signal connectors in “Power and Ground Connections” on page 205.

Introduction

The purpose of this chapter is to describe the usage of a number of tools that can be used to locate and update circuit objects in a design. These include:

The Find tool is used to locate devices, signals or pins in a schematic by name or attribute values. Once the objects are found they may be sequentially selected and viewed or selected as a group to be operated on with some other tool. For example, you might locate all capacitors with a certain Value setting, then use the Browser to display those selected items and enter a modified value. The Find tool is script based, that is, it calls the report generator tool to run a script file and then uses the output of the script to locate the objects that the script describes. For this reason, it has considerably more flexibility than just as a “find by name” tool. Any processing that can be done in a script can be used to locate objects. This chapter describes how to use Find with the predefined scripts provided and also gives details on customizing your own error location scripts.

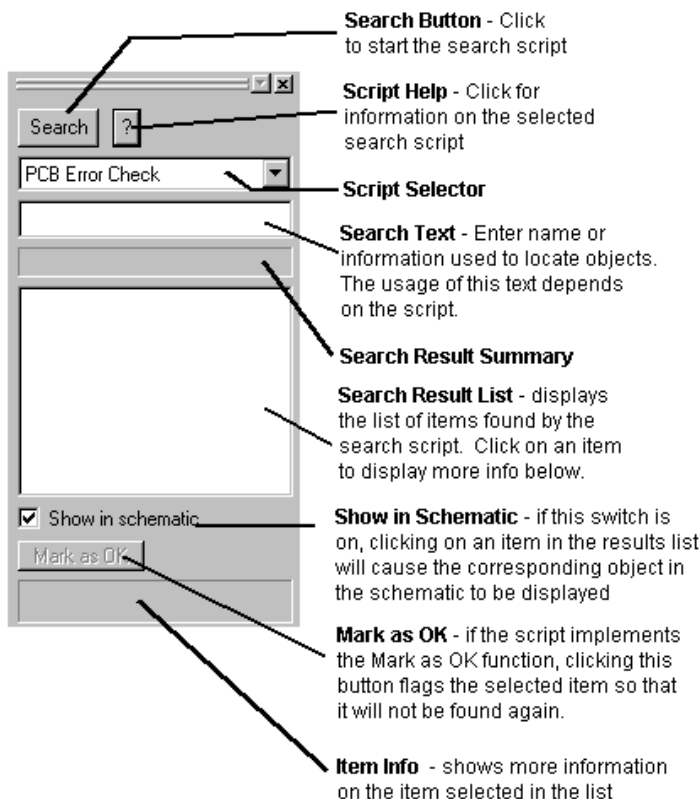
The Browser is a tool used to view, edit and navigate schematic data in spreadsheet-style format. Rows of the spreadsheet display objects in the schematic, either devices, signals, pins or nets. A cell of the spreadsheet displays an attribute value for the object in that row and the attribute field name in that column.

The remainder of the chapter is divided into three sections covering these tools in more detail.

Using the Find Tool

Starting Find

To invoke Find, select Find in the Drawing menu. This opens the find panel:



TIP: The Find palette can be relocated and displayed in different styles as described in "Tool Panels" on page 5.

The following sections provide details on the usage of the various controls in this window.

Search	Click this button to start the search. This runs the search script and displays the results in the item list.
? - Script Help	Click this button to display notes describing the selected search script and how it is used.
Script Selector	This pop-up menu displays a list of all available search scripts. To get more information on a script, select it in this list and click the “?” script help button. See “Where Search Scripts are Placed” on page 259 for more information on how the Find tool locates scripts to put in this list.
Search Text	This box is where you type text that is used by the script to locate objects. For example, if you select a “find devices by name” script, type the name of the device you want to find here. The usage of this text is completely up to the script, and some scripts may ignore it.
Search Result Summary	After a search is completed, the script will display a summary of results in this box.
Search Result List	When a search is completed, this list will contain a list of all items found by the script. This will typically be a list of objects in the design matching some criteria, but the script can make use of this list any way it likes.
Show in Schematic	If this switch is on, the Find tool will attempt to locate the corresponding object in the schematic each time an item in the result list is clicked. If the script has not provided any object information with the result items, this will not be available.
Mark as OK	If the script implements the Mark as OK function (described below), this button will allow you to mark a selected item so that it will not be found again.
Item Info	This box displays more detailed information on an item selected in the results list.

Locating Found Objects on the Schematic

If the “Show on Schematic” box is checked, the current object will be located on the schematic and selected each time an item is selected in the results list.

Some types of circuit objects may not be able to be shown: Bus internal pins and signals have no visible manifestation on the schematic. If one of these is

found by the script, its containing bus will be selected.

If the found object is inside a subcircuit block, the subcircuit will be opened and the object selected. This will not be possible in the specific case where the object is in a physical mode design and another instance of the containing subcircuit is already open for editing and has been modified.

Marking Located Objects as OK

The “Mark as OK” button allows you to indicate that a found object is not actually an error and should not be found again by the same test. This allows you to keep “false alarms” out of the test results so that actual errors will be more easily located in future checks.

NOTE: This feature may not be implemented in all scripts. If the script does not generate the information required, this button will be disabled.

Clearing the Mark as OK Settings

The Mark as OK settings stored with objects by previous error checks can be removed in either of these ways:

Use Export to run a Clear OK Errors script provided. This will not locate any objects, but it will clear the Mark as OK settings for all objects in the circuit.

Manually remove any value from the OKErrors attribute field in the device, signal or pin in question. The numeric value stored by the script is taken as a set of bits that mask out specific errors. Setting the value to “0” or removing it completely indicates that all error checks should be performed on this object again.

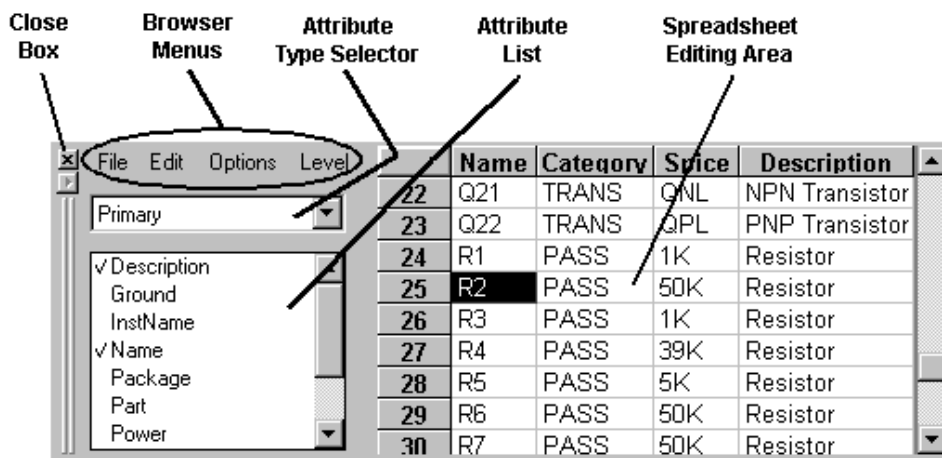
See “Customizing Search Scripts for the Find Tool” on page 255 for information on writing your own search scripts.

Using the Browser Tool

Opening the Browser

To open the Browser, select the Browser item in the View menu. This opens

a spreadsheet window.



Close Box

Click this control to close the Browser window. Another method of closing is to select the Browser item in the View menu.

File Menu

This menu contains the Save As command, which allows you to save the contents of the spreadsheet as a text file.

Edit Menu

This menu contains the standard clipboard editing commands which allow you to operate on selected cells in the spreadsheet.

Options Menu

This menu contains options to control the type of objects displayed and how they are selected.

Level Menu

The menu shows the circuit hierarchy upwards from the current circuit and allows you to move up by selecting an item. NOTE: This menu only appears when a subcircuit in a hierarchical design is current.

Attribute Type Selector

This list allows you to select what type of fields are shown in the Attribute List. By default, only Primary fields are shown, i.e. attribute fields marked as “Primary” in the Define Attribute Fields command. You can also elect to show:

Secondary fields—Fields not marked as “Primary”

Pseudo attributes—Text values that are associated with an item by the system, but are not normal attribute fields, including the part name and graphical information.

Definition fields—All attribute fields not marked “Keep with Instance” in the Define Attribute Fields table.

Instance fields—All attribute fields marked “Keep with Instance” in the Define Attribute Fields table.

By default the Browser will initially display all devices in the current circuit. If any objects are selected in the schematic, the Browser will display only selected devices or signals. Selected devices have priority over selected signals.

Closing the Browser

To close the Browser window, either click in the Close box (the small X button at the corner of the window), or select the Browser item again in the View menu.

Updating the Browser Window

The Browser will update any time a change occurs in the schematic that effects the data displayed in the spreadsheet, or when its scope is set to Circuit and the current circuit changes.

Selecting the Type of Objects Displayed

NOTE: The Browser is controlled by the separate File, Edit and Options menus on the Browser panel itself. The menus in the main application menu bar do not affect the Browser. Specifically, the Cut, Copy and Paste edit menu items in the main menu bar will still operate on the selected circuit items and not on the spreadsheet.

The Object Type submenu under Options allows you to change the type of object being displayed. There are eight different types of objects that may be displayed Devices, Signals, Pins, Nets, and selected Devices, Signals, Pins, and Nets.

If the “Selected” form of an object is chosen to be displayed, e.g. Selected

Devices, then only items which are selected in the schematic will be displayed. Similarly, if Selected Signals is chosen in the Object Type menu, then only signals that are selected in the schematic will be listed in the spreadsheet.

Determining Where to Search for Objects

When looking for objects, the Browser will limit its search scope by using the selection made in the scope menu pop-up. There are four different scopes:

- Design** In this scope the entire design hierarchy is search for objects of the correct type. This scope has a side-effect. Some of the objects attribute values displayed may not be editable because of definition/instance conflict resolution. The uneditable objects and/or fields will be displayed with a gray background.
- Circuit** In this scope the search is limited to the pages of the circuit with the most active window.
- Here Down** In this scope the search for objects is in the circuit with the active most window and in all circuits below it in the hierarchy. Some of the objects and/or fields may not be editable for the same reasons as explained in Design scope.
- Page** This scope only looks for objects in the circuit with the most active window and in addition only looks at the page associated with that window.

Displaying Attributes

To add an attribute field to the spreadsheet, simply double-click its name in the attribute list.

NOTE: The Nets and Selected Nets object types display a different format in the spreadsheet and do not allow attributes to be added.

Attributes are divided into three groups:

- Pseudo Attributes** These are characteristics about the object that the schematic knows but are not true attributes. These attributes may not be edited.
- Primary Attributes** These are attribute that have been selected in the Define Attribute Fields command and marked as “Primary”. This is done to allow easier access to the more common attributes.
- Secondary Attributes** These are all the attributes that are not marked as primary (see above).

The attribute field names appearing in the list will change as the type of object selected for display is changed. Only attributes that apply to a given object type are displayed. For example, the attribute field “Part” is associated with Devices but will not be associated with Signals.

See “Defining a New Attribute Field” on page 177 for more information.

If an attribute name is checked in the attribute list then it will be added as a new column at the end of the spreadsheet. Double-clicking a checked attribute name removes the associated attribute from the spreadsheet.

Changing Attribute Values

Selecting and Editing Cells

To edit a cell, just click on it and start typing. Cells that are shaded gray are displaying values that cannot be edited in the spreadsheet

Modifying the value of a cell will cause the corresponding attribute value of the object in the schematic to change.

WARNING: The schematic data is updated immediately and there is no undo.

Using the Edit Menu Commands

The Edit menu in the Browser panel contains the standard clipboard editing commands Cut, Copy, Paste, and Clear, plus four additional menu items:

- | | |
|---------------------|---|
| Fill Down | The Fill Down command takes the selection rectangle and copies the values in the cells at the top of the selection to all the selected cells below them. |
| Fill Right | The Fill Right command takes the selection rectangle and copies the values in the cells at the left of the selection to all the selected cells to the right of them. |
| Make Visible | The Make Visible command takes the object attributes affected by the selection rectangle and makes them visible in the schematic. This command applies only to attribute values (not pseudo-attributes like the part name) and only to visible objects. For example, it is not possible to display attributes associated with bus internal signals. |

Make Invisible The Make Invisible command takes the object attributes affected by the selection rectangle and makes them invisible in the schematic.

Disallowed Editing Changes

Any editing change that may cause a structural change to the connectivity of the schematic will be disallowed and aborted by the Browser. There are only a few special cases that will cause this to happen. For example:

Renaming a visible signal to the same name as another visible signal on the same page would cause the two signals to become one signal.

Making a signal visible which has the same name as an already visible signal with the same name would cause the two signals to become one signal.

Saving and Printing Data in a Browser Window

The Save As command in the Browser's File menu provides a method of directly saving data that appears in the spreadsheet display. There is no direct printing command to print the contents of the spreadsheet, but you can open the resulting test file in any text editor or word processor to format it for printing.

TIP: If you will be making this kind of transfer frequently, we suggest creating an Export script to write the data to a file in the desired format. This will ensure consistent results and will not be dependent on the current settings in the Browser.

Browser Clipboard Data Format

The Browser exports data in a tab-delimited text format commonly used by spreadsheet and database programs. The following points summarize this format:

Items appearing on one line in the spreadsheet display appear on one text line.

Items on one line are separated by a tab character.

Lines are terminated by a single carriage return character.

If an item contains anything other than alphanumeric characters and spaces it is enclosed in double quotes. If it contains a double-quote

character, that character is doubled, i.e. it is preceded by a second double-quote.

Note that attribute items can themselves contain carriage returns and other control characters. Such items will be enclosed in double-quotes in the clipboard text.

Showing Objects in the Schematic

When the Show in Schematic option in the Options menu is checked, the Browser will always attempt to display the associated schematic object whenever a cell is selected in the spreadsheet.

The Browser may sometimes not be able to show the object. The reason for this is generally that the object was not in an open window, or does not have a visible representation, e.g. bus internal signals.

Sorting Displayed Objects

Items are always displayed in the spreadsheet sorted in order by the first column. If the values in the first column are the same, the second column is used, etc. The sort order can be changed by re-ordering the columns so that the primary sort value is in the first column

Adjusting the Spreadsheet

Resizing Columns

Columns may be resized by moving the cursor into the column heading area and positioning it over the vertical dividing line. The cursor when correctly positioned it will change shape. Press the mouse button and drag the line to size the column to its desired size. Rows cannot be resized.

Moving Columns

Columns are reordered by moving the cursor into the column heading area and positioning it over the column heading to move. Hold the key and drag the label to its new position.

TIP: It can be useful to move columns so that the sorting mechanism will base the sort on the new ordering of the columns.

Customizing Search Scripts for the Find Tool

Find has no ability to locate circuit objects and examine properties on its own. It reads a report file generated by the Export tool in a specific format which lists the objects of interest. It reads this file and then displays the results to the user in an interactive manner. Creating an error script thus requires some knowledge of the particular format read by Find and the scripting techniques required to generate such a file.

Find Data File Format

The Find module reads a text file containing a list of items to display for the user. All the items in the list usually correspond to objects (e.g. devices, signals or pins) in the design, this is not required. Some potential uses of the Find tool may call for the display of data stored in design attributes or even read from external files or software.

For each item, the following information can be supplied:

The item's title to display in the item list. This should uniquely identify this item in as compact a form as possible, generally no more than 30 - 40 characters. This data is required.

The object's locator string. This is used to uniquely locate the object in the schematic, even in hierarchical designs. If this is not provided, the "show in schematic" function will not operate.

A text message to be displayed in the "item info" box when this item is selected in the list. This will normally indicate to the user the type of error found and any other relevant information specific to this item.

An error bit number that can be set to mask out this error. This is used to implement the "Mark as OK" function on the Find control panel. If the user clicks this button, the given error bit will be set in the OKErrors attribute field. It is the script writer's responsibility to check this bit when looking for errors and to skip over this object if it comes up. More information on the Mark as OK function can be found in the DesignWorks Script Language Reference (separate manual on disk).

In addition to the information provided per object, the file can also contain a title line that places a title in the Find results box. This will normally indicate

the type of error check done, the number of errors found, and other status information.

NOTE: Script developers should note that if you use the Mark as OK function in your scripts, you should provide a “Clear OK Errors” script to clear these settings in all marked objects. There is no user interface provided in Find for clearing these flags.

Data File Example

The following text illustrates a simple Find data file.

NOTE: Although the format of the command keywords is similar to a Export command script, an Find data file is not a script file. The Export tool will normally be used to generate the data file, as described in “Generating a Find Data File Using the Export Tool” on page 257.

```
$TITLE PCB error check found 2 errors.
$OBJECT D23D
$OBJTITLE U1*4 - invalid name
$MESSAGE Device U1*4 has an inv name.
$ERRBIT 12
$OBJECT D25D
$OBJTITLE U17 - inv pin #
$MESSAGE Device U17 has one or more invalid pin numbers
$ERRBIT 14
```

This table describes the keywords used in the data file:

\$TITLE	This sets the title displayed at the top of the Find results box. It should be set once at the beginning of the file. The title text starts immediately following the \$TITLE keyword and can continues until any other \$ keyword is found. This means that the title text can be multiple lines, obviously limited by the space available in the output box.
\$OBJTITLE	This sets the short title string used to display the item in the results list. The text follows the keyword on the line.
\$OBJECT	This indicates that start of the data for a new object, i.e. an item in the results list. (This does not have to correspond to an object in the design.) Following the keyword is an optional object locator string. The locator will be used to find the object in the design when the user elects to show this object. The locator string is generated using the \$DEVLOC/\$SIGLOC/\$PINLOC script commands.

- \$MESSAGE** This sets the message displayed in the “item info” box on the Find panel. The format is the same as for \$TITLE, that is, the message text starts after the \$MESSAGE keyword and continues for any number of lines until the next \$ command is found.
- \$ERRBIT** This sets the bit number used to mask this error using the “Mark as OK” function. If this is not supplied, the “Mark as OK” button will be disabled. This must be a decimal integer between 0 and 31.

Note that all keywords must start at the beginning of a new line and must appear with all upper case letters, exactly as shown.

Generating a Find Data File Using the Export Tool

When the user selects a script from the list in the initial Find window, these actions take place:

Find passes the name of the script to the Export tool and asks it to execute it.

Export executes the script, which must generate a data file in the format described earlier in some known disk location and must pass the full name of the file, including directory path name, back to Find.

Find opens the data file and displays the title and the list of results.

The execution of the Export tool in this mode is a bit different than operating it directly from the File menu. In this case, when the script starts executing, there is already a “report output file” open which is actually a memory buffer in Find. Any output generated by the script will be interpreted by Find as the file name. The script must contain an explicit \$CREATEREPORT command to generate the disk data file, followed by a \$CLOSEREPORT, followed by commands to generate the name of the file to pass back.

Here is a simple example to look for devices with no package code that illustrates these points:

```
$NOTES
```

```
Missing Package Codes
```

```
This script locates devices with no code in the Package
attribute field. The text entry box is not used.
```

```
$ENDNOTES  
$CREATEREPORT($TEMPPATH\package report)  
$FIND $DEVICES $NOT(&Package)  
$TITLE $DEVCOUNT devices found. The error file is in $TEMPPATH\hello file.  
$DEVICES\$_OBJECT $DEVLOC$NEWLINE\$_OBJTITLE $DEVNAME\$_MESSAGE Device $DEVNAME  
has no Package code. { Note: this is part of the $DEVICES line }  
$CLOSEREPORT  
$TEMPPATH\package report
```

Note these points:

The \$ keywords required by Find in the output file should always be prefixed with an escape character “\” so that Export does not attempt to interpret them as commands.

You must be careful to ensure that only a single line containing the file name is generated outside the \$CREATEREPORT/\$CLOSEREPORT pair.

Everything between \$CREATEREPORT and \$CLOSEREPORT is output to the data file. The format of this file must meet the specifications given earlier.

It is best to specify a complete directory path name when creating the data file and describing its location to Find. This will ensure that the file does not end up in an unexpected location due to the setting of the system's “current directory”. You can use the script keywords \$DESIGNPATH, \$SCRIPTPATH, \$TEMPPATH and \$PROGPATH to generate complete path names to known locations without specifying the absolute name of your disk. Note that \$DESIGNPATH will be null if the design has not been saved.

When the user clicks the Script Help (“?”) button on the Find panel, the contents of the \$NOTES section (i.e. anything between the \$NOTES and \$ENDNOTES keywords) are displayed in a Help box. This should always be included to describe the function of the script, how it uses the text input box, and any other relevant information.

As long as you keep these points in mind you can use any of the Export capabilities outlined in the DesignWorks Script Language Reference (separate manual on disk) to produce the data file.

Using the User Text Entry Box

The script can optionally make use of the contents of the Search Text box on the Find panel as part of its search. In the script, the contents of this box is passed as an argument to the script and is accessible using the &ARG1 variable. For example, the following script line will locate all devices with names matching the contents of the box:

```
$FIND $DEVICES $EQ(&ARG1, $DEVNAME)
```

Where Search Scripts are Placed

In order to appear in the Find tool's script list, search scripts must be located in a folder specified by the SCRIPTFOLDER keyword in the initialization file DW.INI.

See "Specifying Search Script Location" on page 390 for more information.

Device symbols are an important resource in your design creation process. Whether you primarily use the symbols provided with DesignWorks, or you create special libraries for your own use, the completeness and accuracy of this data has a major effect on your design flow. Library files generally outlast any one design and are used for many years across many projects. In addition, many DesignWorks features, for example, gate packaging, rely on specific steps being taken while creating a symbol. For these reasons, DesignWorks provides a variety of features for creating and editing the symbols themselves and for maintaining symbol library files.

This chapter covers these topics:

- The creation and maintenance of symbol library files.

- The creation and editing of individual device symbols using the device symbol editor tool.

- The attribute fields that are commonly used in symbols to implement gate packaging, auto-naming and other DesignWorks features.

- How to use the standard libraries that come with DesignWorks.

- Schematic operations that affect symbol definitions.

IMPORTANT: Creating a device symbol can involve much more than just drawing the graphics that represent the device. Many DesignWorks features and many features in other ECAD packages that you may want to interface with, rely on correct text attributes being associated with each symbol. Before you start on a project that involves creating many symbol libraries be sure you read Chapter 6—Before Starting a Major Design on page 113. This will assist you in looking at your entire design process and the data that is required to make all parts of the process work correctly.

Working With Symbol Libraries

The symbols and related parameters for DesignWorks devices are stored in data files called symbol libraries. For each device type in a library the following data is stored:

- General information on the type, such as number of pins, number inputs, number of outputs, type name, default delay, default attributes, position, orientation and type of each pin, etc.

- A picture representing the symbol for this part type.

- A polygon outlining the symbol, used for highlighting and erasing the symbol.

- An optional internal circuit definition.

The following sections deal with the creation and maintenance of library files. Later parts of this chapter deal with editing the symbols themselves.

Creating a New Library

To create a new, empty symbol library file is created using either of these methods:

- ➔ Right-click anywhere in the Parts palette. In the pop-up menu that appears, select the New Lib command.

In either case, a standard file save box will appear. Enter the desired name for the library and select a disk directory. The library created and opened automatically so it appears in the Parts palette. If you wish to have the library

Terminology Note

It is important to distinguish between the *definition* of a symbol that is stored in a library, and an *instance* or usage of it in a schematic. In this manual we use several different terms to refer to the definition of a device symbol in a library. If we use the term symbol, part, part type, or just type, we are referring to the definition of a device symbol in a library. The term symbol is used when we are primarily interested in the graphical representation of a device, but a symbol is always stored with related pin definition information and text attributes.

opened automatically at startup when you enter DesignWorks in the future, see “Specifying Libraries to Open at Startup” on page 385.

If you are creating a library as part of a design process that will be

Designs and Libraries

Whenever select a device symbol from a library and use it in a design, all information needed to display and edit that device is retained with the design. (Of course, only one copy is kept, regardless of how many times you use the same symbol in one design.) No further access to the library itself is required. This is done to ensure that a design file is always a complete entity and that future changes to a library will not inadvertently render an old design incorrect.

However, there are frequently times when you would like to update a symbol in a library and then copy the changes into one or several designs. Conversely, you may have edited a symbol or its subcircuit on a single design and wish to save the changes back to a library for use elsewhere.

For these reasons, DesignWorks does retain some information with each symbol about its “home” library and provides a number of features to allow transfer and updating of symbols between designs and libraries. Each time a new symbol is used in a design (i.e. one that hasn’t been used in this design before) the library name, file path and “last modified” date are stored in the attribute fields LibName, LibPath and LibDate, respectively. These are used by the Update from Lib command to locate the original library and determine if it has changed since we used it.

In addition, whenever the definition of a device symbol is modified, a new “checksum” value is calculated. This is also used as a check to verify the equivalence of two symbols whenever one is loaded from a library. This checksum cannot be set by the user, but it can be dumped in report using the \$CHECKSUM script command, described in the DesignWorks Script Language Reference (separate manual on disk). This can be used to tell if inadvertent modifications have been made to a symbol, but not to another that appears similar.

Two utility tools are provided with DesignWorks to allow the mass transfer of all the symbols in a design to a library, or to place one each of all the symbols in a library in a design, for testing or modification. These tools are described in “Copying Symbols from a Design to a Library” on page 265.

used over and over, you may wish to make the library part of a design kit. Refer to Chapter 13—Design Templates and Customization on page 329 for the basics of creating design kits.

Manually Opening a Library

To open any library file on your disk:

- ➔ Right-click anywhere in the Parts palette. In the pop-up menu that appears, select the New Lib command.

In either case, a standard file open box will appear. Locate the desired file in the usual way. The library will be opened and appear in the Parts palette. A small amount of memory is occupied by each open library file.

Automatically Opening Libraries at Startup

Libraries can be opened automatically when the program starts by any of these methods:

Open the library using the Open Lib command and check the “Automatically open this library at startup” checkbox. This places the library on the auto-open list, which can be managed using the Auto Open Libs command.

Select the Auto Open Libs command and click the Add button to add a new item to the list.

Placing the library (or a shortcut to it) in the Libs folder inside the DesignWorks program folder.

Placing a command in the DesignWorks initialization file (dw.ini) to specifically open the library using the LIBRARY or LIBRARYFOLDER setup file keywords. See “Specifying Libraries to Open at Startup” on page 385 for more information on this.

NOTE: By default, when DesignWorks is installed, a folder called Libs is created containing the initial libraries. This default folder is itself specified using the LIBRARYFOLDER keyword in the INI file and can be changed if desired.

Manually Closing a Library

To close any open library file, either:

Slide down the File menu to the Libraries sub-menu. In this sub-menu, select the Close Lib command,

or,

Right-click anywhere in the Parts palette. In the pop-up menu that appears, select the Close Lib command.

In either case, a list of the open library files will appear. You can use the and keys to select multiple files to be closed in one operation and then press the Close Lib button. Alternatively, you can simply double-click on the name of a single library.

NOTE: Any information required for symbols used in any open designs will be automatically retained in memory. Once you have used a symbol in a design, all information required has been copied into the design's data. No further access to the library itself is required.

Copying Symbols from One Library to Another

To copy one or more symbols from one library to another, follow these steps:

- ➔ Make sure the source and destination libraries are open in the Parts palette. If not, follow the steps under “Manually Opening a Library” on page 264.
- ➔ Select the Lib Maintenance command, either in the Libraries sub-menu of the File menu, or by right-clicking in the Parts palette.
- ➔ Select the source library in the pop-up library selection menu above the “Source Lib” list.
- ➔ Select the destination library in the pop-up library selection menu above the “Dest Lib” list.
- ➔ Select the symbols to be copied in the source list. You can select multiple items using the and keys.
- ➔ Click the Copy button.

The copy operation will now proceed, with status messages appearing in the Messages area at the bottom of the box.

Copying Symbols from a Design to a Library

See “Saving a Symbol Definition from a Schematic to a Library” on page 272.

Deleting Symbols from a Library

One or more symbols may be permanently deleted from a library by following these steps:

- ➔ Make sure the target library is open in the Parts palette. If not, follow the steps in “Manually Opening a Library” on page 264.
- ➔ Select the Lib Maintenance command, either in the Libraries sub-menu of the File menu, or by holding the \mathfrak{s} key pressed while clicking in the Parts palette.
- ➔ Select the target library in the pop-up library selection menu above the “Source Lib” list.
- ➔ Select the symbols to be deleted in the source list. You can select multiple items using the \mathfrak{a} and \mathfrak{z} keys.
- ➔ Click the Delete button. You will be prompted to confirm the operation before the items are permanently deleted.

WARNING: The Delete operation cannot be undone!

Duplicating a Symbol Within a Library

You can duplicate a symbol within a single library by following these steps:

- ➔ Make sure the target library is open in the Parts palette. If not, follow the steps in “Manually Opening a Library” on page 264.
- ➔ Select the Lib Maintenance command, either in the Libraries sub-menu of the File menu, or by holding the \mathfrak{s} key pressed while clicking in the Parts palette.
- ➔ Select the target library in the pop-up library selection menu above the “Source Lib” list.
- ➔ Select the symbols to be deleted in the source list. You can select multiple items using the \mathfrak{a} and \mathfrak{z} keys.
- ➔ Click the Duplicate button. You will be prompted for each selected item to enter a new name. Names must be unique within a library.

Renaming a Symbol in a Library

You can rename a single symbol in a library by following these steps:

- ➔ Make sure the target library is open in the Parts palette. If not, follow the steps under “Manually Opening a Library” on page 264.
- ➔ Select the Lib Maintenance command, either in the Libraries sub-menu of the File menu, or by holding the $\text{\$}$ key pressed while clicking in the Parts palette.
- ➔ Select the target library in the pop-up library selection menu above the “Source Lib” list.
- ➔ Select the symbols to be renamed in the source list. You can select multiple items using the $\text{\text{⌘}}$ and $\text{\text{⇧}}$ keys.
- ➔ Click the Rename button. You will be prompted for each selected item to enter a new name. Names must be unique within a library.

Getting Information on a Symbol in a Library

You can display information on a symbol, such as the exact location of the source library and a summary of attribute values, by right-clicking on the item in the parts palette, then selecting the Properties command. Part information cannot be modified in this display. To modify part attributes and other properties, select the Edit Part command and use the symbol editor facilities to make the desired changes.

Reordering Symbols Within a Library

Several options are available in the Library Maintenance box for reordering symbols within a library. These options do not affect the data associated with any symbol, they merely change the order in which they are indexed in the library file, which affects ordering in some internal operations.

NOTE: Items displayed in the Parts palette are always sorted alphabetically, so this procedure will not affect the displayed list.

First, to display this box, select the Lib Maintenance command, either in the Libraries sub-menu of the File menu, or by holding the $\text{\$}$ key pressed while clicking in the Parts palette. Two kinds of reordering operations are available:

The Promote and Demote buttons cause the selected items in the Source Lib to be moved up or down the list, respectively.

The Sort button sorts the entire list alphabetically, with the numeric part of any name treated as an integer. The adjacent arrow buttons determine the direction of the sort.

Compacting a Library

When parts are deleted from a library, the free space in the file is not automatically recovered. In most cases this is not a significant overhead. However, if a large percentage of the parts in a library have been deleted then you may wish to compact the file. To do this:

- ➔ Create a new, empty library which will become the target for the Compact operation.
- ➔ Select the Lib Maintenance command.
- ➔ Select the library to be compacted as the Source Lib.
- ➔ Select the new, empty library as the Dest Lib.
- ➔ Click on the Compact button.

IMPORTANT: Verify that the new destination library is correct before discarding the old copy.

Using Circuit Elements as Library Items

In addition to regular symbol libraries, DesignWorks allows you to create libraries of circuit elements or scraps that can be used in designs in a manner similar to placing a symbol. This can be used, for example, to keep commonly-used graphics like title blocks and background elements, circuit scraps, groups of symbols in a pre-aligned format, etc. Each element in such a library is actually a complete design file and can contain any desired circuit elements. Dragging such an element from the library palette actually copies the entire contents of the source file into the destination design at the indicated location.

Setting Up a Library of Circuit Elements

A circuit element library is simply a folder that contains normal design files. To set up such a folder:

Place a command in the DesignWorks initialization file (dw.ini) to specify the desired folder using the LIBRARYFOLDER setup file keywords. See “Specifying Libraries to Open at Startup” on page 385 for more information on this.

Place design files (.cct) in the folder containing the desired elements. Each design file will appear as one element in the library list with the file name as the element name. Only items on page 1 of the design are copied

when the element is used.

Operations on Symbols in a Schematic

Making a Single Device Into a Unique Type

This command makes the selected device into a unique type. In other words, even if other devices in the same design were originally drawn from the same library, this one will now be considered to be unique. This is primarily intended for hierarchical designs. This allows the internal circuit for the selected device to be modified or deleted without affecting other similar devices.

As an example, you could create a standard template for a Programmable Logic Device in a library, complete with internal simulation circuit. Several of these devices could be placed in the design, even though they will eventually have different internal programming. The Make Unique Type command would then be used to isolate them so that their internal circuits could be separately updated.

See more information on this in the sections “Associating a Subcircuit with a Device Symbol” on page 230.

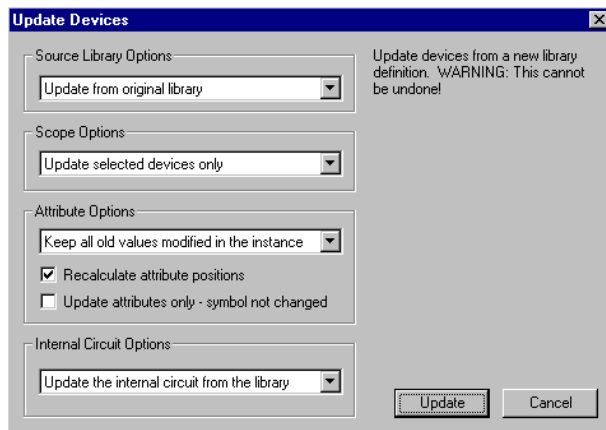
Updating a Symbol from a Library

The Update from Lib command updates the definition of the selected device from its original source library. There are a variety of options available to determine how attributes and subcircuits are updated and how a source symbol definition is selected.

WARNING: Depending upon the options selected, this command may replace the entire internal circuit of the selected device. THIS CANNOT BE UNDONE!

Selecting the Update from Lib command in the Part Type sub-menu of the

Options menu causes the following box to be displayed:



NOTE: The box that appears may be simpler than that shown above. Only options that are relevant to the specific device selected are shown.

The first option pop-up menu allows you to select how the source library will be located:

Original Library

This selection will locate the original library that the device was derived from using information stored in its attribute fields. As long as the library can be located and a part with the same name can be located in the library, you will not be prompted to locate a source.

Select New Source

This selection will cause a sequence of two prompt boxes to be displayed requesting a library and then a part within the library. This new symbol will be used to update the selected device.

Next, you need to select which devices on the schematic you wish to update. The choices are:

Update This Device Only

Only the selected device will be updated. **NOTE:** This in effect redefines the device to be different from others that were originally derived from the same library part.

- Update All Devices of Same Type** This will update all devices in the design that were originally derived from the same part type. Note that this is determined by the internal logical linkage between the device instances and the part definition, not by the type name. If you have multiple type definitions with the same name in your design, this option will only update the devices linked to the same type as the selected one.
- Update All Devices with Same Type Name** This option will update all devices derived from a type with the same name as the selected one, even if the definitions have become separated by some previous operation. This is specifically intended to remedy the case where a design inadvertently contains multiple definitions of similar devices.

The Attribute Update Options pop-up menu allows you to choose how attribute values are updated in the event that the new symbol definition has different values than the existing device instance:

- Keep All Old Values** This selection tells DesignWorks to maintain all attribute values exactly as they are in the current instance, even if the new symbol definition has different values.
- Keep Old Values Modified in Instance** This selection indicates that you wish to keep any attribute values in the device instance that were changed from the default value in the original definition. Any values that still have the default specified in the old symbol will be updated to the new default value in the new symbol.
- Use New Non-null Values** This item indicates that you wish to use the value specified in the new definition, if there is one, but keep any old values that are not overridden by new ones.
- Use All New Values** In this case, the updated instance will have exactly the values in the new definition, even if they are empty. All attribute data present in the old definition will have been replaced by new values.

NOTE: The Name and InstName fields are not modified by the Update operation, regardless of attribute update setting.

Two other options appear in the Attributes section:

Recalculate displayed attribute positions	This checkbox allows you to determine whether displayed attribute values will be left at their old positions or recalculated to fit with the symbol. If the symbol has not changed in shape appreciably and you have manually repositioned the displayed values, you may wish to turn off this option.
Update attributes only - symbol not changed	If this box is checked, only attribute values are affected, the device symbol, pins, subcircuit and other aspects of the part type will not be affected.

Finally, if either the existing instance or the definition in the library, or both, have subcircuits, then you need to choose which circuit you wish to keep.

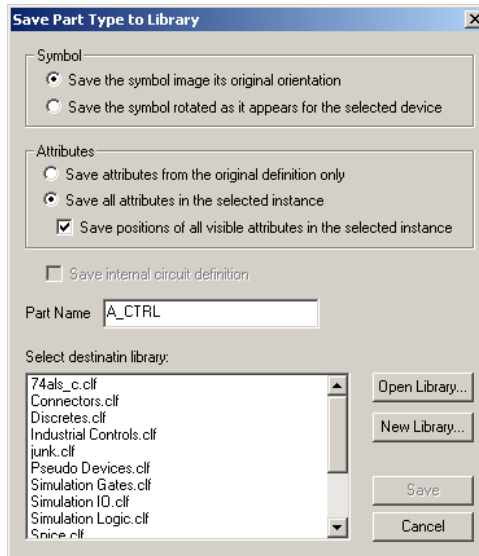
Update Internal Circuit from Library	If the selected device has an internal circuit it is removed. If the new library part definition has an internal circuit as new one is created.
Keep Existing Internal Circuit	Any existing internal circuit is kept and only the symbol is updated. Note that if you make this selection, the part definition in the design will be marked as having been changed since it is not the same as the library definition.

IMPORTANT: Updating Devices in a Hierarchical Design - If your design is in Physical hierarchy mode and you are editing a subcircuit whose parent device is instantiated multiple times, devices that are in other instances of the same subcircuit will not be updated immediately, even if you select "Update All Devices of Same Type". Only the open copy of the circuit will be updated immediately on execution of the command. The other copies of the circuit will be updated when you close the subcircuit and select the Update option.

Saving a Symbol Definition from a Schematic to a Library

The Save to Lib command saves a type definition for the selected device to a

library. The following box will be displayed:



This table summarizes the options available.

Save Symbol Image in its Original Orientation	This saves the type definition with the symbol image as it was originally drawn.
Save the Symbol Rotated as it Appears for the Selected Device	Selecting this option causes the type definition to be saved with the symbol rotated as it currently appears.
Save Internal Circuit Definition	If this box is checked, the internal circuit attached to the selected device will be saved with the part definition.
All attributes in selected instance	If this option is selected, all the attribute values associated with the selected device will be made part of the saved library part.
Save positions of all visible attributes in the selected instance	If this box is checked, then a “.Pt” position field will be created for each attribute that is visible on the instance and for which the associated “.Pt” field is defined in the design’s attribute table. See “Using Default Position Fields” on page 175. for more information.
Attributes in original definition only	If this option is selected, only attributes that were originally defined for the library part will be saved.

Part Name	The part name under which the new library entry will be saved.
Open Library	This button allows you to open any existing library file on disk to add to the destination library list
New Lib	This button will display a standard file save box allowing you to create a new, empty library.

Saving All the Symbols in a Design to a Library

To make a library containing all the symbol definitions used in a circuit:

Ensure that the circuit whose symbols you want to save are in the frontmost window.

Ensure that the library you wish to write the symbols to is selected in the Parts palette. If it is not, either select it from the library selection menu in the Parts palette, or use the New Lib command to create one.

Select the Circuit to Library command in the Part Type submenu of the Options menu.

Click on the OK button.

This command will proceed to write all the unique symbol definitions in the circuit to the selected library. Here are some points to note on the operation of this tool:

In a hierarchical design, only symbols in the current circuit level are written.

Pseudo-devices (i.e. things like power and ground symbols and page connectors) are included in the items written out. Depending on your requirements, you may wish to delete these from the final library.

The symbol written out will include all attribute values defined in the first instance of the symbol that is encountered, except for Name and InstName. I.e. The symbol that is written out is not necessarily identical to the definition used in creating the circuit in the first place.

TIP: One of the purposes of Circuit to Library is to allow you to use the attribute editing tools like the Browser to work on large numbers of symbols at once. You can use Library to Circuit to create a design containing one of each symbol in a library, then editing the resulting

devices and write the modified versions back out. This can be very convenient, for example, if you need to add the same attribute value to all the symbols in a whole library. See “Creating a Design With One of Each Symbol in a Library” on page 275 and “Using the Browser Tool” on page 248.

Editing a Device Symbol in a Schematic

See “Editing an Existing Part on a Schematic” on page 281.

Creating a Design With One of Each Symbol in a Library

There are a number of cases where it is useful to create a design that contains one of each symbol in a library, such as:

- To test a library to ensure that all symbols appear, print and netlist correctly.

- To perform mass modifications on part attributes using the schematic tools such as the Browser. See also “Saving All the Symbols in a Design to a Library” on page 274.

To perform this operation, follow these steps:

- Create a new, empty design using any desired template.

- Open or select the desired library so that it is the current one displayed in the Parts palette.

- Select the Library to Circuit command in the Part Type submenu of the Options menu.

- Click the OK button.

The Library to Circuit command will proceed to place symbols on the schematic in columns based on the sizes of the symbols. If it runs out of room on the page, it automatically creates a new circuit page. The naming of the devices and display of attributes are determined by the template’s settings in the same way as if you had placed the parts manually.

WARNING: This process can require a large amount of memory. Because every device in the resulting design has a unique symbol definition, a large amount of data is loaded into memory while the design is open.

Using the Clipboard in Device Symbol Editor

The standard Edit menu commands Cut, Copy and Paste can be used to move objects inside and between the symbol editor window, DesignWorks circuit windows, and other applications. Some types of graphic objects, notably bitmaps, created by other programs are not supported by the current version of the symbol editor and will not appear if pasted into the editor's drawing area.

Editing Device Symbols

Symbols are created and edited using the device symbol editor tool. In addition to drawing symbols, the device symbol editor can also be used for general graphics (e.g. title blocks or simple mechanical drawings) for use on DesignWorks schematics. It provides a complete, object-oriented drawing environment with standard drawing tools, as well as specific functions tailored for symbol creation.

Creating a New Part from Scratch—Basic Procedure

To create a new device symbol with no initial attribute settings or graphics, either:

Select the New command in the File menu, then choose the Device Symbol document type, or:

Right-click in the Parts palette and select the New Part command.

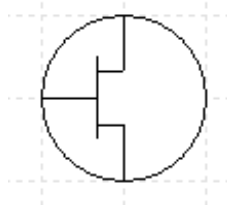
TIP: In many cases, you may wish to start with an existing symbol that has settings similar to the one you require, rather than creating a complete new one.

Step 1—Drawing the Graphics

Draw the graphical shape which represents the part. Do this using the line, rectangle, rounded-rectangle, oval/circle, arc, and polygon tools.

IMPORTANT: Device pins must be added using the pin tools. Do not draw any of the

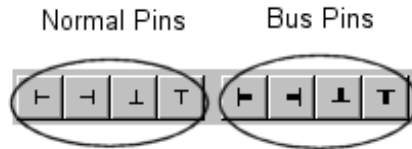
part's pins with the basic graphic tools.



For more detailed information on using the graphical drawing tools, see “Editing Symbol Graphics” on page 288.

Step 2—Adding Pins to the Symbol

Place pins on the part using any of the pin placement tools:



This toolbar provides two types of pins:

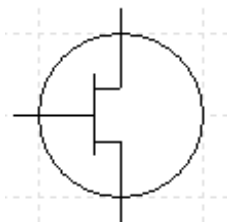
The first group allows placement of normal pins (i.e. not bus pins).

The second group is used to place bus pins, i.e. pins that represent a group of named internal signals.

NOTE: Since we are creating a discrete component symbol with no text on it, we should first disable the pin name display, which is on by default. To do this click in an unused part of the drawing area to make sure nothing is selected, then selected the Properties command in the Objects menu. Select the Pin tab and turn off the Visible switch.

Note that the crossbar portion of the T pin symbol is shown only for alignment purposes and indicates where the pin attaches to the body of the symbol. It is not drawn when the symbol appears in a schematic. Here is what the

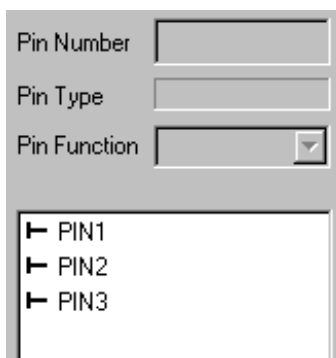
example symbol looks like with the three pin elements placed:



For more background and alternate procedures for pin placement, see “Placing Pins on a Symbol” on page 292.

Step 3—Setting Pin Information

Once you have placed the pins on the symbol, you will see that an entry in the pin list has been added for each pin.



For this example we want to name the pins “Source”, “Gate”, and “Drain”. In addition we want to give them the following pin number “S”, “G”, and “D”.

NOTE: Depending on the order in which you placed the pin symbols, they may not be in the list in the same order as this example. You can check the association of names to pins by clicking on an item in the list and observing which pin on the drawing is highlighted.

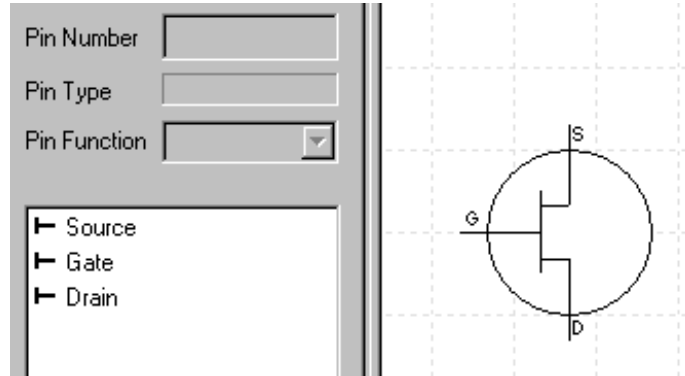
To change the name of a pin:

- ➔ Double-click on the item in the list.
- ➔ Enter the name, in this case “Source” for example.
- ➔ Press the Enter key to terminate editing.

While this first pin is still highlighted in the list, we can enter its pin number by typing it in the Pin Number box. Enter “S” for this example. When you click anywhere else, you will see the pin number appear in the graphics.

Depending on your application, the Pin Type setting may or may not be significant. See “Setting the Pin Type” on page 299 for more information. In this case, we will leave it as the default “input” type.

Set the pin name and number for the other pins in a similar fashion.



NOTE: Some netlist formats, notably SPICE, require that pins appear in a certain order in the output. Unless you specify otherwise using attributes, this order will be determined by the order in which the pins are placed on the symbol. See Chapter 6—Before Starting a Major Design on page 113 for more information. Pins can be reordered by simply dragging them to a new position in the list.

The actual names given to the pins in this simple form of part are not critical. It is a good idea though to use meaningful names for these reasons:

These names will be seen in the Schematic tool's pin properties dialog.

They can be extracted in netlist output.

They are used when binding pins to ports in subcircuits.

In the case of bus internal pins, they are used when connecting bus pins on the part to busses.

Step 4—Saving the Part

To save the part to a library:

➔ Select the Save As item in the File menu.

- ➔ Enter the name for the part. This is the name that will appear in the Parts palette.
- ➔ Choose the desired destination library.
- ➔ Click Save.

IMPORTANT: The procedure given here allows you to produce a symbol with only the simplest graphical and netlisting requirements. Details of creating the symbol and the various settings that may be required for specific applications are covered in later sections of this manual, and in other chapters on specific DesignWorks functions: You may wish to refer to any of the following sections:

A simple procedure for creating simple schematic symbols and hierarchy blocks is given in the tutorial section entitled “Device Symbol Editing and Hierarchical Design” on page 37.

For details on the drawing tools available for creating symbols, see “Drawing the Graphics and Placing Pins on the Subcircuit Symbol” on page 303.

For information on the general entry of part and pin attributes, see “Setting Part and Pin Attributes” on page 285.

For pin settings and how they may affect simulation or netlists, see “Step 3—Setting Pin Information” on page 278.

For information on creating symbols for hierarchical blocks, see “Creating a Block Symbol” on page 222.

For details on adding gate packaging information to a symbol, see “Gate Packaging” on page 314.

For assistance in determining what information should be in a symbol to ensure correct netlist generation and interfacing, see Chapter 6—Before Starting a Major Design on page 113.

For information on creating pseudo-device symbols, see “Creating a Power and Ground (Signal) Connector” on page 308 and the sections that follow it.

Editing an Existing Part in a Library

To edit an existing part in a library:

- ➔ Right-click on the desired item in the Parts palette, then select “Edit Part” from the pop-up menu.

In response to either of these operations, a copy of the symbol definition is loaded into a device symbol editor window. No changes to the source library will be made until you save the symbol back to its original library.

NOTE: Editing a symbol in a library does not automatically update designs that have used that symbol. For more information, see “Designs and Libraries” on page 263.

Editing an Existing Part on a Schematic

To edit the symbol of a device that is already placed in a schematic without modifying the library it came from, right-click on the device and select the Edit Symbol command. This actually saves the symbol to a temporary library, and then opens it in the symbol editor. You can then make any desired changes to the symbol and close the editor. At that point, you will be asked to confirm that you wish to replace the original symbol, and, if applicable, any others of the same type in the schematic.

If the symbol on the schematic has been rotated from the orientation it was originally drawn in, this command will display an option to use either the original orientation or the new one.

NOTE: This process is similar to using the Update from Lib command to update an existing device instance from an updated library and has many of the same options. See “Updating a Symbol from a Library” on page 269.

Closing the Device Symbol Editor Window

An open device symbol editor window can be closed by either clicking in the close box at the upper left corner or selecting the Close command in the File menu. If any changes have been made to the open part, you will be prompted to save or discard the changes.

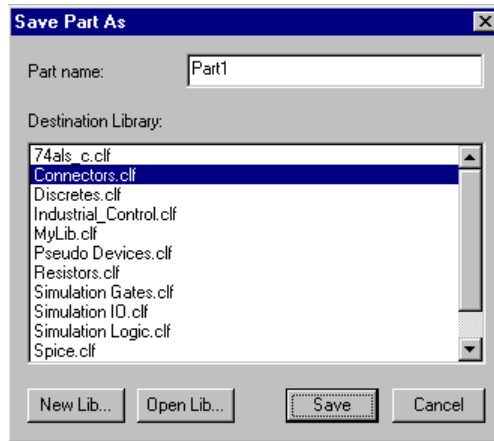
Saving an Edited Part Back to its Original Library

The Save command saves the contents of the current window back to the library it was read from. If the open part was not read from a library (i.e. it was just created), this item will be disabled.

Saving the Part Under a New Name

To save an open symbol under a new name, or to a library other than the one

from which it was read, select the Save As command in the File menu.



When you select this menu item, the “Save As” dialog will appear. It requires that a library name be selected from the list and that a name for the part be entered in the lower box. In this example the Connectors library has been selected and the part's name has been left to default to “Part1”. A name entered will become the name of the part in the library, and the title of the window will be updated to correspond to the new part name.

Only open libraries appear in the list. You can use the New Lib and Open Lib buttons on this box to create a new, empty library, or open any other existing library to complete the save.

WARNING: We do not recommend saving your own parts in the standard libraries provided with DesignWorks, although the program will not prevent you from doing so. When installing future program upgrades, these libraries may be replaced automatically, erasing any changes you have made!

Zooming the Symbol Editor Window

The Reduce/Enlarge/Normal Size commands in the View menu allow you to adjust the scale at which an object is viewed. The default setting for the device symbol editor is to display objects at the same size as they will appear in the schematic at Normal Size.

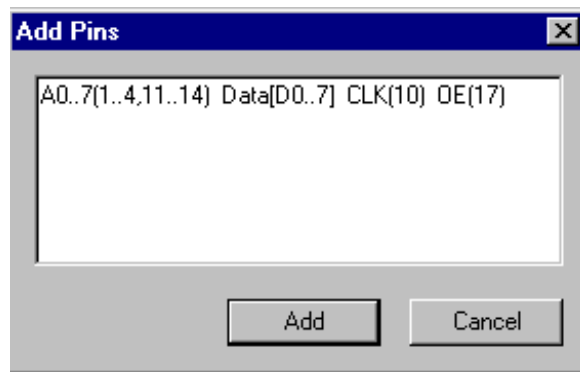
There is also a Magnifying Glass tool  in the toolbar which allows you to select an area of the symbol to zoom in on.

Adding Sequential Pin Names

The basic procedure for adding pins to a symbol is described in “Step 2—Adding Pins to the Symbol” on page 277. However, the basic procedure requires that the pin name and number be entered manually for each pin, which can be rather tedious for large numbers of sequentially numbered pins. To solve this problem, the Add Pins command allows you to enter a sequence of names in a format similar to that used for bus breakout creation and for the Auto Create Symbol command. These pins are added to the pin list, but not placed on the symbol. After the pins have been added to the pin list, you can place the pins sequentially on the graphic with no further typing required.

TIP: You can use the Auto Create Symbol command to specify the pin names and auto-generate a rectangular symbol in one operation. See “Automatically Creating Symbols” on page 303 for more information.

The Add Pins command displays the following box:



Pin names, and optionally their pin numbers, may be entered into the box. The pins are created and merged with the contents of the pin list when a carriage return or enter key is pressed or when the Add button is clicked. The Add Pins palette does not create graphic pins, only pins for the pin list.

The created pins are merged with the names in the pin list. If a like named pin already exists in the list then it may be reordered to appear in the same order as in the Add Pins palette. If a pin being added has a pin number defined then this pin number will replace the pin number in the like named pin.

Syntax for Pin Names in Add Pins

Here are the rules for how a list of pin names is entered:

Pin names may be up to 16 characters long.

Normal pin names (that is, not bus pins or bus internal pins) may be specified as individual names, e.g.: A B C D, or as sequences, e.g.: A0..3. After each pin name or pin name sequence an optional pin number specification is allowed.

A pin number specification defines the pin number(s) associated with the proceeding pin name or pin name sequence. The specification starts with an (and ends with an). For a single pin name the pin number specification should contain a single number. For a pin name sequence multiple numbers and sequences may appear between the brackets.

In the picture above CLK and OE both define normal pin names which have pin numbers 10 and 17, respectfully. The sequence A0..7 defines the pin names A0 through A7. These pin names were also given pin numbers. The pin numbers 1 2 3 4 11 12 13 and 14 were assigned.

It is also possible to skip a pin name when assigning pin numbers to a pin name sequence. Consider the previous example, if we didn't want to assign the number 11 but wanted all other pin names to get the same number, we would do the following: A0..7(1..4,,12..14) instead of A0..7(1..4 11..14). Two commas in a row causes a pin to be skipped when assigning the pin numbers. Three commas cause two pins to be skipped. Four commas, etc.

Bus pin names are denoted by a name followed by [...]. Bus pins do not need to have internal pin names defined and may not have pin numbers.

Bus internal pin names must appear between a [and a]. They have the same format as normal pin names and may have pin numbers. Internal pin names may be added without adding a bus name pin by not placing a bus name in front of the []. For example, [A B C] adds the internal pin names "A", "B", and "C" to the pin list.

The pin function (input, output, etc.) can be specified by placing a "|" character followed by a letter denoting the type. All following pins will have the specified type until another "|" specification is found. The

allowable pin type characters are:

I	Input
O	Output
3	3-state output
B	Bidirectional
C	Open collector
E	Open emitter
L	Low
H	High
N	No connection
P	Power
D	Driver
A	Analog

For example:

```
|I CLK,,D0..7 |O Q0..7
```

Deleting Pins

In a device symbol editor window, pins exist both in the graphic representation of the symbol and in the pin list. They have to be deleted from both places before they are completely removed from the part definition.

To remove a graphical pin from the symbol, simply select it by clicking on it and delete it with the delete key in the usual way. Note that this does not remove the corresponding entry from the pin list. If you wish to replace the pin, for example to use a different orientation, you can simply select the item in the list and place the appropriate pin graphic from the palette.

To remove the selected pins in the pin list and their associated graphic pins:

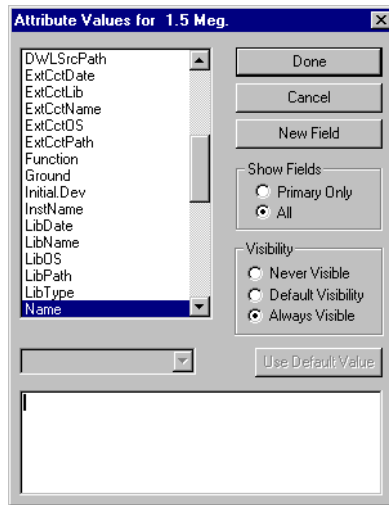
Select the pin or pins in the pin list that you wish to delete. You can select more pins by holding the **Shift** or **Ctrl** keys while clicking in the list.

Hit the Delete key on the keyboard. It will display a confirmation box before proceeding to delete the pin entries.

Setting Part and Pin Attributes

To edit the attributes associated with a part select the Part Attributes com-

mand in the Options menu. To edit the attributes associated with a pin, select a pin in the pin list and then select the Pin Attributes command from the Options menu. The only visible difference between the part attribute dialog and the pin attribute dialog is the addition of “Next” and “Previous” buttons at the bottom of the window.



Attributes defined while in the device symbol editor will be associated with the part in the library. These will become the default attributes for the part when it is placed in a circuit.

IMPORTANT: Only attribute fields that have a value are stored with the symbol. When you save the symbol to a library, any fields with null values are stripped off to save storage. This has the side effect that you cannot create a “place holder” field definition in a symbol for future use without putting a value in it.

Some visibility options appear in this version of the attributes box that are different from what you will see on a device placed in a schematic:

Always Visible This setting indicates that the select field should always be made visible on the schematic when the device is placed, regardless of the visibility setting for this field in the target design. This setting does not prevent the field from being removed after the device is placed, it just sets the initial state.

- Default Visibility** This setting indicates that we wish to make the field visible only if it is defined as Visible by Default in the design in which it is placed.
- Never Visible** This indicates that the field value should not be displayed when the device is placed, regardless of the design's Visible by Default setting for this field. This does not prevent the value from being displayed later, it just sets the initial state.

TIP: You can set the default visibility independently for each attribute field defined in a design. This is done by selecting the Define Attribute Fields command in the Options menu and setting the “Visible by default” option as desired for each field.

For more information on how to create and modify attributes see “Features Requiring Symbol Attributes” on page 313.

Attribute Field Definitions in Symbols and Designs

In DesignWorks, symbol definitions are completely independent of any particular design or library file. For this reason, symbols carry around their own definitions of the attribute fields that are associated with them. By default, when you create a new symbol, you see the attribute definitions for the design that is currently open, or a default set if no design is open. However, the New Field button in the Part Attributes box allows you to add new field definitions that affect only this one part definition. This operation does not automatically add the same field to the current design.

When you place a symbol in a design for the first time, the program compares the attribute fields defined in the symbol to those defined in the design's symbol table. If there are any new fields in the symbol, you will be asked if you wish to add them to the design's table. Your choices in this case are to either:

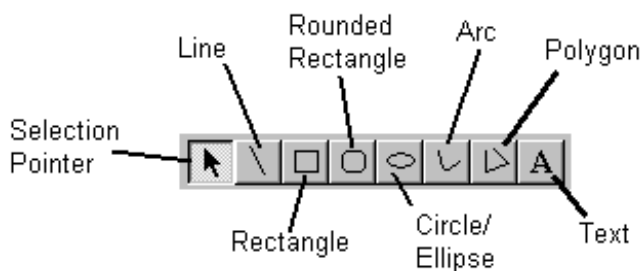
Click the OK button and allow the program to automatically add the field definitions to the design's attribute table. The program will attempt to make reasonable settings for the maximum length, default visibility and other settings, based on the current value, or,

You can click Cancel to stop the device placement and add the fields manually to the design's table using the Define Attribute Fields command. In general, this is preferred, since it gives you more control over the option settings for the field.

Editing Symbol Graphics

Using the Drawing Tools

The 8 items in the top half of the toolbar represent standard drawing tools.



Drawing Lines, Rectangles, Circles, Etc.

The shape tools in the toolbar, and the corresponding commands in the Tools menu, are used to draw standard graphic objects. Here are some notes on the usage of these tools:


Once an item is drawn, it can be repositioned or resized by using the selection pointer tool and clicking and dragging on the “handles” at the ends of the object.

While drawing or adjusting an object, you can use two keys on the keyboard to change the way the object is positioned:

Holding the **Shift** key down forces the object to be vertical, horizontal, or the same on both axes. In other words, if you are drawing a rectangle, it will be forced to a square shape, an ellipse will be forced to a circle, and a line will be forced to vertical, horizontal or 45 degrees, whichever is closest to the current position.

Holding the **Cmd** key down, disables the Snap to Grid option, if it is currently on. This allows you to micro-position graphic items without reference to the grid.

The arrow keys on the keyboard can be used to “nudge” selected objects slightly in any direction to achieve finer positioning than is possible with the mouse.

Holding the  key down while using the arrow keys, adjusts the size of an object in one-screen-dot increments.

Setting Line Width, Line Color and Fill Color

The Line Width, Line Color and Fill Color commands in the Objects menu allow you to control these visible characteristics of any group of selected objects. If there are selected objects when an item is changed, then the selected objects will be given the new characteristic. If there are no graphical objects selected when a change is made, the default setting is changed and graphical objects created in the future will use the new default setting.

Drawing Arcs

Arcs are drawn in a manner similar to the other graphical objects, except that an additional command is available to control the start and stop angles of the arc. In a manner similar to the other object property commands, you can either select an existing arc and then use the Properties command to change its characteristics, or you can select Properties first to set the default settings and then draw the arc with the arc tool.

Drawing Rounded Rectangles

Rounded rectangles are drawn in a manner similar to the other graphical objects, except that an additional command is available to control the radius of the rounded corners. In a manner similar to the other object property commands, you can either select an existing arc and then use the Properties command to change its characteristics, or you can select Properties first to set the default settings and then draw the item with the rounded rectangle tool.

Drawing Text

To draw text items:

- ➔ Click on the text tool (A) in the toolbar.
- ➔ Click in the desired starting location in the drawing area.
- ➔ Type the text on the keyboard.
- ➔ Click anywhere outside the text box to terminate entry.
- ➔ If you want to change the font or size settings, switch to the pointer tool and select the text item. Then choose the Text Font command in the Objects menu, or select any of the available rotation options in the Text Ro-

tation submenu.

Reordering Graphical Objects Front-To-Back

The Bring To Front and Send To Back commands are used to set the front-to-back ordering of the selected objects relative to the other graphic objects.

Grouping Graphical Objects

The Group and Ungroup commands allow you to make multiple graphic objects, except pins, be treated as a single object or visa versa.

Aligning Graphical Objects

The Align sub-menu allows you to pick how the selected objects will be aligned. For example, Align Left causes all of the selected objects to be moved such that their left edges are aligned with the left most selected object's left edge.

Rotating and Flipping Graphical Objects

Any object or group of objects can be rotated 90 degrees or flipped on either axis by one of these methods:

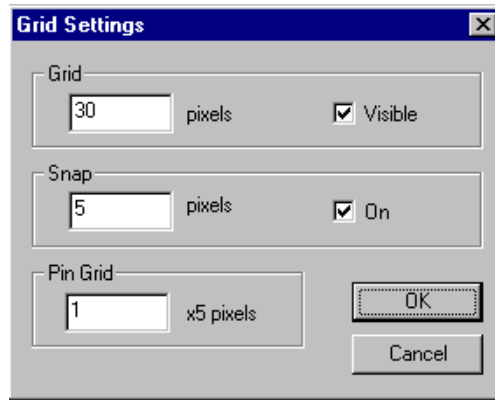
Click on the object to select it, then select the desired command from the Rotate/Flip submenu in the Objects menu.

Right-click on the object and select the desired command from the Rotate/Flip submenu.

Setting Grids

The Grids command allows you to specify the visible grid spacing and the

snap-to grids for objects drawn using the drawing tools.



Display Grid Checkbox This check box determines whether visible grid lines are shown in the drawing workspace of the symbol editor window. The spacing between these grid lines is determined by the value in the “Grid Spacing” field.

Snap On Checkbox This check box determines whether the corners of objects made with the drawing tools are moved to the nearest snap-to grid point.

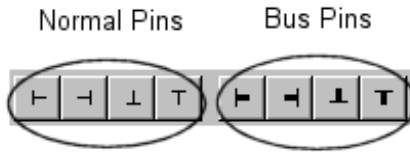
Grid Spacing This number determines the spacing between the visible grid lines. The units are in screen dots at the default zoom level.

Snap Spacing This number determines the spacing between snap-to points for the drawing tools (not including pins). This does not affect objects that have already been placed. The units are in screen dots at the default zoom level.

Pin Snap Spacing This number determines the snap-to grid for device pins. This must be a multiple of 5 to meet the DesignWorks pin grid requirements. The units are in screen dots at the default zoom level.

Placing Pins on a Symbol

The toolbar contains the tools needed to place connection pins on the symbol.



The first group of pin tools is used to create normal pins (i.e. not bus pins) on the part in any of the four orientations.

The second group of pins are used to place bus pins on the part in any of the four orientations. Bus pins are described in more detail in “Placing a Bus Pin” on page 295.

TIP: You can also place pins or groups of pins from the Symbol Gallery window. If you have special types of pin graphics or groupings of multiple pins that you use often, you can place these in the Symbol Gallery for quick access. See “Adding Elements to the Symbol Gallery” on page 297 for more information.

When placing pins, a graphical pin is associated with a name in the pin list. The association is made by applying the following rules and using the first one that matches.

Associate a selected pin name in the pin list which is unplaced and is of the same type. I.e.: normal pins can't be associated with bus pin names or internal bus pin names.

Associate an unplaced pin name of the correct type which follows the first selected pin name. Pin names are examined in a cyclic order so if the bottom of the pin list is reached the search continues from the beginning.

If no association is made then a new pin name of the correct type is created and added to the bottom of the list.

Showing, Hiding, Editing or Moving a Pin's Name

By default, when you place a pin on the symbol, the name of the pin is included in the symbol adjacent to the pin graphic. Here are a number of ways of changing this name display:

The Symbol Editor's Pin List

The pin list box contains a scrollable list of the pin names associated with this device. This list is derived from the following sources:

If the symbol was opened from an existing part, then the initial name list will be the pin names associated with the part.

If a pin tool is clicked in the drawing workspace, a new name may be added with the form “PINxx” where xx is a sequential number. This is only done if there were no unplaced pins in the list.

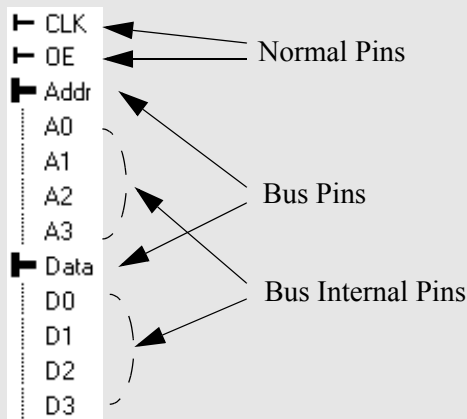
If an open DesignWorks circuit is selected as an internal circuit, or a saved DesignWorks circuit is selected as an external circuit, then the names of the port connectors in the circuit will be merged with the names in the pin list.

The Add Pins command (page 283) can be used to create a list of pins to be merged with the pin names in the pin list.

The Auto Create Symbol command (page 304) can be used to specify all the pin names and pin numbers for a device symbol.

When the automatic “show pin name” feature is used, the pin name that appears in the pin list must be exactly the same as the graphical annotation which appears next to the pin in the part's symbol. The result is that changing a pin name in the pin list will cause the annotation next to a graphical pin to automatically change. For cases where it is necessary to show a pin annotation that is different from the pin's logical name, you can hide the automatic pin name and use a normal text object to add any desired label. See “Showing, Hiding, Editing or Moving a Pin's Name” on page 292.

Any pin that has been NOT been placed (i.e. it has no corresponding graphical pin object which appears as part the symbol) will be shown in red in the list. Bus pins are marked in the list with a bolder pin icon and will be followed by their internal pins.



To show or hide the pin name on one or more pins, select the pins on the symbol or in the pin list, then select the Properties command in the Objects menu. On the Pins tab, turn the Visible switch on or off as desired. Alternatively, you can right-click on a single pin and check or uncheck the Show Pin Name command.

You can change the default name visibility and text font so that you can place a number of pins in a row with the same settings. To do this, click in an unused area of the drawing window to deselect all graphic objects. Next, select the Properties command in the Objects menu and click on the Pin tab. Turn the Visible switch on or off or use the Text Font button as desired. Future pins will then use this setting until it is changed.

To move a displayed pin name relative to its pin, you must first “unlink” it from the pin by right-clicking on the pin and selecting the Unlink Name command. This converts the label to a normal text object which can be moved and set as desired. You then “link” it back to the pin by right-clicking on the text item and selecting the Link to Pin command. This associates it with the pin so that it again follows the pin when moved. **IMPORTANT:** The Link to Pin command searches for a pin with exactly the same name as the given text. You cannot link an arbitrary text item to any pin.

If a placed pin (i.e.: it is shown with a black icon next to its name) is selected then both the pin and the graphical name will be selected. The opposite is also true. If a graphical pin is selected the associated item in the pin list will be highlighted as well. Internal pin names never have graphical associations, therefore selecting an internal pin name never selects a graphical pin.

The relationship between pin names in the pin list and graphical pins is not symmetrical. Every graphical pin must have a pin name, but every pin name does not necessarily have an associated graphical pin. This can lead to some surprises. For example,

Selecting all graphical pins in the drawing workspace may not select all pin names: unplaced and internal pin names will not be selected.

Deselecting all graphical pins in the drawing workspace may not deselect all pin names. Additional unplaced or internal names may have been initially selected.

NOTE: There is no rule that says you have to use the “Show Pin Name” feature to display pin names on the symbol. This is a convenient way of labelling pins, but does restrict the label to be exactly the same as the logical pin name. If you wish to use normal text objects to create pin notations, you can certainly do that, but you will then be responsible for keeping everything aligned if you move the pins.

To edit a pin’s name or function, right-click on the pin on the symbol and select the Properties command. Use the settings on the Pin tab to change the pin name and function. Alternatively, you can modify a pin’s name by double-clicking on it in the pin list. See “Setting the Pin Name” on page 298 for more information on name usage and restrictions.

Setting the Default Pin Name Prefix

By default, when a new pin is added to the symbol, it is named PINx, where x is a number that will be incremented automatically. For bus pins, the default prefix is BUS. There are a couple of ways of controlling this name prefix:

The default name used for new pins can be set in the initialization file. See “Default Pin Name” on page 386 for more information.

When pins are added from the Symbol Gallery, the existing name of the pin (i.e. the name it was stored with) is used as the prefix when creating a new name. See “Using Elements from the Symbol Gallery” on page 296 for more information.

Placing a Bus Pin

Bus pins allow busses to be connected to the part. A bus pin's functionality is determined by the internal pins it contains. These can be specified when the symbol is created, or modified later on the schematic using the Bus Pin Options command described in “Using Bus Pins” on page 196.

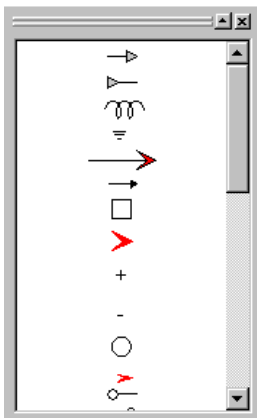
To add a bus pin to a symbol you need to perform two steps:

Use the Add Pins command to add the bus pin to the pin list and specify its bus internal pins. This command is described in “Adding Sequential Pin Names” on page 283.

Place the graphical bus pin using the techniques described in “Placing Pins on a Symbol” on page 292.

Saving Frequently-Used Graphics and Pins

The Symbol Editor has a special graphics palette, called the Symbol Gallery, that can be used to save frequently used graphic elements, customized pin types, and groups of elements for future use. For example, if you frequently need to add pins to a symbol with special graphics indicating the function of the pin, you can add one complete group to the Symbol Gallery and then simply drag and drop it into future symbols when needed.



Displaying the Symbol Gallery Window

To display the Symbol Gallery panel, select the Symbol Gallery item in the View menu.

Hiding the Symbol Gallery Window

To hide the Symbol Gallery panel, select the Symbol Gallery item in the View menu to uncheck it, or click the X “go away” box in the upper right corner of the panel.

Using Elements from the Symbol Gallery

To add an item displayed in the Symbol Gallery list to the symbol editing window, simply drag and drop it at the desired location. Here are some additional notes on this operation:

The complete item dragged from the list will be added to the current symbol as a grouped graphic item. If you then wish to edit the individual elements of the new item, you can right-click on it and select the Ungroup command.

The placed item can be rotated to a new orientation by right-clicking on the item and choosing one of the Rotate or Flip commands.

If the placed item contains one or more pins, these will be added to the pin list or matched with existing, unplaced items in the list. The program attempts to use the name that was assigned to the pin when it was added to the element, but will add or increment a numeric portion of the name in order to create a unique name.

Adding Elements to the Symbol Gallery

The Symbol Gallery is simply a normal symbol library file which is selected to have its contents displayed in the Symbol Gallery list. Thus, all the usual methods for creating, editing and updating symbols can be used on items in the library.

The Symbol Gallery file is not normally open by default, so it must be explicitly opened before you can save items to it or edit it. To open this file, right-click in the parts list and select the Open Lib command. Use the file open box to select the Symbol Gallery file.


In a default DesignWorks installation, this file is called “symbol_gallery.clf” and is located inside a folder called Data Files inside the DesignWorks program folder. If the DesignWorks initialization file has been modified to specify a different file, you will have to locate that file. See “Symbol Gallery Location” on page 387 for information on how to locate this item in the initialization file.

Specifying a Symbol Gallery File

The source of the items in the Symbol Gallery list is a single, standard DesignWorks library file that is specified in the INI file. See “Symbol Gallery Location” on page 387 for information on how to specify this item in the initialization file.

Entering Pin Information

Selecting Items in the Pin List

Clicking on an item in the pin list selects that item and the associated graphic in the drawing area, if any. The  key on the keyboard can be used to add to an existing selection, in the usual way.

Setting the Pin Name

To set a pin name:

- ➔ Double-click on the pin in the pin list.
- ➔ Type the new name.
- ➔ Press Enter to finish name entry.

Here are the rules for pin names:

Names are limited to 16 characters.

Pin names must be unique within a part, except for bus internal pins. Bus internal pins only need to be unique within their own bus.

The program does not restrict what characters can be used in pin names, but we recommend using only alphanumeric characters and a limited set of punctuation characters unless you have some particular reason to use other symbols. Some netlist formats may require the pin name to be exported and may have restrictions on naming. See Chapter 6—Before Starting a Major Design on page 113 for more information.

TIP: You can also set the pin name, number and function by right-clicking on the pin graphic or on the name in the list and selecting the Properties command.

Setting the Pin Number

A four character identifier is displayed in the pin number field. Even though this is referred to as a number it may contain any valid character. For example, “10”, “Q4”, “E123”, and “In” are all valid pin numbers. The value entered for the number is displayed on the stem of a pin.

Setting the Pin Type

The pin type (e.g. input, output, etc.) is selected using the Pin Function pop-up menu. The pop-up will only be displayed for pins of type “Normal” or “Bus Internal”. Pins of type “Bus” do not have their own type.

Using the pop-up menu the following functions may be assigned to the pin: Input, Output, Tristate, Bidirectional, Open Collector, Open Emitter, Tied High, Tied Low, Latched Input, Latched Output, Clocked Input, Clocked Output, Clock Input, and No Connect.

The pin type is important to simulation - is this pin driving or driven, and additionally can be important in error checking and report and netlist generation.

See Appendix C—Device Pin Types on page 373 for more information on pin types.

You can apply a new pin type setting to any number of selected pins. If more than one pin is selected in the pin list then the function pop-up menu will show a current value only if all selected items are the same. In any case, selecting a new setting will affect all selected pins.

Displaying the Pin Name

To automatically display the name of a pin next to the pin graphic, you can either:

Select the pin and choose the Properties command, then check or uncheck the Visible box, as desired.

Right-click on the pin and check or uncheck the Show Pin Name command, as desired.

See “Showing, Hiding, Editing or Moving a Pin’s Name” on page 292 for more information.

Reordering Pins in the Pin List

Pins can be reordered in the list by simply clicking and dragging them to the desired new position. This pin order does not affect the graphical appearance of the symbol, but may affect netlists generated from schematics containing the symbol. For example, the SPICE netlist format depends upon the device pin order matching the order expected by the target simulator.

For more information on pin order and how it affects interfaces to other systems, see Chapter 6—Before Starting a Major Design on page 113.

Creating a Part With a Subcircuit

This section describes how to associate a hierarchical subcircuit definition with a part stored in a library. This is useful if the subcircuit definition will be relatively unchanging and is likely to be used in a number of different designs.

It is also possible to associate a subcircuit with a device after it has been placed on the schematic. See “Creating a Hierarchical Block - Top Down” on page 221 and “Creating a Hierarchical Block - Bottom Up” on page 224.

Creating the Port Interface

A part's subcircuit is a schematic circuit which is associated with the part such that it is considered to be the part contents. A circuit which is to be used as a subcircuit must include parts called port connectors. Port connectors, which are named, allow signals in the circuit to be associated with pins on an enclosing part. Port connectors make this association by name, i.e.: a port connector named “A0” will only associate with a pin with the same name.

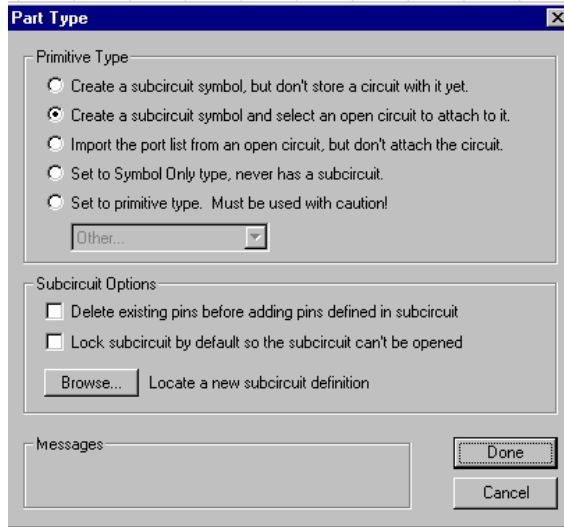
See “Creating and Using Port Connectors” on page 237 for more information on the port interface.

Selecting the Subcircuit

The easiest method to create a part with an associated subcircuit is to begin by selecting the schematic circuit to be used as the subcircuit.

To select the circuit to be used as the part's subcircuit you must also define the type of part being made. Select the Subcircuit and Part Type command from

the Options menu.



This dialog allows you to select among several different options:

Create a subcircuit symbol, but don't store a circuit with it yet

This is the default. It indicates that the part being made has no associated subcircuit, but it doesn't rule out that a subcircuit may be attached by the schematic capture section of DesignWorks.

Import the port list from an open circuit, but don't attached the circuit.

This is similar to the No Subcircuit option. No subcircuit becomes associated with the part, but a port interface is extracted from a circuit for use with the part. Selecting this option prompts you to pick an open schematic circuit. The port names are read from the circuit without attaching the circuit to the part. Like the above option, this doesn't stop a subcircuit from being attached later. Since a subcircuit wasn't attached to the part, the port interface is only associated with the part for the length of the editing session.

Create a subcircuit symbol and select an open circuit to attach to it.

This option prompts for an open circuit to associate with the part being created as its subcircuit. The circuit definition is saved with the part in the library. This operation also imports the circuit's port interface so that the names of the ports appear in the list and you can place the corresponding graphical pins on the symbol.

Set to Symbol Only type, never has a subcircuit.

This option is like the No Subcircuit option except it doesn't allow the schematic capture part of DesignWorks to associate a subcircuit with the part in the future.

Set to primitive type. Must be used with caution!

This option is used to create special DesignWorks part types such as pseudo devices (power, ground and port connectors) and devices for use with the DesignWorks Simulator option. The primitive type options must only be used with a clear understanding of the effect they will have on program operation and are described in "Creating Special-Purpose Symbols" on page 312.

There are some subcircuit options in the lower part of the dialog:

Delete existing pins before adding pins defined in subcircuit

This option is only enabled when the selection made above results in a port interface being imported. If this box is checked, when the "Done" button is pressed all of the old pins in the pin list will be deleted. This allows a new port interface to be brought in without any conflicts with the existing pin list. If this option wasn't checked then port names would be merged with the names in the list. Duplicate pin names and their related properties remain unchanged, except they are now associated with the new port. Unmatched pin names in the pin list remain exactly the same.

Lock subcircuit by default so the subcircuit can't be opened

This option has the effect of saying, “Yes. There is a subcircuit, but in general you don't want to go into it”. This causes the schematic capture part of DesignWorks to prompt to make sure it is really OK to enter the subcircuit before doing so. It also controls if the report generator will list the contents of this device in a netlist. In general this is used for symbols that represent physical parts, but there may be a subcircuit for the simulation purposes.

Locate a new subcircuit definition

This button allows you to replace the subcircuit in a symbol that already has a circuit associated with it.

Drawing the Graphics and Placing Pins on the Subcircuit Symbol

The graphic image of the part may be drawn and the pins placed on it in the same way as was described in “Editing Device Symbols” on page 276.

You can also generate a symbol automatically using the Auto Create Symbol command. See the section Auto Creating a Symbol.

Opening the Subcircuit Associated with a Symbol

If the symbol currently being edited has a subcircuit already stored with it in the library, you can use the Open Subcircuit command in the Options menu to open it for editing. This opens the subcircuit in a design window as if it was an independent design.

IMPORTANT: Modifying and saving the design that was opened with this command DOES NOT automatically update the symbol or the library it was read from. If you wish to update the symbol, you must use the Subcircuit and Part Type command and use the “Create a subcircuit symbol and select an open circuit to attach to it” option to reattach the modified circuit to the symbol.

Automatically Creating Symbols

The Auto Create Symbol command in the device symbol editor tool will gen-

erate a standard, rectangular symbol given a list of names for the pins on each side of the symbol.

Auto-creating Rectangular Symbols

The Auto Create Symbol command will create standard rectangular symbols given a list of the desired input and output pin names. For maximum flexibility, the symbol generated consists of separate graphic objects and is completely editable after it is generated.

The current settings for line width, fill patterns, color, text font, size and style, etc. are used in generating the symbol. The only exception to this is the type name text placed in the center of the symbol, which is written 3 points larger than the current setting and in bold.

Selecting the Auto Create Symbol command displays this box:

The pin name boxes will contain the information entered the last time the device symbol editor was invoked for this part. These can be modified as desired. The new settings will be merged with pin list when the Generate button is pressed.

Entering Pin Names

The four pin name boxes allow you to specify the names of pins to appear on the left, right, top and bottom of the device symbol. Here are the rules for how a list of pin names is entered:

Pin names may be up to 32 characters long.

Pins are specified bottom to top.

Since spaces are used to separate names, it is not possible to specify names containing spaces with this command

Normal pin names (that is, not bus pins or bus internal pins) may be specified as individual names, e.g.: A B C D, or as sequences, e.g.: A0..3.

After each pin name or pin name sequence an optional pin number specification is allowed.

A pin number specification defines the pin number(s) associated with the preceding pin name or pin name sequence. The specification starts with an opening parenthesis “(” and ends with a closing parenthesis “)”. For a single pin name the pin number specification should contain a single number. For a pin name sequence multiple numbers and sequences may appear between the brackets.

Bus pin names are denoted by a name followed by “[...]”. Bus pins do not need to have internal pin names defined and may not have pin numbers.

Bus internal pin names must appear between a [and a]. They have the same format as normal pin names and may have pin numbers. Internal pin names may be added without adding a bus name pin by not placing a bus name in front of the []. For example, [A B C] adds the internal pin names A, B, and C to the pin list.

The pin type can be specified by placing a “|” character followed by a letter denoting the type. All following pins will have the specified type until another “|” specification is found. The allowable pin type characters are:

I	input
O	output
3	3-state output
B	bidirectional
C	open collector
E	open emitter
L	low
H	high
N	no connection

P	power
D	driver
A	analog

Items in a list can be separated by blanks or commas. Placing an extra commas between two items adds extra space between the pins on the symbol. A single comma is ignored (i.e. it serves just to separate the items), 2 commas inserts one extra grid space, etc. Additional space can be added with more commas.

Auto Create Pin List Examples

Here are some examples applying the rules discussed above:

A B C D	Adds 4 input pins A, B, C and D
D0..7	Adds 8 input pins D0, D1, D2 ... D7
D7..0	Adds 8 input pins D7, D6, D5 ... D0
O Q0..7	Adds 8 output pins Q0, Q1, Q2 ... Q7
B DATA[D0..31]	Adds a bus pin with 32 internal pins of type bidirectional
CLK,,,D0..7	Adds a single pin called CLK, then some space, then 8 pins D0 to D7

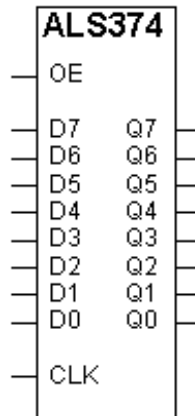
Entering the Symbol Name

The symbol name text box allows you to specify the text that will appear centered at the top of the symbol. This also becomes the new name for the part.

Generating the Symbol

The Generate button causes the current contents of the active drawing window to be erased and replaced by the generated symbol. The pin list will be merged with the new pins described in the dialog. This symbol consists of standard graphic objects so it can be edited using any of the drawing tools provided.

An example of a device produced by the Auto Create Symbol:



Using Pins Already in the List

The Extract Pin List button updates the pin name boxes with the names extracted from the main pin list. Pins which are inputs and busses are placed on the left, outputs are placed on the right. This is typically used when you are creating a symbol for subcircuit and the pin list has already been defined by importing it from the circuit definition.

Creating a Breakout

A breakout is a special device that allows signals to be associated with a bus. It consists of 1 Bus pin, no internal pins, and N normal pins. Because the device type is breakout the normal pins will be connected to like named signals in the bus. Breakouts are normally created using the New Breakout command described in “Using Bus Breakouts” on page 193.

NOTE: Another way to split signals out of a bus is to use a part with a subcircuit as a splitter, which explicitly routes signals between pins. This has advantage of flexibility since signals do not have to be explicitly broken out but may instead be split into busses or any combination of busses and signals. The disadvantage is that since the splitter is a device it will be listed in hierarchical netlists. In flattened netlists it will not appear if marked as a non-protected device. In addition, there is a significant memory and file size penalty.

Creating a Power and Ground (Signal) Connector

A power or signal connector is a special type of device which is generally used to represent a power or ground source, e.g.: +5V, +15V, -15V, -5V, Ground, Vss., Vdd.

These devices have a special properties in the schematic. When a pin on one of these devices is connected to a signal it attempts to assign its pin name to the signal. If the signal doesn't have a name then it gets the name of the pin. If the signal is named, and the name is different from the pin's name, then you will be prompted to select between the signal name and the pin name.

An additional property of signal connectors is that any signal they are connected to is exported across all pages of the schematic.

Power and Ground Connections with the DesignWorks Simulator

The type of the pin must be set correctly if simulations are to make sense. For signal sources like +5V, +15V, and Vss, a normal simulation would expect a logical value of 1 (True). For signal sources like Ground and Vdd, a logical value of 0 (False) is expected. This means the pin type should be “Tied High” or “Tied Low” (See “Setting the Pin Type” on page 299).

If you don't want the signal connector to supply a signal value, but only its name and the fact that it makes the signal global, then the pin type should be set to “Input”.

Creating a Port Connector

Port connectors have the property of associating a signal or bus with a pin on an enclosing part. The association is made by name. The port's Name is compared to the pin names on the parent part. Internal pins in busses are matched by pin name.

See “Creating and Using Port Connectors” on page 237 for more in-

formation on port connectors.

Creating a Signal Port Connector

A port connector for a signal must have a pin which is of the correct type to interface the signal to the parent part's pin. For example, an input pin on the port connector would be correct to connect to a output pin on the parent part. The name of the port connector pin is not important. Only the name assigned to the port connector once it is placed is important; it must be the same as the parents pin.

Creating a Bus Port Connector

A port connector for a bus must have a bus pin which contains pins of the correct type. For example, a bus pin with three internal pins A (input), B (input), and C (output) would be correct to connect to a parent part's bus pin which contained pins A (output), B (output), C (input). The name of the port connector bus pin is not important, but the internal pins must have the same name. Once the port connector is placed in a circuit its reference name is important, it must be the same as the parent's pin name.

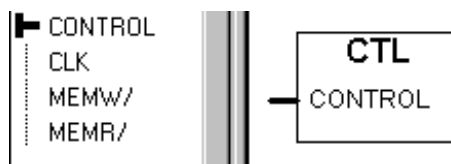
See “Bus Pin Name Matching” on page 241 for more information.

TIP: The easiest way to create port connectors is to use the New Port Connector command described in “Creating and Using Port Connectors” on page 237. The procedure given here is only needed if you want detailed control over the symbol graphics or pin types used in the port connector symbol.

IMPORTANT: The Bus Port Connector does not export all the signals in the attached bus, only the ones for which it has explicit Bus Internal pins

Bus Port Connector Example

For this example, assume that the following simple device has a bus pin called CONTROL containing internal pins CLK, MEMW/, and MEMR/.



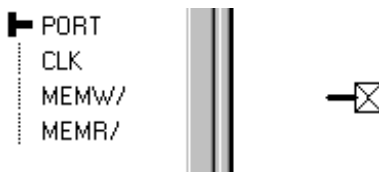
To create the corresponding Bus Port Connector using the symbol editor, follow these steps:

- 1) Select the New command in the File menu and choose the Device Symbol document type.
- 2) From the Options menu select “Add Pins...”
- 3) Enter the following string

PORT[CLK MEMW/ MEMR/]

See “Adding Sequential Pin Names” on page 283 for Add Pins syntax for ranges of numbered signals, etc.

You should now see a pin list like the following:



- 4) For each bus internal pin set the *pin function* as appropriate, i.e. the opposite of the function of the parent pin. We will assume that the CLK signal is an input to the block and that MEMW/ and MEMR/ are outputs:

Select the first internal pin name from the symbol editor's pin list. In this example it is CLK.

Use the *pin function* pop-up menu at the top of the list to select Output for the *pin function*.

Click on the next pin in the list.

For the remaining pins MEMW/ and MEMR/ the default value of Input is correct since the parent pin is an output. Check that the remaining pins are correct.

- 5) Create a symbol for the connector, perhaps a simple rectangle.
- 6) Select the bus pin, PORT in the pin name list.
- 7) Place a bus pin from the device symbol editor's tool palette.
- 8) Set the primitive type for the device.

Select “Subcircuit & Part Type...” from the Options menu.

From the “Subcircuit & Part Type” dialog select “Primitive, Use Caution”

From the pop-up choose “PORT CONNECTOR”.

Select “Done”.

- 9) From the “File” menu choose “Save As...”
- 10) Choose or create a working library to save the part.
- 11) Close the device symbol editor window.
- 12) Place your new Bus Port Connector in your internal circuit.

NOTE: The *pin name* of the bus pin itself (in this case “PORT”) is not important. The association between the Bus Port Connector and the parent bus pin is made by the name applied to the Bus Port Connector symbol itself. I.e. The same Bus Port Connector symbol can be used for any bus with the same internal signal names.

The comments under “Setting the Port Pin Type” on page 237 apply to each internal pin in a bus pin. The bus internal pins do not have to be the same. You can include any combination of names and functions in one bus pin.

Creating a Page Connector

Page connectors have the property of exporting the signal allowing it to be connected to across all pages in a circuit.

See “Inter-page Connections” on page 200 for more information.

Making a Signal Page Connector

The graphical appearance of a signal page connector is not important. The only characteristics it must have are:

It must have only a single input pin. The name of the pin is not important.

It’s primitive type must be set to Page Connector using the device symbol editor’s Subcircuit and Part Type command.

Making a Bus Page Connector

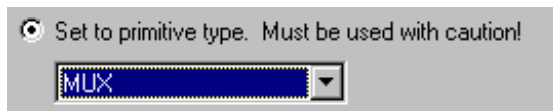
A page connector for a bus is the same as a signal page connector except that it has only a single bus pin with no internal pins.

Creating Special-Purpose Symbols

Assigning a Primitive Type

NOTE: The Primitive type settings should only be used with a clear understanding of their functions. Primitive types are intended primarily for creating pseudo-devices (such as power and ground connectors) and for use with the DesignWorks Digital Simulation option. See Appendix B—Primitive Device Types on page 369.

The primitive type of a part allows DesignWorks to recognize a number of special device types such as pseudo-devices and simulation primitives used by the DesignWorks Simulator option.



To select the primitive type choose the Subcircuit and Part Type command in the Options menu. Then select the “Primitive” radio button. This will cause a pop-up menu to be displayed below it. The pop-up menu will show the primitive type currently selected. Clicking on the pop-up menu will display the other options available. The following manual sections describe the usage of some of these settings. See Appendix B—Primitive Device Types on page 369 for a complete list of primitive types.

Creating Primitive Devices for use with the DesignWorks Simulator

Device symbols for use as simulation primitives with the DesignWorks Simulator must have very specific primitive type and pin type settings and pin orders. Refer to the DesignWorks Simulator Reference Manual for complete information.

Symbol Date Stamping

Whenever a symbol in a library is modified, an attribute field called DateStamp.Symbol is automatically updated with the current time. Since all attributes associated with a symbol are copied into a design when a symbol is used, this can be used to determine whether a symbol has been modified since it was used.

The date stamp is stored using the system's internal integer date format, that is, an unsigned integer representing the number of seconds since January 1, 1970. To create a more human-readable date value, the Export tool has date conversion functions available.

The following operations cause the DateStamp.Symbol field to be updated in the part definition:

- Saving the symbol using a Save or Save As command in the device symbol editor.

- Using the Save to Lib command to save a part from a schematic to a library with the "Use attributes in selected instance" option turned on. If this option is off, the datestamp is not updated since the part is not being modified.

- Saving parts to a library using the Circuit to Library command.

Operations that simply copy the symbol definition intact, such as the Copy button in the Library Maintenance box, do not change the date stamp.

Features Requiring Symbol Attributes

A number of DesignWorks features rely on certain attribute fields being present in a part definition. If you create a part symbol with no default attributes it will have the following characteristics:

- The Packager will assume that it has only one unit per package, i.e. a unique package name will be assigned to each device and the Unit field

will be left empty.

If you have not assigned default pin numbers to the pins, they will have to be entered manually for each device on the diagram.

Part number, component value and package attributes, if needed (e.g. for PCB netlists), will have to be entered manually for each device on the schematic.

No power and ground connections will be made automatically.

The default prefix will be used to auto-generate device name when the part is placed.

The following sections describe the attribute entries that should be made in each symbol you create in order to take advantage of these DesignWorks features.

Gate Packaging

A number of attribute fields are used by the Packager when it assigns a name and unit to a device. If you intend to create your own library symbols that require gate packaging, you will need to be familiar with these fields. If you are only using the libraries provided with DesignWorks, or if you have no requirement for gate packaging, you will not need this information.

See a more detailed description of the packaging mechanism and fields in “Creating a Symbol with Multi-gate Packaging” on page 144.

Auto-Naming

Both the Packager and the Auto-Naming features make use of attribute fields in the generation and positioning of names for devices.

NOTE: DesignWorks allows you to select any device field to be used as the prefix for default name generation. The prefix fields described here are the standard ones included in the DesignWorks libraries, but you can also define your own if you have special naming requirements. See the description of the PrefixField

design attribute elsewhere in this manual.

Field Name	Set In	Description
Name.Prefix	Device	This field gives the prefix to be used in generating a device name or package name. In the DesignWorks standard libraries, this field is empty for most integrated circuit parts. They therefore take the default prefix specified with the design. For discretes, the common prefix character is specified.
Name.Spice	Device	This field is specified only for symbols to be used with a SPICE-based analog simulator. It is an optional name prefix matching the standard required for SPICE models.
Name.Pt	Device	This field contains an X,Y coordinate for the default position of the name when the device is placed. This is specified in 1/1000 inch and is relative to the top left corner of the symbol. E.g. "1000, -200". See "Using Default Position Fields" on page 175 for more options in this field.

See "Setting the Auto-Generated Name Format" on page 131 for more information.

Specifying Part and Package Type Information

If the netlist output from DesignWorks is to be used to generate a printed circuit board, then it will be necessary to associate package type information with each device. This is done by specifying a keyword for each package type in the Package attribute field. Package type fields are included in the standard type libraries, but may be overridden if necessary by specifying a value for the Package field in the device attributes.

The keywords associated with the common package types are described in a technical note supplied with the package.

NOTE: The package code entered in the Package attribute field is used only to pass information to an external PCB layout package. It is not used internally by DesignWorks for gate packaging or any other operation.

See Chapter 6—Before Starting a Major Design on page 113 for more information on passing data to an external PCB package and some other alternative ways of specifying data.

Package Types for Discrete Components

Discrete components do not normally have standard package types. If a package type is required for PCB layout purposes, then the “Package” field in the attributes must be set manually for each device.

Part and Package Fields

The following fields are the standard ones used to specify the part number and package type for netlisting purposes.

NOTE: The standard part and package fields are listed here. You are free to add new fields for special purposes at any time and modify the field names used in the netlist formats.

Field Name	Set In	Description
Part	Device	This field gives the part name that will be used for netlisting purposes. This may be the same as the name as the part is saved under in the library, or it may be in a format required by some external package.
Part.List	Device	This field contains an optional list of part numbers that can be represented by this symbol. If specified, this list will appear in a value list menu in the device pop-up menu on the schematic.
Part.Pt	Device	This field contains an X,Y coordinate for the default position of the Part field when the device is placed. This is specified in 1/1000 inch and is relative to the top left corner of the symbol. E.g. “1000, -200”. See Chapter V - Advanced Schematic Editing for more options in this field.
Package	Device	This field specifies the default package type for use by an external PCB layout package.
Package.List	Device	This field contains an optional list of package codes for the device. IMPORTANT: This field is unusual in that it is linked in DesignWorks to the Part.List field. If you select an item in Part.List, the corresponding item in Package.List is automatically placed in the Package field.

See Chapter 6—Before Starting a Major Design on page 113 for more information on passing data to an external PCB package and some other alternative ways of specifying data.

Fields for Hierarchy Operations

The Restrict attribute field is used to determine whether packaging, report generation and editing operations are allowed to proceed down into the sub-circuit block. The allowable values for this field are discussed in Appendix A—Predefined Attribute Fields on page 361.

Other Fields

The following fields are included in the standard DesignWorks libraries and are frequently used to pass data to external packages in netlists and reports. However, these fields are not required for any specific program features.

See the Read Me file provided with the design kit you are using for specific requirements of the netlists and reports you are planning to use. In addition, you may wish to look at Chapter 6—Before Starting a Major Design on page 113 for an overview of attribute usage and how it may affect interfaces to other systems.

Field Name	Set In	Description
Power	Device	This field contains a list of pins to be connected to the power net in netlist reports. See “Power and Ground Connections in Attributes” on page 207 for more information.
Ground	Device	This field contains a list of pins to be connected to the ground net in netlist reports. See “Power and Ground Connections in Attributes” on page 207.
Function		This field contains a terse symbol function code. A full list of the codes used in the standard DesignWorks libraries is given in “Function and Category Codes” on page 327.
Category	Device	This field contains a terse symbol category code. A full list of the codes used in the standard DesignWorks libraries is given in “Function and Category Codes” on page 327.
Description	Device	This type attribute contains a short description of the symbol type. For instance “Quadruple 2-Input Positive-NAND Gates”.

Permutable	Device	This field contains a list of the swappable pins and gates in this package and is intended for external use by PCB packages. For more information on the format of the Permutable attribute see “The Permutable Attribute” on page 323. Note that this field is not used internally for packaging operations. For the packaging fields, see “Setting Packaging Attribute Fields While Creating a Symbol” on page 144.
Value	Device	This field contains the default component value for the part. It normally applies only to discrete components.
UnusedPins	Device	This field contains a list of the unused pins in this part type. This field is not included in any of the standard DesignWorks report formats, but is included for general usage.

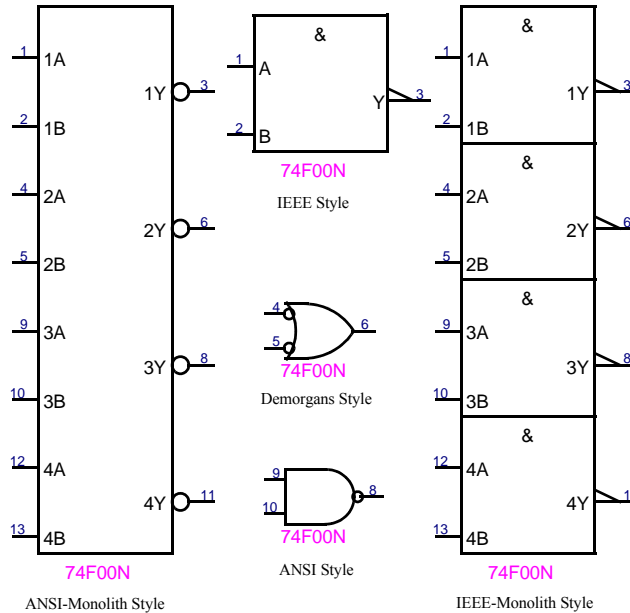
Using the Standard DesignWorks Libraries

The libraries provided with DesignWorks 3.0 contain over 12,000 symbols including the ANSI and IEEE format symbols and a compact ANSI (DesignWorks) format. The purpose of this section is to describe the naming, graphical, attribute and organization conventions used in the libraries. This information will be of interest if you if you need to adhere to some particular drawing standard or if you plan to export data from DesignWorks to another system.

Symbol Format

The DesignWorks libraries include the ANSI and IEEE symbol equivalents for each part. The installation procedure allows you to choose to install only

one symbol type.



Finding a Library

NOTE: This section should be taken only as a general guide as the exact configuration of libraries on your system will depend on which options you chose when you installed the package. In addition, libraries are sometimes added or reorganized in the process of upgrading DesignWorks.

The directory structure used to store the libraries is divided into several folders. The top most folder is separated into Analog, ANSI Compact, ANSI and IEEE.

Within Analog folder you will find:

- Connectors
- Diodes
- Discretes
- National Linear

Transistors

Spice Lib

Misc. Analog

Within the ANSI and IEEE folders you will find:

54 TTL

74 TTL

75 TTL

CMOS

ECL

Memory

Micro Processors

Miscellaneous

PLD

The 54, 74 and 75 series components are divided into their families, so there are separate libraries for the expanded components in the folder “Expanded” and non-expanded components in the “Monolith” folder. Within the Expanded folder where appropriate there is also a “DeMorgans” folder that contains the DeMorgan equivalent symbols for many of the components. All the TTL components are derived from Texas Instruments data.

Within the TTL folders are the following library files:

74, 74S, 74LS, 74AS, 74ALS, 74AC, 74ACT, 74F, 74HC, 74HCT, 74HCU.

54, 54S, 54LS, 54AS, 54ALS, 54AC, 54ACT, 54F, 54HC, 54HCT, 54HCU.

75ALS.

Memories are divided into Motorola, Texas Instruments (TI) and Toshiba.

Micro Processors are divided into Intel, Motorola and Zilog.

Interpreting Library Part Names

A component name is made of up to five parts.

The Logic Family Name

The Device Number

Device Attributes

DesignWorks unit for packaging

A marker indicating the symbol style

Most of these name parts are only used for the TTL family of parts. Other components may have just the manufacturer part name or a descriptive name as the library name, for instance “DIODE” or “2N222”.

Some parts, like diodes, have over 100 different names in the parts list. In this case the library part name will be the first name that appears in the Part.List attribute.

Occasionally the part name is too long to display in the parts palette list, in this case the name will appear truncated.

The Logic Family Name

The logic family name is usually from one of the following:

74, 74S, 74LS, 74AS, 74ALS, 74AC, 74ACT, 74F, 74HC, 74HCT, 74HCU.

54, 54S, 54LS, 54AS, 54ALS, 54AC, 54ACT, 54F, 54HC, 54HCT, 54HCU.

75ALS.

The Device Number

The device number is the middle digits in the component name and corresponds to the symbol type. For example “123” in 74ALS123J.

The Device Attributes

The device attributes are the manufacturer attributes that apply to this symbol. The attributes are enclosed in parentheses “()” such as (W,J).

Unit for Packaging

The presence of a unit indicates that there are multiple symbols required to represent all the units in this package. The unit is appended to the name with a “.” so “.a” would indicate that this component will default to unit “a”.

If there are different symbols for some of the units within a package then the different units will each have their own entry in the library. The unit letter is appended to the end of the library name such as: “5450(J).a” and “5450(J).b”. This component is a Dual 2-Wide 2-Input AND-OR-INVERT Gate (One Gate Expandable). In some cases the part that shows up as xxx.a will have a “c” and “d” units and the “b” unit will be a different entry because it does not share the same symbol.

Symbol Style

The library file name contains a symbol style marker selected from the following:

C	Compact ANSI (DesignWorks format)
A	ANSI style symbol.
I	IEEE style symbol.
D	DeMorgan equivalent symbol.
AM	ANSI Monolith (non-expanded gates).
IM	IEEE Monolith (non-expanded gates).

Library Name Example 1

A 54S00J may be listed in the library selection palette as “54S00(W,J)A”

“54S” is the TTL family name, this part of a library name is only used for the TTL components.

“00” is the component number

“(W,J)” indicates that there are two part suffixes for the 54S00 a “54S00J” and a “54S00W”.

The “A” suffix indicates that this part is an ANSI style symbol.

Library Name Example 2

A 5450J may be listed in the parts palette as “5450(W,J).aI”.

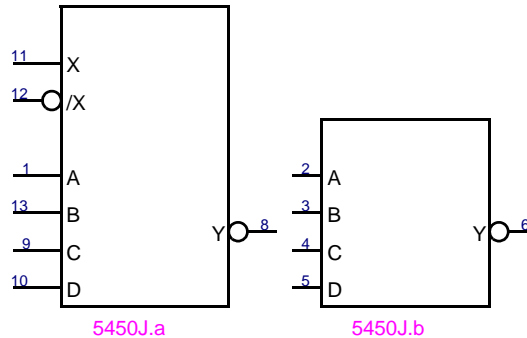
“54” is the TTL family name, this part of a library name is only used for the TTL components.

“50” is the component number

“(W,J)” indicates that there are two part suffixes for the 5450 a “5450J” and a “5450W”.

There may be a “.a” (or “.b”, “.c” etc...) after the part suffix as in “5450(J).a”. This is the unit and it indicates that there is more than one symbol for the gates in a package. Most of the TTL gates have multiple units in a package, the unit name only appears in the library if the graphic symbol for one or more of the units differs from the others.

The “I” suffix indicates that this part is an IEEE style symbol.



The Permutable Attribute

The permutability of pins on a device will be indicated by an attribute in the symbol definition. This field has the following characteristics:

The attribute used is “Permutable”.

The value of the Permutable attribute is a string of bracketed pin numbers.

There are two types of brackets:

'(' ')' enclosing a list of permutable pins.

'[' ']' enclosing a list of non permutable pins.

Pin lists are separated by ',' characters.

Pins that are not in the list are not permutable.

The permutable and non-permutable brackets are nested to indicate conditions where individual pins are not permutable but as a group they may be.

A list of pins inside a () are permutable. e.g. (1,2,3).

A list of pins inside a [] are not permutable. e.g. [1,2,3].

Lists of non permutable pins [1,2,3] inside a () indicates that the

individual non permutable pins may be permuted as a group. e.g. $[(1,2,3],[4,5,6])$

Complex permutations as in Example 3 below may indicate that within a non permutable unit some of the pins may be permuted. Such as the A and B inputs on a NAND gate e.g. (1,2) but the output Y must not be swapped with another pin without swapping both A and B. e.g. $[(1,2),3]$ so we get a list like $[(1,2),3],[(4,5),6])$.

This means pins 1 and 2 may be swapped (they are functionally identical). Pins 4 and 5 may be swapped. If we swap pin 3 or 6 we must swap 1,2,3 or 4,5,6 at once.

Example 1

A 54LS74 has two D type flip flop units each identical. To permute one pin we must permute all pins on one FF with the ones on the other FF. This is indicated as follows:

Permutable = “ $[(14,1,2,3,13,12],[8,7,6,5,9,10])$ ”;

pins 14,1,2,3,13,12 are non permutable with each other.

pins 8,7,6,5,9,10 are not permutable with each other.

But as a group pins $[14,1,2,3,13,12]$ are permutable with pins $[8,7,6,5,9,10]$.

Example 2

A 54LS04 Hex Inverter. The two pins on each device are not permutable since one is an input and the other is an output. But each of the six inverters can be exchanged as a whole with another.

Permutable = “ $[(1,14],[3,2],[5,6],[7,8],[9,10],[13,12])$ ”;

Example 3

A 54LS00 Quad 2-Input Nand Gate. The 4 individual units of the NAND may be permuted as a whole as well the two inputs of each may be permuted. This is indicated as follows:

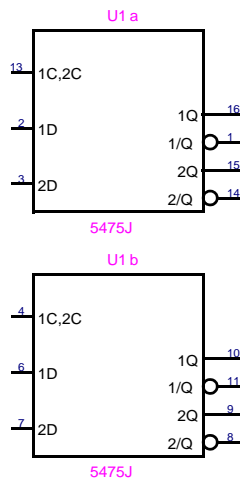
Permutable = “ $[(1,2),3],[(6,7),5],[(9,10),8],[(12,13),14])$ ”;

In this example each non permutable group “ $[(1,2),3]$ ” is one of the NAND gate units for this unit pins 1 and 2 are permutable. Pin 3 is not. However each unit may be permuted as a whole.

Example 4

A 5475J 4-Bit Bistable Latch. There are 4 nearly identical latches in this package. Each has a D input, a Q and \Q output and an enable (C). However there are only 2 enables for the 4 bits. Bits 1 and 2 have a common enable (Enable 1-2) and bits 3 and 4 have a common enable (Enable 3-4). This differentiates bits 1 and 2 from bits 3 and 4. The D,Q,\Q pins on bit 1 may be exchanged with the same pins on bit 2, but if we exchange bit 1 for bit 3 or 4 we must also exchange the Enables. This is indicated as follows:

$$([13, ([2, 16, 1], [3, 15, 14])], [4, ([6, 10, 11], [7, 9, 8])])$$



Package Codes

Package codes have three parts:

- Manufacturer Name
- Number of Pins
- Package Style code

The manufacturer name may be any string of characters although it is advised that no numeric digits should be used since that may confuse the format, the first digits in the package name should be the number of pins used in the package.

Example

TI14DPN

This package is manufactured by Texas Instruments (TI) there are 14 pins on the device and the style is a narrow, plastic dual in line package.

In general the physical dimensions of the packages are not defined by these codes. The DIP package types are standard, others may vary with the manufacturer. The package codes define the sets of foot prints required for PCB layout, it is up to the PCB layout tool to define the actual physical dimensions for the package foot print.

The Package styles used in the DesignWorks libraries are as follows:

Package Type	Code
DIP, plastic narrow	DPN
DIP, plastic medium	DPM
DIP, plastic wide	DPW
DIP, plastic x-wide	DPX
DIP, ceramic narrow	DCN
DIP, ceramic medium	DCM
DIP, ceramic wide	DCW
DIP, ceramic x-wide	DPX
DIP, side brazed narrow	DBN
DIP, side brazed medium	DBM
DIP, side brazed wide	DBW
DIP, side brazed x-wide	DBX
DIP, windowed narrow	DVN
DIP, windowed medium	DVM
DIP, windowed wide	DVW
DIP, windowed x-wide	DVX
MODULE - Leadless/single side	M
MODULE - Leadless/double side	MD
MODULE - Single Sided with Edge Clipped Leads	E
MODULE - Double Sided with Edge Clipped Leads	ED
MODULE - Single Sided with Pin Centered	I
MODULE - Double Sided with Pin Centered	ID
ZIP	Z
Flatpack	F
Flatpack, windowed	FV

Flatpack, Sidebrazed	FB
Quad Flatpack	Q
SOIC narrow	SN
SOIC medium	SM
SOIC wide	SW
SOIC x-wide	SX
Quad SOIC, Gullwing	G
Quad SOIC, Gullwing/Rectangular	GR
SOJ (J-bend SOIC)	J
20/26 pin SOJ	J
SOIC, ceramic	JC
Pin Grid Array	A
Pin Grid Array, windowed	AV
Pin Grid Array, plastic	AP
Pin Grid Array, cavity down	AD
Pin Grid Array, with Heatsink	AH
Leadless chip carrier, ceramic Square	L
Leadless chip carrier, ceramic Rectangular	LR
Leadless chip carrier, sidebrazed Square	LB
Leadless chip carrier, sidebrazed Rectangular	LBR
Leadless chip carrier, windowed Square	LV
Leadless chip carrier, windowed Rectangular	LVR
Leadless chip carrier, plastic Square	P
Leadless chip carrier, plastic Rectangular	PR
Leadless chip carrier, ceramic Square	C
Leadless chip carrier, ceramic Rectangular	CR
Leadless chip carrier, ceramic windowed Square	CV
Leadless chip carrier, ceramic windowed Rectangular	CVR
Leadless chip carrier, plastic with bizarre attributes	PQ

Function and Category Codes

The following table lists the Category and Function values used in the standard DesignWorks libraries. Other values may of course be added as libraries are augmented in future releases.

Category	Function
ARITH	ACCUM, CARRY, ALU, MULT, ADD
BUFF	BUFFINV, DELAY, BUFF, INV

COMP	ADDCOMP, ERRDET, PARITY, COMP
CONN	CONNM, CONNF
CTR	PRESCALE, TIMER, OSC, FREQDIV, RATEMULT, CTR
DIODE	TUNNEL, LED, BRIDGE, ZENER, DIODE
ELMECH	RELAY, SWITCH
FF	DFF, JKFF
GATE	OAI, XNOR, GATE, XOR, AOI, OR, AND, NOR, NAND
LATCH	MULTIVIB, LATCH
LIN	ADCONV, VOLTREG, LIN, OPAMP
MEM	ROM, EPROM, EEPROM, PROM, DRAM, FIFO, SRAM
MISC	LVLCONV, PLL, RCVR, MISC, FILTER, DSWITCH, PLSSYNCH
MUX	AMUX, BITSHIFT, MUX, ENCODE, DECODE
NETSYM	PWR, GND
PASS	XTAL, XFMR, TEST, RES, IND, FUSE, CAP, BAT
PLD	PLD, EPLD, PAL, FPLA
PROC	COPROC, MPERIPH, MCTRL, PROC, MEMCTRL, MEMMAP
REG	REG, SHREG
THYR	UJT, TRIAC, THYR, SCR, DIAC
TRANS	TMOS, PNP, NPN, JFET, MOSFET
XCVR	XCVR

This chapter is about how to customize DesignWorks to fit your particular design application by creating design templates, scripts, libraries and making use of other customization features. Customizations can range from small changes to symbol libraries or design templates to implementing major new applications using DesignWorks as a base.

Some of the topics covered here include:

- How to install and use the design kits supplied with DesignWorks

- How to make simple changes to the design kits supplied (such as adding your company logo to the sheet templates)

- Creating your own design kits from scratch for specific design applications

- How to add new functions to the program using scripts

- How to add custom windows and control panels using HTML

- How to set up custom menu items to give direct access to your added features.

Additional information is also available in separate documentation provided in electronic form, including the **DesignWorks Export Script Language Reference** and the **JavaScript User's Guide** and associated documentation. These items can be found in the Documentation folder inside the DesignWorks program folder.

Creating Design Templates

IMPORTANT: Don't modify the design templates provided with the package without copying them first! You may wish to return to them later if your application changes. The copying procedure is described in "Working from an Existing

Design Template” on page 332.

In DesignWorks, a design template is simply a normal design file that has its sheet size, attribute fields, hierarchy mode and other settings pre-defined for the application at hand. The simplest way to create a template may be to take an existing design that is set up the way you like it, delete all the circuit elements and extra pages out of it and save it in the appropriate template directory.

When you create a new design using one of the template files listed, DesignWorks just reads the file in the normal way, then renames it “Design1” (or the next available number), and disassociates it from the original file so that it cannot accidentally be Saved on top of the template. In all other respects, New Design is the same as doing an Open on the template design. All the settings and contents of the design template file become part of the new design.

Design Templates vs. Sheet Templates

This chapter covers two related topics: *design templates* and *sheet templates*. Both of these things are standard DesignWorks design files from which initial setup data is imported, but they have different usages:

A design template is intended to be used as the source for a New command. When a design template is selected in the New Design box, the design template file is opened and all the circuit data and settings it contains become the starting point for a new design. More information on this process is given in “Creating Design Templates” on page 329.

A sheet template is intended for use with the Import Sheet Info command and is used to set only the sheet border, title block, print setup and other settings related to the printable sheet. The Import Sheet Info command is covered in “Importing Sheet Settings from Another Design, Page or Template” on page 334.

There is no reason why a design template cannot be set up with all the sheet settings you need for your design. This eliminates the need to ever refer to a sheet template. However, sheet templates can be used to apply different border styles and settings to different parts of a single design. In addition, separating out the sheet functions can reduce the number of design templates you need to have available. The design template can be used to select the application type, attribute settings, script setup, etc. A sheet

Contents of a Design Template

The template should be set up so that the user can just start in placing device symbols and signal lines without worrying about whether the border will fit on the printer, the netlist will come out OK, etc. This table defines the items that you may want to include or set up in a template file, and gives you a reference to the section in the manual that discusses the issue.

Design Kit Issue

A title block

A sheet background, or auto-border settings

A printer page setup, possibly with the reduction set as needed. Note that there is a new “Scale to Fit” sheet option that allows automatic scaling to fit your printer

All built-in attribute fields that are not needed should be hidden, i.e. the “In Primary List” box should be off in the Define Attribute Fields box.

All attribute fields required for netlisting and other operations should be pre-defined. Ones that require user entry should be “primary”, others should not be.

A design attribute field can be created to list the device attribute fields that will be used to specify power and ground pins on devices.

The hierarchy mode should be set as appropriate. This may affect the usage of attribute fields, and therefore which items you want to make visible to the user.

The DesignType design attribute field should be set to a unique text string so that the netlisting and other scripts in the design kit can check that they are operating on an appropriate design type.

Optionally, some notes on usage of the design template in a miscellaneous text block on the sheet. These can be deleted by the user when no longer needed.

Where to Look

“Creating a Title Block” on page 339

“Setting Sheet Sizes and Borders” on page 333

“Setting Sheet Sizes and Borders” on page 333

“Defining a New Attribute Field” on page 177

“Defining a New Attribute Field” on page 177

The entry “Reporting Power and Ground Nets” in *the DesignWorks Script Language Reference (separate manual on disk)*.

“Choosing a Hierarchy Mode” on page 218

“Creating Netlists and Reports” on page 346

NOTE: Some users like to have a different format for page 1 of a design, i.e. for a title or index page. It is quite feasible for the template design to have 2 pages with the first one set up differently. The existing Sheet Info mechanism can be used to allow the user to apply different sheet settings to different parts of the design later, if needed.

Naming a Design Template

Technically, the name of a template file can be any valid file name. However, we suggest following the following informal convention in order to minimize user confusion. The name should contain the following items:

Unique keyword identifying application type, e.g. “P Spice”.

A letter or keyword identifying the sheet style, e.g. “ANSI-B”.

A letter or keyword identifying the hierarchy mode, e.g. “Flat”.

For design kits for use on Windows 3.1-based systems, names must be limited to 8 characters, which obviously limits creativity. In these cases, we suggest using single-letter codes for sheet style and hierarchy mode and separating them by hyphens, as in: “PSPCE-BF.CCT”.

Working from an Existing Design Template

If you are modifying an existing template that is used in an application similar to yours, here are the minimum things that you need to change:

If you are also modifying the report and netlist scripts in the design kit, the DesignType design attribute field should be set to different value. This ensures that these scripts can recognize the design type and warn the user if it’s not appropriate. Note that you also need to modify all scripts that check this value to match.

If the template has text notes on the schematic, modify them to reflect the new usage of the template.

The template should be saved under a different name that reflects its new usage.

Where Design Templates are Stored

In order to appear in the template list in the New Design box, template files must be stored in the folder specified in the Templates keyword in the INI file. In the standard DesignWorks installation, this folder is called Templates.

See “Specifying the Location of Design Templates” on page 379 for more information.

Setting Sheet Sizes and Borders

DesignWorks does not have any built-in knowledge of sheet sizes. Sheet border information can either be customized for each design or imported from an existing design file or template. This section describes the methods used for setting the sheet size and border for a schematic and the use of various options for adjusting the schematic border to the available output device.

About Sheet and Border Settings

DesignWorks supports very flexible use of plotters and printers for output on a variety of media. For this reason, there can be complex interactions between the drawing area allocated for each page and the way the schematic is presented when it is output on various kinds of output devices. Various settings allow you to:

- Have the schematic border adjust automatically to the current printer page setup.

- Have the schematic border retain a fixed logical size and have the printer scaling adjust automatically to fit the schematic on a single sheet.

- Have the schematic border retain a fixed physical size and be broken into the required number of printed sheets for output.

The following information is considered to be part of the complete sheet definition:

- The overall height and width of the drawing area.

- The origin and spacing of the location grid.

- The format of text used in the border.

- Whether or not the border size should be linked to the current printer page setup.

- Whether or not the page should auto-expand to multiple sheets.

- Any text and picture objects marked as “border” items.

DesignWorks allows sheet size and border layouts to be set by one of two methods:

Importing an existing layout using the Import Sheet Info command. This is the simplest method of setting the sheet size for a new design or template, but it assumes you have an existing design that has the desired settings.

Creating a custom layout using the Sheet Wizard command. This command, and the associated text and picture capabilities, allows all aspects of the sheet size, drawing grid, border text style, title blocks and graphics to be set up to meet your drawing standards.

Importing Sheet Settings from Another Design, Page or Template

If you have another page in the current circuit, or an existing design or template file that has the sheet size and border settings that you want, you can import those settings into the current design using the Import Sheet Info command. This command performs this sequence of operations:

Deletes all text and picture objects marked as “Border” items from the current page or design.

Copies all text and picture objects marked as “Border” items in the source design or page into the current design or page. You can change this setting on any number of graphic items at once by selecting the desired items, the right-clicking on any one of them and selecting the Properties command

Copies the sheet and border settings (sheet width, grid settings, border text settings, etc.) from the source design or page to the current design or page.

Optionally copies the printer page setup from the source design to the current design.

For more information on the “border” setting in text and picture objects, see “Background and Border Objects” on page 105.

Note these important points regarding this operation:

If the source for the import is a design file, only the items and settings on page 1 of the source design are imported. There is no provision for importing different settings from multiple pages.

The import operation does not affect any circuit items or attribute settings in the destination design.

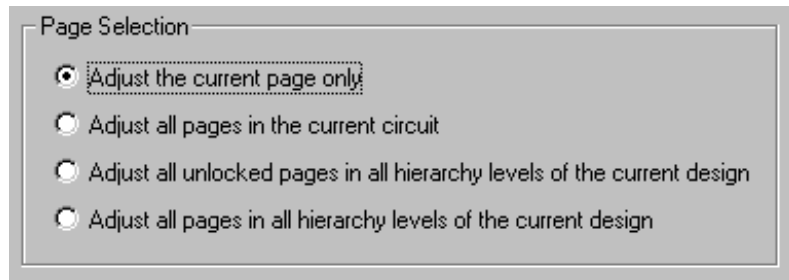
Setting Custom Sheet Size using Sheet Wizard

The Sheet Size Wizard command in the Drawing menu gives you full control over the size of the drawing area that appears on the screen and how it is printed on the available output device. This section will describe the various options that are available.

- ➔ To open the Sheet Wizard, select the Sheet Size Wizard command in the Drawing menu.

Selecting Which Pages to Modify

If your design consists of only a single page and no hierarchy levels, the Wizard will immediately display the first page of sheet settings and you can skip this section. If your design contains multiple pages or circuit blocks, you will first see this panel:



Settings can be applied to individual pages or to the entire design.

- ➔ If the first panel of the Wizard displays the Page Selection choice, then select whether you want to change only the current page or all pages, then click Next. If the selected range contains a variety of settings, you will be warned that they will be overridden and you will have to click Next again.

Selecting Fixed or Auto-Scaled Border

DesignWorks provides two general types of sheet size settings:

Scale border to match printer setup—Selecting this option causes the current printer setup to be used to determine the sheet size. Any changes in print setup (using the Print Setup command in the File menu) will

immediately affect the sheet border. This is the recommended setting for most cases as it provides a guaranteed match between the drawing area and your output device. The only disadvantage of using this setting is that the sheet size will change if you change printers. Most printers have slightly different printable areas, even on the same size paper.

Fixed border size—Selecting this option causes fixed values to be used for the border size and position (as it appears on the screen), regardless of printer page setup. The advantage of this setting is that the sheet configuration will remain fixed regardless of changing output devices. The disadvantages are that the border size may not fit exactly to the aspect ratio of the printable area on your printer.

➔ Choose the desired border scaling method and click Next.

Working with Scaled Border Size

Once you have elected to scale the border to match your printer settings, you will be presented with these options:

Single printed page—When this option is selected, the drawing border on the screen will always match your current printer page setup. Any change in printer setup will automatically be reflected in the sheet you see on the screen. This is the simplest setting and recommended for most case.

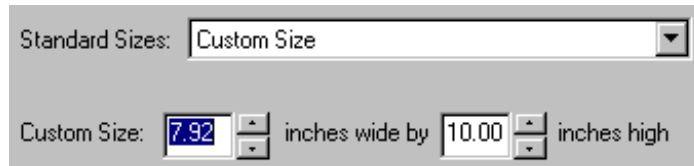
Auto-expand—This setting also adjusts to match the printer page setup, but automatically expands to use multiple pages if you draw outside the current border. This will allow you to assemble larger drawings by piecing together a number of smaller sheets.

Advanced—The Advanced setting is similar to Auto-expand, but allows you to specify exactly how many sheets high and wide you want the printed out put to appear on, plus it allows an additional scale factor to be applied to shrink or expand the output.

Working with Fixed Border Size

When you select the Fixed Border Size option and click Next, you will be pre-

sented with the sheet size controls:



Using these controls, you can either select one of the standard sheet sizes listed, or enter any desired custom size.

NOTE: These settings only determine the size that the sheet will appear on the screen and do not select a corresponding paper size for your printer. Printer settings must be adjusted appropriately using the Print Setup command.

NOTE: The standard sizes that are listed here are determined by settings in the initialization file dw.ini. These settings can be modified to suit your requirements by following the instructions given in “Specifying Standard Sheet Sizes” on page 382.

➔ Enter the desired sheet size and click Next.

The next step in the Wizard allows you to choose how the page will be adjusted for printing. Since you have specified that you wish the drawing to appear a fixed size on the screen, the program needs to know what to do if the visible page does not exactly match the printable area available on the paper. You have two general options:

Scale to fit printed page—In this case, DesignWorks will scale the drawing to fit into the available printed area for the selected printer. You can use the controls to specify that you want the drawing to be stretched onto multiple pieces of paper.

Fixed scale factor—With this setting active, the program will scale the drawing by a fixed amount (default 100%) for printing. This guarantees that the drawing will be a known size on paper, but may require that it be broken into multiple printed pages, depending on the settings.

Setting Border and Background Grid Settings with the Border Wizard

The Border Wizard command in the Drawing menu allows you to set these parameters:

- The size and visibility of the default drawing border.
- The layout of letter and numbers in the reference grid that is used to locate items on a sheet.
- The size and visibility of the background grid.

This table summarizes the first page of Border Wizard controls and options:

Border Orientation	These switches determine the usage of letters and numbers for sheet grid references. This setting will affect the position displayed on Page Connectors, in the Pin List box and in report output.
Show default border on screen	This switch determines whether the default sheet border is drawn around the drawing area on the screen.
Print default border	This switch determines whether the default sheet border is drawn in printed or PICT file output.
Border Font	This button displays the standard text style dialog box allowing you to select the text style for the default border. This has no effect if both Print Default Border and Draw Default Border are off.
Border Width	These switches allow you to specify an exact width for the border or have it calculated automatically based on the text size setting.

When you click Next, you will be presented options for the layout of the major grid lines that are drawn in the background. The major grid lines are the darker lines drawn in the background. You can select later how many minor (lighter) grid lines are drawn in between the major lines.

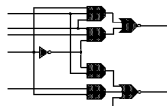
- NOTE:** Even if you choose not to display the drawing grid, these settings affect how the letter/number reference grid is calculated. This information is used and displayed by a number of reports and commands.
- **Specify the number of grid sections** If you choose this option, you can use the controls to specify exactly how many grid sections are displayed. The size of the grid sections will be adjusted automatically to match any changes in sheet size.
 - **Specify the size of the grid sections** This setting forces the grid sections to be a fixed size, regardless of changes in overall sheet size. In this case, the grid section width may not exactly divide into the sheet width, so you must also specify how to handle any leftover space.

The last page of border options includes these settings:

- Show background grid on screen** If this switch is on, the background grid determined by the previous settings is drawn on the screen. For information on changing the colors used for this display, see “Color Settings” on page 378.
- Print background grid** This switch determines whether the background grid is included in printed or graphics file output.
- Number of minor grids per major grid** This number determines how many minor grid divisions are drawn per major division. The default value is 3, entering 1 in effect disables minor grid lines.

Creating a Title Block

DesignWorks has a number of features which can assist in creating title blocks which will contain information that is automatically updated.:

Project Title:
Drawn by:
Orig. Date:
Last Mod. Date:
Page Number: of:
Page Title:
Rev. Level:
Mods:
 <div style="display: inline-block; vertical-align: middle; margin-left: 10px;"> <p>Capilano Computing</p> </div>

Create the graphic on the diagram using the using the built-in drawing tools, or paste an externally-created graphic using Paste command.

Select the graphic and set it to be a background and border object using the Properties command. Once it has been converted to background, it will not be selected again (unless you use the Background Layer Only command) and you can use text notations to draw on top of it.

If desired, use the Position settings in the Properties command to fix the position of the title block relative to the border. This ensures that if the sheet size is changed, the block will stay in the same relative position.

Use text variables to add dates and titles to the block. See “Using Text

Variables” on page 107.

See “Selecting a Background Object” on page 109 for more information on background objects and how to select them.

Creating Custom Sheet Border Graphics

Any graphic or text objects can be made part of the sheet border. To do this:

- select the object.
- choose the Properties command in the Options menu.
- enable the “Make Background” and “Make Border” switches.
- click OK.

The object will now be treated as part of the sheet background.

See “Selecting a Background Object” on page 109 for more information on background objects and how to select them.

Setting Text Styles

Setting Pin Number Text Style

The text style for pin numbers is set globally for the entire design. It cannot be set individually for pins.

To set pin number text style:

- ➔ Select the Design Preferences command in the Drawing menu.
- ➔ Select the Text tab.
- ➔ Click on the Pin Text button.
- ➔ Select the desired text font, size and style in the style box.
- ➔ Click OK on the text box, then OK on the Design Preferences box.

Depending on the size of the design, there may be a short delay at this point while sizes and positions of text items are recalculated.

Setting Attribute Text Style

The text style for attributes is set globally for the entire design. It cannot be set individually for each item or for each field.

To set attribute text style:

- ➔ Select the Design Preferences command in the Drawing menu.
- ➔ Select the Text tab.
- ➔ Click on the Attr Text button.
- ➔ Select the desired text font, size and style in the style box.
- ➔ Click OK on the text box, then OK on the Design Preferences box.

Depending on the size of the design, there may be a short delay at this point while sizes and positions of text items are recalculated.

Setting Border Text Style

Text style can be set for the reference letters and numbers appearing in an auto-generated sheet border. To do this:

- ➔ Select the Border Wizard command in the Drawing menu.
- ➔ Depending on the page settings in your design, you may have to click Next once or twice until the “Border Font” button appears.
- ➔ Click on the Border Font button.
- ➔ Select the desired text style.
- ➔ Click OK on the text style box, then Next and Finish to complete the Wizard.

Setting Text Block Text Style

Text style can be set individually for each text block. To do this:

- ➔ Select the text block by clicking on it. (If it is a background item, *see “Selecting a Background Object” on page 109.*)
- ➔ Select the Properties command in the Options menu.
- ➔ Click on the Text Style button.
- ➔ Select the desired text style.
- ➔ Click OK on the text style box, then OK on the text info box.

Creating Multipage Templates

DesignWorks allows a circuit to be represented on multiple pages with logical connections between pages. When a new circuit is created, it is assumed to have a single page. A new page is added to the circuit by selecting the Pages command in the Drawing menu and clicking on the New Page button.

Sheet Border Setup for Multi-Page Designs

When a new page is added to a circuit, all border information and text and picture objects marked as “border” items are copied automatically from the preceding page (i.e. what was formerly the last page). For this reason, it is most convenient to set up the first page of the circuit with the desired border arrangement before adding other pages.

Borders on all pages in a design can be updated later if needed using the Import Sheet Info command.

Creating Symbol Libraries

The creation of symbol libraries is covered in detail in Chapter 12—Device Symbols and Libraries on page 261.

Creating Custom Menus

DesignWorks allows you to add custom menus to the main menu bar or to the popup menus for various object types to customize commands for your application. A custom menu launches an Export script or JavaScript, which can in turn start another Windows application, generate a report, scan your design for errors, prompt for information, or various other tasks.

Creating a custom menu requires two steps to be taken:

- Add a line to the initialization file `dw.ini` to define the menu.
- Create the script that will be run when the menu is selected.

Defining a Menu in the Main Menu Bar

A custom menu is defined by adding a “MENU” keyword line in the [Export] section of the `dw.ini` file, as in the following example (note that all of the following must be on one line in the `.ini` file):

```
menu= XML, "Generate XML Netlist", "xml_netlist.rfm",  
      "Generate a netlist in XML format", "NETLIST"
```

Here is a definition of these items:

```
menu= menuName, menuItem, script, message, arg1, arg2, ...
```

menuName	This is the name (title) of the menu as it will appear on the program's main menu bar. If this text contains blanks or other special characters, it should be enclosed in double quotes.
menuItem	This is the text of the menu item, as it will appear when the menu is displayed. If this text contains blanks or other special characters, it should be enclosed in double quotes.
script	<p>This is the directory path and file name of the script file. The location of this file is relative to the location of the DesignWorks application. For example, “My Scripts\netlist.rfm” will look in a folder called “My Scripts” inside the DesignWorks program folder. If this field contains blanks or other special characters, it must be enclosed in quotes. More information about specifying file names in INI file entries in “Specifying File and Folder (Directory) Names” on page 375.</p> <p>This file can also be the name of any application or other file on your system. If the file extension is not recognized as one of the DesignWorks support script types, the file name is passed to the Windows system and a "shell open" is requested, which is equivalent to double-clicking on the file in the Windows Explorer.</p>
message	This is text that will appear in the program status bar when the menu item is highlighted. This should provide a description of the command.
arg1, etc.	This is optional text that will be passed to the script. The first argument is available in the script as variable <code>&ARG1</code> , the second is <code>&ARG2</code> , etc.

You can create multiple entries in the same top level menu by specifying the same “menuName” item for each item.

NOTE: Custom menus are added only to the “circuit document” menu bar. If you have another document type (such as a device symbol) open, the custom menus will not appear.

Adding a Menu Item to Popup Device, Signal, Pin or Circuit Menus

You can add your own menu commands to the popup menus that appear when an object in a circuit diagram is right-clicked. The scripts that run in response to these commands have access to information about the selected object and can display or prompt for device information or perform any desired other action.

Here is an example of a menu item in the device popup menu:

```
POPUPSCRIPTDEV = "show Data Sheet"  
                  "Utility Scripts/show_data_sheet.dwj"
```

NOTE: All of the above must be on one line in the [Drawing] section of the INI file.

The popup menu definitions have only two arguments:

- The first argument is the menu item text
- The second argument can be either an attribute field name or a script file name. If the name given is a device attribute field name, then that field is looked up and its contents are used as the script file name. This allows you to have the script response depend on the device type.

Menu items can be added for the following circuit object types:

Object Type	Keyword	Action
Device	POPUPSCRIPTDEV	Displayed in response to a right-click on any device
Signal	POPUPSCRIPTSIG	Displayed in response to a right-click on any signal line
Pin	POPUPSCRIPTPIN	Displayed in response to a right-click on any pin

Object Type	Keyword	Action
Circuit	POPUPSCRIPTCCT	Displayed in response to a right-click anywhere in a circuit diagram that is not on one of the above object types

Creating Scripts for Use with Custom Menus

Any script can be invoked by a custom menu. For example, if you frequently use the same netlist script to generate output from your designs, you could add a menu to give direct access to one of the standard scripts provided with DesignWorks.

The custom menu mechanism provides a method of passing text arguments to the script when it is invoked by a menu. This allows you to use the same script for a number of different menu items by defining different arguments for each case.

For example, a general script is provided with DesignWorks that will open a specified file, as if the user had double-clicked on it from the Windows Explorer. The name of the file to open is passed as an argument, allowing you to add menu items without modifying the script.

Creating Scripts

DesignWorks has several types of scripting technology to address a variety of applications in generating netlists and reports, checking for errors, displaying simulation and analysis results, simplifying data entry, importing and exporting data to and from files, databases and the internet, and many other tasks. Script usage and creation is covered in detail in two separate online documents included with the package:

The DesignWorks Export Script Language is optimized for creating text output from a design and simplifies creation of parts lists, bills of materials, netlists and other such reports that contain device and signal data from the design. All the netlist generation scripts provided with the

package are written in this language. The DesignWorks Export Script Language Reference covers creating netlists and reports using this language.

The standard JavaScript language is supported in DesignWorks for powerful, general-purpose scripting. JavaScript objects and methods are provided to give access to essentially all design data and program functions as well as a powerful set of file and internet operations and general programming support. The DesignWorks JavaScript User's Guide describes the implementation of this language.

DesignWorks provides several ways of creating windows containing an instance of Internet Explorer which has access to the full range of HTML and web features, plus it can access design data and function within DesignWorks. Scripts written in JavaScript or other scripting languages can be embedded in these HTML pages, allowing substantial new functions to be implemented. With this powerful combination, the package can be substantially extended into new application areas. The DesignWorks JavaScript User's Guide describes this feature.

The remainder of this section outlines some issues that are relevant to the overall design kit structure.

Creating Netlists and Reports

The Export tool can be used to generate netlists and reports in a wide variety of formats. In creating such reports, you might want to consider incorporating these features that tie in with other elements of your design kit:

- Check the "DesignType" attribute value in the design to make sure the report is being generated on an appropriate type of design.

- Create a custom menu (in the design kit's Setup file) to fire up the script instead of having to locate the file each time.

- Consider adding error checking, or breaking out the error checking capabilities into a separate script. You can enforce or suggest that a user run the error check script before generating other reports.

- Consider the optional generation of a "transcript" file that will allow the user to trace any automatic assignments or error conditions that may occur.

Error Checking

Many of the scripting features are intended to implement design error checks. You may wish to consider using the following script features in your error checking scripts:

- Check for duplicate or invalid pin numbers.

- Use regular expressions to check the general format of names or values.

- Count items meeting certain criteria and generate totals.

- Use value tables to check whether attributes are drawn from an allowable set.

Error checking scripts can be made more interactive by developing them for use in conjunction with the Find tool. This allows the user to locate objects on the schematic one at a time and view error information in a more user-friendly manner. See a complete description of Find in Chapter 11—Searching and Browsing Tools on page 245.

Note to Users of pre-4.0 DesignWorks Versions

In general, netlists and reports are generated in the same way they were in previous versions, although you may wish to note these points in converting to the new version:

There should be a high degree of compatibility with existing report forms, although they should be tested carefully in the new version. Some keyword and command changes may affect a few report formats. See more information in the DesignWorks Script Language Reference (separate manual on disk).

Many new report features have been added to enhance error checking and other capabilities. You may want to check existing report forms and consider implementing the following changes using new command language features:

See if any parts of the report script can be merged. In older versions, the only way of performing different formatting on different types of devices was to use the \$FIND \$DEVICES command multiple times. In many cases the new conditional operations mean that many of these sections can be merged into one.

Consider adding more error checking, or breaking out the error checking capabilities into a separate script. You can enforce or suggest that a user run the error check script before generating other reports.

Data Entry

The Export tool allows simple warning alert boxes or simple text entry boxes to be displayed as part of script operation. The values generated by these inputs can be used just like any other text values. This data can be assigned to design, device, signal or pin attributes or can be used to control the flow of script operation.

For more complex data entry and display applications, the JavaScript tool allows arbitrarily complex properties boxes and display panels to be created.

Applications of this feature includes:

- Prompting for global design data, e.g. the revision level, text parameters to pass to external tools, etc.

- Prompting for values for selected devices, e.g. PCB package codes, SPICE parameters, etc.

- Performing display and analysis operations on design entities

- Linking to external design data on other sites

Back Annotation

The Export tool has the ability to read line-oriented text files and extract values using the regular expression features. These values can then be used to locate objects and assign attribute values. This can be used to back-annotate attribute data from a variety of external systems.

See “Using Back Annotation” on page 138 for more information.

Invoking Scripts

Scripts can be invoked directly by the user in a number of ways or automatically in response to other user actions:

- Selecting the Export item in the File menu will prompt for a single Export script file to execute.

- A script can be directly executed from a custom menu item defined in a setup file. See Creating Custom Menus on page 342.

- The Find tool will allow scripts to be executed and the results viewed in specific ways. See “Using the Find Tool” on page 246.

- JavaScript files can be opened directly in the script editor and executed

using the Run Script command.

Scripts can be included in HTML pages loaded into custom panels.

Using Custom Panels

DesignWorks Professional has a Custom Panel facility that allows a separate window to be opened, usually in the tabbed status area at the bottom of the DesignWorks main window. Custom Panels can be used to display design data in application-specific ways, link to external sites and add simple or complex functions to the package.

The Custom Panel window is actually an instance of Internet Explorer and displays any desired HTML or other supported type of file. JavaScript (or other scripting language supported by Internet Explorer) code can be embedded in the page. These scripts can access or modify essentially any design data or program function that is available to the user.

Creating an HTML Page for the Custom Panel

Since the displayed page and the scripts embedded in it are actually running inside a copy of Internet Explorer, all the usual facilities designed for web pages are available, and are well described in many books and web tutorials.

HTML pages for use in the Custom Panel can be created by these methods:

By direct coding in HTML using a text editor. This works well if you are very familiar with HTML coding and if the layout needs of your application are simple. This also allows you the most direct access to the scripting environment.

Using an HTML web page editor. This allows you to create more sophisticated layouts without much knowledge of HTML. In order to make use of the integration with DesignWorks, you will have to understand how scripting functions are placed on the page and how they interact with your layout.

How the Custom Panel is Displayed

In order to display the Custom Panel, you have to specify the URL (i.e. web address or file path) of the HTML file to be displayed. You can display the Custom Panel using any of these methods:

The user can display the panel directly using the Show Custom Panel command in the View menu.

The panel can be displayed at application startup using the `-bp` command line option. See “`-bp` (Browser Panel) Option” on page 395 for more information.

The panel can be displayed using the `showCustomPanel` top-level JavaScript method. See the JavaScript Object Reference (separate manual on disk) for more information.

For more information on Custom Panels and other JavaScript topics, see the JavaScript User's Guide, provided as a separate manual on disk.

Introduction to the Export Tool

This chapter provides basic information on using the Export tool for netlisting and error checking with existing scripts. Some information is provided on simple modifications to report generation scripts that can be made without an in-depth knowledge of the command language. Detailed information on the script language can be found in the DesignWorks Export Script Language Reference (separate manual on disk).

NOTE: Many statements about the operation of netlist generation scripts must be considered to be generalizations. Since the script itself is essentially a program that gets executed, almost any aspect of its behavior can be altered to suit the application. If you are using one of the scripts supplied with the package, check the notes provided in the script for more information (see “Viewing Format Notes” on page 355).

General Information on Export

The Export tool provides powerful and customizable text report generation, error checking, back annotation and design analysis capabilities. The tool works from a script defined in a text command language which allows you to read data from the design and from external text files, and to generate text data out to files and back into the design itself. DesignWorks includes a number of scripts for generating common industry netlist formats and bills of materials, and performing error checking tasks. For cross-platform applications, the Export can directly generate files compatible with DOS, Windows, Macintosh or Unix systems.

The Export tool also works in conjunction with other DesignWorks tools, notably Find, to perform a variety of design searching and checking tasks.

The Find tool is described in “Locating Found Objects on the Sche-

matic” on page 247.

Output File Formats

Export creates text files which are intended to be incorporated into other documentation or transmitted to other systems. Files are saved with a file type of “TEXT”, so they are accessible to almost any word processing program or text editor.

For creating human-readable printed reports, you will generally get the best results by converting the file to a font such as Courier, which has fixed character spacing or by using tabs to space out columnar information. No font or formatting information is stored with the files.

Output File Name

The default output file name is the design file name with “ RPT” added. However, the script file can set any desired file name and optionally prompt the user for a new name before the file is saved.

Generating Standard Netlist and Report Formats

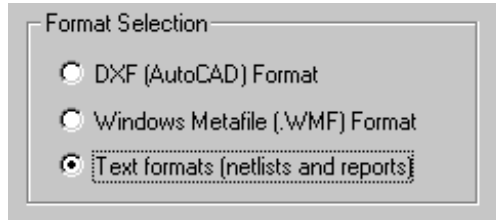
A netlist or report is generated by opening the design you wish to work from, then using the Export command to run the appropriate script

IMPORTANT: Many of the standard formats provided with DesignWorks rely on specific attribute data being entered in the design. Use the Notes button described below to get specific details on each script file, or check the ReadMe file provided with the design kit. If you use a report script that is not from the same design kit as the design template that you started with, make sure you know the requirements of the netlist script.

Basic Report Export Procedure

Select the Export item in the File menu. The following selection will ap-

pear:



Choose the Text Formats option and click Next.

Choose the desired format script from the list. By default, the last item used will be selected in the list.

To run a script, click the Finish button.

The procedure after that is determined by the script file. Most script files will put up a standard file creation box to allow you to specify the name of the output file.

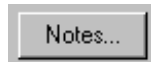
For information on the graphics export options, see “Saving a Circuit Page in WMF, DXF or PDF Graphics Formats” on page 57.

Adding a Script to the Script List

If the script you want to run does not appear in the list, click the Browse button to choose a script file to add to the list of available scripts. Clicking this button will bring up a standard file box displaying all text files with names ending in “.RFM”. Select the desired script file by double-clicking on it.

Viewing Format Notes

The standard script files supplied with DesignWorks will contain a number of comments explaining the usage and requirements of the script. Some formats may rely on certain attribute fields being present in devices in the circuit. These notes can be viewed using the Notes button in the Export box:



Opening the Report on Completion

The “Open finished report on completion” option causes DesignWorks to open the file in the built-in text editor when the report is completed.

NOTE: Some report scripts may generate multiple files or possibly none at all. This option will attempt to open the last file created, if any.

Invoking Export Using Custom Menus

The Export tool can also be invoked using custom menu commands. This allows you to create a convenient command on the application menu bar which gives direct access to frequently-used scripts.

See “Creating Custom Menus” on page 342 for more information.

Device Reporting Options

The listing of devices and device internal circuits can be controlled using the Report Options button in the Properties command for a device. The following options are available:

Omit from report	Nothing about this device or hierarchical block will appear in the netlist.
Report this device	This device or block will appear in the netlist as if it has no internal circuit.
Report subcircuit	The internal circuit of this device will appear in the netlist. I.e. in a hierarchical netlist, the internal circuit will be defined, in a flattened netlist, the internal circuit will be substituted and this device will not appear at all.

For more details on setting device information and options, see “Displaying and Setting Device Information” on page 82. For more information on setting the “report subcircuit” option while creating a hierarchical block symbol, refer to the Restrict attribute field in Appendix A—Predefined Attribute Fields on page 361.

Signal Reporting Options

The listing of signals can be controlled by setting the Omit from report option on a selected signal. For more information, see “Getting and Setting Signal Information” on page 93.

Common Changes to Standard Report Forms

It may be desirable to make minor modifications to one of the standard report scripts in order to tailor it to your design needs. This section covers some common changes that can be made without an in-depth understanding of the script file format. Each of these features is covered in more depth in the DesignWorks Script Language Reference (separate manual on disk).

IMPORTANT: Before modifying any script that comes with DesignWorks, make a backup copy of the original so you will have a reference to go back to.

Script files can be viewed and edited using a text utility such as Notepad or any standard word processor. If using a word processor, be sure to save the file using a “Text Only” option so that no extra formatting information is included.

General Rules

Report scripts consist of text with commands embedded in it. Items starting with a “\$” are commands to the Export and items starting with “&” are references to attribute fields or script variables. Any other text encountered, including blanks, punctuation, and control characters, is simply passed through to the output file. This means that you can place almost anything in a script and it will become part of the output file. The only place where Export is fussy about format is in the arguments to “\$” commands, which usually have to meet some requirements. Anything appearing in braces “{}” is a comment.

Default File Name

The default output file name for a report script is determined by the \$CREATE-REPORT command in a script file. Following is a typical example:

```
$CREATEREPORT($DESIGNNAME.NET) $PROMPT
```

The contents of the parentheses determines the default name of the output file. The keyword \$DESIGNNAME substitutes the name of the current design file. All other characters inside the parentheses are used verbatim. The \$PROMPT keyword indicates that a standard file box should be displayed to confirm the

file name. If this keyword is not present, the file is saved in the same folder as the design without any prompt.

If no \$CREATEREPORT command is present in the script (or if the script generates some output before it is encountered), a name will be generated by adding the word “Report” to the design name and prompting the user.

Attribute Field Usage

Attribute field data is heavily used in generating netlist output files. Most of the standard script files are set up to use the pre-defined attribute fields. These are easily modified to use customized fields which you have added to your design.

Attribute fields are referenced in a script file using the form:

```
&fieldName
```

Wherever this format is used, any other field name can be substituted. For example, a typical device listing will appear as follows in a script file:

```
$DEVICES$DEVNAME &Part
```

The “&Part” item will be substituted with the contents of the Part attribute field. Any other field name can be referenced as desired, or additional ones can be added to the line.

Extracting Power and Ground Connections from Attributes

The Export tool provides two general methods of specifying an attribute field that is going to be used to list power and ground pins. The first, and most direct, method is the \$SIGSOURCE command. You will frequently see a line like this one in a report script:

```
$SIGSOURCE(Ground)
```

This line indicates that any pins listed in a field named Ground on a device should be added to the net with the same name in the netlist. If you have power and ground fields that are commonly used in all your designs, you can simply add more lines like the one above to the report script. See the Design-Works Script Language Reference (separate manual on disk) for more options on this keyword.

Another method which provides more flexibility in specifying different power and ground fields for each design is the \$DESIGNSIGSOURCE command, as in:

```
$DESIGNSIGSOURCE(SigSources)
```

This command specifies that a *design* attribute field named SigSources should be searched for a list of power and ground fields. This is in most ways equivalent to putting a list of \$SIGSOURCE lines in a special report script for that design. See the entry for "\$DESIGNSIGSOURCE" in the DesignWorks Script Language Reference (separate manual on disk) for more information.

For a more complete discussion of the various methods for making power and ground connections, see "Power and Ground Connections" on page 205 and the DesignWorks Script Language Reference (separate manual on disk).

Script Errors

Two general types of errors can occur in generating a report:

Script syntax or execution errors—The Export command language is such that any items in the script file that are not recognized as commands are transmitted to the output file unmodified. Thus, most errors do not result in an explicit error message, but generate unexpected results in the output file. If an error is detected in an area of the script that requires a specific format, a message box is displayed showing the line number of the error. If such an error appears while running a script provided with DesignWorks, contact our Technical Support department for assistance. If you have created or modified the script yourself, refer to the DesignWorks Script Language Reference (separate manual on disk) for assistance in correcting the problem.

Design data errors—A number of circuit data errors can occur during report generation, including unnumbered pins, excessive string length, missing or incorrect attribute values, etc. It is the responsibility of the script to check for the errors that may affect the type of output being generated. Errors can be displayed immediately in the form of an alert box, or may be logged to an output file, depending on the design of the script. See the Format Notes or ReadMe file supplied with the design kit you are using for more information.

Cross-Platform Compatibility of Scripts and Report Output Files

One of the well-known aggravations of the computer industry is the lack of agreement on line termination characters in text files across various computers and operating systems. In order to make it easy to work in a mixed operating environment, Export provides a great deal of flexibility in reading input and generating output.

Line Termination in Script Files

When reading a script input file, Export accepts a single carriage return, a single line feed, or a combination of these in either order as a line terminator. This means that a text file from any Windows, Macintosh or Unix system can be used as input without affecting the output format. Script files created on a Macintosh can be used as input to Export in DesignWorks for Windows and vice versa, even if they do not display correctly in a text editor on the executing platform.

Line Termination in Output Files

The line termination format that is used in the report output generated by Export is determined by the \$LINETERMINATOR command in the script file and does not depend on the line termination used in the script file. If no such command is present in the script file, then the output files will be terminated in the native format of the machine the script is being run on. Note that Export also has the capability of generating a secondary “transcript” file for error logging, etc. This file always uses the line termination of the host computer system.

See more information on line termination in the section “Line Terminators” in the DesignWorks Script Language Reference (separate manual on disk).

Appendix A—Predefined Attribute Fields

The following tables list fields that are predefined for each new design that is created. These fields cannot be removed because many of them are used internally by specific DesignWorks features.

For more information on defining custom attribute fields see “Defining a New Attribute Field” on page 177.

For convenience, the fields marked as “primary” by default are listed first since they are most commonly entered directly by the user. Thus, fields that are not used in a given application can be effectively hidden.

For more information on setting the Primary/Secondary option, see “Primary vs. Secondary Fields” on page 158.

The following information is provided in the table:

- Field Name**—The name of the field as it appears in the Define Attributes box.
- Used In**—The types of objects this field is used in.
- Instance/Definition**—Specifies whether the field data is kept with the definition or instance or a sub-circuit.
- Description**—How the field is used.

NOTE: Many of the predefined fields are used for specific DesignWorks features. More details on the usage of these fields can be found in the manual sections covering those features.

Field Name	Used In	Inst/Def	Description
AutoSym.Bottom	Devices	Def	The string used in the “bottom pins” box in the Auto Create Symbol command in the device symbol editor. See “Automatically Creating Symbols” on page 303.

AutoSym.Left	Devices	Def	The string used in the “left pins” box in the Auto Create Symbol command in the device symbol editor. See “Automatically Creating Symbols” on page 303.
AutoSym.Right	Devices	Def	The string used in the “right pins” box in the Auto Create Symbol command in the device symbol editor. See “Automatically Creating Symbols” on page 303.
AutoSym.Top	Devices	Def	The string used in the “top pins” box in the Auto Create Symbol command in the device symbol editor. See “Automatically Creating Symbols” on page 303.
BusInfo	Pins	Def	Used only on bus breakouts to notate the names of the attached signals. Normally only updated using the Bus Pin Options command. See “Displaying a Bus Pin Annotation” on page 200.
Category	Devices	Def	Contains a terse device category code in the DesignWorks libraries. Can be used as an alternate device prefix by entering the name of this field in PrefixField. Not used internally.
CctName	Design	Def	Design file name. Sets the window title and name of next saved file. Normally only set by the Open and Save commands.
CctOS	Design	Def	The system (Mac or Windows) that was used to create the design file. This is used only for creating external circuit references for sub-circuit parts. Changing it will not affect the current design file. Normally only set by the Open and Save commands.
CctPath	Design	Def	The directory path name for the design file. This is used only for creating external circuit references for sub-circuit parts. Changing it will not affect the current design file. Normally only set by the Open and Save commands.
DateStamp.Cct	Design	Def	This field is set by the program to the current date and time each time the circuit is modified. This is intended to allow scripts to determine if a design has been modified since the last time the script was run.

DateStamp.Dev	Devices	Def	This field is set by the program to the current date and time when a device is placed in the circuit and not modified thereafter. This is intended for back annotation processes. See “Device Date Stamping” on page 154.
DateStamp.Last	Design	Def	This field is used by the program as part of the date stamping process. See “Device Date Stamping” on page 154.
DateStamp.OS	Design	Def	This is set to the system that the date stamping was performed on, to ensure cross-platform compatibility of dates. See “Device Date Stamping” on page 154.
DateStamp.Symbol	Devices	Def	This field is set by the program when a symbol (part definition) in a library is modified. See “Symbol Date Stamping” on page 313 for more information.
Delay.Dev	Devices	Def	Used with the DesignWorks Simulator option.
Delay.Dev.Max	Devices	Def	Used with the DesignWorks Simulator option.
Delay.Dev.Min	Devices	Def	Used with the DesignWorks Simulator option.
Delay.Dev.Typ	Devices	Def	Used with the DesignWorks Simulator option.
Delay.Pin	Pin	Inst	Used with the DesignWorks Simulator option.
Delay.Pin.Max	Pin	Def	Used with the DesignWorks Simulator option.
Delay.Pin.Min	Pin	Def	Used with the DesignWorks Simulator option.
Delay.Pin.Typ	Pin	Def	Used with the DesignWorks Simulator option.
Depth	Devices	Def	Indicates nesting depth of device sub-circuit. Only used temporarily by the Report tool while generating hierarchical reports. This value is not maintained during editing operations. See the entry Depth Ordering in Pure Netlists in the DesignWorks Script Language Reference (separate manual on disk).
Description	Devices	Def	A short description of the device type. This is included in all standard DesignWorks libraries. Not used internally.
Designer	Design	Def	Intended to be used to display the designer’s name on the sheet. Not used internally. See “Creating a Title Block” on page 339.

DesignType	Design	Def	A value used to distinguish which design template was used to create the design. This is used to check that the appropriate netlist scripts are being used, etc. See “Contents of a Design Template” on page 331.
DevPrefix	Design	Def	Default prefix for auto-assigned device names. Normally set using the Device Naming and Packaging Options command. “Enabling Naming and Packaging Options” on page 123.
ExtCctDate	Devices	Def	Modified date of external design file containing device internal circuit. Normally set by the device symbol editor.
ExtCctLib	Devices	Def	Name of library to search for an internal circuit.
ExtCctName	Devices	Def	Name of external design file containing device internal circuit. Normally set by the device symbol editor.
ExtCctOS	Devices	Def	Source system type for design file containing device internal circuit. Normally set by the device symbol editor.
ExtCctPath	Devices	Def	Directory path for the design file containing the device internal circuit. Normally set by the device symbol editor.
Function	Devices	Def	Contains a terse device function code. Can be used as an alternate device prefix by entering the name of this field in PrefixField. Not used internally.
Ground	Devices	Def	List of pins to be hooked to ground net in netlist, separated by commas. See “Power and Ground Connections in Attributes” on page 207.
HierNameSep	Design	Def	Contains the separator string used in generating hierarchical device and signal names for Report output. Normally set only by commands in a report script file. See the entry for the \$HIERNAMESeparator command in the DesignWorks Script Language Reference (separate manual on disk).
InstName	Devices, signals	Inst	Name applied to physical instance. This is normally used for the package name in hierarchical designs. Normally not used in Flat or Pure mode designs. See “Name vs. InstName” on page 136.
Initial.Dev	Devices	Inst	Used with the DesignWorks Simulator option.

Initial.Pin	Pin	Inst	Used with the DesignWorks Simulator option.
Initial.Sig	Signals	Inst	Used with the DesignWorks Simulator option.
Invert.Pin	Pin	Def	Used with the DesignWorks Simulator option.
LibDate	Devices	Def	“Last Modified Date” of the library the device was read or updated from, stored in encoded form. This is used by the Update from Lib command. See “Updating a Symbol from a Library” on page 269.
LibName	Devices	Def	Name of the library the device was read or updated from. This is used by the Update from Lib command. See “Updating a Symbol from a Library” on page 269.
LibPath	Devices	Def	Directory path of the library the device was read or updated from. This is used by the Update from Lib command. See “Updating a Symbol from a Library” on page 269.
LibOS	Devices	Def	Source system of the library the device was read or updated from. This is used by the Update from Lib command. See “Updating a Symbol from a Library” on page 269.
LibType	Devices	Def	The original name of the part when it was read from a library. This is used by the Update from Lib command. See “Updating a Symbol from a Library” on page 269.
Name	Devices, signals	Def	Definition name, i.e. reference designator. This is the field set using the text tool on the schematic. See “Name vs. InstName” on page 136.
Name.Prefix	Devices	Def	Prefix used to create default Name value. Used by Packager and Auto-name. See “Setting the Name Prefix for a Symbol” on page 133.
Name.Pt	Devices	Def	Default position of Name attribute relative to device. See “Using Default Position Fields” on page 175.
Name.Spice	Devices	Def	Alternate prefix for SPICE-based simulators. See “Selecting an Alternate Prefix Field” on page 133.
OKErrors	Devices, Signals, Pins	Def	Used by the Find tool to mark errors set as OK by user. Should not be set manually. See the entry Implementing Mark as OK in Error Checking Scripts in the DesignWorks Script Language Reference (separate manual on disk).

Package	Devices	Def	Device package code. This is included in all standard DesignWorks libraries. Not used internally. See “Selecting the Part and Package Type” on page 87.
Package.List	Devices	Def	List of package types that apply to this part type. See “Using Value List Fields” on page 173.
PageRef	Devices	Def	Used in Page Connector pseudo-devices only. Contains the auto-generated page references. See “Automatic Display of Page References” on page 201.
PageRefFormat	Design	Def	Format string for page references. Normally modified only using the Design Preferences command. See “Automatic Display of Page References” on page 201.
PageRefWidth	Design	Def	Max. number of items per line in page references. Normally modified only using the Design Preferences command. See “Automatic Display of Page References” on page 201.
Part	Devices	Def	Part type code. Used by Packager to assign gates to packages and in external netlists and bills of materials to identify the part type. See “Selecting the Part and Package Type” on page 87.
Part.List	Devices	Def	List of part types possible for this symbol. See “Using Value List Fields” on page 173.
Part.Pt	Devices	Def	Default position of Part attribute relative to device. See “Using Default Position Fields” on page 175.
Permutable	Devices	Def	This field contains a list of the swappable pins and gate sections in this part type. Parentheses “()” enclose items that can be swapped, while brackets “[]” enclose items that cannot be swapped. This is intended for use by external PCB layout packages. Not used internally.
Pin Sequence	Devices	Def	This field is used to determine pin order in netlist output formats that require specific device pin ordering, e.g. SPICE. This is <u>not defined</u> in the standard DesignWorks libraries. Not used internally. See the entry for the \$DEVPINSEQUENCE command in the DesignWorks Script Language Reference (separate manual on disk).

PkgLevel	Devices	Def	<p>An integer 0 to 3 specifying packaging action: 0 = normal 1 = lock and check 2 = lock and don't check 3 = ignore Normally set using the Properties command. See "Setting Device Packaging Options" on page 134.</p>
PkgPrefix	Design	Def	<p>Default prefix used by Packager to create a package name. Normally set using the Device Naming and Packaging Options command. See "Design Attribute Fields Used By the Packager" on page 151.</p>
Power	Devices	Def	<p>List of pins to be hooked to power net in netlist, separated by commas. See "Power and Ground Connections in Attributes" on page 207.</p>
PrefixField	Design	Def	<p>Contains the name of the field to use as a name prefix for device packaging and device auto-naming. See "Design Attribute Fields Used By the Package" on page 151.</p>
Restrict	Devices	Def	<p>An integer value from 0 to 7 used to set access restrictions for the device's internal circuit. The value is the sum of the following: 1 = Don't report 2 = Don't package 4 = Don't Push Into Normally set using the Properties command. See "Locking and Unlocking Subcircuits" on page 221.</p>
Revision	Design	Def	<p>Intended to be used to display the designer's name on the sheet. Not used internally. See "Creating a Title Block" on page 339.</p>
Script.Dev	Design	Def	<p>The name of a script to run each time a device is placed.</p>
Script.Open	Design	Def	<p>The name of a script to run each time the design file is opened.</p>
Script.Pin	Design	Def	<p>Reserved</p>
Script.Sig	Design	Def	<p>Reserved</p>
SigPrefix	Design	Def	<p>Prefix used to generate default signal names. Normally set using the Signal Naming Options command. "Using Signal Auto-Naming" on page 210.</p>
Sim.InputMap	Design	Def	<p>Used with the DesignWorks Simulator option.</p>

Spice	Devices, Design	Def	Holds simulation parameters for SPICE-based simulators. Not used internally. See “Using DesignWorks with SPICE-based Simulators” on page 45.
TestVectors.Cct	Devices	Def	Used with the DesignWorks Simulator option.
TestVectors.Dev	Devices	Def	Used with the DesignWorks Simulator option.
Unit	Devices	Inst	Gate unit, e.g. a, b, c, etc. Set and checked by Packager or can be set manually. See “Attribute Fields Set By the Packager” on page 145.
Unit.All	Devices	Def	List of all units available in this part type. Used by Packager. See “Required Packaging Attributes” on page 145.
Unit.List	Devices	Def	List of all units in this part type having this symbol. Used by Packager. See “Required Packaging Attributes” on page 145.
UnusedPins	Devices	Def	List of unused pins in this part type. Not used internally.
Value	Devices	Def	Component value to appear in netlists. Not used internally. See “Selecting the Part and Package Type” on page 87.
VisPin.List	Pins	Def	List of pin numbers for this pin corresponding to the gate units in “Unit.List”. Used by Packager. See “Required Packaging Attributes” on page 145.

Appendix B—Primitive Device Types

Every device on a DesignWorks schematic has a characteristic known as its *primitive type*. The primitive type is set when the part entry in the library is created and cannot be changed for individual devices on the schematic.

Primitive types fall into three general groups:

Schematic symbols: The two primitive types SUBCIRCUIT and SYMBOL fall into this category and are the normal primitive types used for creating schematic symbols. SUBCIRCUIT is the default type for symbols created using the device symbol editor. There are no restrictions on the ordering or type of pins on these symbols.

Pseudo-device types: These are the symbols used for page connectors, power and ground symbols, etc.

IMPORTANT: DesignWorks has very specific requirements for the order and type of pins on pseudo-devices. Refer to the following table for information. These rules ARE NOT CHECKED by the device symbol editor.

Simulation types: The majority of the primitive types defined in the following table are simulation primitives and are intended for use with the DesignWorks Digital Simulator option.

IMPORTANT: The simulation primitive types should not be used for user-created symbols without a clear understanding of their function. See the Digital Simulator manual for more information.

Schematic Symbol Primitive Types

Primitive Type	Pin Requirements	Comments
SUBCIRCUIT	No restrictions	Symbol having an optional internal circuit. This is the default for symbols created using symbol editor.
SYMBOL	ENo restrictions	Symbol with no internal circuit.

Pseudo-Device Primitive Types

IMPORTANT: The pin requirements listed in the following table must be followed when creating pseudo-device symbols. These rules ARE NOT CHECKED by the symbol editor.

Primitive Type	Pin Requirements	Comments
BREAKOUT	Pin 1 is Bus Pin, followed by N Normal Pins, set to Input	Splits signals out of or into a bus.
SIGNAL CONNECTOR	Exactly 1 normal pin, normally set to Input	Used for power and ground connections.
PORT CONNECTOR	Signals - exactly 1 pin, Busses- exactly 1 bus pin with any number of internal pins	Makes a connection between the signal it is connected to and a like named pin on the parent device.
PAGE CONNECTOR	Signals - exactly 1 pin, Busses- exactly 1 bus pin and no internal pins	Exports the signal or bus globally across all pages at one level of a schematic.

Simulation Primitive Types

WARNING: Do not use these without referring first to the Digital Simulation manual for more information!

Primitive Type	Pin Requirements	Comments
NOT	1 Input, 1 Output, ...	Output complement of Input. Same for each gate.
AND	N Inputs, 1 Output	Output <- AND of all Inputs
OR	N Inputs, 1 Output	Output <- OR of all Inputs
XOR	N Inputs, 1 Output	Output <- XOR of all Input
XNOR	N Inputs, 1 Output	Output <- XNOR of all Input
D-FF	/Set, Data, Clock, /Reset, Q, /Q	
JK-FF	/Set, J, /Clock, K, /Reset, Q, /Q	
D-FF (non inv)	Set, Data, Clock, Reset, Q, /Q	Set and Reset are active high.
JK-FF (non inv)	Set, J, /Clock, K, Reset, Q, /Q	Set and Reset are active high.
BUFFER	1 Input, 1 Output, ...	Output is Input. Same for each gate.
MULTI- PLEXER	N Data, M Select, /E, Output	Routes 1 of N inputs to the output. M is the smallest value such that $2^M \geq N$
DECODER	N Select, M Output, /E,	Selects 1 of M outputs. N is the smallest value such that $2^N \geq M$
REGISTER	N Inputs, N Outputs, Clock, Clear	Inputs are latched when clock is high, clear is async.
SHIFT REG	N Inputs, N Outputs, Clock, Shift/~Load, Shift-In	Shifts or loads data on clock.
ADDER	N A-Inputs, N B-Inputs, N Outputs, Carry In, Carry Out	Adds A and B w/Carry
COUNTER	N Inputs, N Outputs, Clock, Count/~Load	Counts up or loads on clock.
X-GATE	2 Bidirectional, Control, ...	The 2 bidirectional pins are connected if control is active. Repeated for each gate.
RESISTOR	2 Bidirectional, ...	Converts high drive to low drive for each resistor.
PULL_UP	1 Bidirectional, ...	Outputs a high with a low drive level for each pin.

ONE SHOT	Clock, /Clear, Q, /Q	Retriggerable. Delay is 1 by default. Delay setting is used as pulse width.
CLOCK	1 Output	The delay setting is used as the high and low period of the clock.
PROBE	1 Input	Displays Boolean input value on the schematic
SPST-SWITCH	2 Bidirectional Pins	Makes a connection between the two pins when switch is closed, otherwise both pins are undriven.
SPDT-SWITCH	3 Bidirectional Pins	Has pins A, B, and C; A and C are connected or B and C are connected.
HEX KEYPAD	Q0..3 Outputs, Strobe Output	Outputs a 4 bit hex value, strobe goes high when key pressed.
PUSH BUTTON	3 Bidirectional Pins	A, B, and C Pins; A and C are connected or B and C are connected. Switch is momentary.
SETUP & HOLD	Clock, 1 Data (Input), Error (Output)	Error = X if Data changed after Setup or before Hold time, Error = Z otherwise.
GLITCH DETECTOR	Sense (Input), Error (Output)	Error = X if sense changes faster then delay setting, Error = Z otherwise.
UNKNOWN DETECTOR	Sense, Output	If Sense = 0 or 1 then Output = 0, Output = 1 otherwise.
SIMULATOR STOP	1 Input	Performs requested task (stop simulator, generate report, ...) on rising edge.
LOGICBOX		Represents LogicBox hardware interface. See LogicBox Manual.

Appendix C—Device Pin Types

Every device pin has a characteristic known as its *pin type*. The pin type is set when the part entry in the library is created and cannot be changed for individual device pins on the schematic.

See “Setting the Pin Type” on page 299 for information on how to set the pin type while creating a device symbol.

What Pin Types are Used For

For many general schematic editing purposes, the pin type will be unimportant and can be ignored. However, pin type settings are important in the following cases:

Pin type settings are crucial if you plan to use the DesignWorks Digital Simulation option. The pin type of each pin determines the direction of signal flow and must be appropriately set for each symbol.

Pin type information is required in many netlist file formats for FPGA layout and digital simulation.

Correct pin type settings allow error checking scripts to check for fanout and multiple drive situations.

Other future analysis tools may use this information for timing and loading analysis.

Pin Types Table

The following table lists the function of each of the pin types available in DesignWorks.

Pin Type	Description
IN	Input - this is the default for pins created using DevEditor. This setting is used for all pins on discretes except those with some digital function.
OUT	Output - always enabled.
3STATE	Output - can be disabled (i.e. high-Z)
BIDIR	Bidirectional
OC	Open collector output - i.e. pulls down but not up
BUS	Bus pin - This does not represent a physical signal but is a graphical representation of a group of internal pins, each having its own type.
LOW	Output - always driving low
HIGH	Output - always driving high
LTCHIN	Input to a transparent latch - this is used for calculating cumulative setup and hold times.
LTCHOUT	Output from a transparent latch - this is used for calculating cumulative setup and hold times.
CLKIN	Input to an edge-triggered latch - this is used for calculating cumulative setup and hold times.
CLKOUT	Output from an edge-triggered latch - this is used for calculating cumulative setup and hold times.
CLOCK	Clock input - this is used for calculating cumulative setup and hold times.
OE	Open emitter output - i.e. can pull up but not down.
NC	A no-connect pin
POWER	Power supply pin
DRIVER	A high-current or high-voltage driver pin
ANALOG	An analog input or output

Appendix D—Ini File Format

The “DW.ini” file is divided into sections. Each section starts with a section heading contained within square braces, e.g. [Drawing]. Within a section each non-blank line is either a statement or a comment. A statement is a keyword, which specifies an option, followed by an equal sign, which is a separator, followed by the option's value. Statements are terminated by a carriage return. A comment is any line that starts with “/”.

Specifying File and Folder (Directory) Names

A number of the INI file keywords are used to specify file and folder locations for various purposes. This section describes a number of features available to make it easier and more flexible to specify the locations of items on disk.

More information on file and directory locations and search paths is given in Appendix F—Installation on page 397

Absolute and Relative Path Names

Folders can be specified using either an *absolute* (i.e. starting with a drive name) or *relative* path. If the path begins with a drive or server name, the path is absolute, otherwise it is relative to the location of the DesignWorks application. The following are valid statements:

```
Directory = Libs
```

specifies a folder called Libs but does not give its absolute location. In this case, the program looks first in the root folder (See “Specifying a Root Directory” on page 397.), if any, then in the “start in” folder specified when the program was started, then in the folder containing the application itself. In most installations, these three locations are all the same and only the application folder is searched.

```
Directory = d:\4000 Series
```

specifies a folder called “4000 Series” on disk D.

```
Directory = ..\Libs\7400 Counters
```

specifies a folder called “7400 Counters” which is contained inside a folder called Libs which is in the same folder one level up from the application.

For some file types, such as “.exe” application files, DesignWorks will also search in the system folder. This allows commonly-used applications such as NotePad to be specified without a directory path.

Using Environment and Registry Variables

Most places in the INI file (as well as Export scripts) that allow you to specify a file or folder name, also allow references to Windows environment variables and registry entries as sources of information about file location. This is primarily intended for use in cases where references are needed to other applications or data files that may not be installed with DesignWorks and are not in a fixed location.

It is beyond the scope of this section to discuss how environment or registry entries are set up. Please refer to the appropriate Windows documentation.

The following example makes reference to an environment variable called “DataDir”:

```
menu= XML, "Generate XML Netlist", "%DataDir%xml_netlist.rfm"
```

IMPORTANT: The value of the variable is placed in the resulting name strictly by text substitution without any checking. It is the user's responsibility to ensure that this results in a valid file name with appropriate name separator characters (e.g. backslashes between directory names), etc. For example, in the above case, the variable DataDir should contain a terminating backslash character in order to create a valid file name when it is combined with the other text in the file name. Also note that some commonly-used system environment variables such as PATH may contain a list of alternate directories to search. When used in this context, DesignWorks DOES NOT automatically break out such lists and search all the specified directories.

Here is a sample that refers to a registry entry in order to locate an outside application to execute in response to a custom menu (note that all of the following must be on one line):


```
menu= Simulate, "Run Simulator", "%HKEY_LOCAL_MACHINE\
SOFTWARE\JoesApps\SuperSim\AppPath%"
```

Since the Windows registry allows multiple values to be associated with a separate name for each value, this format also allows a value name to be specified by appending a colon ":" and the name, as follows:

```
Library = "%HKEY_LOCAL_MACHINE\SOFTWARE\Capilano\
DesignWorks:InstallPath%\Libs\thing.clf"
```

Using Common System Locations

File names can be specified with a variable that indicates one of a number of common system locations.

%my documents%	The document folder for the current user
%program files%	The Program Files folder on this computer
%desktop%	The Desktop folder on this computer
%windows%	The Windows system folder on this computer

For example:

```
%windows%NotePad.exe
```

Section [Drawing]

Initial Directory Settings

```
Directory = dir_name
```

This statement specifies the initial working directory. The default working directory if this statement does not appear in the Ini file is the same as set by Window's Program Manager.

Font Settings

```
XXX_Font = font_name font_size [BOLD ITALIC]
```

This statement specifies the font for text items appearing in a schematic document. “font_name” is the name of a TrueType font, only TrueType fonts are supported. “font_size” is the point size to use. There are two optional style keywords which may be applied, “BOLD” and “ITALIC”. The possible items which may have there font specified are:

Default_Font, Attribute_Font, Border_Font, MiscText_Font,
Pin_Font, Symbol_Font, PrintPage_Font

Color Settings

XXX_Col =ColorName

This statement specifies the color for items appearing in a schematic document. All items in a schematic, except the page background, default to black. The page background defaults to white. The possible values for XXX_Col are:

Default_Col

DeviceAttrs_Col, SignalAttrs_Col, BusAttr_Col,
PinNumber_Col, PinNumber_Selected_Col

Device_Col, Signal_Col, Signal_Selected_Col, Pin_Col,
Pin_Selected_Col, Bus_Col, Bus_Selected_Col, BusPin_Col,
BusPin_Selected_Col

Page_Col, Boundary_Col, GridMajor_Col, GridMinor_Col,
RandowText_Col, RandowTextFrame_Col

The available colors are (letter case is not significant in color names):

Red
Green
Blue
Cyan
Magenta
Yellow
Black
DkGray
Gray
LtGray
White
VltGray
LtBlue
LtGreen
LtRed
DkBlue

DkGreen
 DkRed
 Violet
 Orange

Specifying the Location of Design Templates

Templates = folder

The Templates keyword specifies the location of the design template files that will be displayed in the New Design box. This keyword can be specified multiple times to point to multiple template folders, if desired.

See “Where Design Templates are Stored” on page 332 for more information.

Specifying the Location of Example Files

Examples = folder

The Examples keyword specifies the location of the example files that will be displayed when the Examples button in the New Design box is clicked. This keyword can be specified multiple times to point to multiple template folders, if desired.

Adding Custom Menu Commands to Popup Menus

Keywords: POPUPSCRIPTDEV, POPUPSCRIPTSIG, POPUPSCRIPTPIN, POPUPSCRIPTCCT

See “Adding a Menu Item to Popup Device, Signal, Pin or Circuit Menus” on page 344.

Adding Default Attribute Field Definitions

The setup file keyword ATTRFIELD allows you to specify attribute field definitions that will be included in every new design that is created. The format of this keyword is as follows:

```
ATTRFIELD = fieldName
ATTRFIELD = fieldName/options;
```

"fieldName" can be any valid attribute field name, i.e. consisting of letters, digits and the period and underscore characters “.” and “_”. The maximum field name length is 16 characters and spaces are not allowed.

If no options are specified, the field will be a device field with default maximum length and switch settings.

A number of switch setting options can be specified as letters and digits after a “/” following the field name. If any digits follow the “/”, they are taken to be the maximum data length for this field. The default is 32000. Any combination of other option letters can be selected from the following table. The order of the option letters is not important.

Any number of new fields can be added by using multiple ATTRFIELD lines. Any line specifying the name of an existing field will be ignored.

Option Letter	Meaning
A	Allow Carriage Return character in data
C	Allowed in design
D	Allowed in devices
F	Fixed, i.e. user cannot remove field definition
G	Group with Name
I	Instance data, i.e. can have different value inside different instances of the same hierarchical block
L	Location fixed on schematic
N	Only allow name characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
P	Allowed in pins
R	Rotate with object
S	Allowed in signals
T	Do not display value list, even if a .List field is present
U	Only allow numeric characters in field value (NOTE: This is not enforced in the current version of DesignWorks)
V	Visible by default
W	Show Field Name by default
X	Read only value
Y	In Primary list by default

ATTRFIELD examples:

```
ATTRFIELD = Aperture/SY10
ATTRFIELD = InputDelay/PYU10
ATTRFIELD = ModelText
```

Enabling Auto-Backup and Timed Auto-Save

Backup = On

This statement specifies that you want an automatic backup created each time an existing circuit file is resaved. The default is On.

```
AutoSave = 15
```

This statement specifies that you want to be prompted to save your work whenever the specified number of minutes has passed without a save. The default is 0, i.e. no auto save.

See “Backup Procedures” on page 61 for more information.

Disabling Device Date Stamping

This statement controls the application of a date stamp to devices whenever they are edited.

```
NODATESTAMP
```

This keyword does not take any value. If the keyword appears, date stamping will be disabled. If it does not appear, date stamping is enabled.

See “Device Date Stamping” on page 154. for more information.

Specifying Standard Sheet Sizes

The SheetSize keyword is used to specify the standard sheet sizes that are available in the Sheet Size Wizard command using the following format:

```
SheetSize = name width height
```

Where:

name is the text that you want to appear in the sheet size list. This should be surrounded by double quotes if it contains any non-alphanumeric characters.

width and **height** are decimal numbers expressed in internal units:

```
1 inch = 1008
1 cm = 397
```

For example:

```
SheetSize = "ANSI A Portrait - 8-1/2W x 11H" 8568 11088
SheetSize = "ANSI A Landscape - 11W x 8-1/2H" 11088 8568
SheetSize = "ANSI B - 17W x 11H" 17136 11088
```

```
SheetSize = "ANSI C - 22W x 17H" 22176 17136
SheetSize = "ANSI D - 34W x 22H" 34272 22176
SheetSize = "ANSI E - 44W x 34H" 44352 34272
```

See “Setting Custom Sheet Size using Sheet Wizard” on page 335. for more information.

Solid Grid Lines

The SOLIDGRID Ini file keyword determines if grid lines are drawing with solid lines. If not on then the grid is drawn with the default dotted lines. On some platforms dotted lines are not supported correctly or may be slow.

```
SOLIDGRID = on
```

Zoom Factors

```
SCALES = n1..n11
NORMALSCALE = index
```

The SCALES statement is used to specify the magnification levels used by the Reduce and Enlarge commands. The keyword is followed by 11 decimal integers separated by blanks and sorted in ascending order. The 1:1 scale level (at which externally created pictures appear in their original size) is 14. Enlargements are specified by smaller numbers (e.g. 7 gives 200%) and reductions by larger numbers. The default values are:

```
SCALES = 4 7 10 14 18 24 28 42 63 98 140
NORMALSCALE = 3
```

The NORMALSCALE statement is used to specify which of the scale steps specified in the SCALES line will be used as the “Normal Size” setting. The index must have a value in the range of 1 to 11.

Pin Spacing

```
PINSPACE = n;
```

The PINSPACE keyword is used to specify the spacing between adjacent pins when breakout symbols are created. This can also serve as the default for symbols created by other tools. The value must be a single decimal integer which is used as a multiple of the standard grid space of 5 dots. The default value is 2.

Breakout Parameters

```
BREAKOUT = dth dtv;
```

The BREAKOUT keyword lets you control the creation of bus breakout symbols generated by the program. This does not affect any breakouts in existing files as these symbols are already created and stored with the file.

The BREAKOUT keyword is followed by two numbers for the following parameters:

- dth** the horizontal offset (in screen dots at 100% scaling) for placement of text names on a breakout
- dtv** the vertical offset (in screen dots at 100% scaling) for placement of text names on a breakout

Disabling “Loose End” Markers on Signal Lines

The NOLOOSEENDS Ini file keyword disables the cross-markers that are normally displayed on the screen at the ends of unconnected line segments. The format of the command is:

```
NOLOOSEENDS = off
```

Undo Levels

The UNDO Ini file keyword indicates the number of levels of undo which should be maintained. A value of zero means that there is no undo. The default value is 10.

```
UNDO = n
```

NOTE: Undo data is held in memory during program operation. A higher value here will increase the memory space occupied by the program.

Fine-Tuning Pin Number Text Display

The PINTEXT keyword allows you to adjust the display position of pin numbers on devices. The format of this keyword is as follows:

```
PINTEXT = dth dtv
```

“dth” and “dtv” are positive or negative integer offsets which adjust the text position by the given amount in screen dots at Normal Size screen scale. All devices in the design are equally affected.

IMPORTANT: This adjustment should not be required in normal use and should be used

with caution. No checking is done on the range of these settings.

NOTE: Changing these numbers in the Ini file will not automatically recalculate the positions of pin numbers in existing designs. You can force a recalculate by using the Design Preferences command to change the pin text font or size, then change it back to the original setting.

Internal Error Checking

```
NOERRORCHECK = off
MSGCHECK = on
```

The first statement disables the internal error checking described elsewhere in the Appendix. The second statement increases what is checked.

NOTE: This should only be done in consultation with our Technical Support department as it will prevent warnings from being issued for internal or unusual system errors. In many cases these errors can be easily recovered if corrected immediately but may cause data corruption if left undetected.

Section [Libraries]

Specifying Libraries to Open at Startup

Specifying the Default Library Folder

```
FOLDER = directory_path
```

This specifies the folder/directory that will contain the libraries to be specified in following LIBRARY statements. This statement can be omitted if the libraries are located in the same directory as the DesignWorks executable, or if you prefer to specify a complete file “library_path” in each library statement.

Specifying a Single Library

```
LIBRARY = library_path
```

This specifies a single library to open. The “library_path” can be simply the name of the library if the library is in the current directory, or a relative path to the library, or a fully qualified path from the root. For example:

```

LIBRARY = Lib1.clf
LIBRARY = lib\74LS00.clf
LIBRARY = \mylibs\blocks\controls.clf

```

Opening All Libraries in a Folder

```

LIBRARYFOLDER = directory_path

```

This names a folder/directory to be searched for libraries. All libraries in this folder will be opened. Folders inside this folder are not checked. The format of the folder name is the same as for the FOLDER keyword above.

Section [DevEditor]

This section contains items affecting the device symbol editor tool.

Default Font

```

Font = "font_name" font_size [BOLD ITALIC]

```

This statement specifies the default font for text items appearing in a device symbol editor document. “font_name” is the name of a TrueType font, only TrueType fonts are supported. “font_size” is the point size to use. There are two optional style keywords which may be applied, “BOLD” and “ITALIC”.

Grid Settings

```

GridColor =RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW
            BLACK, DKGRAY, GRAY, LTGRAY, or WHITE
GridSize = grid
SnapSize = snap
PinSnapSize = pinsnap

```

The GridColor statement specifies what color to use when drawing the symbol editor’s Grid. GridSize, SnapSize and PinSnapSize are all expressed as a 5 pixel multiplier. GridSize specifies the number of 5 pixel spaces between displayed grid lines. SnapSize sets the grid snap for all graphical objects, except pins, and PinSnapSize sets the snap used when positioning pins.

Default Pin Name

```

PinName = PIN1

```

```
BusName=BUS1
```

These entries set the default name used for normal pins and bus pins, respectively. When a new pin is added to a symbol, the exact name given is tried first. If this name is already in use, a numeric part is added, or any existing numeric part is incremented until a unique name results. You can force names to always have leading zeros by specifying an initial name such as PIN001.

Symbol Gallery Location

```
SymbolGallery = Data Files\Symbol Gallery.clf
```

This statement specifies the location of the library file to be used as the "Symbol Gallery", i.e. the list of graphic items displayed while using the symbol editor. Only one symbol gallery file can be specified. The file must be a valid DesignWorks symbol library (.clf) file.

Section [Export]

Specifying Predefined Script Variables

```
ScriptVar = path, "d:\mydocs"
```

The ScriptVar keyword allows you to predefine variables that can be accessed from within an Export script. This can be used to set values such as the location of certain directories that may vary from user to user but will be fixed on that one computer. The format is:

```
ScriptVar = varName, value
```

varName is the variable name, consisting of any letters, numbers, underscore or period, up to 16 characters long. This variable can be referred to withing the script using the &varName format.

value is any text. If the value contains any non-alphanumeric characters, the value should be enclosed in double quotes.

Creating Custom Menus

See *"Creating Custom Menus"* on page 342.

Specifying the Location of Export Scripts

```
ScriptFolder = Export Scripts
```

This statement defines where to look for the scripts to display in the Export script list. The specified location is relative to the directory containing the DesignWorks application. This keyword can be specified multiple times to add multiple directories, if desired.

Section [System]

Tools Directory

```
ToolFolder = c:\dw\tools
```

This statement defines where to look for the external code modules. If this value is not specified then modules are loaded from the tools subdirectory under wherever the DesignWorks executable is located.

Default System Font

```
Font = "font_name" font_size [BOLD ITALIC]
```

This statement specifies the default font which the DesignWorks system will use when no other font has been specified. Certain program modules may by default display text using this default font. If no font is specified an attempt is made to use a Courier type face. If no font size is specified then 10 point is used.

“font_name” is the name of a TrueType font, only TrueType fonts are supported. “font_size” is the point size to use. There are two optional style keywords which may be applied, “BOLD” and “ITALIC”.

Example:

```
Font = "Courier New" 10 Bold
```

Printer Line Scaling

```
PrinterScaleLines = None, All, or OverOnePixel
```

This allows the user to specify whether lines are to be scaled when printed.

“None” is the default, it indicates that no scaling will occur. With this setting a line's width is printed with the same number of pixels as on the screen. When printing to high resolution printer, i.e.: ≥ 300 dots per inch, this will cause thick screen lines (busses) to be reproduced as printed lines which do not appear to be much thicker than thin screen lines (signals). This setting is most useful when printing to dot-matrix printers where the printer's resolution is similar to the screen's resolution.

“All” indicates that every line will be scaled so that its printed width appears the same as on the screen.

“OverOnePixel” indicates that lines that have a screen width of greater than 1 pixel will be scaled when printed. The result is signal lines being drawn very finely, but busses appearing as thick lines.

Printer Color Mapping

`PrinterColors = Yes or ToBlack`

This keyword controls translation of colors to black and white during printing. This is used in cases where the printer driver being used does not properly handle color requests. If `PrinterColors=ToBlack` is specified then all colors are translated to black for printing. If this keyword is missing or an invalid option has been specified then no translation of colors will occur.

Clipboard Color Mapping

`ClipboardColors = Yes or ToBlack`

This is used to control translation of colors to black and white when placing an image on the clipboard. If `ClipboardColors=ToBlack` is specified then all non-white colors are translated to black. If this keyword is missing or an invalid option has been specified then no translation of colors will occur.

Section [System Font Translations]

`Old_Font_Name = Replacement_Font_Name`

Font translations are used when the fonts imbedded in a file are not available

on the current platform. This section allows the user to define which fonts (available on the current platform) are to be used instead of the fonts specified. The replacement font must be a TrueType font.

Each line in this section specifies a font mapping. For example:

```
Bookman = Courier New
Times = Times New Roman
```

specifies that anytime the font Bookman is displayed then Courier New should be used as its replacement, and when Times is displayed or requested that Times New Roman should be used.

Section [Find]

Specifying Search Script Location

```
ScriptFolder = dir_name
```

This keyword specifies the location of search scripts that will be available to the user in the Find tool. All files with the “.RFM” extension that are located in the specified directory will be added to the Find tool’s list. This keyword can be specified multiple times (one per line) if desired, to add more directories.

Section [TextEdit]

Specifying Additional Text Document Types

You can specify additional file extensions to be handled by the built-in text editor. For example, if you frequently create a text report from your designs that has the file extension "rpx", you can add this extension to the list of handled file types. This means that this file type will appear in the New and Open commands, allowing easier access to those files.

NOTE: These additional file types are not registered with the Windows system. I.E.

you will not be able to open them in DesignWorks by double-clicking on a file in the Windows Explorer.

For example, the following line adds handling for report script files having the .RFM extension.

```
TextDoc = "\nReport Script \nReport Script\nReport  
Script Files (*.rfm)\n.rfm"
```

(All of the above text must appear on one line in the INI file)

Notes on this format:

The string consists of 5 sections that provide information about the file type. The sections are separated by a newline marker "\n".

The first section (before the first \n) specifies ###

The second section specifies the name prefix used when creating new files of the given type. A number is appended to this string to give the complete file name.

The third section provides the name that will be used to show the file type in the New document type list. If this is omitted, then it will not be possible to create this file type using the File/New command.

The fourth section provides the string that appears in the "Files of type" selection list in the Open file box.

The last section specifies the default file extension applied when a Save As is done on this file type.

Appendix E—Command Line Arguments

The DesignWorks Professional package can be started from a command line with arguments specifying which file to open and other actions. This can be useful, for example, to perform a sequence of actions from a batch file or to create a Windows shortcut which opens a specific file or generates a report.

Specifying File Names on the Command Line

A command line can contain multiple arguments, which are separated by blanks. For this reason, any file name which contains blank characters must be enclosed in double quotes. For example, the following line starts the program and opens the given file:

```
"C:\Program Files\DesignWorks Professional 4\
DesignWorks.exe" "C:\Documents\Control System.cct"
```

NOTE: The above must be entered as a single line.

Within the quotation marks, blanks are significant, i.e. if the file name contains two blanks in a row, exactly the same must be specified in the command line.

-exp (Export) Option

The -exp option can be used to invoke the Export command from the command line. The -exp must be followed immediately by the name of the export script file to use. For example:

```
"C:\Program Files\DesignWorks Professional 4\DesignWorks.exe"
"D:\My Schematics\System1.cct"
-exp "D:\My Scripts\Export Script.rfm"
```

As before, all of the above example must appear on a single line.

NOTE: Depending on the script invoked, the Export command may require user interaction, so you may wish to modify the script for batch use. For example, the \$CREATEREPORT \$PROMPT command will stop and require the user to confirm the location of the saved report. This can be modified by removing the \$PROMPT option to run silently.

-exit (Exit Immediately) Option

The -exit option causes the program to terminate after the preceding actions are completed. If this flag is not included, the program will stay active after the command line is executed. For example:

```
"C:\Program Files\DesignWorks Professional 4\Designworks.exe"  
"D:\My Schematics\System1.cct"  
-exp "D:\My Scripts\Export Script.rfm"  
-exit
```

-hide (Hide Window) Option

The -hide option causes the program to run without displaying its main window. This is intended, for example, for cases where you want to use the program to open a file, generate a report and quit without any visible action.

-nodoc (No Document) Option

The -nodoc option causes the program to start up without opening any document window or prompting with the usual options box.

- js (JavaScript) Option

This option causes the given JavaScript to be run. The -js must be immediately followed by a the script file name. The file name can be optionally followed by any number of arguments that will be passed to the script. Note the following:

The argument value cannot start with a hyphen "-" as this character indicates another command line option.

These arguments can be accessed within a JavaScript using the getScriptArg method.

For example:

```
"C:\Program Files\Designworks Professional 4\Design-  
works.exe"  
"D:\My Schematics\System1.cct"  
-js "D:\My Scripts\Export Script.dwj" "FORMAT1"
```

-bp (Browser Panel) Option

This option causes a custom panel to be displayed with the embedded browser navigated immediately to the source document specified immediately after the -bp.

See "Using Custom Panels" on page 350 for more information.

Appendix F—Installation

Installing on a Write Protected Server

For normal operation, DesignWorks Professional needs access to a writable directory to store state information and temporary files. This “root” directory is also the default location for searching for user documents, the “INI” settings file, and other files.

Specifying a Root Directory

In most single-user installations the root directory is the same as the location that the application itself is installed in. However, if the program is to be installed on a server and accessed by multiple users, it can be useful to specify a root directory that is different from the program location. This has two benefits:

- The server can be write-protected, ensuring that no single user can make changes that will affect other users.

- Each user can have their own program settings while sharing the program and data files.

Two features are built-in to DesignWorks to support this situation:

- When the program is run for the first time on a given system, it attempts to create a file in the directory containing the program. If this operation fails, it prompts the user to select a root directory. This should be a writable directory on a local hard disk.

- If at any time, the root directory needs to be changed, you can select the Help/Root Directory command to manually change the setting.

Files in Root Directory

There is no single file that has to be in the root directory, however the pro-

gram will create the settings file “dwstate.ini” in this location. This file contains current settings for controls and other program state information.

File Search Paths

Whenever the program attempts to locate a file, such as the INI file, or any file whose location has not been completely specified, it looks in the following directories, in this order:

- The root directory

- The application’s initial working directory

- The directory containing the main application file.

If the requested file is an executable program, it also looks in all directories specified in the system PATH environment variable.

There is more information on specifying file and directory names in the INI file given in “Specifying File and Folder (Directory) Names” on page 375.

Installing and Locating Symbol Library Files

DesignWorks comes with a variety of different symbol library files and many more can be downloaded from our web site. By default, only a small number of these are opened automatically at startup. You can open and close libraries manually or customize which libraries open automatically when the program is started, to suit your application.

DesignWorks symbol library files normally have the “.CLF” file name extension.

Location of Libraries

When DesignWorks is first installed, the libraries will be located in the Libs folder inside the DesignWorks program folder. Inside that folder may be other sub-folders containing additional families of libraries.

Opening Libraries Manually

To open library files manually, right click in the parts palette window, or move to the Libraries submenu of the File menu, and select the Open Lib command. This command allows you to locate a symbol library file in the usual way. When the file is opened, it will be added to the list of available libraries in the parts palette.

Opening Libraries Automatically at Startup

When DesignWorks is first installed, it is configured to open automatically any library file that it finds in the Libs folder inside the DesignWorks program folder. You can add more libraries to be opened automatically by simply placing them in this folder. Sub-folders are NOT scanned, so files that are not immediately needed can be “hidden” in nested folders.

If you wish to create a more sophisticated layout of folders containing libraries for your projects, you can use the LIBRARY and LIBRARYFOLDER keywords in the DesignWorks initialization file (DW.INI) to create additional library folders.

See “Specifying Libraries to Open at Startup” on page 385 for more information.

Appendix G—Technical Support

Our goal is to ensure that you get reliable and productive use out of your DesignWorks package. If you have any problems or questions, please contact us at:

Technical Support,
Capilano Computing,
2631 Viking Way, Unit 218,
Richmond, B. C.
Canada, V6V 3B5

Voice (604) 522-6200

Fax (604) 273-9397

email tech@capilano.com

WWW <http://www.capilano.com>

Internal Error Detection

DesignWorks makes use of a number of complex internal data structures in order to maintain an up-to-date image of your circuit at all times. To assist in detecting problems due to hardware failures, program errors or operating system errors, a code module has been added which checks these structures for consistency. This is done in the "background" while the program is idling and should normally be invisible to the user.

Should a problem be detected, a warning box will be displayed. The "State" value is a code that specifies the type of problem detected, and the "Address" value is the memory address of the object that was in error. It is beyond the scope of this manual to discuss the meaning of all possible error codes and an

estimate of their severity. In general, a State < 100 indicates a structural problem that is likely to cause a serious program malfunction if you proceed with editing. The warning box will only appear once if this type of error is detected, even if other errors occur later. The error detection mechanism is reset when all design files are closed. States > 100 indicate unexpected situations detected in connection with some specific function and may or may not be serious. If the error disappears after a later check (due to an offending object being deleted, for example) you will be notified. This may occur if you delete the corrupted part of the circuit or if some other internal check succeeds in correcting the problem.

If you see this box in the course of normal program operation, then save your design file immediately under a different name (so as not to wipe out your last good backup). Quitting the program and rereading the saved file may result in the problem being corrected. If you can isolate the problem down to one specific object, then try deleting that object and recreating it. In any case, please contact our Technical Support department and provide as much information as you can about the situation which created the problem. We will help you in any way we can to recover any lost data.

Error checking can be disabled using the NOERRORCHECK ini file option, described in “Internal Error Checking” on page 385. This should not be used under normal circumstances.

Index

Symbols

\$ report generator commands
357

\$ variables in text 107

& in report scripts 357

&Attribute variables
in text 108

.List fields 169, **173**

.Pt fields **175**, 315, 316

A

Add Bus Sigs button **199**

Add Pins command **283**, 293,
295

Align commands 112, 290

Allow Carriage Returns option
180

arrow keys 8, 71, 78, 79, 82

Associated Internal Signal op-
tion 101

Attach Subcircuit command
225, **232**

ATTRFIELD INI keyword 379

Attribute Probe command 10,
158, **169**, 181
tutorial 41

attributes **157**

Allow Carriage Returns op-
tion 180

default position fields **175**

default value **177**, 271, 286

defining 177

tutorial 39

defining in INI file 379

definition vs. instance **159**,
179, 217

Delete function 182

design 75, 108, 162, **162**

devices 84

Duplicate function 182

Group with Name option
179

in symbol editor 285

instance 179

Justification command 167

list menu 169, 173

maximum length 182

package type 315

pins 101, 163, 285

planning 114

power and ground 205

predefined fields 158, 361

primary 10, **158**, 169, 181

read only fields 181

Rotate Text With Object op-
tion 180

rotation 163

setting 160

tutorial 39

setting options 178

Show Field Name com-
mand 168

Show Field Name option
180

signals 94

symbol 267

temporary fields 159, 181

text style 165

value list fields 173

variables 108, 162, 165

Visible by Default option
180

Attributes command **160**, 170,
171

devices 84

pins 101, **163**

signals 94

Auto Create Symbol command
234, 293, **304**

tutorial 42

Auto Open Libs command **264**

Auto-assign package and unit
option 135

auto-creating
subcircuits 224
symbols 303

automatic page references **201**,
201, 366

auto-naming **121**
assignment order 137
choosing options 122
devices 121
prefix field 133

\$AUTONUMBER 102, 154

auto-packaging **121**
limitations 128
name format 131
options 123
prefix field 133

reenabling after manual ed-
its 128

\$NONAME option 133

auto-save 62, 382

AutoSave INI keyword 382

AutoSym.xxx attribute fields
361

B

Back Annotate command 138

back annotation 115, 120, 127,
129, **138**, 154, 348

and packaging 139

background grid 290, **338**, 378

Background Layer Only command 106, **109**, 111, 339
background objects **105**, 106, 339
 selecting 109
Backup INI keyword 381
backups **61**, 381
bidirectional pin 299, 374
bill of materials 119, 131, 152, **353**
bitmap pictures 67
border 118, 331, 333, **337**, 338
 importing 334
border objects **105**, 106, 111, 333
-bp command line option 395, 351, 395
Breakout command 194
breakout device 189, **193**, 214, 370
 custom 307
 INI file parameters 384
 tutorial 27
BREAKOUT INI keyword 384
Bring To Front command 290
Bring to Front command 112
Browser tool 56, 72, 103, 130, 220, 245, **248**
 tutorial 34
Bus Info command **191**
bus page connector 202, **202**, 214
 creating 311
Bus Pin Info command 197, **197**
Bus Pin Options command 362
bus pins **196**
 adding internal pins 198
 adding to symbol 284, **295**
 annotation 200
 text style 165
 changing connections 197
 connecting 189, **191**, 192
 definition 51, 374

 deleting internal pins 199
 on hierarchy blocks 223, **240**
 selecting internal pins 92
BusInfo attribute field 362
BUSNAME INI file keyword 387
busses **189**, 202
 adding signals to 192
 bus pins 295
 creating 191
 tutorial 27
 hierarchy 240
 Page Connector 311
 pin annotation 200
 Port Connector 240, 309
 tutorial 27

C

Category attribute field 362
CctName attribute field 362
CctPath attribute field 362
Center in Page command **73**, 105, 106, 111
change count 75
\$CHANGECOUNT 75
Circuit to Library command **274**, 313
\$CIRCUITNAME text variable 108
circuits
 definition 51
 opening 76
 read-only 75
 showing info 73
 subcircuits 215
 zooming 54, 73
Clear Change Count option 75
Clear command 73, 82
 attribute pop-up menu 97
 attributes 164
clipboard **65**, 66, 105

 color mapping 389
 in dialog boxes 4
 keyboard shortcuts 4
 tutorial 29
ClipboardColors INI keyword 389
Close command 62
Close Lib command **264**
Close Part 281
color
 clipboard color mapping 389
 in device symbols 289
 printer color mapping 389
 settings in INI file 378, 386
 signals 93
 symbol editor grid 386
Color command 93
Compact library option 268
compatibility
 design files 54
connectors
 packaging 152
 page connector 200
 power and ground 207, 208
 signal connector 207, 208
Copy command 4, **67**, 73, 81
 color mapping 389
 symbol editor 276
Create Subcircuit Block command 222, **226**
\$CREATEREPORT 357
cursor keys 8, 71, 78, 79
cursor usage 9
custom menus 342
custom panels **350**
Cut command **67**, 73
 symbol editor 276

D

databases
 exporting to 119

- date and time
 - circuit date stamp 362
 - device date stamp 363
 - library modified date 263, 365
 - symbol date stamping 313
 - text variables 107
- date stamping **154**, 362
 - disabling
 - NODATESTAMP INI keyword 382
- \$DATECREATED text variable 107
- \$DATEMODIFIED text variable 107
- \$DATENOW text variable 107
- DateStamp.Cct attribute field 362
- DateStamp.Dev attribute field 155, 363
- DateStamp.Last attribute field 363
- DateStamp.OS attribute field 363
- DateStamp.Symbol attribute field **313**, 363
- default border 338
- default font 388
- default names
 - auto-naming 121, **210**
- default prefix 124
- Define Attribute Fields command 131, 158, 174, **177**
- Delay.Dev attribute field 363
- Delay.Dev.Max attribute field 363
- Delay.Dev.Min attribute field 363
- Delay.Dev.Typ attribute field 363
- Delay.Pin attribute field 363
- Delay.Pin.Max attribute field 363
- Delay.Pin.Min attribute field 363
- Delay.Pin.Typ attribute field 363
- delete key 26, 82, 91
- Delete Page option 77
- Delete pins option 285
- Demote
 - library part 267
 - page 77
- Depth attribute field 363
- Description attribute field 317, 363
- Design Attributes command 134, **162**, 228
- Design Preferences command 92, 366
 - auto-naming 212
 - page reference options 201, **202**
 - print scope 229
 - Show Page Breaks 60
 - Show Printed Page Breaks 60
 - signal auto-naming 210, 211
 - text style 166, 340
- DESIGN setup file keyword 382
- design templates
 - creating 330
 - naming 332
 - vs. sheet templates 330
- Designer attribute field 363
- designs
 - auto-save 382
 - backups 61, 381
 - change count 75
 - closing 62
 - creating 52
 - disposing 62
 - opening 54
 - printing 59
 - references to libraries 263
 - saving 57
 - saving in DXF or WMF format 57
 - setting attributes 75
 - structure 50
 - templates 52
- DesignType attribute field 346, 364
- Detach Subcircuit command **233**
- DevEditor INI section 386
- Device Naming and Packaging
 - Options command **126**, 131, 364, 367
- Device Naming and Packaing
 - Options command tutorial 36
- device symbol editor tool 52, 78, 102, 153, 174, 206, 209, **276**
 - Auto Create Symbol command 304
 - creating a new part 276
 - hierarchical blocks 222
 - page connectors 204
 - setting pin numbers 102
 - setting primitive type **312**, 369
- devices
 - auto-naming 99, 121, **210**, 314
 - connectors 152
 - date stamping 154, 382
 - default name prefix 133
 - definition 51
 - deleting 82
 - discrete components 154
 - duplicating 81
 - finding 56
 - Flip commands 82
 - libraries 78, **262**
 - moving 82
 - names **84**, 171
 - naming 85
 - package type 87, 315
 - packaging **127**, 172, 219, 313
 - tutorial 35
 - packaging options 84, 134, 367
 - part type 87
 - pin type **373**
 - placing 77, 79

- primitive type 83, 312, **369**
- report options 356
- rotation 8, 71, **78**, 79, 82
- setting info 82
- Show Pin Numbers option 83
- symbol creation 276
 - tutorial 38
- token numbers 152
- DevPrefix attribute field 133, 364
- Directory INI keyword 377
- Discard Subcircuit command 234
- discrete components 154
 - name prefix 132, 315
 - package types 316
 - pin numbers 83, 101, 154
 - pin type 374
 - tutorial 21
 - Value attribute field 318
- Draw Bus command 9
- Draw Sig command 10
- Duplicate command **71**, 81
 - attributes **168**
- DXF file 57
 - creating 57

E

- Edit command 169
- Edit Part command 280
- Edit Symbol command **281**
- EMF clipboard data 67
- Enhanced Metafile format 67
- Enlarge command 55, 383
- Enter key 85
- environment variables 376
- error checking
 - internal 385, 401
 - using Export 119
 - using Find 246

- error codes
 - packaging 151
- error reports 119
- Examples INI keyword 379
- exit command line option 394
- exp command line option 393
- Export command 119, **348**, **353**
 - attribute fields in reports 358
 - creating scripts 345, 357
 - custom menus 342, 356
 - device reporting options 356
 - file formats 354
 - line terminators 360
 - pin numbering 154
 - signal reporting options 356
 - specifying script location 388
 - tutorial 48
- ExtCctDate attribute field 364
- ExtCctLib attribute field 364
- ExtCctName attribute field 364
- ExtCctOS attribute field 364
- ExtCctPath attribute field 364
- Extract Pin List button 307

F

- fanout 373
- file auto-save 382
- file backups 61, 381
- file compatibility 54
- file name variable 108
- \$FILENAME text variable 108
- \$FILEPATH text variable 108
- Fill Color command 289
- Fill Down command 252
- Fill Right command 252
- Find tool 56, 72, 172, 220, **245**
- Fit to Single Sheet When Printing option 118
- flat hierarchy mode 117, **218**
 - names 132, 171

- packaging 123, 145, 172
- Flip Horizontal command 82
 - symbol editor 290
- Flip Vertical command 82
 - symbol editor 290
- FOLDER INI keyword 385
- Folder Keyword 385
- font
 - attributes 165, 340
 - border 341
 - default settings 378, 386
 - pin numbers 340
 - planning 115
 - text blocks 106
 - translations 389
- FONT INI keyword 378, 386, 388
- front-to-back ordering 112
- Function attribute field 134, 364

G

- gate packaging 127
- Generate button 306
- Get Info command
 - attributes 84
 - busses 190
 - circuits 73
 - design 73
 - devices 230
 - packaging options 130
 - page **73**
 - page connectors 205, 209
 - selection 71
 - signals **93**
 - text 105, **106**
- Go To Selection command **73**
- graphic objects
 - front-to-back ordering 112
- graphics files 57
- graphics objects
 - pasting 66

- Grid 383
 - grid
 - background 334, 338, 383
 - reference 202
 - symbol editor 290, 386
- GRIDCOLOR INI keyword 386
- Grids command 290
- GRIDSIZE INI keyword 386
- ground and power connections
 - 205**, 208, 308
 - in hierarchy 236, 242
- Ground attribute field 317, 364
- Group command **111**, **290**
- Group with Name option 179

H

- hide command line option 394
- Hide command 97
- hide command line option 394
- hierarchy **215**
 - bottom up design 224
 - definition vs. instance 217
 - example 215
 - flat mode 117, **218**
 - mode 218
 - netlists 227
 - physical mode 172, **218**
 - planning 116
 - printing 229
 - pure mode 172, **218**
 - top down design 221
 - tutorial 43
- HierNameSep attribute field 228, 364
- HTML 346
 - in Custom Panel 350

I

- Ignore packaging option 130, **135**, 135, 367
- Import Sheet Info command 105, 106, 111, 334, **334**, 342
 - sheet templates 330
- In Primary List option 181

- Initial.Dev attribute field 364
- Initial.Pin attribute field 365
- Initial.Sig attribute field 365
- input pin 299
- instance data 179
- Instance fields 159
- InstName attribute field 127, 145, **170**, 172, 228, 364
 - packaging 136
- interfacing 119
- Invert.Pin attribute field 365

J

- JavaScript **346**
 - custom menus 342
 - in Custom Panel 350
 - invoking 348
- Join bus pin option 198
- Join Sequential bus pin option 198
- js command line option 395
- Justification command 167

K

- Keep with Instance option 179
- keyboard shortcuts 7

L

- Lib Maintenance command 265, 267
- LibDate attribute field 365
- LibName attribute field 365
- LibOS attribute field 365
- LibPath attribute field 365
- libraries 78, **262**
 - compaction 268
 - creating 78, **262**
 - introduction 78
 - maintenance 265
 - opening automatically 385
 - planning 116
 - shortcut 264
- Libraries sub-menu 262
- LIBRARY INI keyword 385

- LIBRARYFOLDER INI file
 - keyword 386
- LibType attribute field 365
- Line Color command 289
- line terminators 360
- Line Width command 289
- Line Width option 94
- Link to Pin command **294**
- Lock and Check packaging option 129, **135**, 135, 147, 367
- Lock and Don't Check packaging option **135**, 135, 147, 367
- Lock and Don't Check packaging option 129
- Lock Opening Subcircuit option 84, 221, 230, 303
- Lock packaging option 130
- logical name 172

M

- Magnify command 10, **55**
 - tutorial 26
- Make auto-assigned names visible option 124, 125, 126
- Make Invisible command 253
- Make Unique Type command 231, **269**
- Make Visible command 252
- MENU INI keyword 343
- menus
 - custom 342, 356
 - pop-up 8

N

- Name attribute field 127, **170**, 228, 365
 - auto-naming 210
 - default position 315
 - in hierarchy 172
 - packaging 136
 - signals 94, 98
- Name command 86

- devices 86
 - signals 96, 97
- Name.Prefix attribute field 131, **133**, 133, 146, 315, 365
- Name.Pt attribute field **175**, 315, 365
- Name.Spice attribute field 133, 315, 365
- names **170**
 - applying multiple 98
 - auto-naming 99
 - default prefix 124, 125, 126
 - devices **84**, 171
 - editing 97
 - hierachical names
 - separator char 228
 - hierarchical names 228
 - invisible 95, 96, 171, 208
 - moving 98
 - Name vs. InstName 136, **170**, 172
 - pin names 293
 - removing 86, 97
 - repositioning 86
 - restrictions 120
 - signals **94**
- netlists 119, 152, 212, **353**
 - flattened 227
 - hierarchical 227
 - omitting signals 356
 - pin order 299
 - tutorial 24
- New Breakout command 190, 191, **194**, 199
 - tutorial 27
- New command 52
 - design templates 330
 - example files 379
 - tutorial 43
- New Lib command **262**
 - tutorial 37
- New Page option 76

- New Port Connector command
 - bus ports 241
 - signal ports 239
- NODATESTAMP INI file key-word 155
- nodoc command line option 394
- NOERRORCHECK INI key-word 385, 402
- NOLOOSEENDS INI keyword 384
- \$NONAME prefix option 133
- Normal Size command 55
- nudge 8, 82
- \$NUNPAGES text variable 107

O

- OKErrors attribute field 365
- Omit from Report option
 - devices 132
 - signals 94
- open collector pin 299, 374
- Open Design command 54
- open emitter pin 299, 374
- Open Lib command 78, **264**, 264
- Open Page option 76
- Open Subcircuit command **303**
- Option key 72, 78, 99, 103
- Orientation command **79**
 - Paste command 71
- output pin 299

P

- Package attribute field **87**, 315, 316, 366
- Package.List attribute field 316, 366
- packaging **127**, 172, 313, 314
 - creating a symbol 144
 - default name prefix 124, 125, 126
 - error codes 151

- flat mode 123
- manual 130
- name format 131
- options 123
- physical mode 123
- PkgLevel attribute field 135
- prefix field 133
- pure mode 123
- required attributes 145
- tutorial 35
 - vs. auto-naming 121
- Packaging Options 84, 134
- packaging options 367
- page
 - opening 76
 - page title 77
 - printer page setup 60
- Page Connector device 51, 172, **200**, 201, 205, 311
 - connectivity rules 214
 - page reference format 202
 - tutorial 28
- page references 201, 366
 - format 202
- \$PAGENUM text variable 107
- PageRef attribute field 366
- PageRefFormat attribute field 366
- PageRefWidth attribute field 366
- pages
 - adding 342
 - tutorial 29
 - creating a new page 76
 - deleting 77
 - multipage templates 342
 - opening 56, 76
 - page number variables 107
 - PAGETITLE variable 77, 107
 - references 201
 - reordering 77
 - title block 339

- Pages command 7, 56, **76**, 105, 107, 342
 - tutorial 29
- \$PAGETITLE text variable 77, 107
- paper size 51, 60
- Part attribute field 83, **87**, 146, 173, 316, 366
- Part Attributes command 174, 285, **285**
 - tutorial 39
- Part.List attribute field 87, 316, 366
- Part.Pt attribute field **175**, 316, 366
- Parts palette 77, **79**
- Parts Palette command 81
- Paste command **67**, 81
 - auto-connection 70
 - rotation 8, 71, 79
 - symbol editor 276
 - text 51, 105
 - title blocks 339
- Paste Special command 66, **68**
- PDF file 57
 - creating 59
- Permutable attribute field 366
- physical hierarchy mode
 - netlists 227
- physical hierarchy mode **218**
 - back annotation 138
 - InstName attribute field 170, 228
 - Keep With Instance option 179
 - names **172**, 228
 - packaging 123, 124, 136, 145
 - printing 229
- physical name 172
- pictures 52
 - background 105, 339
 - border items 333
 - Get Info 66
 - pasting 66
- Pin Attributes command 285, **286**
- pin function
 - specifying in Add Pins 284
- Pin Info command 100
- Pin Info option 84
- Pin List command **91**, 94, 205, 210
 - tutorial 20
- pin name
 - displaying in symbol 277
- pin names
 - default in IN file 387
 - displaying on symbol 38, 292
 - editing 295, **298**
 - when adding from Symbol Gallery 295, 297
- pin numbers **102**
 - auto-incrementing 99, 103
 - default 298
 - default pin numbers 102
 - editing 102
 - invisible 101
 - position 384
 - rotation 104
 - Show Pin Numbers option 83
 - text style 104
 - visible 101
- Pin Ordinal Number option 101
- pin spacing
 - breakouts 196, 383
- pin type 299, **373**
- PINNAME INI file keyword 387
- pins 100
 - adding to symbol 277, 284, **292**
 - attributes 101, 163, 285
 - bus internal 92, 191, 197, 284
 - bus pins 51, 92, 165, 189, 192, **196**, 197, 223, 240, 277, 284, 295, 374
 - definition 51
 - function 284
 - getting info 84
 - input 374
 - order in netlists 299
 - output 374
 - pin number **102**, 298
 - pin type 237, 299, 373, **373**
 - placing on symbol 292
 - reordering 299
 - selecting 93
 - type 101
- PinSequence attribute field 366
- PINSNAPSIZI INI keyword 386
- PINSPACE INI keyword 383
- PINSPACE setup file keyword 383
- PINTEXT INI keyword 384
- PkgLevel attribute field 135, 147, 367
- PkgPrefix attribute field 146, 151, 367
- Place Subcircuit command 231
- plotting 51, 57, 118
- Point command 9
- Pop Up command **220**
- pop-up menus 8
 - devices 87
 - Parts palette 77
 - value lists 162
- port connectors 232, 234, 236, **237**, 300
 - bus pins 240
 - creating 308
 - name 172
 - name matching 239
 - pin type 237
 - port pin type 237
 - tutorial 44
- power and ground 116, **205**, 208
 - in hierarchy 236, 242
 - tutorial 18, 22
- Power attribute field 317, 367
- power net 308
- predefined fields 361

prefix field 131, 133, 211
 prefixes
 default name prefix 124, 125, 126
 Name.Prefix attribute field 146
 PrefixField attribute field 132, 133, **134**, 151, 367
 primary fields **158**, 181
 primitive type 83, 312, **369**
 Print command 59
 page numbering 108
 Print Setup command 60
 Printer Scaling 389
 PrinterColors INI keyword 389
 PrinterScaleLines INI keyword 389
 printing **59**, 59
 bus lines 389
 color mapping 389
 document standards 114
 fitting to a single sheet 60
 in hierarchy 229
 line widths 389
 paper size 51
 planning **118**
 sheet sizes 118, 335
 specifying a page range 59
 \$PRINTNUMPAGES text variable 108
 \$PRINTPAGENUMBER text variable 108, 230
 Promote
 library part 267
 page 77
 Properties command
 arcs 289
 attributes 160, 163
 border setting 334
 breakout pins 196
 busses 191, **191**, 192
 devices **82**, 84, 87, 134, 356,

 367
 tutorial 21
 displaying pin name 277
 displaying pin names 38
 graphic objects 334, 339
 locking subcircuits 221
 packaging options 36, 137, 147, 367
 pictures 66, 105
 pins 100, 103, 154, 163
 report options 356
 signal connectors 209
 signals 93, 96
 tutorial 28
 symbols 38, 267
 text 341
 text items 106
 pseudo-devices 51, 92, 366
 pure hierarchy mode 117, **218**
 names **172**
 netlists 227
 packaging 123, 136, **145**
 restrictions 117
 Push Into command **220**
 autocreating subcircuit 224

R

Read Only option 75
 Reassign Device Names command **132**
 Redo command **65**
 Reduce command 55, 383
 Reduce to Fit command 55, 56
 tutorial 25, 26
 registry variables 376
 Repackage Design command
 127, 129
 tutorial 36
 report generation 119, **353**
 attribute fields 358
 line terminators 360

Rescan Design command 69, 128, **129**, 130, 139
 Restrict attribute field 317, **367**
 locking subcircuits 221
 restricted opening 230
 Revert command 57
 Revision attribute field 367
 Rotate Left command 82, **163**
 symbol editor 290
 Rotate Right command 82
 attributes **163**
 symbol editor 290
 Rotate Text With Object option 180
 rotation 71
 attribute text 180
 devices 79, 82
 Paste command 71
 pin numbers 104
 symbol text 289

S

Save As command 57, 282
 Save command 281
 timed auto-save 62
 Save Design As command 57
 Save Design command 57, 61
 Save to Lib command **272**, 313
 SCRIPTFOLDER INI keyword 388
 SCRIPTVAR INI file keyword
 Export tool
 predefined variables 387
 secondary fields **158**
 Select All command 72, 73
 selecting objects 71, **87**
 multi-page selections 72
 Send to Back command 112, **290**
 Set Design Attributes command 160

- ul style="list-style-type: none;">
- sheet 51
 - border 330, 337
 - size 60, 118, 335
 - tutorial 31
- standard sizes 382
- Sheet Size Wizard command 60, **335**
 - tutorial 31
- sheet templates
 - borders 333
 - multipage 342
 - sheet sizes 333
 - vs. design template 330
- SHEETSIZE INI keyword 382
- shift key 72, 82, 87, 99, 103
- symbol editor 298
- shortcut
 - libraries 264
- shortcuts
 - Cut/Copy/Paste 4
 - keyboard 7
- Show Bus Pin Annotation option 200
- Show Custom Panel command **350**
- Show Field Name command **168**, 180
- Show Field Name option 180
- Show Pin Numbers option 83
- Show Printed Page Breaks option 229
- signal connector device 95, 207, **208**
 - creating 308
 - tracing connections 209
- Signal Info command 200
- Signal Naming Options command 367
- signals **89**
 - attributes 94
 - auto-naming 210
 - checking connections 91
 - color 93, 378
 - connecting 70
 - connecting by name **95**, 171, 172, 208, 213
 - connections on Paste 70
 - creating **89**
 - definition 51
 - editing 91
 - finding 56
 - Get Info command 192
 - in busses 192
 - interconnecting 89
 - line width 94
 - loose end markers 384
 - Name command 97
 - names **94**
 - invisible 95
 - naming 95, 96
 - Omit from Report option 94, 356
 - page connectors 311
 - pin list 94
 - Pin List command **91**
 - Properties command 93
 - removing 91
 - selecting 93
 - sequential naming 99
 - token numbers **212**
- SigPrefix attribute field 367
- \$SIGSOURCE
 - in hierarchy 243
- Sim.InputMap attribute field 367
- SNAPSIZE INI keyword 386
- SOLIDGRID INI keyword 383
- SPICE 121, 365, 366
 - tutorial 45
- Spice attribute field **46**, 368
- spreadsheets
 - Browser tool 245
 - exporting to 119
- SUBCCT primitive type 220, **223**
- Subcircuit & Part Type command 234, **300**, 303, 312
- subcircuits 215
 - example 215
 - internal **230**, 231
- locking 83, 84
- primitive type 369
- symbol editor 300
 - tutorial 43
- symbol editor
 - default font 386
- symbol editor tool **276**
 - grid color 386
 - tutorial 37
- symbol gallery **296**
 - file location 387
 - specifying in INI file 387
- SYMBOLGALLERY INI file
 - keyword 387
- symbols
 - creating 37, **276**
 - displaying information 267
 - gate packaging 144
 - hierarchical block 215
 - name prefix 133
 - rotation 8, 71, 78, **82**
 - standards 116
 - tutorial 37
 - using 77
- ## T
- templates 330
 - creating 330
 - example files 379
 - New Design command 52
 - specifying location 379
 - Templates INI keyword 379
 - TestVectors.Cct attribute field 368
 - TestVectors.Dev attribute field 368
 - text **104**
 - background **105**, 106
 - border **105**, 107
 - border items 333
 - document types 390
 - editing 104
 - pasting 66
 - selecting 109

- style
 - attributes 50, **165**, 167, 340
 - border 334, 341
 - pin numbers 104, 340
 - text blocks 33, **106**, 341
- tutorial 32
- variables 107
- Text command 10, 85, 95
- TEXTDOC INI keyword 390
- \$TIMECREATED text variable 107
- \$TIMEMODIFIED text variable 107
- \$TIMENOW text variable 107
- title blocks 51, 107, 276, **339**, 382
- token numbers
 - devices **152**
 - signals **212**
- tool palette
 - magnifying glass 55
 - signal tool 90
 - zap tool 91
- toolbar 8
- TOOLFOLDER INI keyword 388
- transcript files 346
- tristate pin 299
- type name 78, 83, 92, **171**, 209, 262, 304

U

- Undo command **65**, 384
 - tutorial 15
- UNDO INI keyword 384
- Ungroup command **290**
- Unit attribute field 127, 130, 131, **145**, 169, 173, 368
- Unit sub-menu 169
- Unit.All attribute field 145, 368
- Unit.List attribute field 146, 368
- Unlink Name command **294**
- UnusedPins attribute field 318, 368
- Update from Lib command 269, 365
- Use Default Value button 177
- Use Page Setup option 118

V

Value attribute field 318, 368

tutorial 22

value list fields 162, 169, **173**, 174

variables

in text **107**

scripts 357

Visible by Default option 131, 180

VisPin.List attribute field 146, 368

W

Window menu 7

windows

circuit 52

opening 56

scrolling 54

zooming 54

schematic 51

WMF file

creating 57

Z

Zap command **10**, 66, 82, 86, **91**, 97

zoom

Go To Selection command 73

Magnify command 10

magnifying glass tool 55