

PADS[®] I/O Designer[™] for FPGA User Guide

Release PADS VX.2.6 and later

**© 2004-2019 Mentor Graphics Corporation
All rights reserved.**

This document contains information that is proprietary to Mentor Graphics Corporation. The original recipient of this document may duplicate this document in whole or in part for internal business purposes only, provided that this entire notice appears in all copies. In duplicating any part of this document, the recipient agrees to make every reasonable effort to prevent the unauthorized use and distribution of the proprietary information.

This document is for information and instruction purposes. Mentor Graphics reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Mentor Graphics to determine whether any changes have been made.

The terms and conditions governing the sale and licensing of Mentor Graphics products are set forth in written agreements between Mentor Graphics and its customers. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Mentor Graphics whatsoever.

MENTOR GRAPHICS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MENTOR GRAPHICS SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF MENTOR GRAPHICS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

U.S. GOVERNMENT LICENSE RIGHTS: The software and documentation were developed entirely at private expense and are commercial computer software and commercial computer software documentation within the meaning of the applicable acquisition regulations. Accordingly, pursuant to FAR 48 CFR 12.212 and DFARS 48 CFR 227.7202, use, duplication and disclosure by or for the U.S. Government or a U.S. Government subcontractor is subject solely to the terms and conditions set forth in the license agreement provided with the software, except for provisions which are contrary to applicable mandatory federal laws.

TRADEMARKS: The trademarks, logos and service marks ("Marks") used herein are the property of Mentor Graphics Corporation or other parties. No one is permitted to use these Marks without the prior written consent of Mentor Graphics or the owner of the Mark, as applicable. The use herein of a third-party Mark is not an attempt to indicate Mentor Graphics as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A current list of Mentor Graphics' trademarks may be viewed at: mentor.com/trademarks.

The registered trademark Linux[®] is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777
Telephone: 503.685.7000
Toll-Free Telephone: 800.592.2210
Website: mentor.com
Support Center: support.mentor.com

Send Feedback on Documentation: support.mentor.com/doc_feedback_form

Table of Contents

Chapter 1

Introduction to PADS I/O Designer	11
PADS I/O Designer Capabilities	11
PADS I/O Designer Integration	12

Chapter 2

Licensing and Configuration	15
License Options	15
Environment Variables	15
User Definable Variables	15
Configuration File (IODESIGNER.XML)	16
Starting PADS I/O Designer	16
Starting PADS I/O Designer from Windows	16
Selecting a Scheme	17

Chapter 3

PADS I/O Designer Workspace	19
Workspace Windows	19
Customizing Your Workspace	19
Selecting Objects	20
Signals List	20
Pins List	22
Symbol Window	25
Symbol Window Settings	27
TextEditor Window	27
Device Window	27
Pin Display	29
Bank Display	30
Clock Display	31
DQ DQS Pins Display	32
DCI Cascade Region Display	32
Show Layout in Device View	32
Scrolling and Zooming in the Device Window	33
Device Window Modes	34
Console Window	35
Shortcuts in the Console Window	36
Timings Window	36
Properties Window	40
Signal Properties	42
Pin Properties	43
Pin Properties for Layout Database	44
Port Properties	45

Symbol Properties	48
Graphic Properties	48
Text Properties	48
Synchronization Wizard	49
Synchronizing Files	49
Synchronization Check	50
GUI Startup Preferences	50
Errors and Warnings	50
Customizing Files and Directories	51
Changing the Appearance of Elements	51
Source Control Settings	51
Advanced Settings	52
 Chapter 4	
Typical Design Flows	53
Supported Schematic Design Creation Tools	53
PADS Designer Flows	53
 Chapter 5	
Working with Projects and Databases	57
What is a Project?	57
Project Window	57
Opening a Project	58
Creating a New Project	58
Adding a Board to a Project	59
Adding Databases to a Board	60
Viewing Settings	61
 Chapter 6	
Adding an FPGA to the Design	65
FPGA Database Wizard	65
FPGA Device Setup	67
Selecting the Device	67
Importing FPGA Vendor Files	67
FPGA Device Library	68
Vendor Library Updates	68
FPGA Library Update Notifications	68
Preliminary Device Support	69
Design Rule Check	69
Defining I/O Signals	69
Importing I/O Signals Using HDL	70
Importing I/O Signals Using a Spreadsheet	71
Creating and Editing Signals in the Signals List	72
Design for Multiple Devices in a Common Package	77
PCB Design Wizard	78

Table of Contents

Chapter 7	
User-defined Rules	79
Creating a New Rule	79
Chapter 8	
Making Pin Assignments	83
Assigning Signals to Pins	83
Mark to Assign	83
Assigning Signals to Pins Using Drag and Drop	84
Assign Mode	84
Changing Assignments in Lists	84
Drag & Drop	84
Choosing Signals/Pins to Assign from a List	85
Combining Pins into a Bus Pin	85
Pin Swapping	85
Pin and Signal Types	86
Clock Assignments	88
Differential Pins Support	89
VREFs.	89
Multi-Gigabit Transceiver Pins.	90
Special Signal/Pin Assignments	91
Chapter 9	
Creating, Editing and Updating Symbols and Schematics	95
Symbol Generation	95
Symbols Generator	96
Power Symbol Generation	100
Building a New Symbol	104
Adding an Image to a Symbol	107
Editing Symbol Elements	107
Moving and Changing Elements	108
Changing Symbol Backgrounds	108
Printing Symbols.	108
Deleting Symbols	108
Symbol Editor Settings	109
Creating Generic Symbols for Use With PADS I/O Designer	112
Importing Symbols from Independent Libraries	112
Exporting Symbols and Schematics.	112
Hierarchical Schematic Generation	113
Flat Schematic Generation	114
Placing Symbols in an PADS Designer Schematic	115
Importing Symbols and Schematics.	117
Schematic Update	118
Additional Information	119
Chapter 10	
Optimizing the I/O Assignments	121
I/O Optimization Methods	121

Fan-out Visibility	121
Layout Database.	122
I/O Optimization	122
Synchronizing the Layout Database with a Design	122
Layout Window	124
Connectivity List Window	125
Adjusting Component Orientation	125
Unravel Nets	126
Layout Scenarios Window	129
Creating and Applying a Layout Scenario	129
Multi-chip PCB Optimization.	130
 Chapter 11	
Data Exchange.	133
HDL and EDIF/XML Files	133
Recognizing Differential Signals	134
Synthesis Constraints Files	134
Place and Route Tools	134
Place and Route Constraints Files.	135
Pin Report Files	136
FPGA Xchange Files	136
FPGA Device Library Path.	137
Timing Report Files	137
Design Rule Check for Altera Devices.	137
SSO Check	137
Generating Device List	138
Exporting Symbols to a Library	138
Integration with PADS Designer	138
Importing an Existing PADS Designer Project	139
Importing Symbols from PADS Designer	139
Exporting Symbols/Schematics to PADS Designer	140
Updating Symbols/Schematics from PADS Designer	142
PADS Designer Usage Scenarios	142
Using the Interconnectivity Table Editor	146
Integration with Pads Layout.	147
Constraints Bus Members Generation	149
 Chapter 12	
FPGA Updater	151
FPGA Updater Introduction.	151
Manually Launching FPGA Updater	151
Automatically Launching FPGA Updater	151
Configuration	152
Log file.	153
Update Process.	155
Checking for Updates	155
Critical Release Update.	158
Downloading Updates.	158

Table of Contents

Backup of Current FPGA Library.	159
Installation.	160
Summary Page	162
FPGA Updater Log Viewer.	163
Backup and Rollback of FPGA Library.	163
Updating FPGA Libraries in a High Security Design Environment	166
 Chapter 13	
PADS I/O Designer Tool Reference	169
Status Bar	169
Pop-up Menus	169
Undo and Redo	169
Navigating List Windows	170
Operations on Columns.	170
Row Selection.	170
Sorting Rows.	170
Filtering Rows.	171
Filter Rules	172
Changing the Pin Name Font Size	176
Keyboard Shortcuts	177
Global Shortcuts	178
Shortcuts in Signal and Pins Lists.	179
Shortcuts in the Symbol Window	180
Shortcuts in the Device Window	180
Managing Keyboard Shortcuts	181
Modifying Shortcuts	181
Active Keyboard Shortcuts View	183
Toolbars	184
Main Toolbar	184
View Toolbar	185
Mode Toolbar	186
Layout View Toolbar	186
Unravel Toolbar	187
Zoom Toolbar	187
Symbol Toolbar	188
Source Control Toolbar.	189
 Chapter 14	
TCL Interface	191
Creating and Running TCL Scripts	191
PADS I/O Designer TCL Commands	191
PADS I/O Designer Defined Scalar TCL Variables	217
PADS I/O Designer Defined TCL Array Variables	219
 Chapter 15	
Design Versioning	221
Source Control Configuration	221
Source Control Usage	221

Getting Existing Databases	222
Adding Databases to the Repository	222
Additional Files in the Repository	222
Getting Latest Database Version	222
Checking Out the Database	222
Checking In the Database	222
Undo Check Out	223
Setting Labels	223
Browse Changes History	223
Showing Differences	223
Refreshing Source Control Status	223
Chapter 16	
Dialog Boxes and Field Reference	225
Settings Dialog Box	227
Database Properties Dialog Box	228
Layout Database Properties Dialog Box	231
Add Signal Dialog Box	232
Edit Primitive Value(s) Dialog Box	233
Import Design Wizard Dialog Box	234
xDX IOD Components Dialog Box	236
Library Components Dialog Box	237
Rule Editor Dialog Box	238
Rules Wizard	239
Unravel Nets Dialog Box	241
External Tools Dialog	242
Appendix A	
Preferred Devices List	245
Appendix B	
Vendor Support Information	247
Appendix C	
Device-dependant Assignment Rules	251
Actel Assignment Rules	251
Altera Assignment Rules	251
I/O Pin Placement with Respect to LVDS I/O Pins	252
Xilinx Assignment Rules	254
Lattice Assignment Rules	254

Table of Contents

Appendix D
Generic IO Standards..... 257

Index

Third-Party Information

End-User License Agreement
with EDA Software Supplemental Terms

Chapter 1

Introduction to PADS I/O Designer

PADS I/O Designer Capabilities

- **Broad FPGA Vendor Device Support**

Users will benefit from the broad device support from the leading FPGA vendors (Actel, Altera, Lattice, Xilinx) that PADS I/O Designer offers. This device support is kept up-to-date through periodic Library Updates.

- **Rules Engine**

An automatic Rules Engine supports standard rules and enable users to customize assignment rules.

- **Correct by Construction I/O Assignment**

PADS I/O Designer allows you to assign and optimize I/O assignments with confidence that they are correct. PADS I/O Designer does not require you to verify pin assignments made in PADS I/O Designer.

- **Automation of Error Prone Manual Processes**

Manual symbol and schematic creation for high pin count devices can be time consuming and error prone. PADS I/O Designer automatically generates symbols and schematics to efficiently incorporate the FPGA board design into the PCB process.

- **Improved Quality of Results**

PADS I/O Designer will improve overall PCB quality by optimizing the I/O assignments based on actual PCB component orientation.

A major benefit of system I/O assignment optimization is the reduction of PCB routing layers, through counts and trace lengths. Faster and easier PCB routing and overall improved PCB quality implies lower fabrication costs and better signal integrity and timing margins.

Note

PADS I/O Designer does not support multiple instances of the same PADS I/O Designer component.

PADS I/O Designer Integration

PADS I/O Designer generates Place & Route constraints, based on the HDL design and mapping process and then allows you to create the necessary symbols, schematics and hierarchical associations based on the “post-route” pin data.

PADS I/O Designer allows you to:

- Read signals from entities in VHDL files, and modules in Verilog files.
- Define signals within the PADS I/O Designer environment, and generate VHDL entities, or Verilog modules.
- Import/export CSV files.
- Choose an FPGA device from devices supported by vendors such as Xilinx, Altera, Lattice or Actel.
- Map HDL signals to FPGA pins, and generate FPGA constraints.
- Update mappings based on the files generated by Place & Route software.
- Create and edit functional-level, and board-level symbols in the advanced built-in Symbol Editor.
- Generate symbols and schematics.
- Work in teams through the built-in version control systems interface.
- Optimize I/O pin assignments improving efficiency on the PCB and drastically reducing design time.

PADS I/O Designer does not run in standalone mode. It is used in conjunction with other design solutions such as PADS Designer.

Settings Consolidation with PADS Designer

When you are using PADS I/O Designer with PADS Designer, PADS I/O Designer derives settings from both PADS I/O Designer and PADS Designer, as follows:

- PADS I/O Designer reads grid and object styles from DxDesigner.xml, and does not change them.
- PADS I/O Designer reads PADS I/O Designer-only settings from several PADS I/O Designer files, in the following order:
 - iod.xml - in the installation directory
/<mgc_home>/<release>/SDD_HOME/IODesigner/resources/iod.xml.
 - IODesigner.xml - in the corporate working directory (WDIR), if the file exists.
 - IODesigner.xml - in the user working directory (WDIR).

PADS I/O Designer saves settings only to the IODesigner.xml file in the user working directory.

PADS I/O Designer Design Process

PADS I/O Designer takes its initial settings from a default set in the iod.xml file in the installation directory. PADS I/O Designer then reads in your company's more specific settings, and where there are differences, the last read settings win. PADS I/O Designer then reads the user's IODesigner.xml file for environment specific settings. Finally, PADS I/O Designer reads your schematic design tool's .xml files to ensure PADS I/O Designer output has a matching look and feel for the schematic environment. See the reference guide for your schematic design tool for more detail about schematic settings.

The initial FPGA signals and/or assignments are typically imported by PADS I/O Designer from the FPGA vendor tools (for example, Actel, Altera, Lattice, Xilinx). After the PCB process integration and I/O optimization, the new signal assignments are exported to the respective FPGA vendor tool. PADS I/O Designer is knowledgeable of the FPGA vendor specific file formats making data movement very easy.

PADS I/O Designer starts by importing a signal list in the form of an HDL file or an FPGA vendor netlist. The signals may or may not be assigned at this point. I/O assignment can be easily done in a correct by construction fashion within PADS I/O Designer. PADS I/O Designer provides a device library that incorporates most of the vendor-specific pin assignment rules. The benefit comes from having the ability to assign and optimize pin assignments in the PCB design process.

Once an initial pin assignment is made, PADS I/O Designer is used to generate a symbol set for the FPGA. Symbols can be fractured based on a number of different parameters. The symbol set along with the schematic can be exported directly to the schematic tool. The benefit comes from automating two manually intensive activities - symbol and schematic creation.

After a preliminary PCB layout is complete, PADS I/O Designer can import the layout to begin the I/O optimization phase. You can view the actual component orientation and netlines as it appears in the layout tool. The goal is to use PADS I/O Designer to optimize the pin assignment for this particular component orientation. This is done by moving pin assignments to shorten netlines and remove cross-overs. Once you feel that this is the best I/O assignment for this component orientation, PADS I/O Designer will update the symbols and schematics to reflect the new pin assignments. The quality of the PCB is dramatically improved based on the optimized I/O assignment in the form of fewer layers, shorter traces, less vias and as a derivative improved signal integrity. This also lowers PCB costs.

Chapter 2

Licensing and Configuration

License Options

You must have an PADS I/O Designer for FPGA license to open a project in PADS I/O Designer. Other license options include:

- **Multi-Chip PCB Optimization Option**

Allows you to simultaneously optimize multiple devices. Without a multi-chip license, you can work with only one device at a time when applying optimization scenarios in a [Layout Database](#). For more information, see “[Multi-chip PCB Optimization](#)” on page 130.

- **PADS PCB View Option.**

Allows you to load a PADS layout in PADS I/O Designer.

- **PADS PCB Optimization Option.**

Allows you to create a layout database in PADS I/O Designer.

When you perform an action that requires a license for which you have permission, PADS I/O Designer automatically requests the license and holds it until you close PADS I/O Designer. You can also select **Setup > Licensing Options** to manually request or release product license options at any time during your session.

Environment Variables

On Windows platforms, all environment variables necessary to successfully run PADS I/O Designer are set automatically during the installation process.

User Definable Variables

The following variables are automatically set to default locations (determined from, for example, \$MGC_HOME) upon installation of the tool. You can choose to manually set these variables to point to a different location. When PADS I/O Designer is invoked and these variables are set, then the tool replaces the background default values with the values that you specified.

To verify what variables are set, click **Setup > Settings + Paths**.

If some of these variables do not work, it is possible that you have some paths manually set, so check your configuration file (IODESIGNER.XML) for any preference settings. See “[Configuration File \(IODESIGNER.XML\)](#)” on page 16 for more details.

To set these variables in Windows, use **Control Panel > System > Advanced Tab > Environment Variables**.

Table 2-1. User Definable Environment Variables

Variable	Description
IOD_PATH_FILE_VD_DX	DxDesigner executable file
IOD_PATH_FILE_CONSOLE_LOG	Console log file
IOD_LIBRARY_PATH	FPGA libraries directory

Configuration File (IODESIGNER.XML)

The IODESIGNER.XML configuration file contains preference settings (for example, paths, color schemes, shortcuts) for PADS I/O Designer. You can edit this file, found in your WDIR directory, with any text editor. PADS I/O Designer saves configuration settings automatically into this file on exit.

Starting PADS I/O Designer

How you start the PADS I/O Designer tool depends on your operating system. The following sections describe the methods for each supported operating system.

Starting PADS I/O Designer from Windows

The PADS I/O Designer start method varies with the version of Windows you are running. The following table shows the start methods for the different Windows versions.

Table 2-2. PADS I/O Designer Windows Start Methods

Windows 7 and earlier	Start > All Programs > <release> > Design Entry > PADS I/O Designer <release>.
Windows 8	From the Windows 8 Apps view, under <release>, click PADS I/O Designer <release>.

To invoke PADS I/O Designer, select **Start > All Programs > <release> > Design Entry > PADS Designer <release>.**

When PADS I/O Designer displays the Startup dialog, you can choose to do one of the following:

- Create a new empty project (see “[Creating a New Project](#)” on page 58).
- Open an existing project (see “[Opening a Project](#)” on page 58).
- Open a recently used project.

You can set preferences for GUI (graphical user interface) startup, see [GUI Startup Preferences](#) for more information.

Caution



You may see the following warning message while loading databases created with versions of PADS I/O Designer prior to IOD7.1:

```
# Some signals cannot be matched with nets on the layout:
# signal_name1, signal_name2, signal_name3 . . .
# Flow synchronization may be needed.
```

If you do see this message, you should synchronize your flow using the [Synchronization Wizard](#).

Selecting a Scheme

To choose a scheme, click **Setup > Settings** and select a scheme from the **Scheme** drop-down box. The scheme you choose should match the workflow you are using.

To create a new scheme, click **New**. PADS I/O Designer makes a copy of the currently-selected scheme. Specify a name for the new scheme and click **OK** to save it. Only user-defined schemes can be renamed. If a standard scheme is selected, the **Save Scheme** checkbox is unchecked and disabled. If a user-defined scheme is selected, this checkbox is enabled and checked by default. If checked when you click **OK**, the scheme is saved in addition to any other changes made in the dialog.

Chapter 3

PADS I/O Designer Workspace

Workspace Windows

The PADS I/O Designer workspace is made up of a number of windows which are used to display and manipulate information. The windows available depend upon the type of database you are currently viewing.

Common windows (available in all database types):

- [Pins List](#)
- [Project Window](#)
- [Console Window](#)
- [Properties Window](#)

FPGA database windows:

- [Timings Window](#)
- [Device Window](#)
- [Signals List](#)
- [Symbol Window](#)
- [TextEditor Window](#)

Layout database windows:

- [Connectivity List Window](#)
- [Layout Scenarios Window](#)
- [Layout Window](#)

Customizing Your Workspace

Windows can be displayed or hidden at any time using the menu item **View > Windows** then selecting or deselecting the window, or by using the buttons in the [View Toolbar](#).

The layout of windows within your workspace is customizable; they can be docked, undocked, put into tabs, and moved around. The show/hide state of each window is preserved between invocations of the tool.


The currently active window is distinguished by a differently colored title bar. Input key presses are sent to the active window first.

Selecting Objects

When objects like signals, pins, and symbol elements are selected in one window, that selection is maintained across all relevant windows. For example, the connection between signals and pins is based on pin assignments, so whenever a signal is selected in the [Signals List](#), its assigned pin is highlighted in the [Pins List](#). Similarly, whenever a pin is selected in the [Pins List](#) and this pin is assigned to a signal, the signal is highlighted in the [Signals List](#).

Graphical elements of symbols do not have their counterparts in other windows, however, ports in symbols do have counterparts in signals or pins. When you select a port in the [Symbol Window](#), the appropriate signal and/or pin is also selected in all other windows.

Signals List

The Signals List displays a list of signals. These signals can be read from an external source file (for example an HDL file) or entered manually. The Signals List may be closed and opened at any time by clicking **View > Windows > Signals List**, or by clicking the  icon on the [View Toolbar](#).

The tabs at the bottom of the list allow you to view signals by their assignment or placement status.

- All
Lists all available signals.
- Assigned
Lists signals that have assigned pins.
- Unassigned
Lists signals that do not have assigned pins.
- Unplaced
Lists signals that:
 - Have not been placed on a PCB / functional symbol.
 - Are not assigned.

Figure 3-1. Signal List Window

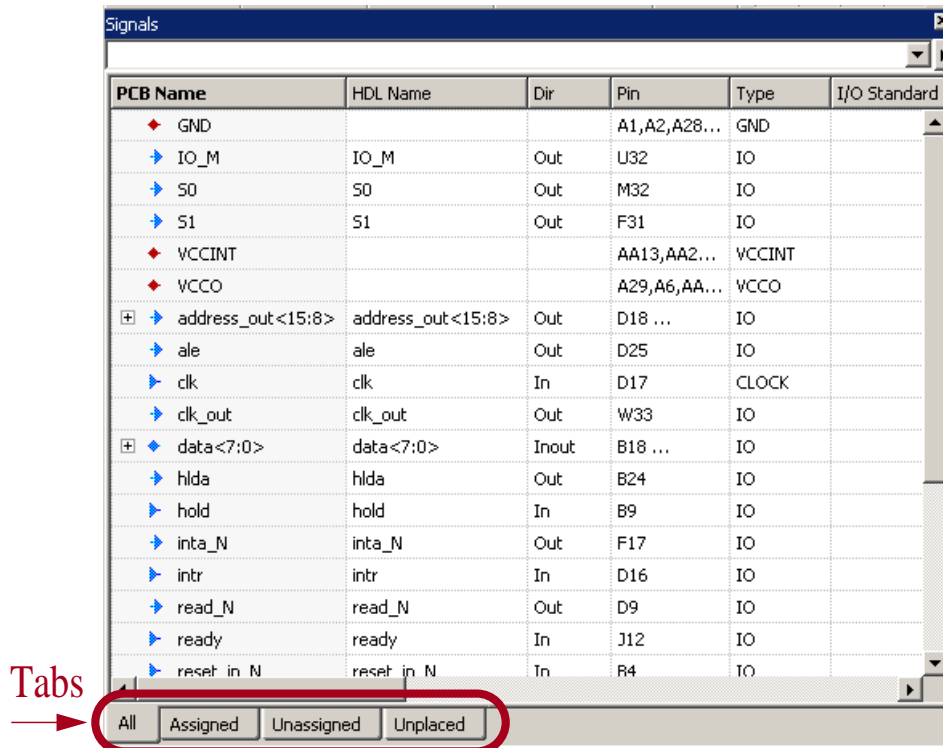


Table 3-1 describes the information listed in the Signal List for each signal:

Table 3-1. Signals List Contents

Column	Description
PCB Name	Signal PCB name
HDL Name	Signal HDL name
Dir	Signal direction
Pin	Pin currently assigned to the signal
Type	Signal type
I/O Standard	The chosen I/O standard for the currently assigned pin, such as GTL or HSTL
Swap Group	Swap group of the currently assigned pin
Locked By	Indicates whether a signal is locked
Symbol	The functional and/or PCB symbol belonging to that signal
Drive Str.	The signal's drive strength
Termination	The signal's termination type for Xilinx devices
Slew Rate	Programmable slew rate control

Table 3-1. Signals List Contents

Column	Description
Delay	Buffer Delay
PCI Clamp	On Chip PCI-clamp diode
Pullmode	Pin Termination for Lattice devices
OPENDRAIN	On/Off Open Drain for Lattice devices
Input termination	On Chip Input Termination type for Altera devices
Output termination	On Chip Output Termination type for Altera devices
DQ Group Size	DQ Group Size
DQS for DQ	DQS for DQ information


Copying Signal Data to the Clipboard

When using the [Signals List](#), the **Edit > Copy** command allows you to copy the content of the selected rows to the clipboard. The content is copied as text, one line for each row. All visible columns are copied, and their contents are separated with the tab character. In this format, the data can be pasted into a spreadsheet application.

Related Topics

- “[Defining I/O Signals](#)” on page 69
- “[Making Pin Assignments](#)” on page 83

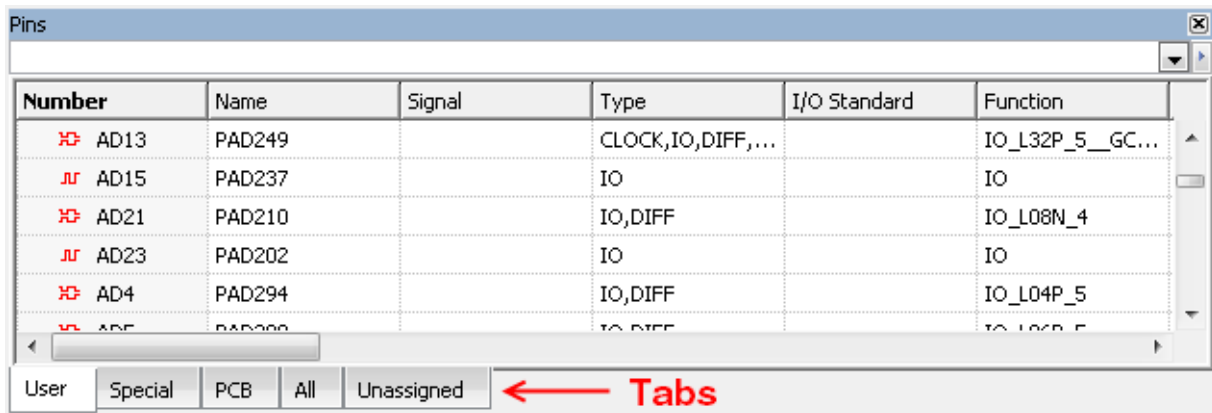
Pins List

The Pins List displays the pins available in the currently selected device. The Pins List can be closed and opened at any time, by selecting the menu item **View > Windows > Pins List**, or by pressing the  button on the [View Toolbar](#).

Note



The number of columns in the Pins List can vary depending on the vendor.

Figure 3-2. Pins List

Number	Name	Signal	Type	I/O Standard	Function
AD13	PAD249		CLOCK,IO,DIFF,...		IO_L32P_5_GC...
AD15	PAD237		IO		IO
AD21	PAD210		IO,DIFF		IO_L08N_4
AD23	PAD202		IO		IO
AD4	PAD294		IO,DIFF		IO_L04P_5
AD5	PAD200		IO,DIFF		IO_L08P_5

User Special PCB All Unassigned ← Tabs

In an FPGA database, tabs at the bottom of the list are used to view different pins:

- User
Lists normal pins used for I/O.
- Special
Not used. Tab displaying special pins, e.g. MGT, PLL pins that might have been assigned to signals.
- PCB
Lists PCB pins.
- All
Lists all pins.
- Unassigned
Lists unassigned pins.

Table 3-2 describes the information listed in the Pins List for each pin on the device.

Table 3-2. Pins List Contents

Column	Description
Number	The pin's number.
Name	The pin name.
Signal	The signal to which the pin is assigned.
Type	All possible types for the pin, such as IO, GND, VCC, VSS or VDD. The default type is listed first.
I/O Standard	I/O Standard chosen for a pin, such as GTL or HSTL.

Table 3-2. Pins List Contents

Column	Description
Function	An additional description of a pin, such as GND or CLK.
Bank	The power bank that a pin belongs to.
Swap Group	The swap group to which the pin belongs.
VccIO	The I/O VCC supply
Vref	Input reference voltage
Vref Group	Voltage Reference Group Name
Drive Str.	The pin's drive strength.
Symbol	The PCB symbol that the pin belongs to.
Termination	Pin termination type for Xilinx devices
Slew Rate	Programmable slew rate control
Delay	Buffer Delay
MGT Channel	Serial Transceiver Channel for Altera devices
PCI Clamp	On Chip PCI-clamp diode
Pullmode	Pin Termination for Lattice devices
OPENDRAIN	On/Off Open Drain for Lattice devices
Input termination	On Chip Input Termination for Altera devices
Output termination	On Chip Output Termination for Altera devices
Clock Regions	Clock Regions for Xilinx devices
DQ DQS Regions	DQ DQS pins regions information

Copying Pin Data to the Clipboard

When using the [Pins List](#), the **Edit > Copy** command allows you to copy the content of the selected rows to the clipboard. The content is copied as text, one line for each row. All visible columns are copied, and their contents are separated with the tab character. In this format the data can be pasted into a spreadsheet application.

Printing Signals and Pins


To print a list of signals or pins, select **File > Print > Signals** or **Pins**. This displays your system's standard Print dialog.

Related Topics

- [“Making Pin Assignments”](#) on page 83

- “Combining Pins into a Bus Pin” on page 85
- “DQ DQS Pins Display” on page 32

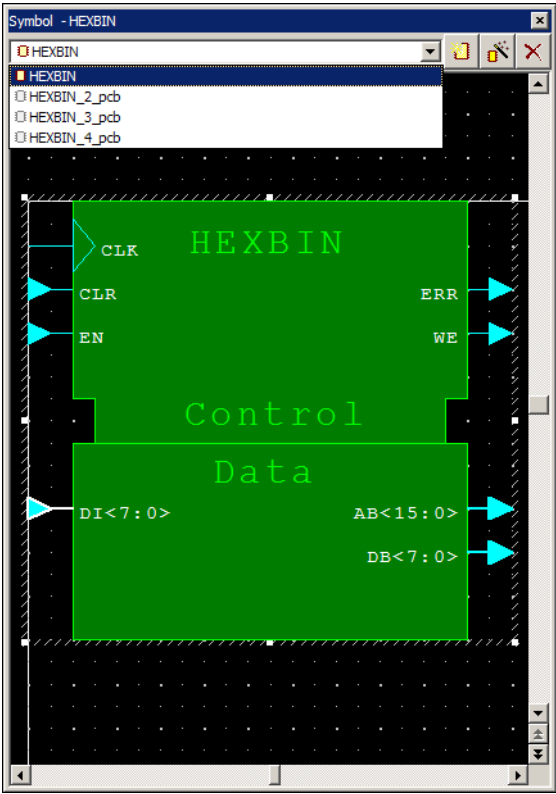
Symbol Window

The Symbol Window is available when working with FPGA databases, and displays the current symbol form. You can open or close this window at any time from **View > Windows > Symbol**, or by clicking the  icon on the [View Toolbar](#).

In addition to displaying generated symbols, the Symbol Window can be used to draw and edit symbols, as described in “[Building a New Symbol](#)” on page 104.

Use the scrollbars on the right and bottom of the window to navigate around the window.

Figure 3-3. Symbol Window



You can zoom-in on the Symbol Window. To zoom-in on the Symbol Window, use the controls on the main toolbar, or right-click in the Symbol Window and use the zoom controls.

Table 3-3. Symbol Window Zoom Controls


	Zoom In Increases zoom by 50% 3-button mouse: Click the middle mouse button.
---	--

Table 3-3. Symbol Window Zoom Controls









	<p>Zoom Out Decreases zoom by 50% 3-button mouse: Press SHIFT and click the middle mouse button.</p>
	<p>Zoom to Fit Shows the symbol in its entirety</p>
	<p>Zoom to Selection Fills the window with the selected symbol elements. 3-button mouse: To zoom to an area, hold down SHIFT, and drag in the symbol window with the middle mouse button or the right mouse button. If you move to the right, zoom in will be performed. If you move to the left, zoom out will be performed. Press Esc, or drag into the cancel box in the upper-left corner of the dragging rectangle to cancel the zoom.</p>
	<p>Previous Zoom Restores the zoom to the state of the previous zoom operation.</p>
	<p>Next Zoom Cancels the Previous Zoom operation.</p>

Table 3-4 describes the Select, Zoom, and Pan modes within the Symbol Window.

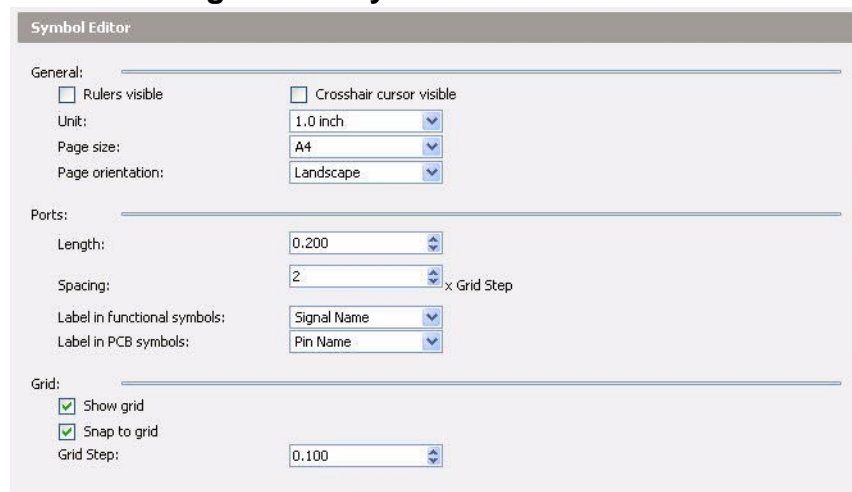
Table 3-4. Symbol Window Modes

	<p>Select Mode Allows you to select ports and graphical elements in the Symbol Window. For details, see the section “Editing Symbol Elements” on page 107.</p>
	<p>Zoom Mode When enabled, zooming in is performed by a single mouse-click. Zooming out is performed by SHIFT+click. To zoom in on a particular area, click and drag a rectangle around it. Upon releasing the mouse button, the selected area fills the entire Symbol Window.</p>
	<p>Pan Mode The Pan Mode allows you to easily scroll in the Symbol Window. This option is also called a Hand Tool. The mouse cursor in Pan Mode has the hand shape. Clicking and dragging in Pan Mode moves the entire Symbol Window and its contents.</p>

Symbol Window Settings

To customize the appearance and behavior of the Symbol Window, click **Setup > Settings + Symbol Editor** to display the Symbol Editor window as shown in [Figure 3-4](#). Use this window to modify settings, such as rulers, crosshair cursor, grid visibility, snapping mode, units, and so on.


Figure 3-4. Symbol Editor Window



Related Topics


- [“Creating, Editing and Updating Symbols and Schematics”](#) on page 95
- [“Symbols Generator”](#) on page 96
- [“Building a New Symbol”](#) on page 104
- [“Editing Symbol Elements”](#) on page 107


TextEditor Window

The TextEditor Window opens an XML file defining a device for editing. You can open and close the file menu item **View > Windows > TextEditor**, or with the  button on the View Toolbar.


You can also use this icon to reopen a window when you have closed it by clicking the x in the upper right corner of the window.

Device Window

The Device Window displays a graphical representation of the physical device. You can close it and open it with the menu item **View > Windows > Device**, or with the  button on the View Toolbar.

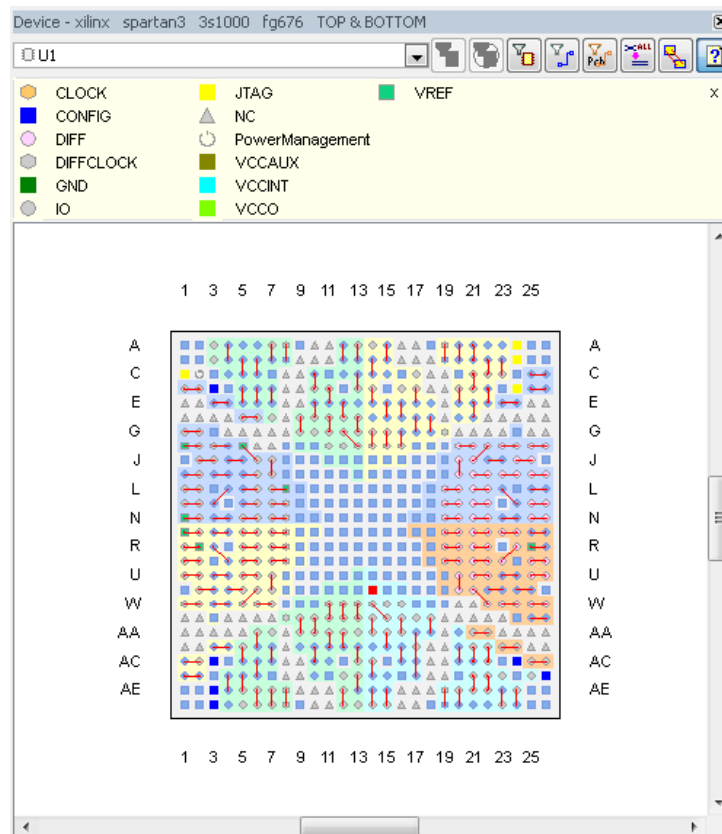
You can select and filter any set of components within the Device Window. To view individual components in the Device Window, press **Ctrl+Alt**, and **left-click** on the component. The Filter PCB Nets button  allows you to filter out PCB nets in the Device view, useful when unravelling.

Note

 PCB signals are filtered only if they are on the signal list. PCB nets still show up in the Signals list, even though they are turned off in the Device Window.

You can also change the active FPGA by double-clicking directly from the Device Window (or from the Project Window).

Figure 3-5. Device Window



You can view the Top, Bottom, or Both Sides of a device. To switch between the views, select from the **View > Device** menu options, **Top View**, **Bottom View**, or **Both Sides View**.

To rotate the view of a device and its layout, click **View > Device > Rotate View**, or right-click on the device, and select the required rotation (possible values are 0, 90, 180, and 270 degrees).

Right-click in the Device Window and select Rotate View by PCB Layout to switch to a PADS view format. When an active device can be rotated on the board, this option allows you to rotate







the view of the entire board in the device window. When the Rotate View by PCB Layout option is turned off, the device window displays the FPGA and its layout with the same rotation established in a PADS.

To enable the display of differential pins, choose **View > Device > Show Differential Pairs** option from the menu. The differential pairs are then connected with red lines (by default).

Pin Display

Pins in the Device Window are displayed in different colors depending upon their current status. [Figure 3-5](#) shows the default colors for a sample of pins.

Table 3-5. Pin Display Colors

	Assigned Pin (a pin that is assigned to a signal) (Color: light blue)
	Unassigned Pin (a pin that is free to be assigned to a signal) (Color: gray)
	USER or SPECIAL unassigned pin (Color: black)
	Selected pin (Color: red)
	Pin to be assigned using drag and drop (see “ Assigning Signals to Pins Using Drag and Drop ” on page 84) (Color: Yellow on light blue)
	Pin marked for assignment (see “ Mark to Assign ” on page 83) (Color: green)

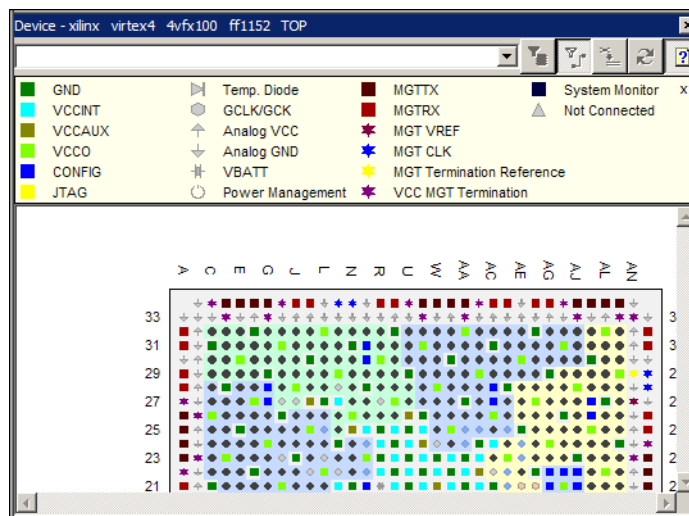
To change pin colors, do the following:

1. Select **Setup > Settings + Appearance**.
2. Choose **Device Window** from the drop down list in the **Category** field.
3. Select the pin type from the **Subcategory** list and edit the colors using the drop-down controls.

FPGA Pin Types

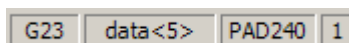
The FPGA pin types are described in the Pin Legend box, located at the top of the Device Window:

Figure 3-6. Pin Legend Box



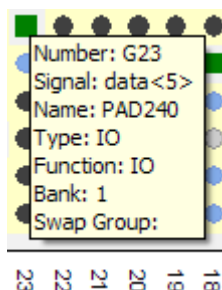
When the mouse pointer is moved over a pin, the pin information (pin number, name, bank and assigned signal name) is displayed in the status bar.

Figure 3-7. Pin Information Status Bar



Additionally, if you hold the mouse pointer over a pin for a while, you will see a tool tip window containing the pin number, name, type, bank, function, swap group and assigned signal name (if assigned).

Figure 3-8. Pin Tool Tip Window



Bank Display

Pins that belong to different power banks are displayed with a different background color, which improves visual distinction between pins from different banks. The banks are displayed in three colors in such a way that banks 1, 4, 7, etc. are displayed in one color, banks 2, 5, 8, etc.

are displayed in another color, and banks 3, 6, 9, etc. in yet another color. To disable bank display, uncheck the **View > Device > Show Banks** menu option.

Clock Display

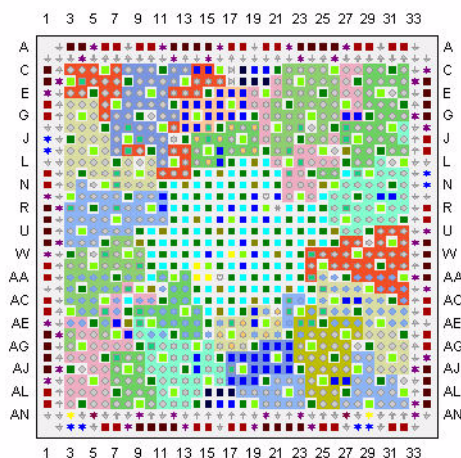
PADS I/O Designer allows you to filter the Device Window for various clock information. The option is available for Xilinx Virtex families, as specified below. To use clock filtering, select from the menu options:

- **View > Device > Show Clock Regions:** Show Clock Regions (Xilinx Virtex4, QVirtex4, QRVirtex4, Virtex5).
- **View > Device > Show Local Clocks - IOB Columns:** (Xilinx Virtex2, Virtex2p, Spartan3, Spartan3l, Spartan3e, Spartan3a, Spartan3adsp) Show possible clock pins. Move the mouse over the selected pins to see the corresponding clock regions.
- **View > Device > Show Local Clocks - Interface to CLB Array:** (Xilinx Virtex2, Virtex2p, Spartan3, Spartan3l, Spartan3e, Spartan3a, Spartan3adsp) Show possible clock pins. Move the mouse over the selected pins to see the corresponding clock regions.

In order to select an entire clock region for a given pin, follow these steps:

- Select **Setup > Settings + Filters**, and enable **Regular expressions**.
- In the [Pins List](#), right-click the **Local clocks** column and choose **Filter Column** from the popup menu.
- In the [Pins List](#)'s filter edit box enter the desired pin number and press **Enter**.

Figure 3-9. Clock Region Example



DQ DQS Pins Display

PADS I/O Designer allows you to filter the Device Window for DQ DQS pins regions information. This option is available for Lattice and Altera families, as specified below. To use the DQ DQS regions filtering, select from the menu options:


- **View > Device > Show DQ DQS Pins in X4 Mode:** (Altera: Arria GX, Arria II GX, Hardcopy II, Stratix II, Stratix II GX, Stratix III, Stratix IV; Lattice: LatticeEC, LatticeECP, LatticeECP2, LatticeECP2M, LatticeECP2MS, LatticeECP2S, LatticeECP3, LatticeXP, LatticeXP2).
- **View > Device > Show DQ DQS Pins in X8X9 Mode:** (Altera: Arria GX, Arria II GX, Cyclone, Cyclone II, Cyclone III, Hardcopy II, Stratix, Stratix GX, Stratix II, Stratix II GX, Stratix III, Stratix IV).
- **View > Device > Show DQ DQS Pins in X16X18 Mode:** (Altera: Arria GX, Arria II GX, Cyclone II, Cyclone III, Hardcopy II, Stratix, Stratix GX, Stratix II, Stratix II GX, Stratix III, Stratix IV).
- **View > Device > Show DQ DQS Pins in X32X36 Mode:** (Altera: Arria GX, Arria II GX, Cyclone III, Hardcopy II, Stratix, Stratix GX, Stratix II, Stratix II GX, Stratix III, Stratix IV).

DCI Cascade Region Display

PADS I/O Designer allows you to show DCI Cascade Regions information in the Device Window. This option is available for Xilinx families, as specified below, and should be read from a constraint file. To use DCI Cascade Regions, select the menu option:

- **View > Device > Show DCI Cascade Regions (read from constraint file):** Virtex-5 Qvirtex-5, Virtex-6, Qvirtex-6, Virtex-6I

Show Layout in Device View

You can turn layout view on and off using the  button at the top of the Device window. Layout view is off by default. The first time you click this button, PADS I/O Designer loads and displays the layout.

Note


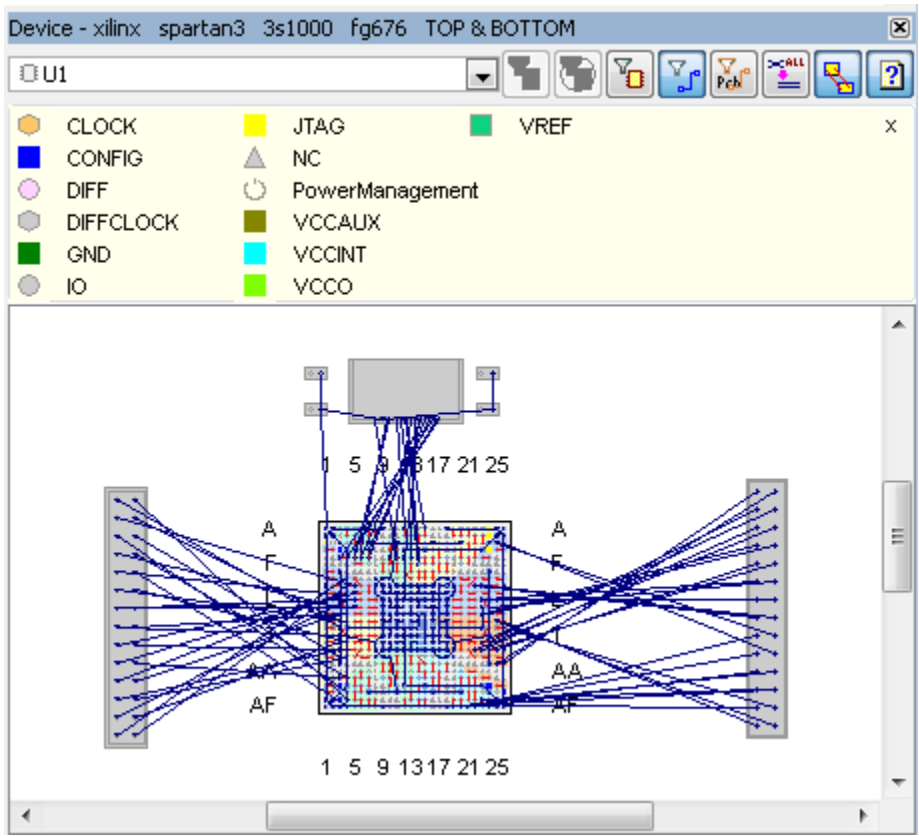
 If the layout has changes made to it in the layout tool, synchronizing the layout in PADS I/O Designer loads and displays the layout automatically.

Figure 3-10. Device View - Show Layout



Note
Device View can ONLY load a layout. The result of a .lpc or a schematic load is NOT supported.



Scrolling and Zooming in the Device Window

You can scroll through the Device Window using the controls on the main toolbar, or by right-clicking in the window to access the zoom commands on the pop-up menu.

Table 3-6. Device Window Zoom Controls

	Zoom In Increases zoom by 50%
	Zoom Out Decreases zoom by 50%.
	Zoom to Fit Shows the device in its entirety.
	Zoom to Selection Fills the window with the selected pins.

Table 3-6. Device Window Zoom Controls

	Previous Zoom Restores the zoom to the previous state.
	Next Zoom Cancels the Previous Zoom operation.

Quick Mouse Zoom Operations

You can use the mouse to change the zoom.

Table 3-7. Quick Mouse Zoom Operations

Zoom In	Click + middle button
Zoom Out	SHIFT + click + middle button
Zoom Area	SHIFT + drag + middle button
Zoom Area In	SHIFT + drag right + right button
Zoom Area Out	SHIFT + drag left + left button
Cancel Zoom	Esc

Device Window Modes

The Package Window operates in one of the following modes.

Table 3-8. Device Window Modes






	Select Mode Select Mode allows you to select pins in the Device Window. To select a pin, click on it. To add a pin to the existing selection, CTRL+click it. Dragging allows you to select more than one pin. While dragging, you will see the selection rectangle displayed. All pins within the rectangle are selected. To zoom in on an area in Select Mode, use SHIFT+click , and drag a rectangle around the required area. When the mouse button is released, the window zooms in on the selected rectangle to fill the entire window.
	Zoom Mode. When enabled, zooming in is performed by a single mouse-click. Zooming out is performed by SHIFT+click . To zoom in on a particular area, click and drag a rectangle around it. Upon releasing the mouse button, the selected area fills the entire window.
	Pan Mode allows you to scroll in the Device Window. In Pan Mode, the mouse cursor has the hand shape. Clicking and dragging in Pan Mode moves the entire Device Window and its contents.

Table 3-8. Device Window Modes

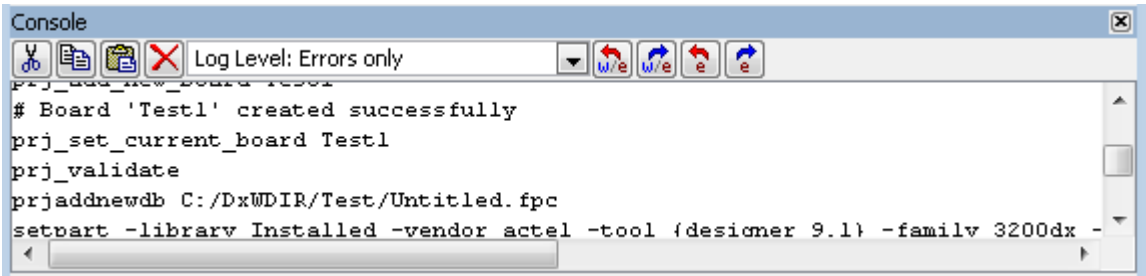
	Assign Mode allows you to assign signals to pins visually. This mode is useful when assigning a bus signal. Select a bus signal in the Signals List and click in the Device Window to start assigning bus items. You will notice a special mouse cursor in the Assign Mode. By default, the bus items are assigned MSB to LSB. Press the Shift key to assign the items the other way. The selection in the Signals List is updated as consecutive items are assigned.
---	--

Console Window


The Console Window displays commands, output messages, error messages, and warnings. Commands can be entered and executed directly from the Console Window. The Console Window can be closed and opened at any time from **View > Windows > Console**, or the  button on the View Toolbar.


A toolbar at the top of the window provides access to common Console Window commands. You can close and reopen the toolbar from a popup menu. You can also open the console log file by selecting **Show console log** from the popup menu.

Figure 3-11. Console Window



You can enter commands in the last line of the Console Window. Press Enter to execute the command. See “[TCL Interface](#)” on page 191 for descriptions of the commands available.

The four buttons to the right of the dropdown window  allow you to jump forward or back through the file in the console window, one error or warning at a time.

To clear the contents of the Console Window, click the clear console button , or right-click inside the Console Window and select **Clear Console**.

Errors and Warnings

If you choose to work with the Console Window closed, you can tell PADS I/O Designer to open the window automatically if a warning error is reported. To do this:

1. Select **Setup > Settings + Advanced**.

2. Select **Open console at warning**, and/or **Open console at error**.

Shortcuts in the Console Window

For operations such as repeating previous commands in the Console Window, the following shortcut keys are available:

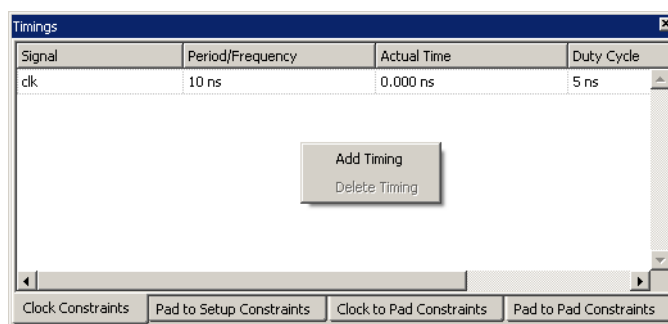
- Up Arrow - If the cursor is in the last line, it displays the previous command. Otherwise it moves the cursor up.
- Down Arrow - If the cursor is in the last line, it displays the next command after using the **Up Arrow** key. Otherwise it moves the cursor down.

Timings Window

The Timings Window allows you to add timing constraints for signals in the [Signals List](#). There are 4 categories of timing constraints you can specify: Clock Constraints, Pad to Setup Constraints, Clock to Pad Constraints, and Pad to Pad Constraints.

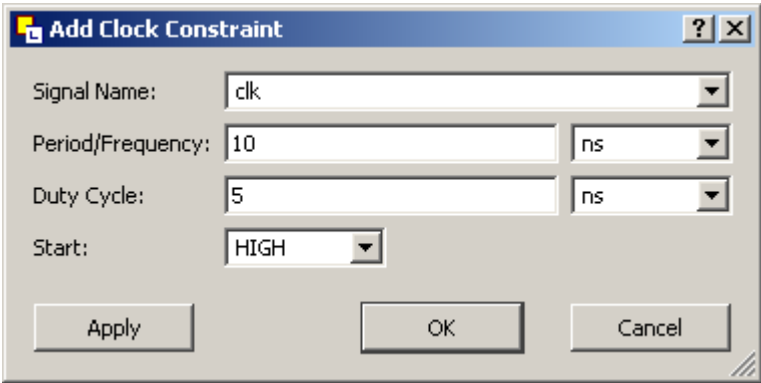
1. Clock Constraints

Figure 3-12. Clock Constraints



You can add or delete constraints by using the Add Timing and Delete Timing options from the popup menu. Only signals with CLOCK and DIFFCLOCK type are displayed in the Signal Name list.

Figure 3-13. Add/Delete Constraints



2. Pad to Setup Constraints

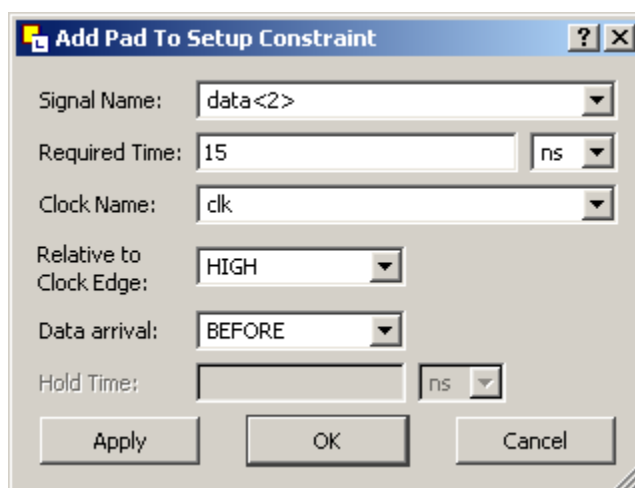
Figure 3-14. Pad to Setup Constraints

Timings					
Signal	Required Time	Actual Time	Slack (ns)	Clock	Relative to Clock
data<0>	10 ns	0.000 ns	10	clk	HIGH
ready	7 ns	0.000 ns	7	clk	HIGH

Clock Constraints
Pad to Setup Constraints
Clock to Pad Constraints
Pad to Pad Constraints

Only signals with IN and INOUT type are displayed in the Signal Name list and only signals with CLOCK and DIFFCLOCK are displayed in the Clock Name list.

Figure 3-15. Add Pad to Setup Constraint

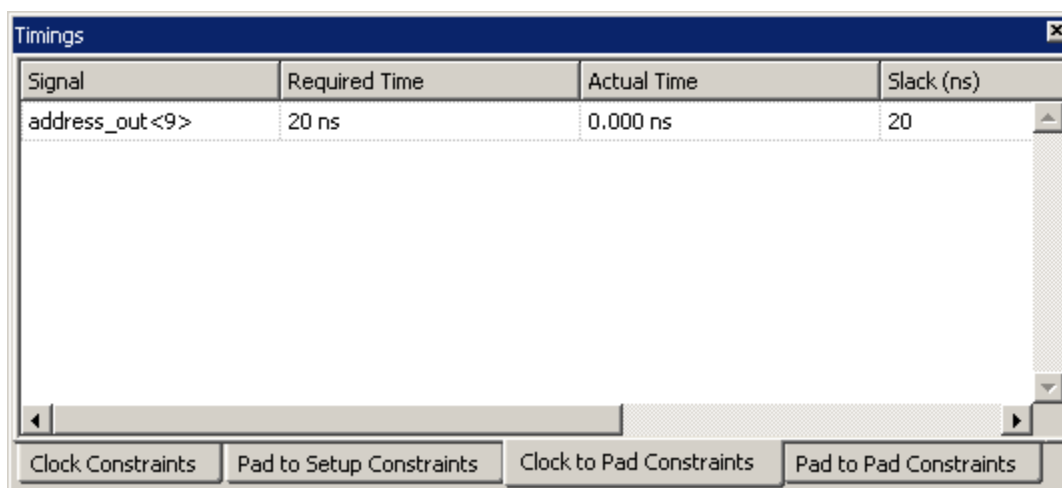


The dialog box titled "Add Pad To Setup Constraint" contains the following fields and controls:

- Signal Name: data<2>
- Required Time: 15 ns
- Clock Name: clk
- Relative to Clock Edge: HIGH
- Data arrival: BEFORE
- Hold Time: ns
- Buttons: Apply, OK, Cancel

3. Clock to Pad Constraints

Figure 3-16. Clock to Pad Constraints



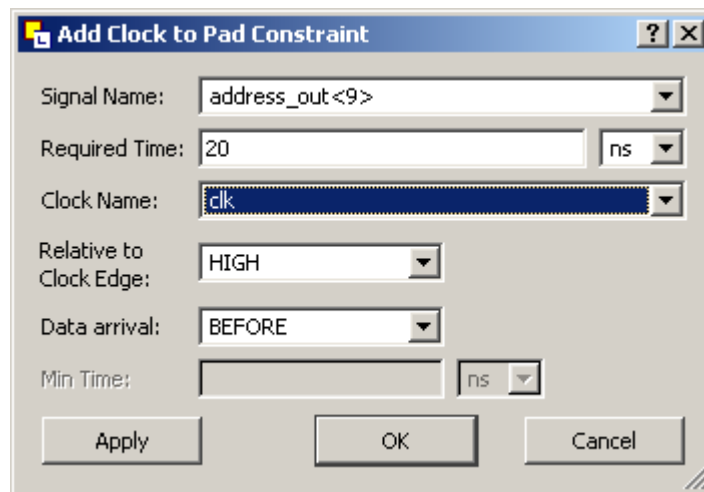
The Timings window displays a table with the following data:

Signal	Required Time	Actual Time	Slack (ns)
address_out<9>	20 ns	0.000 ns	20

At the bottom of the window, there are four tabs: Clock Constraints, Pad to Setup Constraints, Clock to Pad Constraints (selected), and Pad to Pad Constraints.

Only signals with OUT and INOUT type are displayed in the Signal Name list and only signals with CLOCK and DIFFCLOCK are displayed in the Clock Name list.

Figure 3-17. Add Clock to Pad Constraint

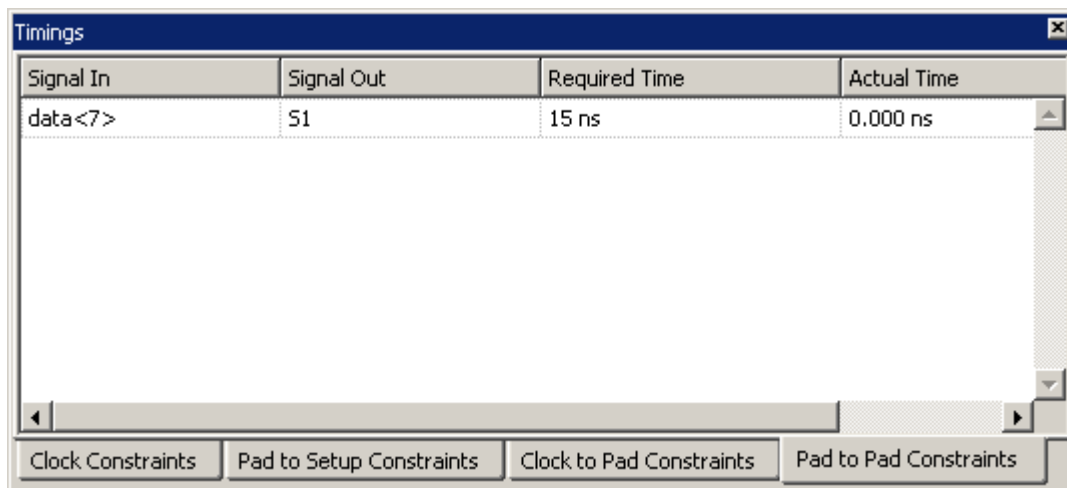


The dialog box 'Add Clock to Pad Constraint' contains the following fields and controls:

- Signal Name:** address_out<9>
- Required Time:** 20 ns
- Clock Name:** clk
- Relative to Clock Edge:** HIGH
- Data arrival:** BEFORE
- Min Time:** ns
- Buttons:** Apply, OK, Cancel

4. Pad to Pad Constraints

Figure 3-18. Pad to Pad Constraints



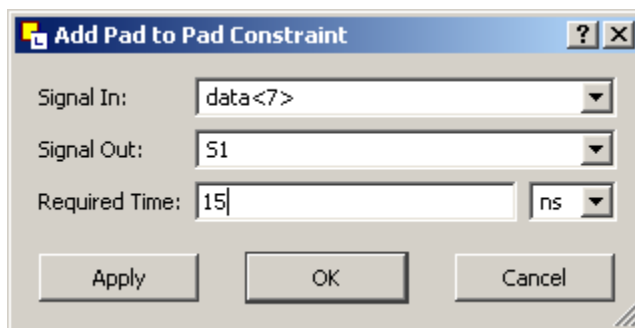
The 'Timings' window displays a table of constraints. The 'Pad to Pad Constraints' tab is selected.

Signal In	Signal Out	Required Time	Actual Time
data<7>	S1	15 ns	0.000 ns


At the bottom of the window, there are four tabs: Clock Constraints, Pad to Setup Constraints, Clock to Pad Constraints, and Pad to Pad Constraints.

Only signals with IN and INOUT type are displayed in the Signal In list and only signals with OUT and INOUT are displayed in the Signal Out list.

Figure 3-19. Add Pad to Pad Constraint



Properties Window

The Properties Window displays and enables editing of the attributes of signals, pins, symbols and their elements. To open the Properties window, use the **Edit > ... Properties** command, the  button on the **View Toolbar**, or by right-clicking on the selected element and choosing **... Properties**. Additionally, double-clicking the selected elements in the **Symbol Window** opens the Properties Window. The name of the menu command displaying the Properties window depends on the active window. For example, if the **Signals List** is active, the command is named **Signal Properties**.

The contents of the Properties Window depends on the active window and the current selection. If the **Signals List** is active, signal properties are displayed. If the **Pins List** is active, pin properties are displayed. If the **Symbol Window** is active, selected elements properties are displayed, or the attributes of the whole symbol when nothing is selected.

Some attributes displayed in the Properties window are read-only, for example, signal name for ports. Other attributes may be modified by double-clicking the attribute field on the right side of the Properties window. Some attributes are presented in groups, with a + sign to the left. For example, all attributes associated with displaying the Signal Name besides the port are in the Signal Name group.

In the Properties window, you can simultaneously change attributes for a group of selected items. For example, to change properties for several pins, select the pins, open the Properties window, and edit the desired attribute.

Any change made to an attribute will only be carried through to all elements supporting that attribute. For example, if ports are selected together with an arc, changing the **Inverted** attribute will set its new value for all selected ports, leaving the arc unchanged.

PADS I/O Designer interprets most of the symbol attributes internally. However, it is possible to add custom symbol attributes. To add a custom attribute, click on the right side of the Properties window next to the text New attribute, and type in the name of the new attribute. It will then appear on the list, and can be edited as other attributes. Custom attributes are generated in symbol and schematic files in the form of properties.

For each type of element, the following tables list all the attributes available in the Properties Window:

- [Signal Properties](#)
- [Pin Properties](#)
- [Pin Properties for Layout Database](#)
- [Port Properties](#)
- [Symbol Properties](#)

Signal Properties

Table 3-9. Signal Properties

Name	Value
PCB Name	Signal PCB name
HDL Name	Signal HDL name
Dir	Indicates the direction of the signal (In/Out)
Pin	Shows the pin assigned to the signal
Type	Shows the type of signal
I/O Standard	Shows the I/O standard of the signal
Swap group	Shows the signal swap group
Locked by	Indicates whether the signal is locked
Symbol	Shows the symbol name that the signal is placed upon. If a signal is placed on more than one symbol, their names are separated by '&'.
STRENGTH	Shows the pin's drive strength.
TERMINATION SLEW_RATE DELAY	Additional signal properties for Xilinx devices
WEAK_PULL_UP_RESISTOR TREAT_BIDIR_AS_OUTPUT SLOW_SLEW_RATE POWER_UP_LEVEL PCI_IO INCREASE_DELAY_TO_OUTPUT_PIN FAST_OUTPUT_REGSITER FAST_OUTPUT_ENABLE_REGISTER FAST_INPUT_REGISTER ENABLE_BUS_HOLD_CIRCUITRY DECREASE_INPUT_DELAY_TO_INTER NAL_CELLS STRENGTH	Additional signal properties for Altera devices

Pin Properties

Table 3-10. Pin Properties

Name	Value
Number	Shows the pin number. This property is read-only.
Name	Shows the pin name. This property is read-only.
Signal	Shows the signal to which the pin is assigned.
Swap group	Shows the pin swap group.
Type	Indicates the pin type.
I/O Standard	Shows the I/O Standard selected for the pin.
Function	Shows the pin function. This property is read-only.
Bank	Shows the power bank the pin belongs to. This property is read-only.
VCCIO	Shows the I/O VCC supply level.
STRENGTH	Shows the pin's drive strength.
TERMINATION SLEW_RATE DELAY	Additional pin properties for Xilinx devices.
SLOW_SLEW_RATE POWER_UP_LEVEL FAST_OUTPUT_REGISTER FAST_INPUT_REGISTER DECREASE_INPUT_DELAY_TO_I INTERNAL_CELLS	Additional pin properties for Altera devices.
DQDQS X32X36 Region	Shows the X32X36 region number the pin belongs to. This property is read-only.
DQDQS X16X18 Region	Shows the X16X18 region number the pin belongs to. This property is read-only.
DQDQS X8X9 Region	Shows the X8X9 region number the pin belongs to. This property is read-only.
DQDQS X4 Region	Shows the X4 region number the pin belongs to. This property is read-only.
Vref Group	Shows the vref group name/number the pin belongs to. This property is read-only
Lab Row	Shows the row number of a Lab where the pin exists. This property is read-only.

Table 3-10. Pin Properties (cont.)

Lab Column	Shows the column number of a Lab where the pin exists. This property is read-only.
Local Pad Number	Shows the pin position inside of a Lab. This property is read-only.
Global Pad Number	Shows the global internal pin number. This property is read-only.

Pin Properties for Layout Database

Table 3-11. Pin Properties for Layout Database

Name	Value
Number	Shows the pin number. This property is read-only.
Name	Shows the pin name. This property is read-only.

Port Properties

Table 3-12. Port Properties

Name	Value
Port Label	<p>Label Visible Indicates whether the port label is visible or hidden. The default is True.</p> <p>Label Position Indicates the position of the label relative to the port. The property is available only when Label Visible is set to True. This can be one of the following: Over, Under, Side, and Moved. Moved signifies that you have moved the label. The default is Side.</p> <p>Label Type One of the following: Signal Name, Pin Name, and Pin Number. Indicates what is used as the port label. The default value of this property may be set under Setup > Settings, on the Symbol Editor page. The Label in Functional block option is used to set the default for functional symbols. The Label in PCB symbol option is used to set the default for PCB symbols.</p>
Signal Name	<p>Displays the signal name as well as the main port label.</p> <p>Label Visible Indicates whether the signal name is visible or hidden. The default is False.</p> <p>Label Position Indicates the position of the signal name relative to the port. The property is only available when Label Visible is set to True. This can be one of the following: Over, Under, Side, and Moved. Moved signifies that you have moved the signal name. The default is Side.</p>
Pin Number	Displays the pin number as well as the main port label.

Table 3-12. Port Properties (cont.)

Pin Name	<p>Label Visible Indicates whether the pin number is visible or hidden. The default is False.</p> <p>Label Position Indicates the position of the pin number relative to the port. The property is only available when Label Visible is set to True. This can be one of the following: Over, Under, Side, and Moved. Moved signifies that you have moved the pin number. The default is Over. Displays the pin name as well as the main port label.</p> <p>Label Visible Indicates whether the pin name is visible or hidden. The default is False.</p> <p>Label Position Indicates the position of the pin name relative to the port. The property is only available when Label Visible is set to True. This can be one of the following: Over, Under, Side, and Moved. Moved signifies that you have moved the pin name. The default is Side.</p>
Pin Function	<p>Displays the pin function as well as the main port label.</p> <p>Label Visible Indicates whether the pin function is visible or hidden. The default is False.</p> <p>Label Position Indicates the position of the pin function relative to the port. The property is only available when Label Visible is set to True. This can be one of the following: Over, Under, Side, and Moved. Moved signifies that you have moved the pin function. The default is Over.</p>
Signal Name	<p>Displays signal name.</p> <p>Label Visible Indicates whether signal name is visible or hidden. The default is False.</p> <p>Label Position Indicates the position of the label relative to the port. The property is available only when Label Visible property is set to True. Label Position may be one of the following: Over, Under, Side, and Moved. Moved indicates that you have moved the label. The default value is Over.</p>
Pin Type	Shows the pin type. This property is read-only.
Dir	Shows the direction of the signal. This property is read-only.
Port Type	Shows the port type. The available types are: default, IN, OUT, BI, ANALOG, OCL, OEM, TRI, POWER, GROUND, TERMINAL.

Table 3-12. Port Properties (cont.)

Port Shape	Shows the shape of the port. The available shapes are: default, ARROW, and CLK.
Inverted	Indicates whether the port is inverted. The default value is False.
Length	Specifies the port length.
Port Binding	<p>Specifies port binding behavior during signal reassignment. There are two possibilities:</p> <p>Signal: If a given port has the Signal binding type, it means that the port is bound to a signal. On reassignment, the port's pin properties change according to the reassignment. However, the port's signal properties remain unchanged.</p> <p>Pin: If a given port has the Pin binding type, it means that the port is bound to a pin. On reassignment the port's signal properties change according to the reassignment. However, the port's pin properties remain unchanged.</p>
Attributes	Lists the attributes defined for the port.

Symbol Properties

Table 3-13. Symbol Properties

Symbol Name	Indicates the name of the current symbol.
PCB Symbol	Indicates whether the current symbol is a PCB or functional symbol. This property is read-only.
Background	The background color of the current symbol. The default empty value is a standard rectangular background.
Attributes	Lists attributes of a symbol. Symbol attributes may be locked to make sure they are not overwritten during the export/import process. To lock/unlock a given attribute, click the padlock icon.
Instance Attributes	Lists instance specific attributes.
Automatic Symbol Changes	Used by the Symbol Generator to determine whether automatic symbol update is allowed to modify this symbol.
Symbol Group	Used by the Symbol Generator during the update process to determine the group of pins to add to a particular symbol.

Graphic Properties


- Color - the color of the selected graphical elements.
- Fill Style - the fill style of the selected graphical elements; solid or grid, for example.
- Line Style - the type of line; solid or dashed, for example.
- Line Width - the width of the line in common units.



Text Properties

- Text - shows the characters within the text.
- Font - shows the font in which the selected text is displayed.
- Font Size - shows the font height of the selected text.
- Text Color - shows the color of the selected text.

Synchronization Wizard

The Synchronization Wizard synchronizes the database with all external files associated with it, such as input files specified to supply source data for the component, and the schematic to which you want to export the component.




To access the Synchronization Wizard, select **Import / Export > Synchronize**, click the  icon on the main toolbar, or click the set of three circles in the lower right corner of the status bar.

Use the  and  icons to switch between Logical view and Files view.

Logical view lists items that need to be synchronized in a logical manner. Files view lists all items that may need to be synchronized. The **Status** column reports the status of each tracked item.

Synchronizing Files

Use the following procedure to run the Synchronization Wizard and update some or all files used in the database:

1. Enable the desired action in the **Action** column. For example, import data , export data  or take no action (blank selection). If both import and export are needed, the conflict icon  displays, and you must choose which action to take. If tracking is disabled, you must select the required action from the **Action** column.

2. Click **Next**.

This starts the process of synchronizing items that have been marked for update in the **Action** column.

3. When the progress is complete, click **Next**.

A summary of what has been updated is displayed.

If there are conflicts in assignments, the Synchronization Wizard allows you to merge the assignments manually, or select a file from which to read the assignments. The following files contain the assignment information:

- Synthesis Constraints Files
- P&R Constraints Files
- Pin Report Files
- FPGA Xchange Files
- PCB Design Files

4. Click **Finish**.

Synchronization Check

PADS I/O Designer checks for updated files at a specified time interval, and prompts you to run Synchronization if it finds them. By default, this check is performed every 3 seconds, but you can change this time interval on the **Setup > Settings + Synchronization Check** page.

PADS I/O Designer tracks only the files selected in the **Track Status** column of the [Synchronization Wizard](#) in this check. Select the checkbox in the **Track Status** column for each item you want to check with the Synchronization checker.

Click **Apply** to save the Synchronization Wizard settings (tracking) for the current database. The settings are stored in the database file following a **Save** action.

GUI Startup Preferences

PADS I/O Designer GUI (graphical user interface) startup preferences can be set under **Setup > Settings + Gui Startup**.

Three startup options are available:

- Do not show startup dialog — PADS I/O Designer does not prompt you for the project to open. You select the file to open by selecting **File > Open Project**, or **File > New Project**.
- Load last project — PADS I/O Designer starts and automatically reopens the most recently opened project.
- Show startup dialog — PADS I/O Designer opens and shows a list of recent projects to select from, as well as options to start a new project or browse to an existing project.

Note



When PADS I/O Designer launches from within a PADS Designer project, PADS I/O Designer opens with that design project's design and associated PADS I/O Designer project databases.


For more information on project and databases, see “[Working with Projects and Databases](#)” on page 57.

Errors and Warnings

PADS I/O Designer reports errors and warnings in the [Console Window](#) and displays them in dialogs. You can customize this behavior in the **Setup > Settings** dialog. On the Advanced page you can find the options **Show warnings in dialogs** and **Show errors in dialogs**. Separate options for errors and warnings give you more control over what to present in the dialog, and what to present in the [Console Window](#) only.

Customizing Files and Directories

The **Setup > Settings + Paths** page enables you to specify paths to files and directories necessary for tool and project integration.

To change a directory path, double-click on the path and select the  icon. This allows you to browse and select a new location.

Changing the Appearance of Elements

The **Setup > Settings + Appearance** page enables you to change the appearance of some elements of PADS I/O Designer.

The settings are grouped by categories, and a number of subcategories are available for each category. For each subcategory, the color, font, and other attributes can be changed (the content depends on the type of scheme specified).

Note



Font size in the Symbol Window can be set either in standard points, or relative to the symbol port length. If the option Size relative to port length is selected, then font size is set in relation to Port Length. Port Length can be changed on the Symbol Editor page in the **Setup > Settings** dialog.

Source Control Settings

The Source Control page where you set the source control system. PADS I/O Designer supports the following:

- Microsoft SourceSafe
- CVS
- RCS
- Clearcase
- Custom

Depending on your choice of source control system, various options are available on the Source Control page. To display the Source Control page, click **Setup > Settings + Source Control**.

Microsoft SourceSafe

To configure PADS I/O Designer to work with Microsoft SourceSafe, you must provide the following information:

- Database - path to SourceSafe initialization file
- Root - root node of the SourceSafe repository

CVS

To configure PADS I/O Designer to work with CVS (Concurrent Versioning System), you must enter a path to the CVS repository.

Custom Source Control System

The Custom page allows you to enter any number of variable-value pairs that can later be used by custom Source Control System TCL commands.

Advanced Settings

You can set the following options on the **Setup > Settings + Advanced** page:

- Show warnings in dialog (default: off).
- Show errors in dialog (default: on).
- Open [Console Window](#) on warning (default: off).
- Open [Console Window](#) on error (default: off).
- Differential signals postfixes (defaults: `_P` and `_N`).
- Recognize differential signals: Turns on extracting differential structures info from HDL files. The option by signal postfixes tells PADS I/O Designer to use the specified postfixes during differential pairs search.
- HDL source: Enables the extraction of differential signals.
- Default bus direction: Specifies a default bus direction for scalar signal groups.
- Separate bus pins: Does not create bus pins during update from the constraints file.
- Recognize differential signals by postfixes: Enables the extraction of differential signals during constraints file update, based on defined postfixes.
- Autodetect clocks: Turns on recognizing clock signals based on a regular expression. Autodetection applies to reading signals from files (HDL source files, constraint files).
- Overwrite schematics instead of update (DesignView/DesignCapture flow only). If this option is enabled (the recommended setting), the entire PADS I/O Designer generated schematic is overwritten with the new version during export.

Automatic SSO Check: Automates running of the Xilinx SSO (Simultaneous Switching Output) Check.

Chapter 4

Typical Design Flows

Supported Schematic Design Creation Tools

PADS I/O Designer creates FPGAs that can be used in [PADS Designer Flows](#).

PADS Designer Flows

IOD supports the following DxDesigner Flows:

- [Schematic Export Flow for PADS Designer \(FPGA First\)](#)
- [Schematic Update Flow for PADS Designer \(FPGA First\)](#)
- [Schematic Update Flow for PADS Designer \(PCB First\)](#)

Schematic Export Flow for PADS Designer (FPGA First)

The following steps describe how to create an FPGA component database in PADS I/O Designer, create and export symbols to DxDesigner, perform I/O planning and layout optimization using data from PADS Layout.

1. Create a new project containing a new FPGA database as described in “[Creating a New Project](#)” on page 58 and “[Adding Databases to a Board](#)” on page 60.
2. Build the FPGA database either using the [FPGA Database Wizard](#) or by doing the following:
 - a. Select **File > Database Properties** and click the **Vendor and Device** page to specify an FPGA device. See [Selecting the Device](#).

You can also specify FPGA vendor files such as [Place and Route Constraints Files](#) or [Pin Report Files](#) at this time. See “[Importing FPGA Vendor Files](#)” on page 67.
 - b. Select the **Signal Source** page to import I/O signals for the FPGA from an HDL file or spreadsheet. See “[Defining I/O Signals](#)” on page 69.
3. Set the I/O Standard for signals in the [Properties Window](#). See “[Set the I/O Standard](#)” on page 75.
4. Add user-defined rules (*optional step*). See “[User-defined Rules](#)” on page 79.


5. Select **Assign > Assign Pins** or use the shortcut key **F4** to assign signals to pins as defined in the HDL source specified. For information on other ways to assign pins, see [Making Pin Assignments](#).

The pin assignments will follow device-specific rules as well as user-defined rules. See “[Device-dependant Assignment Rules](#)” on page 251 and “[User-defined Rules](#)” on page 79.

The FPGA design is now ready to insert into the PCB schematic and go to Layout. The next step is to build a symbol set for the I/O signals and then export them to the schematic.

Note

At this stage, the pin assignments are not optimized for the PCB layout.

6. Create symbols using the Symbols Generator. See “[Symbols Generator](#)” on page 96.
 - a. Invoke the **Symbols Generator** by selecting **Symbol > Symbols Generator** or by clicking the  icon on the toolbar or at the top of the [Symbol Window](#).
 - b. Create a hierarchical symbols for I/O pins.

This flow makes use of implicit power connections, so there is no need to create power and ground symbols. PADS I/O Designer will attach the necessary properties to the component.

7. Select **Export > Schematic and Symbols** to export the symbols to the schematic. See “[Exporting Symbols and Schematics](#)” on page 112.
8. In PADS Designer, run netlist generation and open PADS to view the optimized board results.
9. I/O Optimization (optional)
 - a. In the PADS I/O Designer layout database, select **Import > Layout** to load the PCB layout.

The PCB layout and netlines are visible in the [Layout Window](#).
 - b. Review the layout and create new layout scenarios if further optimization is needed.
10. Select **Export > Schematic and Symbols** to update the schematic with the optimized pin assignments. See “[Exporting Symbols and Schematics](#)” on page 112.

If you specified FPGA vendor files when you set up the database, you can use the FPGA database’s **Export** menu to update them with the new pin outs. See “[Place and Route Constraints Files](#)” on page 135 and “[FPGA Xchange Files](#)” on page 136. To create new files, select **File > Database Properties** and click the **Place and Route** page to enter file locations for these files.

With that step, the loop is completed and the FPGA design will be updated to reflect the optimized I/O assignments.

Schematic Update Flow for PADS Designer (FPGA First)

The following steps describe how to create an FPGA component database in PADS I/O Designer for use with [Schematic Update](#) to wire-up schematic components and optimize connections during layout creation.

1. Open the project in PADS Designer and place component symbols on the PADS Designer schematic.
2. To assign Ref Des, run **Create Netlist for Layout** or **Assign Reference Designator (REFDES)** in the PCB Interface.
3. Open the project in PADS I/O Designer.
4. Build an PADS I/O Designer database, including signals and pin assignments related to the component, from the schematic:
 - a. Use HDL and/or Pin Report File for the specific device.
 - b. Use spreadsheet file with signals and assignments.
5. Ensure proper PCB Part Number and Ref Des settings on the Database Properties - PCB Flow tab. (See “[Database Properties Dialog Box](#)” on page 228.) PCB Part Number and Ref Des are used to match components on the schematic.
6. Run Export - [Schematic Update](#) to wire-up the component symbols on the schematic according to assignments in PADS I/O Designer. Schematic update converts any signal name to a net name.
7. When the schematic update completes, perform forward annotation to PADS Layout and create layout.
8. Import the layout to the PADS I/O Designer FPGA database / layout database.
9. Optimize assignments using PADS I/O Designer's [Unravel Nets](#) functionality.
10. Run Export - [Schematic Update](#) to update net connections on the schematic according to the optimized assignments in PADS I/O Designer.

Schematic Update Flow for PADS Designer (PCB First)

The following steps describe how to create an FPGA component database in PADS I/O Designer for use with [Schematic Update](#) to optimize connections for a component which has already been placed on a PADS Designer schematic and used in the layout.

1. Open the existing project in PADS I/O Designer.

2. Build an PADS I/O Designer database, including signals and pin assignments related to the component, from the schematic:
 - a. Use the PCB Design Wizard option “Import signals and assignments only” to import signals and assignments into PADS I/O Designer from the schematic without importing the symbol.
 - b. Use HDL and/or Pin Report File for the specific device.
 - c. Use a spreadsheet file with signals and assignments.
3. Ensure proper PCB Part Number and Ref Des settings in Database Properties - PCB Flow tab. PCB Part Number and Ref Des are used to match components on the schematic.
4. Import the layout to the PADS I/O Designer FPGA database / layout database.
5. Optimize assignments using PADS I/O Designer's [Unravel Nets](#) functionality.
6. Run Export - [Schematic Update](#) to update net connections on the schematic according to the optimized assignments in PADS I/O Designer.

Chapter 5

Working with Projects and Databases


What is a Project?

After PADS I/O Designer launches, the first step is to open or create a project. A project is an environment in which you can work with multiple databases which form part of the same design.

The project file (*.prj*) should be created within the Mentor flow tool, for example PADS Designer, and this project file is then opened in PADS I/O Designer (see [“Opening a Project”](#) on page 58). If you are using the PADS Designer PADS flow, projects can be created within PADS I/O Designer first (see [“Creating a New Project”](#) on page 58), and then opened within PADS Designer and used as normal. For other flows, if you create the project in PADS I/O Designer first, it will not be able to connect to the iCDB and you will be unable to work interactively.

An PADS I/O Designer (for PADS Designer flows) project can contain one or more “Designs” which correspond to the top-level design or designs in the Mentor flow tool project. When working in PADS I/O Designer, databases are created within the project, and are associated with a particular top-level design in PADS Designer. (For WG flows, only one design per project is allowed.)

Project Window


The Project Window allows you to organize databases for a board in the project environment. All database files within a project are listed by board with their path and Reference Designator (if applicable). Each board in the project can be selected using the drop-down box. The Project Window can be closed and opened at any time, either using the menu item **View > Windows > Project**, or the  button on the view toolbar.

Right-clicking in the Project Window pops up a menu of commands allowing you to create a new project, open or close an existing project, add new boards to the project, or add new or existing databases to the current project.

Related Topics

- [“What is a Project?”](#) on page 57
- [“Creating a New Project”](#) on page 58.
- [“Adding a Board to a Project”](#) on page 59.
- [“Adding Databases to a Board”](#) on page 60

Opening a Project

1. Open the [Project Window](#) by selecting **View > Windows > Project**, or click the  button on the view toolbar.
2. Do one of the following:
 - Select **File > Open Project**.
 - Right-click in the [Project Window](#) and select **Open Project**.
3. Browse to the existing project file (.prj), select it and click **Open**.

The project opens in the [Project Window](#). Top-level designs can be selected from the drop-down box, and any associated PADS I/O Designer databases are listed in the window. Double-click on a database in the list to open it.

Creating a New Project


Projects can be created within PADS I/O Designer and then opened with the appropriate tool in your design flow and used as normal.

Note



For the procedure on creating a new project in Cadence OrCAD®, see “[Creating a Project in Cadence OrCAD® Flow](#)” on page 59.

To create a new project:

1. Open the [Project Window](#) by selecting **View > Windows > Project**, or click the  button on the view toolbar.
2. Do one of the following:
 - Select **File > New Project**.
 - Right-click in the [Project Window](#) and select **New Project**.

The **New Project** dialog is displayed.

3. In the **Project Name** field, enter a name for the project. PADS I/O Designer automatically suggests a project folder with the same name in the Project Location field. If this is acceptable, skip the next step.
4. In the **Project Location** field, enter the path to the location to which the project should be saved (or click **Browse** to navigate there) and enter a name for the project.

A project file (.prj) will be saved in that location.

5. Click **OK** to display the **New Board** dialog.


6. In the **Board Name** field, enter a name for the board.
7. Click **OK**.

An empty project is created in the [Project Window](#). You can now add boards to the project, and then add PADS I/O Designer databases to each board.

- See “[Adding a Board to a Project](#)” on page 59
- See “[Adding Databases to a Board](#)” on page 60

Creating a Project in Cadence OrCAD® Flow

To create an PADS I/O Designer project in the Cadence OrCAD® EDIF flow:

1. Open the [Project Window](#) by selecting **View > Windows > Project**, or click the  button on the view toolbar.
2. Do one of the following:
 - Select **File > New Project**.
 - Right-click in the [Project Window](#) and select **New Project**.

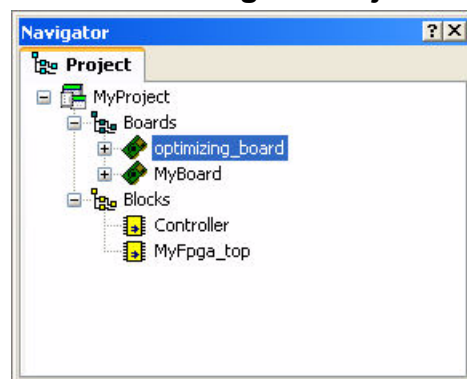
The **Choose a filename to save project** dialog is display. Select location of *.prj* file.

3. Click **Save** to display the New Board dialog.
4. In the **Board Name** field, enter a name for the board.
5. Click **OK**

Adding a Board to a Project

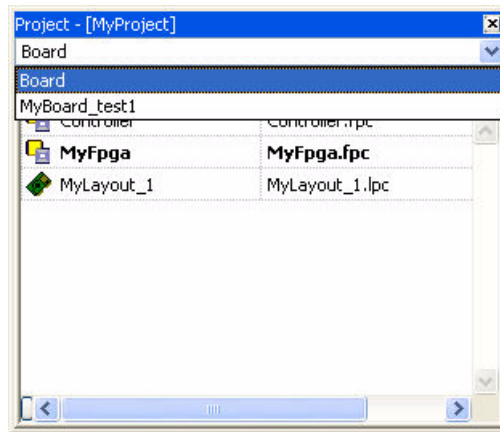
In PADS Designer, a project may contain a number of designs, which correspond to different PCB boards in the project. [Figure 5-1](#) shows the PADS Designer Project Navigator where boards are listed underneath the project name (in this case: *MyProject*).

Figure 5-1. PADS Designer Project Navigator



When opening a PADS Designer project in PADS I/O Designer, you can select each top-level design using the drop-down box on the [Project Window](#) as shown in [Figure 5-2](#). When a board is selected, all of the PADS I/O Designer components residing within it are listed in the [Project Window](#).

Figure 5-2. Selecting a Board in the PADS I/O Designer Project Window



New boards can be added to the Project in PADS I/O Designer using the following procedure:

1. Do one of the following to add a new board to the project:
 - Select **File > Add New Board**.
 - Right-click in the [Project Window](#) and select **Add New Board**.
2. Enter a name for the new board and click **OK**.

Adding Databases to a Board

In PADS I/O Designer you create a database file for each FPGA device, and a [Layout Database](#) file in which to perform layout optimization:

- FPGA Database (*.fpc*)
See “[FPGA Database Wizard](#)” on page 65.
- Layout Database (*.lpc*)
See “[Optimizing the I/O Assignments](#)” on page 121.

When multiple databases are open, the workspace contains multiple tabs to display concurrently opened databases. The database icon on each tab indicates the database type (FPGA, or layout).



Databases are created by adding them to designs that reside within a project. Once you have added a database to a board, you can begin building or modifying that component or layout.

You can use the Database Wizard to add a new FPGA database to a board, by doing one of the following:

- Select **File > Add to Board > New FPGA from Wizard**.
- Right-click on the [Project Window](#) and select **Add to Board > New FPGA from Wizard**.

See “[FPGA Database Wizard](#)” on page 65.

You can also add a new database to your project without using the wizard by doing one of the following:

- Select **File > Add to Board > New FPGA**, or **New Layout**.
- Right-click on the [Project Window](#) and select **Add to Board > New FPGA**, or **New Layout**.

The **Save As** dialog appears. Enter a name for the new database and save it to your working directory.

The new database opens and is listed under the design in the [Project Window](#). A newly created database contains neither signals nor symbols, but a default FPGA (last used) device is chosen.

You can add existing databases to the project by doing one of the following:

- Select **File > Add to Board > Existing Database**.
- Right-click on the [Project Window](#) and select **Add to Board > Existing Database**.

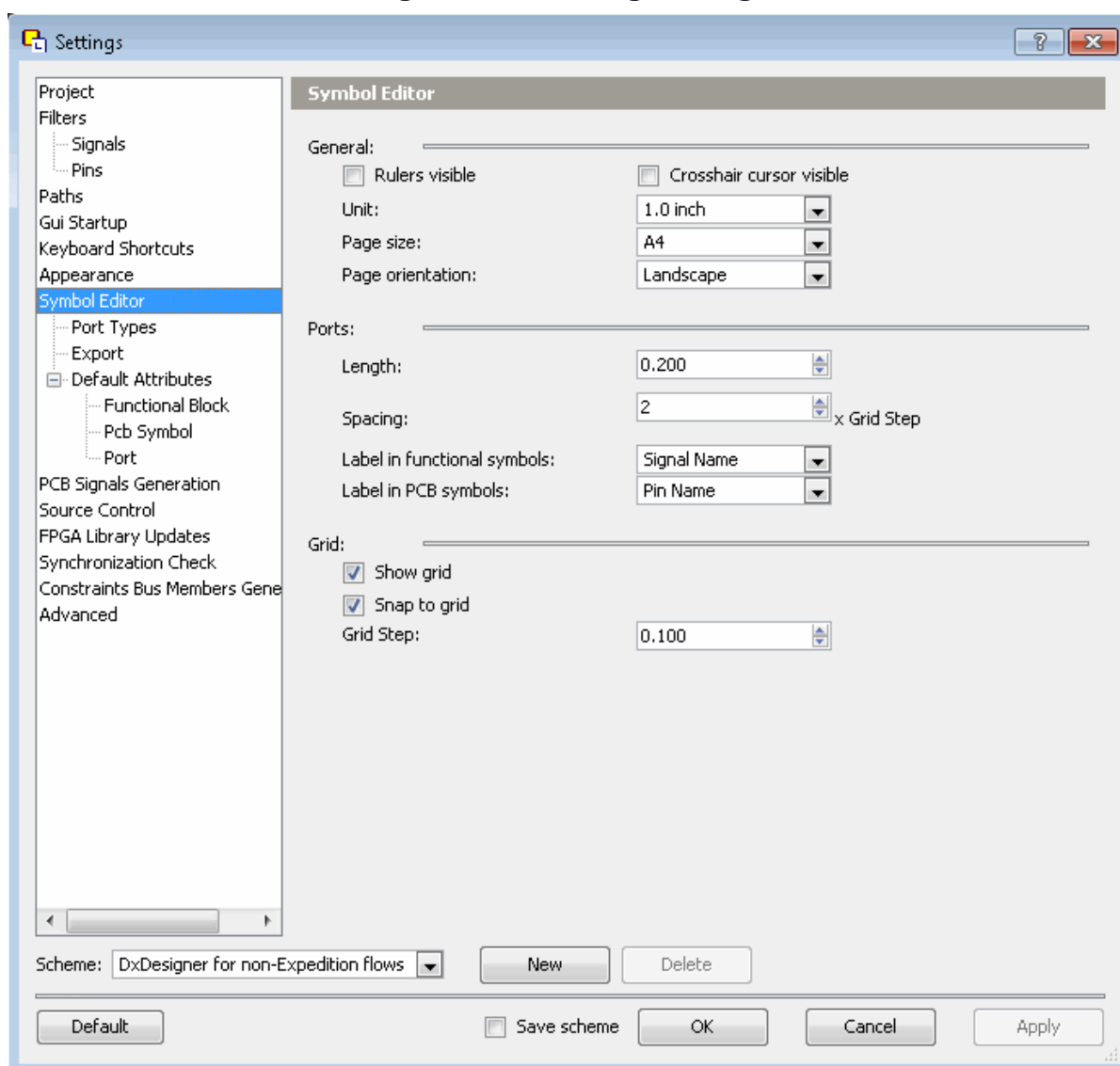
Related Topics

- “[What is a Project?](#)” on page 57
- “[Adding a Board to a Project](#)” on page 59
- “[Optimizing the I/O Assignments](#)” on page 121

Viewing Settings

To view and edit project settings, click **Setup > Settings** and select the appropriate node in the left pane. The right pane displays the current settings for the selected node (see [Figure 5-3](#)).

Figure 5-3. Viewing Settings



Related Topics

- For Project, see [“Settings Dialog Box”](#) on page 227.
- For Filters, see [“Regular Expression Filters”](#) on page 173.
- For Paths, see [“Environment Variables”](#) on page 15.
- For GUI Startup, see [“GUI Startup Preferences”](#) on page 50.
- For Appearance, see [“Changing the Appearance of Elements”](#) on page 51.
- For Symbol Editor, see [“Symbol Window Settings”](#) on page 27.
- For Source Control, see [“Source Control Settings”](#) on page 51.

- For Synchronization Check, see “[Synchronization Check](#)” on page 50.
- For Advanced, see “[Advanced Settings](#)” on page 52.
- For Constraints Bus Members Generation, see “[Constraints Bus Members Generation](#)” on page 149.
- For Keyboard Shortcuts, see “[Keyboard Shortcuts](#)” on page 177.
- For PCB Signals Generation, see “[PCB Signals Generation](#)” on page 110.
- For FPGA Library Updates, see “[FPGA Library Update Notifications](#)” on page 68.

Chapter 6

Adding an FPGA to the Design

This chapter describes in detail the steps necessary for building an FPGA device database.

FPGA Database Wizard

The FPGA database wizard guides you through the steps necessary in order to specify the information required to build a new FPGA database in PADS I/O Designer.

1. Do one of the following to invoke the FPGA database wizard:
 - Select **File > Add to Board > New FPGA from Wizard**.
 - Right-click in the project window and select **Add to Board > New FPGA from Wizard**.
2. Specify a name (for example, *MyFpga*) and location for the new FPGA database and click **Save**. PADS I/O Designer displays the Start page.
3. Click **Next** to display the Signals Source page.
4. Specify the type of source file and the file path:
 - **VHDL:** This is the file that is analyzed by PADS I/O Designer and from which signals are read. If the file contains multiple top-level units, you are asked to choose one of them.
 - **Verilog:** If you have a Verilog file that references some external files, you are asked to enter the directory in which to search for those additional files. See “[Additional HDL Files](#)” on page 70 for more information.
 - **Spreadsheet:** If you specify this option, you are asked to specify the type of delimiter.
5. Click **Next**.
6. If necessary, specify additional source file data and click **Next** to display the Vendor and Device page.
7. Specify vendor, device, and default I/O standards for use in the new database.

You can select one of over 5000 devices from a library that is kept current with those of the FPGA vendors.
8. Click **Next** to display the Place and Route page.

9. Specify the Place & Route constraints file and the Place & Route pin report file. You can also specify additional data exchange files on this page.

See “[Place and Route Constraints Files](#)” on page 135 and “[Pin Report Files](#)” on page 136.

10. Click **Next** to display the Synthesis page.
11. Specify the tool name from the drop-down list and specify the Synthesis constraints file. See “[Synthesis Constraints Files](#)” on page 134.
12. Click **Next** to display the PCB Flow page.
13. Specify a unique part number in the **Part number** field.

Note



For OrCAD flows, specify additional information (**EDIF file**).

14. Specify a cell name.
15. Click **Next** to display the Finish page. This page provides a summary of the settings as shown in [Figure 6-1](#).
16. Click **Finish** to close the wizard.

Figure 6-1. Summary of FPGA Wizard Settings

```
Database path: C:\MentorGraphics\WDIR\MyProject\MyFpga.fpc

Signals Source:
Type: VHDL
Path: C:\MentorGraphics\WDIR\Workshop\HDL\Controller.vhd
Entity: Controller
Additional files: none

FPGA device:
Vendor: xilinx
Tool/Library: ISE 11.4
Family: acr2
Device: xa2c128
Package: vq100
Speed: -7

Place and route:
Constraints file name: C:\MentorGraphics\WDIR\Workshop\Place and Route\imageproc.ucf
Pin report file name: C:\MentorGraphics\WDIR\Workshop\Place and Route\imageproc.pad

Synthesis:
Tool: Xilinx Synthesis
Constraints file name: C:\MentorGraphics\WDIR\HPotter.xcf

PCB Flow:
Part number: harrypotter
Cell name: 1X680_BC_B0
```

To make changes to information entered here, click **File > Database Properties**, and make changes in the Database Properties dialog.

Related Topics

- “[FPGA Device Library](#)” on page 68

- [“Place and Route Constraints Files”](#) on page 135
- [“Pin Report Files”](#) on page 136

FPGA Device Setup

Create a new project containing a new FPGA database as described in [“Creating a New Project”](#) on page 58 and [“Adding Databases to a Board”](#) on page 60.

Once you have added a new FPGA database to a project, you can begin to build the FPGA either using the [FPGA Database Wizard](#) or by doing the following:

- [Selecting the Device](#)
- [Importing FPGA Vendor Files](#)

Selecting the Device

1. Select **File > Database Properties** and click the **Vendor and Device** page.
2. Specify the vendor from the drop-down box.
3. The other fields in the Device section are populated with the options available, allowing you to choose the device from a particular tool/library and family, and to specify a package and speed.

Importing FPGA Vendor Files

FPGA vendor files can be imported directly into PADS I/O Designer either during setup of a new FPGA database using the [FPGA Database Wizard](#), or using the [Database Properties Dialog Box](#).

1. Right-click on an FPGA database in the [Project Window](#), select **Database Properties** and click the **Place and route** page.
2. Specify FPGA vendor files such as [Place and Route Constraints Files](#) or [Pin Report Files](#).

FPGA vendor files typically have a pin number assigned and the I/O Standard assigned. These two items do not exist in the HDL file. PADS I/O Designer re-assigns the existing signals to the new pins assignments in the vendor file as well as making the I/O Standard assignment.

Related Topics

- [“Place and Route Constraints Files”](#) on page 135
- [“Pin Report Files”](#) on page 136

FPGA Device Library

The PADS I/O Designer built-in library contains information about each FPGA device and the rules that apply to it. Mentor Graphics receives this data from the FPGA vendors either directly or by using an automated process which translates that data directly from the FPGA vendor library formats.

Vendor Library Updates

In the case where these mechanisms fail to deliver the complete set of data, a request for more data will be discussed with the appropriate FPGA vendor through the partnership channel that Mentor Graphics has with all supported FPGA vendors. In some cases this data can be delivered quickly, while in other cases it takes some time for the FPGA vendors to make this data available.

Since the PADS I/O Designer device library has a one-to-one relationship with the same version of the FPGA vendor tool, any errors that were introduced in the FPGA vendor library will also exist in the PADS I/O Designer library unless they were found during the translation process. As soon as the FPGA vendor fixes their errors they will automatically make their way into the PADS I/O Designer device libraries. This could be as soon as the FPGA vendor releases their new software since Mentor Graphics is planning to make separate library updates for the PADS I/O Designer device libraries available through SupportNet more frequently than the major software releases.

PADS I/O Designer cannot guarantee that after changes have been made to the pin assignment the design:

- Is still routable due to the available FPGA surface given the new re-located signals. For example it could be difficult to route an FPGA for which 90% of the surface is used with a pin assignment that is not optimal for routability.
- Meets their timing requirements since the pin assignment also influences the timing paths and therefore the actual timing. So when a design is timing critical it could be that a new pin assignment will violate certain timing requirements.
- Meets all the I/O Design rules. The PADS I/O Designer built-in library with device and rule information is used on-the-fly to test the design. However, Mentor Graphics is dependent on the FPGA vendors for these rules and in some cases it takes time before the data becomes available.

FPGA Library Update Notifications

PADS I/O Designer has the capability to keep you informed of updates to your FPGA library, and automatically install updates when they become available. For more information on setting up and using this capability, see [“FPGA Updater”](#) on page 151.

Preliminary Device Support

Preliminary devices can be added to the library on request. They are not added by default. Preliminary devices reside in the installation library along with regular devices, and when you update the library with a new version, preliminary parts can be overwritten with released data.

To distinguish preliminary parts from released parts, each preliminary part is prefixed with the word “Preliminary” in the PADS I/O Designer Database Properties window, Device window, and Status bar.

Design Rule Check

It is recommended that you check the I/O design against any of the violations mentioned above by running it through the FPGA vendor tools before proceeding to the PCB process. PADS I/O Designer offers two ways to achieve these checks:

Note



PADS I/O Designer DRC supports Altera, but does not work with Xilinx.

1. After generating the Place and Route constraints, file the DRC utility from the chosen FPGA vendor can be run from within PADS I/O Designer through the **Tools > Design Rule Check** menu. This ensures that the design is not violating any of the I/O design rules.
2. When the Place and Route constraints file has been generated, continue the regular FPGA process by taking the design including the Place and Route constraints file through the FPGA vendor tool flow to implement the device. This way the design can be verified against any risks of not being able to implement the design.

Note



In both steps, the chosen FPGA vendor tools have to be installed in order to complete the checks.

Related Topics

- [“Preferred Devices List”](#) on page 245
- [“Vendor Support Information”](#) on page 247
- [“Device-dependant Assignment Rules”](#) on page 251
- [“Generic IO Standards”](#) on page 257

Defining I/O Signals

I/O signals and signal assignments can be defined in an FPGA database in the following ways:

- “[Importing I/O Signals Using HDL](#)” on page 70
- “[Importing I/O Signals Using a Spreadsheet](#)” on page 71

I/O signals can also be created manually within PADS I/O Designer:

- “[Creating a New Signal or Signal Bus](#)” on page 72

Importing I/O Signals Using HDL

PADS I/O Designer includes integrated VHDL, Verilog, EDIF and XML parsers. The following procedure describes how to import I/O signals into a database from source files in those formats using the [Database Properties Dialog Box](#):

1. Select **File > Database Properties** to display the [Database Properties Dialog Box](#).
2. Select the **Signals source** page.
3. Select VHDL or Verilog.
4. Specify the path to the file in the **File path** field. You can enter the path manually, or click **Browse** to navigate to the file’s location.
5. Go to the VHDL definition page.
6. Select the required entity, module or cell in the **Entity** dropdown box.
(Optional step) If appropriate, specify additional VHDL file(s) or Verilog search path(s) in the box below. See “[Additional HDL Files](#)” on page 70 for more information.
7. Click **OK**.
8. Open the [Synchronization Wizard](#).
9. Click **Next** and **Finish** on the [Synchronization Wizard](#) to import the data from the specified file.

The [Signals List](#) will be populated with the signals read from the source file.



Note

If the file contains errors that prevent the signals from being read, then the errors found by the integrated parser are displayed. To continue reading in signals from the file, the errors have to be corrected first.

Additional HDL Files

Before the file containing the entity declaration is analyzed, it is sometimes necessary to analyze additional files in advance. This may be the case if, for example, the selected HDL file uses some constants declared in packages. PADS I/O Designer allows parsing any number of additional HDL files before the main one.

The VHDL definition page on the [Database Properties Dialog Box](#) contains the list of these additional HDL files.

1. To add a file to the list, use the **Add File** button located on the right side of the list.
2. To remove a file from the list, use the **Remove Selected** button.
3. Files are processed in the same order as presented on the list. To move a file up or down the list, click on it, and use the **Move Up** or **Move Down** button.

Since PADS I/O Designer installation already includes standard VHDL libraries (IEEE and STANDARD), there is no need to manually add these files for analysis.

Note that the main VHDL file is analyzed at the very end, after all of the additional files have been analyzed. So for example, if you have multiple entity architectures in separate files, the architecture that you want to use in PADS I/O Designer needs to be analyzed at the very end. To do this, you need to set the architecture file as the main file.

HDL Names and PCB Names

In the [Signals List](#), each signal has an HDL Name and PCB Name. Upon loading an HDL netlist, both names assume the HDL name. The PCB Name can be used as the pin name during symbol creation by specifying **Signal Name** as the **Port label** for the PCB symbol in the [Symbols Generator](#).

PCB names and HDL names can be changed by doing one of the following:

- Select the signal in the [Signals List](#) and edit the **PCB Name** or **HDL Name** fields in the [Properties Window](#). DIFF children are also renamed.
- **Right-click** in the [Signals List](#) and select rename. A dialog appears into which the new PCB name and/or HDL name can be entered.

A checkbox can be used to select whether DIFF children are also renamed.

Changing the PCB name of a signal only affects the PCB names of its children. Similarly, changing the HDL name of a signal only affects the HDL names of its children.

Importing I/O Signals Using a Spreadsheet

The following procedure describes how to use the [Settings Dialog Box](#) to import I/O signals and signal assignments into a database from a file with comma delimited values.

1. Select **File > Database Properties** to display the [Database Properties Dialog Box](#).
2. Select the **Signal source** page.
3. Select **Spreadsheet** as the signal source.

4. Specify the path to the spreadsheet file in the **File Path** field. You can enter the path manually, or click **Browse** to navigate to the file's location.
5. Select the Spreadsheet definition page, and enter a character in the **Delimiter** field (for example, a comma) which defines the start of each entry.
6. In the **Column assignments** window, the numbering should match the spreadsheet column numbers where I/O Designer can locate information for items such as Signal name, Signal type, Signal direction, and so on. Use the column index column to move the assignments up or down in the list.
7. Click **OK**.
8. Open the [Synchronization Wizard](#).
9. Click **Next** and **Finish** on the [Synchronization Wizard](#) to import the data from the specified file.

The [Signals List](#) will be populated with the signals listed in the spreadsheet.

Creating and Editing Signals in the Signals List

In an FPGA database, signals can be added, removed and renamed in the [Signals List](#) using the Add Signal dialog.

- “[Creating a New Signal or Signal Bus](#)” on page 72
- “[Removing Signals](#)” on page 73
- “[Renaming Signals](#)” on page 73
- “[Combining Signals into a Bus Signal](#)” on page 73

Creating a New Signal or Signal Bus

Use the following procedure to create a new signal and add it to a database in the [Signals List](#):

1. Do one of the following to display the Add Signal dialog:
 - a. Right-click in the [Signals List](#) and select **Add Signal...**
 - b. Select the menu item **Edit > Add Signal...**
2. Enter a name for the signal in the **Name** field.

Caution



Signal names must not start with a hyphen (-) and cannot contain any of the following characters: square brackets ([]), parentheses (()), angle brackets (< >), commas (,), tabs, or spaces.

3. Select the signal's direction (In, Out, Inout, Buffer or Linkage) in the **Direction** field.
4. Select the signal's type in the **Type** field.
5. Enter an I/O standard if you don't want the default value.
6. Specify a range of values in order to create a signal bus (*optional*).

Removing Signals

Use the following procedure to remove a signal from a database:

1. From the [Signals List](#), select the signal or signals to be removed and do one of the following:
 - Right-click on the signal or signal selection in the [Signals List](#) and select **Remove Signals**.
 - Select the menu item **Edit > Remove Signals**.

The signal(s) will be removed from the [Signals List](#).

Renaming Signals

To change the name of a signal in a database, use the following procedure:



Caution

Signal names must not start with a hyphen (-) and cannot contain any of the following characters: square brackets ([]), parentheses (()), angle brackets (< >), commas (,), tabs, or spaces.

1. From the [Signals List](#), select the signal to be renamed and do one of the following:
 - Right-click in the [Signals List](#), select **Rename** and enter the new name for the signal in the dialog box. You can also use the dialog box to change the bus range.
 - Select the menu item **Edit > Rename** and enter the new name for the signal in the dialog box.
 - Enter the new name for the signal in the [Properties Window](#).

Combining Signals into a Bus Signal

In the [Signals List](#), signals can be grouped into buses, known as *bus signals*. Bus signals are created automatically for vectors read from an HDL file. Additionally, user-defined bus signals may be created at any time. To create a bus, use the following procedure:

1. In the [Signals List](#), select the signals which will form the bus.

2. Right-click on the selection and select **Combine**.
3. Enter a PCB name and an HDL name for the bus signal. Optionally, you can use the controls on this dialog to adjust the range of signals in your bus.

A bus signal is created and can be identified in the [Signals List](#) by the plus sign at the left of the bus name.

Click the plus sign to expand or collapse the bus in the list, displaying or hiding its elements. Alternatively, select **View > Expand**, or **View > Collapse** to do the same thing. To expand or collapse all buses together, select **View > Expand All**, or **View > Collapse All**.

Note

Busess cannot be nested - bus signals cannot be combined into another bus.

A bus can be split or replaced with the flat list of its elements. To split a bus right-click on it and select **Split**. To split all buses at once, right-click in the List Window and select **Split All**.

Bus signals can be renamed in the same way as other signals. See “[Renaming Signals](#)” on page 73.

Related Topics

- “[Combining Pins into a Bus Pin](#)” on page 85
- “[HDL Names and PCB Names](#)” on page 71

Differential Pairs

To create a differential signal pair from existing signals, select the required signals in the [Signals List](#), right-click and choose **Create Differential Pair**.

To split a differential pair back into separate signals, right-click the selection and choose **Split Differential Pair**.

When adding signals to the [Signals List](#) manually, you can specify that they are a differential signal pair by selecting the signal type as **DIFF** in the **Type** field on the Add Signal dialog.

In an FPGA database, differential signal assignments assign the whole DIFF signal, and member assignment is done automatically.

Differential signals are displayed as shown in [Figure 6-2](#).

Figure 6-2. Differential Pairs Displayed in the Signals Window

PCB Name	HDL Name	Dir	Pin	Type
▶ CLK	CLK	In	P27	IO
☐ ▶ DIFF_BUS<0:3>	DIFF_BUS<0:3>	In		DIFF
☐ ▶ DIFF_BUS<0>	DIFF_BUS<0>	In		DIFF
▶ DIFF_BUS_P<0>	DIFF_BUS_P<0>	In		DIFF
▶ DIFF_BUS_N<0>	DIFF_BUS_N<0>	In		DIFF
⊕ ▶ DIFF_BUS<1>	DIFF_BUS<1>	In		DIFF
⊕ ▶ DIFF_BUS<2>	DIFF_BUS<2>	In		DIFF
⊕ ▶ DIFF_BUS<3>	DIFF_BUS<3>	In		DIFF
☐ ▶ RIGHT	RIGHT	In	AH27,AG28	DIFF
▶ RIGHT_P	RIGHT_P	In	AH27	DIFF
▶ RIGHT_N	RIGHT_N	In	AG28	DIFF
▶ STOP	STOP	In	W22	IO

PCB Signal Assignments

PADS I/O Designer supports PCB signal assignments. Special signals, for example ANALOG_GND, may be assigned to pins from the PCB page of the [Pins List](#). PADS I/O Designer allows you to add and use PCB signals directly in the [Signals List](#). You can distinguish HDL signals from PCB signals by their icons.

For automatic PCB Signal assignment, you can use [Update Power Signals](#).

Set the I/O Standard

FPGAs are very flexible and PADS I/O Designer fully supports the variety of I/O standards available. To change a signal's I/O standard (IOS), select the signal in the [Signals List](#), and then in the [Properties Window](#), you can select the IOS from the drop-down list.

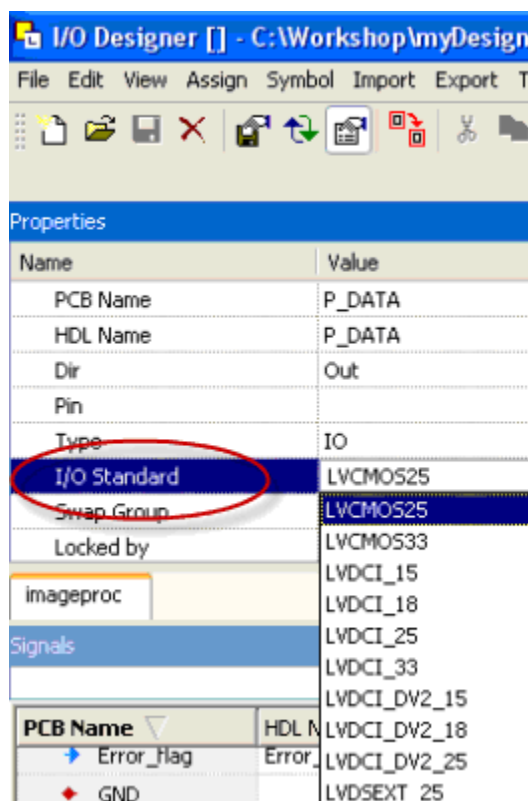
Note



PADS I/O Designer supports a default IOS. You can change the default IOS in the database settings. You must not leave the IOS field empty, with the exception of MGT signals, whose IOS field can be left empty.

The example in [Figure 6-3](#) shows the bus PDATA[0:63] being assigned an I/O standard of LVCMOS25. The “25” indicates this is a 2.5 Volt signal and will require that the Bank supplies be set to 2.5 Volts typically using the VCCO pins.

Figure 6-3. Setting the I/O Standard



When a signal is assigned a specific voltage, the entire bank is affected and unused pins are changed to the same I/O standard. A conflicting voltage level or I/O standard cannot be applied to that bank. As part of the I/O Standard assignment, PADS I/O Designer will also manage the power assignment for any affected banks. PADS I/O Designer creates a new power signal for the voltage and assigns it to the VCCO pins on the affected banks. In this example, the signal name would be V_2_5, but you can configure the signal names from **Setup > Settings + PCB Signal Generation**, as described in “[PCB Signals Generation](#)” on page 110.

Signal Locking

Locking signals is useful when multiple designers are sharing a database, or when you want to apply security to a signal. PADS I/O Designer does not allow you to delete, change the pin assignment, or change any other attribute of a locked signal. The Signal locking feature does not depend on any source control system. The locked/unlocked state of a signal is stored in the database.

Note

Locking signals does not prevent the import of a new signal definition, such as from HDL, from removing a signal.

To lock a signal or signal selection, right-click on it in the [Signals List](#), and choose **Lock Signals**.

To unlock a signal or signal selection, right-click on it in the [Signals List](#), and choose **Unlock Signals**.

A small padlock icon in the pin/signal list identifies locked signals.

Pin Locking

Locking pins is useful when multiple designers are sharing a database, and when you want to apply security to a pin. PADS I/O Designer does not allow you to change the pin assignment or any other attribute of a locked pin. The Pin locking feature does not depend on any source control system. The locked/unlocked state of a pin is stored in the database.

To lock a pin or pin selection, right-click on it in the [Pins List](#), and choose **Lock Pins**.

To unlock a pin or pin selection, right-click on it in the [Pins List](#), and choose **Unlock Pins**.

A small padlock icon in the pin/signal list identifies locked pins.

Design for Multiple Devices in a Common Package

PADS I/O Designer supports designing for multiple devices, allowing you to switch to a larger or smaller device without changing the pin assignment or PCB routing. This is accomplished by overlaying multiple devices typically in a common package and enabling only those pins that are common. Pin assignments are therefore limited to only those common pins. This provides the flexibility to swap devices while maintaining I/O assignments and leaving the PCB unchanged.

PADS I/O Designer provides a possibility to show only pins common to the current device and some other devices. To turn it on, select **View > Pins from Other Devices**. The displayed window presents the list of devices from the current vendor with the same package as the current device. The list can be easily modified by the use of the **Add** and **Remove** buttons.

If the option is turned on by the OK button, only pins which have the same functions in the current device and all selected devices are displayed. For instance, if a pin is I/O in the current device, and the pin in the same position is *not connected* in one of the devices from the list, the pin is not displayed. In this way the option helps you to plan the design for different parts.

To access pins hidden by the **Pins from Other Devices** option select **View > Show Hidden Rows**. If that setting is checked, all rows are displayed. The rows that are hidden will be displayed in a gray color.

PCB Design Wizard

Used to import symbols with signals and pins into an FPGA database directly from the design.

1. In an FPGA database, select **Import > PCB Design Wizard**. The [Import Design Wizard Dialog Box](#) is displayed.
2. Enter the path to the project file containing the required design in the **Project Path:** field, or click **Browse** to navigate to the file's location.
3. In the **Design Name** field, select the top-level design containing the required component.

(Optional Step) Enter the Reference Designator for the required component in the **Ref Des:** field and click **Next**.

4. Select the required symbol from the list.

You can use [Regular Expression Filters](#) to filter the list of symbols. Click the ... button for a list of popular filters. Click **Apply** to apply the filter to the list.

(Optional Step) If you want to **Import signals and assignments only**, select the checkbox for this option, located beneath the list of symbols. This will not import the symbol. This option can be used with [Schematic Update](#) in order to preserve existing symbols.

(Optional Step) If you want to **Generate Functional Blocks for PCB Symbols**, select the checkbox for this option, located beneath the list of symbols.

5. Click **Next**.
6. Specify information about the FPGA device by selecting the options from the drop-down boxes. Click **Next**.
7. Verify that the signal mapping from the imported PCB signals and any existing HDL signals found in the database is correct and click **Finish**.

Chapter 7

User-defined Rules

PADS I/O Designer provides a mechanism for defining and managing user-specified rules for an FPGA device, which can supplement the vendor device rules. Every rule can contain a set of simple conditions or “primitives” concatenated by logical operators. Conditions can be set for each primitive such that the rule is executed when the primitive is, is not, contains, does not contain, matches or does not match the value(s) specified.


This section describes how to create rules see “[Creating a New Rule](#)” on page 79, and then refer to the following example:

- “[Setting a Rule for an FPGA Device](#)” on page 80

Creating a New Rule

1. Select **Tools > Rule Editor**.

The [Rule Editor Dialog Box](#) is displayed.

2. Click the New Rule  icon. The [Rules Wizard](#) is displayed.

3. Enter a name for the new rule in the **Name** field.

4. Enter a description for the rule in the **Description** field.

5. In the **Failure answer** field, enter some text that will be returned if the attempted assignment breaks this rule.

6. Click **Next**.

7. Set whether the rule will allow or forbid placements.

This information appears in the window at the bottom of the [Rules Wizard](#).

8. Click **Next**.

The next pages allow primitives and conditions for signals and pin conditions to be defined.

9. Double-click the required primitive or use the arrow controls to move it from the **Available Primitives:** list into the **Rule Primitives:** list.

The [Edit Primitive Value\(s\) Dialog Box](#) is displayed.

10. In the **Operator:** field, select the required operator for the primitive.

The rule is executed when the primitive **is, is not, contains, does not contain, matches, or does not match** the values defined in the **Operand:** field or selected from the list of available values (present depending upon the type of operator).

11. Enter value(s) into the **Operand:** field or select them from the list of available values (present depending upon the type of operator). This field is case sensitive.

[Regular Expression Filters](#) may be used to specify values. Clicking the arrow next to the **Operand:** field gives suggestions of commonly used regular expression filters.

12. Click **OK**.

Primitive details are added to the window at the bottom of the [Rules Wizard](#).

13. Click Add Term to make more rule assignments for a different set of signals (*optional*).

Note




When setting “Allow” rules, any signals or pins which do not qualify the rule will be disallowed by default. If you want all signals and pins to be usable, ensure that you set additional terms to allow those signals and pins to be assigned. This is demonstrated in “[Setting a Rule for an FPGA Device](#)” on page 80.

14. Repeat step 10 to add further primitives to the rule if required.
15. Click **Next**.
16. Click **Finish**.

The rule is created and will be present in the Rules: field in the [Rule Editor Dialog Box](#).

Example 7-1. Setting a Rule for an FPGA Device

The following example creates locate rules for an FPGA device. The intent of the rule is to locate the ALU busses and strobe to banks 6 and 10 in the top left quadrant of the Xilinx Virtex 4 part.

1. In an FPGA database, select **Tools > Rule Editor** to open the Rule Manager dialog.
2. Click the **New Rule**  icon to create a new rule.

The **Set rule name and description** page allows you to enter a name for the new rule in the **Name** field, a description for the rule in the **Description** field, and in the **Failure answer** field, you can enter some text that will be returned if the attempted assignment breaks this rule.

3. Click **Next**.
4. On the **Rule scope** page, set the following primitives by double-clicking them from the list of **Available Primitives**:
 - Vendor name is **Xilinx**

- Family name is **Virtex 4**
 - Device name is **4vfx40**
 - Package name is **ff1152**
5. Click **Next**.
 6. Select **Allow assignments**.
 7. On the **Pins condition** page, set the following primitives by double-clicking them from the list of **Available Primitives**:
 - Pin type is **IO**
 - Pin bank is **6** or **10**.
 8. On the **Signal conditions** page, double-click **Signal Name** from the list of **Available Primitives**. Set the primitive to:
 - Signal Name is **^ALU.***
- This sets the primitive to use all signals that start with “ALU”.
9. The final step is to add an additional term that states “do nothing” for the remaining signals. This in effect “releases” the remaining signals to be assigned as needed.
 - a. Click **Add Term**.
 - b. On the **Signal conditions** page, double-click **Signal Name** from the list of **Available Primitives**.
 - c. Set the primitive to: Signal Name not matches **^ALU.***

This adds a term to the rule to allow use of signals that don’t start with “ALU”.

Additional terms can also be set up to make more rule assignments for a different set of signals.

Chapter 8

Making Pin Assignments


Assigning Signals to Pins

Use the following procedure to assign all unassigned signals to available pins in one step:

1. Ensure that no signals are selected in the [Signals List](#).

If no signal is selected, then PADS I/O Designer will attempt to assign all signals in the list or those which are marked to assign (see “[Mark to Assign](#)” on page 83).

2. To assign all signals to available pins, do one of the following:

- Select **Assign > Assign Pins** or click the  icon on the toolbar.

Signals are assigned to available pins, preserving any existing assignments.

- Select **Assign > Assign Pins with Overwrite** or click the  icon on the toolbar.

Signals are assigned to available pins, overwriting any existing assignments.

Signals are assigned in the same order that they are listed in the [Signals List](#) to the pins in the order listed in the [Pins List](#). If the number of signals differs from the number of pins, the remaining signals or pins are ignored.

Mark to Assign

The Mark to Assign command is used to make a selection of signals or pins and assign just those signals or pins when **Assign > Assign Pins** or **Assign > Assign Pins with Overwrite** is chosen. To mark signals or pins for assignment, use the following procedure:

1. Select the required signals or pins in the [Signals List](#), the [Pins List](#) or the [Device Window](#).
2. Right-click on the selection and choose **Mark to Assign**.

Marked signals are displayed in the [Signals List](#) and [Pins List](#) with an animated boundary. In the [Device Window](#) marked pins blink continuously. Marked signals or pins remain marked until the assignment is made.

Marked signals or pins can be unmarked by right-clicking on the selection and choosing **Remove Marks**.

Assigning Signals to Pins Using Drag and Drop

To assign signals to pins with drag and drop, use the following procedure:

1. Select the signals that are to be assigned in the Signal List. Selecting a bus will assign pins to all bus elements.
2. Click the selected signals, and drag them to the [Pins List](#) or [Device Window](#).

Pins that are to be assigned are highlighted in yellow. In the [Pins List](#), the signal names are displayed in the Signal column.

When more than one signal is being dragged onto the device, pins are highlighted in horizontal rows. The horizontal row may be toggled to a vertical column by holding down the **CTRL** key while dragging.

To restrict the assignment to one power bank only, hold down the **ALT** key while dragging.

Existing assignments are preserved, since signals that already have pins assigned to them are not considered for assignment. To force changes in existing assignments, hold down the **SHIFT** key while dragging. New assignments are then given independently of the existing assignments. Conflicting assignments will be removed.

3. Drop the dragged signals to perform the assignments.

If too many signals are selected, as many signals as pins available will be assigned.

Assignments using drag and drop may also be performed in the opposite direction, by dragging pins in the [Pins List](#) or the [Device Window](#) to the [Signals List](#). Signals that are to be assigned to are highlighted in yellow, and holding down the **SHIFT** key overwrites existing assignments.

Assign Mode

PADS I/O Designer allows you to assign pins visually in Assign Mode. This mode is especially useful for assigning bus signals. For a detailed description see “[Device Window Modes](#)” on page 34.

Changing Assignments in Lists

Drag & Drop

Selected rows in the Signal List and Pin List may also be dragged within the same list window. This operation changes pin assignments. To reassign pins select the signals in the Signal List, and drag the rows to a different position. During the drag and drop, the pin numbers are displayed in new positions. Dropping the pins assigns them. Similarly it is possible to select - in the Pin List - pins that are assigned, and then drag the signals assigned to them to a new position

in the Pin List. Note that this feature, as opposed to other pin assignment options, always changes existing assignments.

Choosing Signals/Pins to Assign from a List

Another way of changing assignments within List Windows is to double click on the Pin column in the Signal List, or on the Signal column in the Pin List. Then the list of all suitable pins or signals is displayed. Selecting an item from the list changes the assignment.


Combining Pins into a Bus Pin

The [Pins List](#) allows pins to be grouped into buses, known as *bus pins*. To create a bus, use the following procedure:

1. In the [Pins List](#), select the pins which will form the bus.
2. Right-click on the selection and select **Combine**.
3. PADS I/O Designer will ask you to enter a name for the bus pin, or you can accept the default name. Optionally, you can use the controls on this dialog to adjust the range of pins in your bus.

A bus pin is created and can be identified in the [Pins List](#) by the + at the left of the bus name.

Click + to expand or collapse the bus in the list, displaying or hiding its elements. Alternatively, select **View > Expand**, or **View > Collapse** to do the same thing. To expand or collapse all buses together, select **View > Expand All**, or **View > Collapse All**.

 **Note** Buses cannot be nested - bus pins cannot be combined into another bus.

A bus may be split or replaced with the flat list of its elements. To split a bus right-click on it and select **Split**. To split all buses at once, right-click in the List Window and select **Split All**.

To change the name of a bus pin, right-click it in the [Pins List](#) and select **Rename**, or with the bus selected in the [Pins List](#), select the menu item **Edit > Rename**.

You can also combine signals into bus signals, see “[Combining Signals into a Bus Signal](#)” on page 73.

Pin Swapping

PADS I/O Designer offers you advanced swapping capabilities, available in Signal List and Pin List context menus:

- **Swap > Swap Two Pins:** Swaps two selected signals/pins.
- **Swap > Swap Entire Bus:** Swaps two selected buses. The selected buses must have the same width.
- **Swap > Swap Entire Swap Group:** If the selected pins/signals belong to two different swap groups, this command swaps all assignments for these two swap groups.
- **Swap > Swap Entire Bank:** If the selected pins/signals belong to two different banks, this command swaps all assignments to pins from these two banks

Pin and Signal Types

The Type column in the Pin List displays allowed pin types, such as IO, CLOCK, etc. Some types may be combined. For example, for differential clock pins DIFFCLOCK is displayed.

[Table 8-1](#) lists the available pin and signal types. Unassignable pins have no associated HDL signal. Clock Capable I/O pairs are regular I/O pairs with special hardware connections to nearby regional clock resources.

Note



For information on specific pins, refer to the vendor data sheet for the part.

Table 8-1. Pin and Signal Types

Pin Type	Description
AnalogTRTN	Unassignable temperature monitor return pin.
AnalogGND	Unassignable ground for an analog circuitry pin.
AnalogIO	Unassignable analog I/O pin.
AnalogPF	Unassignable analog power filter pin.
AnalogOFEP	Unassignable other front-end pin. (Actel only.)
AnalogVCC	Unassignable power for an analog circuitry pin.
AnalogVREF	Unassignable analog reference voltage pin.
CCLOCK	Assignable Clock Capable pin.
CCLOCKDIFF	Assignable differential Clock Capable pin.
CLOCK	Assignable clock pin (see Clock Assignments below).
CONFIG	Unassignable configuration pin.
DIFF	Assignable differential pin (see Clock Assignments below).
DIFFCLOCK	Assignable differential clock pin.

Table 8-1. Pin and Signal Types (cont.)

Pin Type	Description
DQ	Assignable Double Data Rate memory interface data pin.
DQS	Assignable Double Data Rate memory interface data strobe pin.
DQSDIFF	Assignable Double Date Rate memory interface differential data strobe pin.
GND	Unassignable ground pin.
IO	Normal assignable pin.
JTAG	Unassignable JTAG pin.
MGT	Multi-Gigabit Transceiver pin.
MGTCALRES	Unassignable MGT calibration resistor pin.
MGTCLK	Assignable differential reference clock of MGT pins.
MGTRX	Assignable MGT receiver pin.
MgtTerminationReference	Unassignable precision reference resistor pin.
MGTTX	Assignable MGT transmitter pin.
MGTVCC	Unassignable power-supply pin for transceiver mixed signal circuitry of the MGT.
MGTVCCAUX	Unassignable analog power supply pin for circuitry of the MGT.
MGTVCCIO	Unassignable power-supply pin for the MGT output drivers.
MGTVCCPLL	Unassignable power-supply pin for PLL MGT.
MGTVCCTRX	Unassignable input/output buffer power supply pin for channel on left or right side of device.
MGTVREF	Unassignable threshold voltage pin for MGT.
MGTVTTRXC	Unassignable power-supply pin for the resistor calibration circuit of the MGT.
MSS	Microcontroller subsystem pin. (Actel only.)
NC	Unassignable not connected pin.
OTHER	Unassignable pin of some other category.
PLL	Assignable phase-locked loops pin.
PLLCAP	Unassignable phase-locked loops capacitor pin.
PLLDIFF	Assignable differential phase-locked loops pin.
PowerManagement	Unassignable power management pin.

Table 8-1. Pin and Signal Types (cont.)

Pin Type	Description
REFRES	Unassignable reference resistor pin.
RESERVED	Assignable pin with some additional features.
SystemMonitorADC	Unassignable system monitor pin.
TemperatureDiode	Unassignable temperate diode pin.
VBATT	Unassignable decrypter key memory backup supply pin.
VCC	Unassignable power pin.
VCCAUX	Unassignable power-supply pin for auxiliary circuits.
VCCINT	Unassignable dedicated internal core logic power supply pin.
VccMgtTermination	Unassignable power-supply pin for TX/RX circuitry of a transceiver.
VCCO	Unassignable power-supply pin for the output drivers.
VCCPD	Unassignable dedicated programming power supply pin.
VREF	Unassignable threshold voltage pin.
VTT	Unassignable dedicated terminating supply pin.

You can change the pin type for assigned pins, and change the default pin type for unassigned pins. Clicking inside the Type column presents a list containing all possible types for the selected pin. Changed pin types are stored in the database. Changes made to pin types are always associated with appropriate changes of the I/O Standards. It works the other way around too; if you switch a pin to an I/O Standard that does not support differential pins, the pin type is switched to non-differential.

In general, pin type usage is limited to what is hard-coded into the part libraries and allowed by pin assignment rules. For example; based on library data and a rule, you are not allowed to assign an output signal to a clock pin, since clock pins can be used only as input pins.

Clock Assignments

You can set PADS I/O Designer to automatically detect clock signals. To enable this functionality, select **Setup > Settings + Advanced**, then select **Detect Clocks** in the **Clock autodetection** section. PADS I/O Designer uses regular expressions to recognize lock signals. In most cases, clock signals are uniformly named, with names such as CLK. While parsing an HDL file, PADS I/O Designer looks for signals with similar names, and marks them as clock signals.

PADS I/O Designer can also recognize differential clock signals in an HDL design that has differential buffers instantiated in the top-level design. To enable this functionality, select **Setup > Settings + Advanced**, then select **Recognize differential signals** under **HDL source**

section. PADS I/O Designer then sets DIFFCLOCK type on any detected differential clock signals.

You can modify clock assignments later, using the Type column in the Signal List. To mark or unmark a signal as a clock, locate it in the **Signal List**, click in the **Type** column, and select the CLOCK or IO option from the list. The clock property of a signal is stored in the database.

Differential Pins Support

PADS I/O Designer checks for assignments to differential pins, i.e., pairs of pins that should be assigned to one signal, representing its positive and negative part, respectively. Differential pins are by default treated as normal. For any differential pin it is possible to turn on the differential pins support built in PADS I/O Designer. To do that, switch the type of such pin to DIFF. Then the type of the second pin of the pair will be switched to DIFF too. Similarly as for clock signals, a signal may be marked as being differential. Such signals are to be assigned to differential pin pairs.

To mark a signal as differential, locate it in the [Signals List](#), click in the Type column, and in the displayed list select the DIFF option.

To unmark a signal as differential, locate it in the [Signals List](#), click in the Type column and, in the displayed list, select the IO option. Then, whenever a signal is assigned to one pin of the pair, the other pin is assigned to the same signal automatically.

You can also choose as the type of a signal DIFFCLOCK, which means that such a signal is both a clock and a differential signal.

To easily distinguish differential pins, right-click on the [Pins List](#) and select **Combine All Differential Pins**. Two-element bus pins are created from all differential pin pairs. If one of a pair of differential pins is selected, right-click on the [Pins List](#) and select **Goto Complementary Differential Pin** to locate another one from the pair.

VREFs

Some I/O Standards require a VREF source with a specified value. If there's a pin assigned with such an I/O Standard the VREF value is established for the entire bank.

In order to support vendor or custom rules, Xilinx DCI Cascading for example, PADS I/O Designer can also establish a VREF value across multiple banks.

PADS I/O Designer does not allow you to assign a different value to a pin than that required by the established VREF.

Multi-Gigabit Transceiver Pins

PADS I/O Designer supports working with MGT (Multi-Gigabit Transceiver) pins by offering the following features:

Note



I/O Designer does not perform automated pin swapping on Multi-Gigabit Transceivers signal types due to specific pin placement considerations.

- Combining MGT pins

To combine the MGT pins in the Pin List, go to **Edit > Buses > Combine All Multi-Gigabit Transceiver Pins** or choose the Combine All Multi-Gigabit Transceiver Pins from the Pin List popup menu.

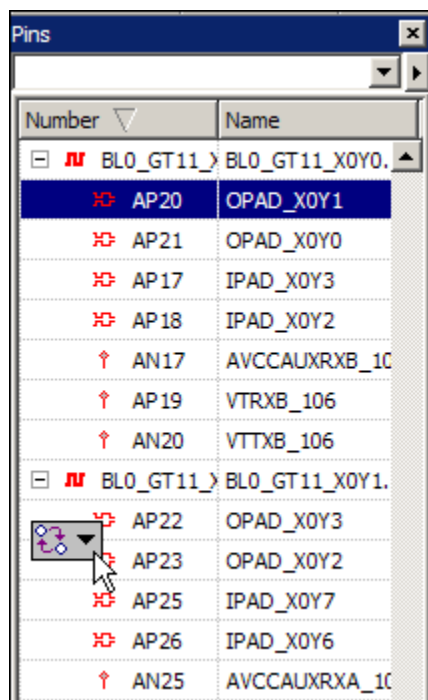
- MGT channel reassignment

PADS I/O Designer allows you to reassign an entire MGT channel within a single operation. To reassign an MGT channel:

- Drag and drop a single MGT pin onto a pin from a different channel

A smart tag should appear.

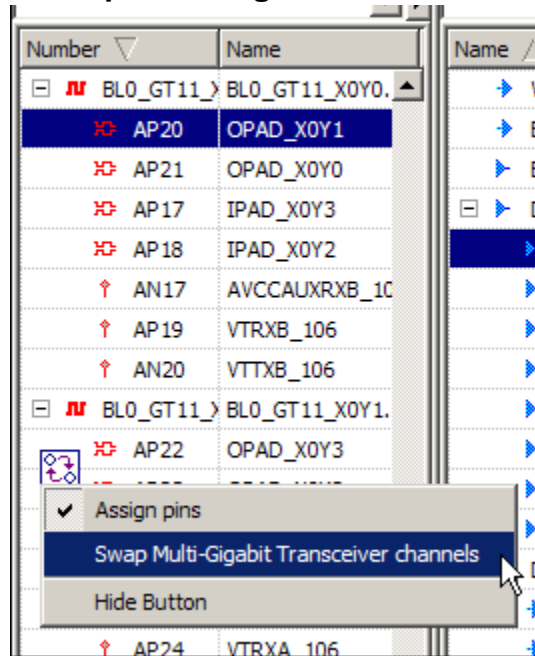
Figure 8-1. MGT Channel Reassignment



- Move the mouse pointer over the smart tag, and click the dropdown button

- c. From the dropdown menu, choose the Swap Multi-Gigabit Transceiver channels

Figure 8-2. Swap Multi-Gigabit Transceiver Channels



- **PCB Symbols**

The Symbols Generator allows you to generate symbols for MGT channels:

- One common symbol for all MGT channels
- Separate symbol for each MGT channel
- One common symbol for all MGT blocks
- Separate symbol for each MGT block

For more information, refer to “[Symbols Generator](#)” on page 96.

Special Signal/Pin Assignments

It is possible to define compatibility between different pin and signal types such that special assignments may be made.

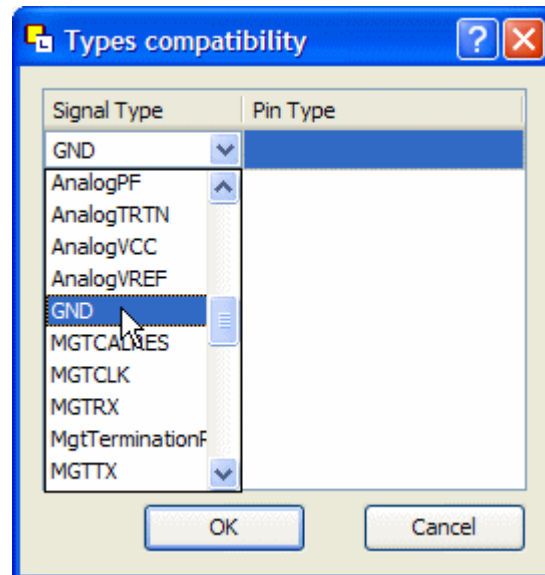
- An association between signal and pin types is made by defining a signal type and a pin type using the Types Compatibilities dialog.
- Special assignments mode has to be entered while assigning by holding down CTRL + SHIFT while dragging signals to the pins list.

Setting Types Compatibilities

1. Select **Tools > Types Compatibility**. The Types Compatibility dialog is displayed.
2. **Right-click** in the dialog and select **Insert**. An entry appears in the list.
3. Double-click in the **Signal Type** column and select the required signal using the drop down box.
4. Double-click in the **Pin Type** column and select the required signal using the drop down box.
5. Click **OK**.

The signal/pin association is now enabled and assignments between signals and pins of these types can now be made. See “[Making Special Assignments](#)” on page 92.

Figure 8-3. Setting Types Compatibility



Making Special Assignments

Once Types Compatibilities have been set (see “[Setting Types Compatibilities](#)” on page 92) assignments can be made between signals and pins of those types by holding down **CTRL** + **SHIFT** whilst dragging signals to pins.

Special FPGA assignments can only be made when assigning signals that have not been previously assigned.



Note

When making special assignments in this way, any [User-defined Rules](#) which have been created are ignored.

Special assignments can be unassigned in the same way as ordinary assignments - it is **not necessary** to hold down **CTRL + SHIFT** to remove assignments.



Tip: This operation could be used after the device is optimized, to allow connection of unused I/O pins to a PCB signal (i.e. GND) such that the pins will be automatically connected to the appropriate net (i.e. GND) on the layout.


Chapter 9

Creating, Editing and Updating Symbols and Schematics


Symbol Generation


Once a device has been built in an FPGA database, PADS I/O Designer can generate Functional block and PCB symbols to represent it. You can then export these symbols to the schematic or to the central library.

PADS I/O Designer also generates part information. You can generate a complete part and export it to the local schematic or the central library. When you export parts to the central library, it is recommended that you place every pin explicitly on a PCB symbol, since central library parts are intended to be generic, available for use in multiple designs.

 **Note** The PADS I/O Designer tool requires read/write access to library partitions in the central library. If the PADS I/O Designer tool displays an error message regarding library object partition access, use system tools to modify the central library directory permissions and library object partitions to allow full access. See [Configuring Library Permissions](#).

See “[Typical Design Flows](#)” on page 53 for guidance on which symbols to generate for your design process.

 **Functional Block Symbols** (or simply Functional Symbols) are used on functional-level schematics. Ports in functional blocks are equivalent to signals in an HDL unit (VHDL entity or Verilog module).

 **PCB symbols** are used on board-level schematics, as the representation of a physical device. Ports in PCB symbols are equivalent to pins in the physical device.

There are two methods which can be used to create new symbols in PADS I/O Designer:

- Generate symbols using the [Symbols Generator](#).
- Build symbols in the [Symbol Window](#) using the symbol editing tools. See “[Building a New Symbol](#)” on page 104.

PADS I/O Designer supports heterogeneous symbols; several symbols belonging to one logical unit. All functional blocks share signals from the database, and the signals are not normally duplicated on different functional blocks. Each PCB symbol in PADS I/O Designer belongs to a functional block and should only contain pins that are assigned to signals placed on that

functional block. There may be one or more functional blocks, and each of them may contain one or more PCB symbols, forming a hierarchy, which is directly represented if symbols with schematics are generated. Each functional block is then generated as a symbol with attached schematic. The schematic contains all PCB symbols belonging to the functional block, and the wires attached to the PCB symbols pins in the hierarchical schematic represent the pin assignments.

Only one symbol may be edited at a time. To switch between symbols, you can use the list at the top of the Symbol Window. You could also browse through the available symbols by using the two arrow buttons in the bottom-right corner of the [Symbol Window](#).

Note




In earlier versions of PADS I/O Designer, Symbols Generator was referred to as Symbol Wizard.

Symbols Generator

The Symbols Generator is a full-featured tool provided to assist you in creating and updating both functional block symbols and related PCB symbols.

The default binding type for ports created by the Symbols Generator is **Pin**.


Creating a New Symbol Using the Symbols Generator

1. Invoke the Symbols Generator by doing one of the following:
 - From the main menu in PADS I/O Designer, select **Symbol > Symbols Generator**.
 - Click the  icon on the toolbar or at the top of the [Symbol Window](#).
2. On the **Basic Settings** page, enter information for:
 - **Symbol name**
The naming convention for PCB symbols is *<Symbol name>_pcb*. Do not use names with characters following *_pcb*, as this may cause problems when creating functional symbols split by PCB afterwards. Functional symbols should always be named differently to PCB symbols (for example, without the *_pcb*) to avoid overwriting them.
 - **Use signals:** During symbol generation, whether to use **All** signals or **Selected** only.
 - **Pin Limits for All Symbol Types:**
 - Always split symbols larger than page size: This option applies to functional and PCB symbols.

Buses are not split between symbols even if the resulting symbol is larger than the page size.

- Always split symbols with more pins than <number>: This option applies to functional and PCB symbols.

3. Click **Next** to display the **Symbol Settings** page.

 **Note** The information in the right hand column of the Symbol Settings page is specific to the selected (highlighted) pin type in the central column. Any changes you make in the right hand column apply to the selected pin type only. When you select a different pin type, the information in the right hand column changes to reflect any differences in the information for that pin type.

4. On the **Symbol Settings** page, enter information for:

- **Pins usage:**
 - Create separate symbols for these pins
 - Add these pins to I/O symbols
- **Design type:** Specifies whether or not the design type is hierarchical or flat:
 - Hierarchical: Enables all controls below
 - Flat: Reduces the symbol type to PCB and selects it
- **Symbols type:** Functional, PCB or Both.

5. To specify advanced symbol settings, click **More >>** to display the **Fracturing**, **Labels**, and **Appearance** tabs. The options available under these tabs depends on the selected pin group - see [Figure 9-1](#). These advanced settings are described in the following.

Fracturing tab

Select whether the symbol should be split. The options available here depend on whether you have chosen to create Functional Symbol, PCB Symbol, or both. Following is a list of symbol fracturing options with explanations ([Figure 9-1](#) shows how the list changes depending on the selected pin group).

- **Split PCB and Functional Symbols:**
 - Do not split: Create a single functional/PCB symbol depending on the Symbol Type selected.
 - Split only PCB symbols: This option enables you to specify a more detailed split method for PCB symbols.

- Split functional and PCB symbols: This option enables you to specify more detailed split method for both functional and PCB symbols. It is only available if both symbols are created (Design Type = Hierarchical).
- **Fracturing Scheme for PCB and Functional Symbols:**
 - Split by power banks: Create separate symbols for groups of pins that belong to different power banks.
 - Separate data and control signals: Create separate symbols for buses and scalar signals.
 - Separate signals connected to different components: Create separate symbol for groups of signals that belong to different HDL units.
 - Split by signal: Only available for PCB power pins.
 - Split by channel and Split by block: Only available for MGT pins.
 - Split by PCB symbols: Create functional symbols split by existing PCB symbols (that is, imported from the central library).
 - Split by functional symbols: Create functional symbols split by existing functional symbols (that is, imported from the central library).

Labels tab

Label settings are available for all types of pins. Once defined, these settings do not normally need to be changed (as is in the case of symbol generation).

- **Functional block:** Sets port labels for the functional block. The default port label depends on the pin type.
- **PCB symbol:** Sets port labels for the PCB symbol. The default port label depends on the pin type.

Appearance tab

These settings are available for all types of pins. Once defined, these settings do not normally need to be changed (as is in the case of symbol generation).

- **Symbols:** Specifies the following:
 - Symbol name
 - Symbol description
 - Functional symbol background
 - PCB symbol background
- **Ports:** Sets the port position for the pin. It is used even if pins of the current type are added to other symbols. For example, select **Left** to insert the port to the left of the symbol. This option relates to the Clk pin position or the VCC/GND pins position only.

Selecting the Left and Right option can assist in schematic updating and maintenance by allowing the symbol to grow vertically, leading to fewer connection issues.

6. Click **Next >** to display the **Summary** page.
7. On the **Summary** page, review the changes you have made. The page summarizes symbols and ports that will be created, deleted, or changed after clicking **Finish**. Click **More >>** to view advanced summary information.
8. Click **Finish**.

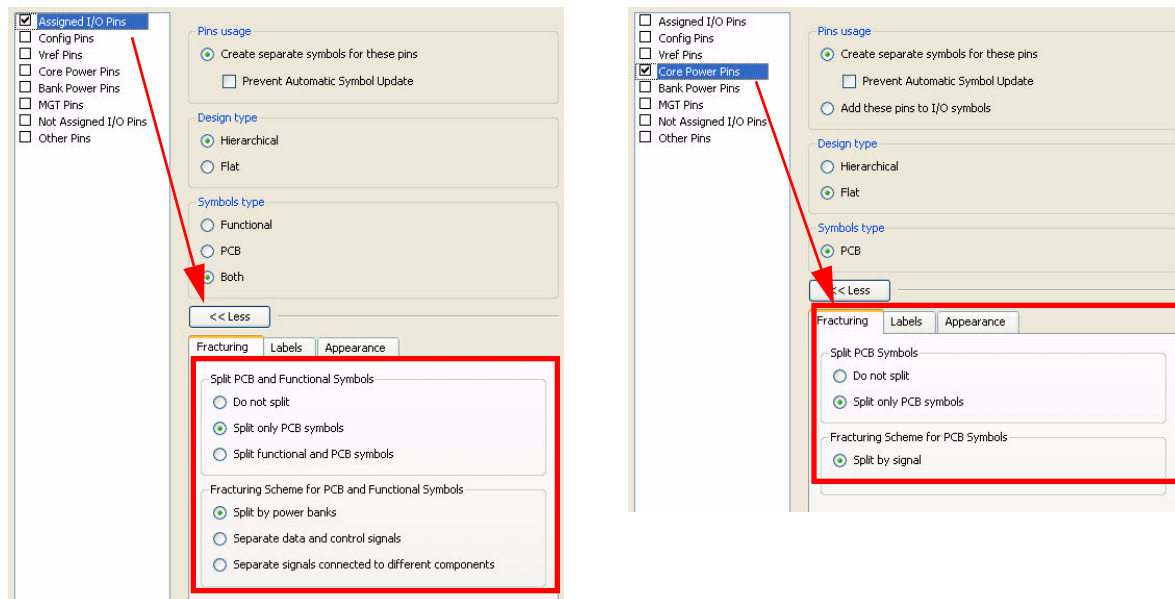
The symbols are created and appear in the [Symbol Window](#). You can click **Finish** at anytime to create a symbol using the current settings.

To display the **Divide Ports** page, check **Adjust symbols manually** on the Summary page and click **Next**. Use this page to add unplaced signals and pins to symbols (existing or created by the Symbols Generator). The page provides an easy way to handle signals and pins that have not been assigned and therefore are not part of any symbol.

Several pages contain the buttons **Load Defaults** (loads the default settings) and **Save as Defaults** (saves the current settings as defaults).


Figure 9-1 shows the different option available depending on the selected pin group.

Figure 9-1. Symbol Settings Page (Advanced Options)



Updating Symbols Using the Symbols Generator

Once symbols have been created, the Symbols Generator can be used to update them. In this case, it is not necessary to re-enter the information for symbol fracturing, appearance or port labels; PADS I/O Designer uses the settings entered when creating the symbol.

1. Invoke the Symbols Generator by doing one of the following:
 - From the main menu in PADS I/O Designer, select **Symbol > Symbols Generator**.
 - Click the  button at the top of the [Symbol Window](#).
2. The Symbols Generator opens on the Basic Settings page. Select **Update symbols** under **Symbols** and click **Next** and **Finish**.

Limitations

It is not possible to update symbols using the Symbols Generator when:

- No symbol exists
- All existing symbols have the **Automatic Symbol Update** flag set to **Prevent**.
- All existing symbols were created manually
- All existing symbols were created semi-automatically (with multiple steps in Symbols Generator) for Selected pins only or when functional and PCB symbols were not created in a single event.

Updating Symbols in Databases Created Using Older Versions of PADS I/O Designer

The Symbols Generator does not provide update functionality for all symbols created in older versions of PADS I/O Designer (pre-9.1 versions). (The reason for this is that symbols from older versions have a *prevent* flag set after loading.) In order to use the update symbols function, symbols must be created using the Symbols Generator.

To support updating symbols from PADS I/O Designer versions prior to 9.1, the tcl command `symbolwizard` is supported in PADS I/O Designer 9.2.

Power Symbol Generation

FPGA power can be managed as implicit connections in the PDB or explicitly in the generated symbols.

A list of default power signal name assignments can be viewed under **Setup > Settings + PCB signals generation**.

If an I/O Standard is not selected with a specific voltage (for example, 2.5, 3.3, 1.8 volts) the top window on the dialog is ignored and the power signal names shown in the bottom window are used. If an I/O Standard is selected (for example, LVCMOS33), PADS I/O Designer will detect a 3.3 volt signal and apply the Default PCB Signal in the top window named 'v_3_3'. PADS I/O Designer will automatically configure the Bank where this pin resides.

Explicit Power Connection Symbols

Explicit power management requires the generation of power symbols. By selecting power symbol creation in the [Symbols Generator](#), PADS I/O Designer automatically creates new power signals in the [Signals List](#). New power signals are automatically assigned to the correct power pins in the [Pins List](#).

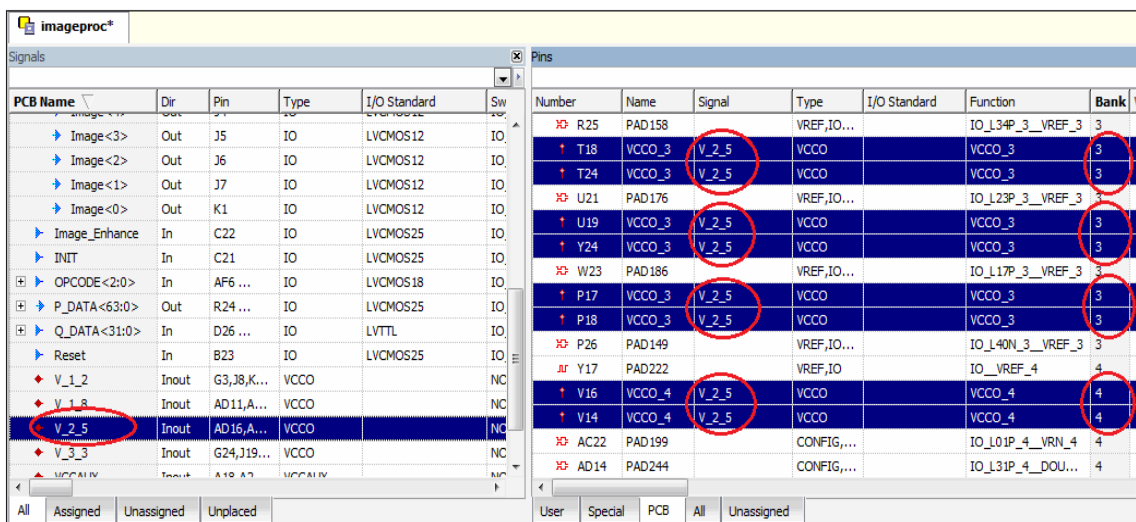
Power signals are **not** part of the functional block that is created as part of the [Symbols Generator](#) process. But power symbols are exported as part of the [Exporting Symbols and Schematics](#) step. Power symbols must be placed and the proper power and ground signals must be manually attached.

PADS I/O Designer will automatically create new power signals based on the I/O Standard assignment and makes the appropriate assignments for the bank VCCO pins. In [Figure 9-2](#), the *P_DATA* bus was assigned a LVCMOS25 standard which requires a 2.5V VCCO supply. PADS I/O Designer automatically creates the signal during symbol generation and makes the assignments.

Figure 9-2. Power Signal Generation Based on I/O Standard Assignment

PCB Name	Dir	Pin	Type	I/O Standard	Sw
Image<3>	Out	J5	IO	LVCMOS12	IO
Image<2>	Out	J6	IO	LVCMOS12	IO
Image<1>	Out	J7	IO	LVCMOS12	IO
Image<0>	Out	K1	IO	LVCMOS12	IO
Image_Enhance	In	C22	IO	LVCMOS25	IO
INIT	In	C21	IO	LVCMOS25	IO
OPCODE<2:0>	In	AF6...	IO	LVCMOS18	IO
P_DATA<63:0>	Out	R24...	IO	LVCMOS25	IO
Q_DATA<31:0>	In	D26...	IO	LVTTL	IO
Reset	In	B23	IO	LVCMOS25	IO
V_1_2	Inout	G3,38,K...	VCCO		NC
V_1_8	Inout	AD11,A...	VCCO		NC
V_2_5	Inout	AD16,A...	VCCO		NC
V_3_3	Inout	G24,J19...	VCCO		NC
W22	Inout	A10,A7	IO	LVCMOS12	IO

Number	Name	Signal	Type	I/O Standard	Function	Bank
W25	PAD177		IO,DIFF		IO_L22N_3	3
W22	PAD188		IO,DIFF		IO_L16P_3	3
W23	PAD186		VREF,IO...		IO_L17P_3_VREF_3	3
W24	PAD185		IO,DIFF		IO_L17N_3	3
W25	PAD184		IO,DIFF		IO_L19P_3	3
W26	PAD183		IO,DIFF		IO_L19N_3	3
AC25	PAD192	P_DATA<58>	IO,DIFF	LVCMOS25	IO_L03P_3	3
AC26	PAD191	P_DATA<60>	IO,DIFF	LVCMOS25	IO_L03N_3	3
Y25	PAD190		IO,DIFF		IO_L14P_3	3
Y26	PAD189		IO,DIFF		IO_L14N_3	3
AF24	PAD201	P_DATA<41>	IO,DIFF	LVCMOS25	IO_L04P_4	4
AE23	PAD203	P_DATA<13>	IO,DIFF	LVCMOS25	IO_L05N_4	4
AC20	PAD214	P_DATA<8>	IO,DIFF	LVCMOS25	IO_L09P_4	4
AC21	PAD209	P_DATA<17>	IO,DIFF	LVCMOS25	IO_L07P_4	4
AA14	PAD241		CONFIG,...		IO_L30P_4_D3	4



Configuration Symbol Generation

The **Symbols Generator** may be used to create configuration symbols such as JTAG by checking the **CONFIG Pins** group in the Symbol Settings page. When you complete the Symbols Generator, this creates symbols for all the pins with type CONFIG (except Vrp and Vrn which are put onto the Vref pins symbol).

If more control is required, CONFIG signals can be added and the symbol created manually. To do this, use the following steps:


1. After the JTAG signals are added to the FPGA database, assign them to the JTAG pins by dragging the JTAG pin to the JTAG signal.
2. Create a new empty symbol by selecting **Symbol > New** or by clicking the  icon on the toolbar or on the **Symbol Window**.
3. Manually create the symbol by dragging the pins from the **Pins List** to the new symbol in the **Symbol Window**. The result is a symbol specifically for JTAG purposes.

Figure 9-3. Manually Created JTAG Symbol



The JTAG symbol is typically not part of the hierarchical block. Make sure the [Symbol Properties](#) do not list a functional block attribute.

This symbol can now be exported and placed in the schematic. See “[Exporting Symbols and Schematics](#)” on page 112.

Update Power Signals

The Update Power Signals functionality (**Tools > Update Power Signals**) checks the correctness of GND, VCC, VCCO, VREF, VRN, and VRP pin assignments. In the case of VCCO and VREF pins, PADS I/O Designer checks whether or not all pins are connected to the correct voltage levels. It also checks if other power pins are connected.

Checking power signals involves setting up GND, VCC signals to power the device and VREF and VCCO signals to set thresholds for I/O standards used in the device. There are also VRN and VRP pin connections that are necessary to set up DCI I/O standards. To ensure signal integrity before creating symbols, update power signals is run automatically (if not disabled) before every:

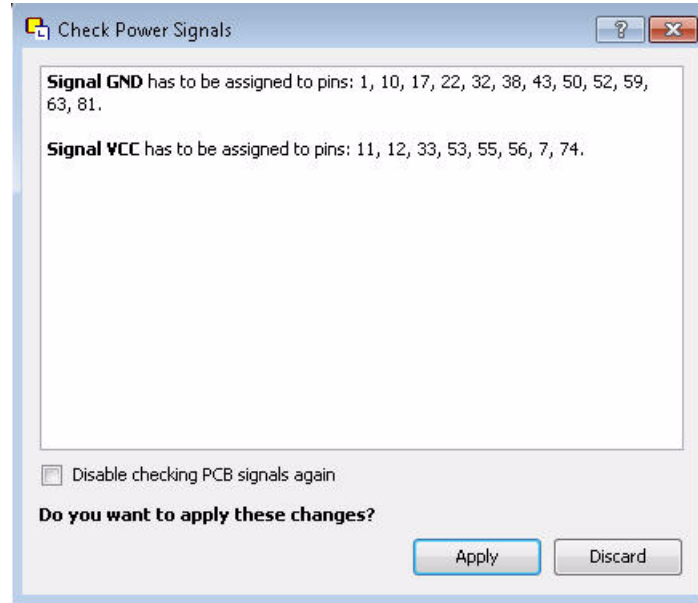
- [Symbol Generation](#)
- [Schematic Update](#) / Schematic Update for All Components
- Export Schematic and Symbols / Export Schematic and Symbols for All components
- Export All Symbols / Export Current Symbol Only
- Export Interconnectivity Table / Export Interconnectivity Table for All Components

Update Power Signals creates power signals from the information in the Pins window’s VCCIO and VREF columns. PADS I/O Designer fills these columns automatically, taking into consideration signal I/O standards and direction.

In the event that power signals are not assigned correctly, PADS I/O Designer suggests the correct assignments as shown in [Figure 9-4](#). You can then choose to **Apply** or **Discard** the

suggested assignments using the buttons on the dialog. Select “Disable checking PCB signals again” to disable automatic checking. If you disable automatic checking, you can still manually check PCB signals by selecting **Tools > Update power signals**.


Figure 9-4. Check Power Signals Dialog



Building a New Symbol

PADS I/O Designer incorporates an advanced built-in symbol editor which can be used to create and edit PCB-level symbols or functional-level block symbols.

To create a new symbol in PADS I/O Designer:

1. Do one of the following:
 - Select **Symbol > New**.
 - Click the  button on the [Zoom Toolbar](#).
2. Enter a name for the symbol in the **Symbol name:** field.
3. To create a PCB Symbol, check the PCB symbol option. To create a functional block symbol, uncheck this option.

If this option is checked, the list of all existing functional blocks in the **Functional block:** field is activated.
4. For PCB symbols, select the functional block to which the new PCB symbol will belong, in the **Functional block:** field.


5. Select a background for the symbol from the list of those available in the **Symbol background:** field.
6. Click **OK**.

The new symbol is displayed in the [Symbol Window](#).

7. Add ports to the symbol by dragging and dropping signals and pins from the [Signals List](#) and [Pins List](#) to the symbol in the [Symbol Window](#).

You can create several ports by selecting a group of ports / signals and dragging the group to the symbol. Ports created by dragging signals have their binding type set to Signal. Ports created by dragging pins have their binding type set to Pin.

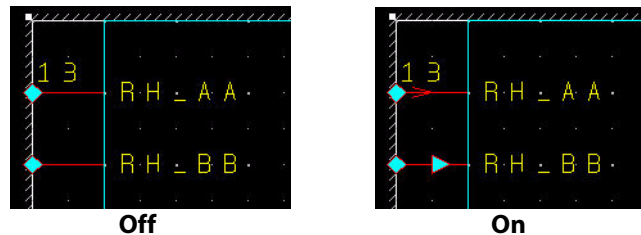
Note

 If you drag a bus to a PCB symbol, bus members are added as separate ports, since in most cases you do not want buses on PCB symbols. To override this feature, drag a bus to a PCB symbol with the **CTRL** key down. Then it will be added as one bus port.

Port Types and Shapes

PADS I/O Designer provides support for several port types. By default, port types are visualized by means of different port shapes. If you prefer straight lines without any decorations for ports of all types, uncheck the **View > Symbol > Show Port Type** option. [Figure 9-5](#) shows the result of switching port types on and off.

Figure 9-5. Displaying Port Types




Further customization of the appearance of ports is possible. All available port types are listed under **Setup > Settings + Port Types** (under **Symbol Editor**) together with shapes assigned to them. There are several shapes to choose from. For available port types, see “[Port Types](#)” on page 109.

Drawing Tools

Use the drawing tools to add [Arcs](#), [Circles](#), [Lines](#), [Rectangles](#) or text to a symbol. The following sections describe all operations specific to these modes.


Arcs

To draw an arc, use the  button on the toolbar. Then click the position where the arc should begin, and drag to its desired end.

Additional features available while drawing an arc are:

- Orientation of the arc may be toggled with the **Tab** key.
- Its middle point may be anchored with the **Space** key.
- If the arcs middle point is anchored, it may be unanchored with the **Backspace** key.


Circles

To draw a circle, use the  button on the toolbar. Then click on the position where one of the corners of circles bounding rectangle should lie, and drag to the opposite corner.

Additional features available during circle drawing are:

- Hold down **SHIFT** while drawing to force an ellipse instead of an arbitrary circle.


Lines

To draw a line or polyline, use the  button on the toolbar. Then click the position where the line should begin, and drag to its desired end.

Additional features available during line drawing are:

- Press the **Spacebar** while drawing to add a new segment to the line. Added segments may be removed with the **Backspace** key.
- Press **CTRL** while editing (dragging one of the handles) to add a new segment.
- Press **SHIFT** while dragging the middle handle to split the line into two parts.


Rectangles

To draw a rectangle, use the  button on the toolbar. Then click the position where one of the corners of rectangle should lie, and drag to the opposite corner.

Additional features available during rectangle drawing are:

- Hold down **SHIFT** while drawing to force a square instead of an arbitrary rectangle.

Text

To add text to a symbol, use the  button on the toolbar, then click the position where the text should begin and start typing. The text cursor (caret) is displayed, and normal editing

operations, limited to a single line, may be performed. Editing is completed after the Enter key has been pressed. The **Esc** key may be used to cancel editing without adding the text to the symbol.

Snap to Grid

While drawing and editing symbols, Snap to Grid is enabled/disabled under **Setup > Settings + Symbol Editor**. If the Snap to Grid option is selected, all drawing operations obey this setting. However, snapping to grid may be temporarily disabled with the ALT key. For instance, to start drawing a line outside grid, it is not necessary to disable snapping to grid, but it is enough to hold the ALT key down while drawing.

Adding an Image to a Symbol

In the PADS Designer flows, you can add an image to each symbol by associating it with a graphics file. Graphics files are stored in the /OLE directory within the project. To do this, either:

- Select **Symbol > Add Picture**
- or
- Right-click in the **Symbol Window** and select **Add Picture** from the pop-up menu.


You can then browse for a graphics file, which can be in the format *.bmp* or *.jpg*.

Note



Adding an image file to a symbol is available in the PADS Designer flow only.

Editing Symbol Elements

To select elements in a symbol, enable Select Mode by clicking the  icon in the toolbar. Select an element by clicking on it. To add an element to the existing selection, **CTRL+click** on the element. More than one element may be selected together by dragging. While dragging, you will see the selection rectangle displayed. All elements within the rectangle are selected.

To zoom in on an area, drag the mouse with the **SHIFT** key down. The rectangle is displayed when you drag. After dragging the selected rectangle will be zoomed to fill the entire window. See “[Symbol Window Zoom Controls](#)” on page 25 for more information on zoom controls.

The Symbol Window supports the usual clipboard operations: Cut, Copy, and Paste. The options are available in the View menu, in the toolbar, and in the pop-up menu.

In Select Mode, the selected elements can be duplicated by dragging them with the CTRL key held down. The selected elements remain unmoved, and the copy of the selection may be dragged to the desired position.

To delete symbol elements, select them, right-click and select Delete. This command is also available in the Edit menu.

Moving and Changing Elements

To move symbol elements, select them, and drag them to the desired position. To resize a symbol element, select it, and make sure that it is the only selected element. A resizable element will now display handles, that is, small squares. Dragging handles will resize or reshape the symbol element.

To edit graphical text, use the Edit command located in the selected text pop-up menu. PADS I/O Designer will switch to text-editing mode. The text cursor is displayed, and usual operations for editing a single line of text are available. The Enter key completes editing, while **Esc** cancels all changes.

Changing Symbol Backgrounds

PADS I/O Designer supports different symbol backgrounds. In addition to the default rectangle, you can choose among several other backgrounds, such as *mux*. To change the background, open the Properties window for the symbol, and change the Background property there.

Backgrounds may be customized. To edit the background of the current symbol, check the **Symbol > Edit Background** option. All symbol elements, except for the background, are hidden. In this mode you can use any graphical elements to change the appearance of the background. To switch back to normal symbol editing, uncheck the Edit Background option.

You can save the changed background in order to use it in other symbols. Use the command **Symbol > Export Background**. This option displays the standard Save dialog, in which you can choose the background file name. Backgrounds should be saved to the shapes subdirectory of the PADS I/O Designer installation directory, since in the [Properties Window](#) only backgrounds found in this subdirectory are accessible.

Printing Symbols

PADS I/O Designer allows printing symbols created and edited in the program. To print the current symbol, use the **File > Print > Symbol** command. This displays your computer's standard Print dialog.

Deleting Symbols

To delete the symbol displayed in the [Symbol Window](#), use the **Symbol > Delete** command. If you want to delete all symbols, then use the **Symbol > Delete All Symbols** command.

Symbol Editor Settings

Select **Setup > Settings + Symbol Editor** to define properties for symbols created in PADS I/O Designer as well as the page size and orientation for building schematics.

The Ports section defines pin length as a function of the units defined in the General section. Also it defines the spacing between pins as a function of grid steps.

Note



When you use PADS Designer as your schematic tool, PADS I/O Designer reads the grid value from PADS Designer settings when you open the project. You can change the value of the grid used by PADS I/O Designer, but the grid value reverts to the PADS Designer setting when you next open the project.

The Symbol Editor page allows you to alter the following [Symbol Window](#) options:

- Show or hide rulers.
- Show or hide the crosshair cursor.
- Default unit of measurement. Possible values are: 0.01mm, 0.05mm, 0.1mm, 0.5mm, 1.0mm or 0.005inch, 0.01inch, 0.05inch, 0.1inch.
- Page size. Available page sizes are: A0, A1, A2, A3, A4, A, B, C, D, E.
- Page orientation; portrait or landscape.
- Port length.
- Spacing between ports.
- Label format for ports in functional symbols and pins in PCB symbols (Signal Name, Pin Name, Pin Number or Pin Function).
- Show or hide grid.
- Snap [Symbol Window](#) elements to grid.
- Grid Step.

Port Types

The **Setup > Settings + Port Types** (under **Symbol Editor**) page allows you to alter the following options:

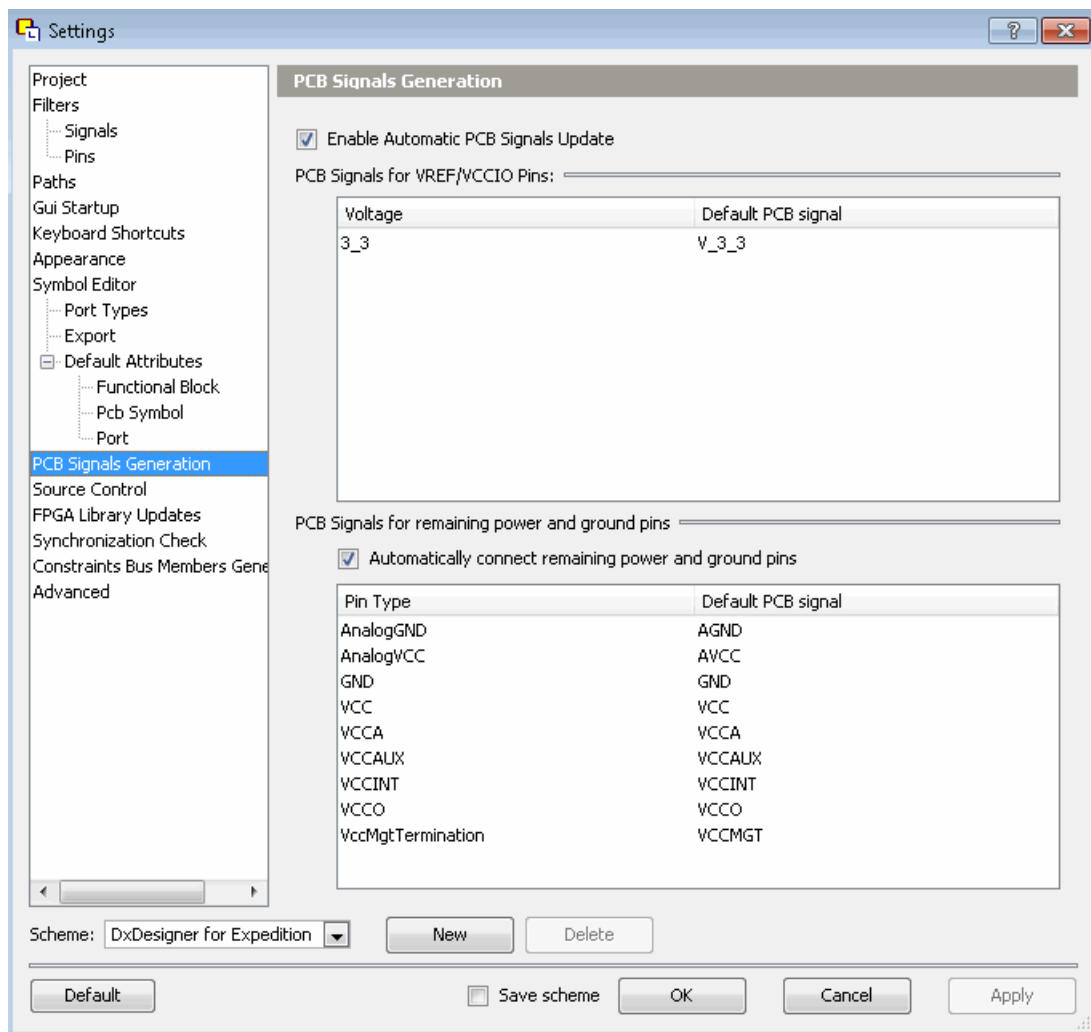
- Show or hide port type on a symbol in the [Symbol Window](#).
- Set port size of a symbol in the [Symbol Window](#).
- Associate port type with a shape.

- Available built-in port types include: TERMINAL, GROUND, POWER, TRI, OEM, OCL, ANALOG, BI, OUT and IN.
- Available built-in shapes include: ANALOG, ARROW, BI, CLK, DIAMOND, IN, INV, OCL, OEM, OUT and TRI.

PCB Signals Generation

The **Setup > Settings + PCB Signals Generation** page contains settings for PCB signals (power, ground and Vref). These settings are used by [Update Power Signals](#).

Figure 9-6. PCB Signals Generation Window



If the 'Enable Automatic PCB Signals Update' option is selected then PADS I/O Designer will perform check for power signals before the following actions;

- Symbols Generation

- Export Schematic and Symbols / Export Schematic and Symbols for All components
- Export Interconnectivity Table / Export Interconnectivity Table for All components
- Export All Symbols / Export Current Symbol Only
- Schematic Update / Schematic Update for All components

The PCB Signals for VREF/VCCIO Pins list contains default PCB signal mappings for VREF/VCCIO pins. The value for a given VREF/VCCIO pin from the list is used only if the pin does not have a manually assigned PCB Signal.

The remaining power and ground pins list contains default PCB signal mappings for power and ground pins other than VREF/VCCIO. The automatically connect remaining power and ground pins option allows you to control whether PADS I/O Designer should generate signals for the selected pins.

Symbol Export Settings

The **Setup > Settings + Export** (under **Symbol Editor**) page contains settings for symbol export:

- Run viewer after export: Select whether to run a viewer after exporting a symbol.
- Export graphical attributes: This option allows you to export graphical attributes, such as colors, fonts and graphics.
- Create multipage schematic: This option allows you to split generated schematics into multiple pages. When this option is enabled, schematics that do not fit into the chosen page size will be split.
- Add dangling nets: A generated schematic contains nets that connect PCB symbol ports with functional block ports. This option tells PADS I/O Designer to generate nets for ports that do not exist in the functional block.
- Export port direction shape: This options adds graphical elements representing port types to the exported file.
- Export swap groups: Export swap group information. Option is enabled by default.
- (PADS Designer only) Automatically add page border to the generated schematic: This group allows you to add a page border to the PADS Designer schematic.
- (Orcad flow only) Extend functional symbol with _func and keep base symbol name for PCB symbols.

Creating Generic Symbols for Use With PADS I/O Designer

Generic symbols are symbols for devices, stored in a Library, whose pin names, pin numbers, fracturing scheme and symbol name do not change from one design to another. PCB-based pin swapping is not used in this flow because FPGA pin swapping rules are design and device-based. The generic symbols in the library are placed directly into the schematic and are not touched by PADS I/O Designer.

When using generic symbols, any pin-assignment changes made by PADS I/O Designer are migrated to the schematic using **Export > Schematic Update**. This implements the pin assignments made in PADS I/O Designer by adding net name stubs to the symbol on the schematic. The symbol itself is not modified using this process.

Generic Symbols are PCB symbols and not functional blocks. The reason for this is that PADS I/O Designer reads the FPGA vendor delivered libraries and interprets the number of pins, pin numbers, and pin functions for each of the devices in the library. PADS I/O Designer automatically creates basic symbols with appropriate fractures and create the associated parts database (PDB) library files for these symbols with all the appropriate pin mapping information.

The initial settings for the symbol editor may be configured by editing [Symbol Editor Settings](#) under **Setup > Settings + Symbol Editor**.

The label for functional and PCB signals should be set to “Pin Function” for creating generic symbols.

Importing Symbols from Independent Libraries

PADS I/O Designer supports the use of symbols from independent libraries with the **Import > Symbols from Independent Libraries** menu selection.

This option opens a dialog from which you select the partition from which to import the symbol, and, once a partition is selected, lists the symbols for that partition. The list of partitions contains the independent libraries defined in PADS Designer.

Note



If you want PADS I/O Designer to place the symbols on generated schematics as library symbols, you must import the symbols as “Read only”. If you import the symbols as anything other than “Read only,” PADS I/O Designer treats them as local symbols.

Exporting Symbols and Schematics

PADS I/O Designer allows you to generate symbol and schematics data for exchange with external design definition tools, such as PADS Designer.

When exporting to PADS Designer, PADS I/O Designer puts PADS I/O Designer symbols on separate sheets called a reserved sheet that should not be modified. Reserved sheets contain only PADS I/O Designer symbols and nets. Connection to the rest of the schematics is guaranteed by net names. During schematics export, sheets with the IOD_RESERVED_SHEET attribute are automatically overwritten. In the remaining sheets, symbols with the IOD_GENERATED attribute are moved to a new Reserved Sheet.

All sheets are placed into separate blocks in the schematic tool. When you add a sheet inside a block generated by PADS I/O Designer, then the sheet will not have the IOD_RESERVED_SHEET attribute. PADS I/O Designer protects against deleting of this additional sheet during the symbol update process.

Hierarchical Schematic Generation

A hierarchical schematic is generated using the following generation methods:

- **Export > Schematic and Symbols**

Generates a functional block along with the underlying schematics containing PCB symbols for the current database.

- **Export > Schematic and Symbols for all Components**

Generates functional block along with the underlying schematics containing PCB symbols for all component databases in the design.

PADS I/O Designer automatically exports the symbol set along with the necessary number of schematic pages. When complete, the new schematic pages appear in PADS Designer under Blocks. The hierarchical block can be placed by selecting the local **Symbols** and then **local** for the exported functional block. Place the functional block with Added Nets and Net Names so the functional block is fully wired. For more information see “[Placing Symbols in an PADS Designer Schematic](#)” on page 115.

The underlying schematic contains the PCB symbols and the necessary wires and ports to make the connection to the Functional block.

Note



When exporting schematics from PADS I/O Designer, the following message can occur:

“iCDB Error: Running server for the project has changed (Could not create new session - project GUID mismatch) I/O Designer is not connected to iCDB database”

This problem may be connected with copying a project while it is already opened in PADS Designer. Close PADS Designer to clear the problem.

Note



When exporting schematics from PADS I/O Designer, the following message can occur when working on a PADS Designer project created in PADS I/O Designer:

*"Cannot export schematic: 'Special Components' are not set:
PORT_BI,PORT_IN,PORT_OUT"*

*To change offpage connector settings, run PADS Designer with current project, and in PADS Designer choose **Setup > Settings**, select **Project**, then select **Special Components**.*

Flat Schematic Generation

A flat schematic is created when only the symbols are exported from PADS I/O Designer. The following generation methods are available:

- **Export > All Symbols**

Generates all symbols without an underlying schematic.

- **Export > Current Symbol Only**

Generates a single symbol without an underlying schematic.

The symbols can then be placed in the schematic where desired. Turn on the Add Nets and Add Net Name option so the placed symbols are fully wired. For this PADS Designer feature to work correctly, the symbol pin label must be the name of the wire. This is a 'custom' symbol for this implementation of the component.

Symbols can be added to the independent library in the following steps:

1. Export symbols as described above.
2. From PADS Designer, export local symbols to the independent library as ASCII files.

Note



The PADS I/O Designer tool requires read/write access to library partitions in the central library. If the PADS I/O Designer tool displays an error message regarding library object partition access, use system tools to modify the central library directory permissions and library object partitions to allow full access. See [Configuring Library Permissions](#).

After pin changes in PADS I/O Designer, only the symbols need to be updated in PADS Designer. **Note:** This assumes that the symbols are not changed (that is, a signal is added or deleted from symbols).

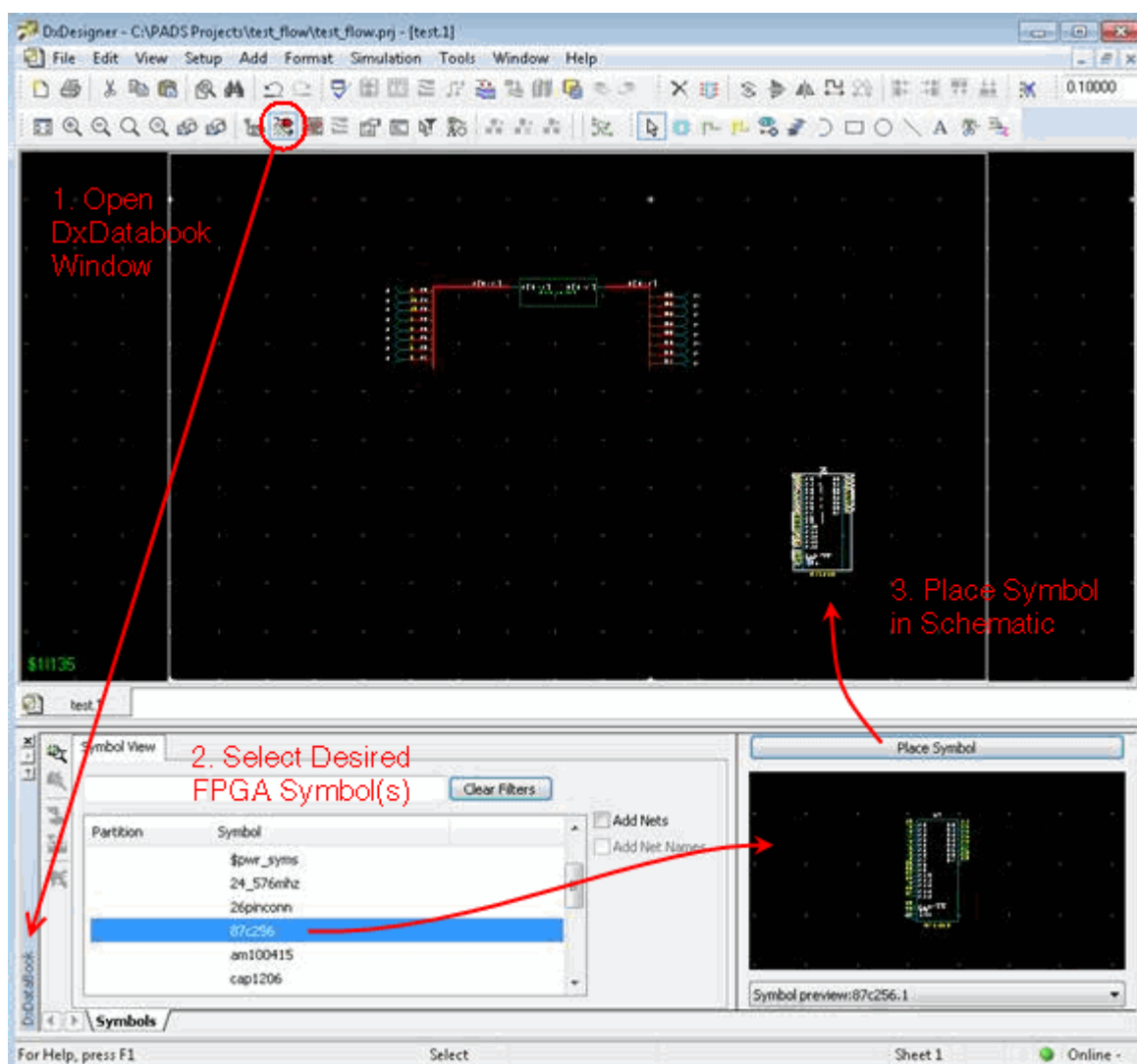
Once symbols have been exported, they are available in the local symbols partition, ready to be placed into the schematic. See “[Placing Symbols in an PADS Designer Schematic](#)” on page 115.

Placing Symbols in an PADS Designer Schematic

Use the following procedure to place symbols created in PADS I/O Designer into a PADS Designer schematic, and generate reference Designators for those symbols.

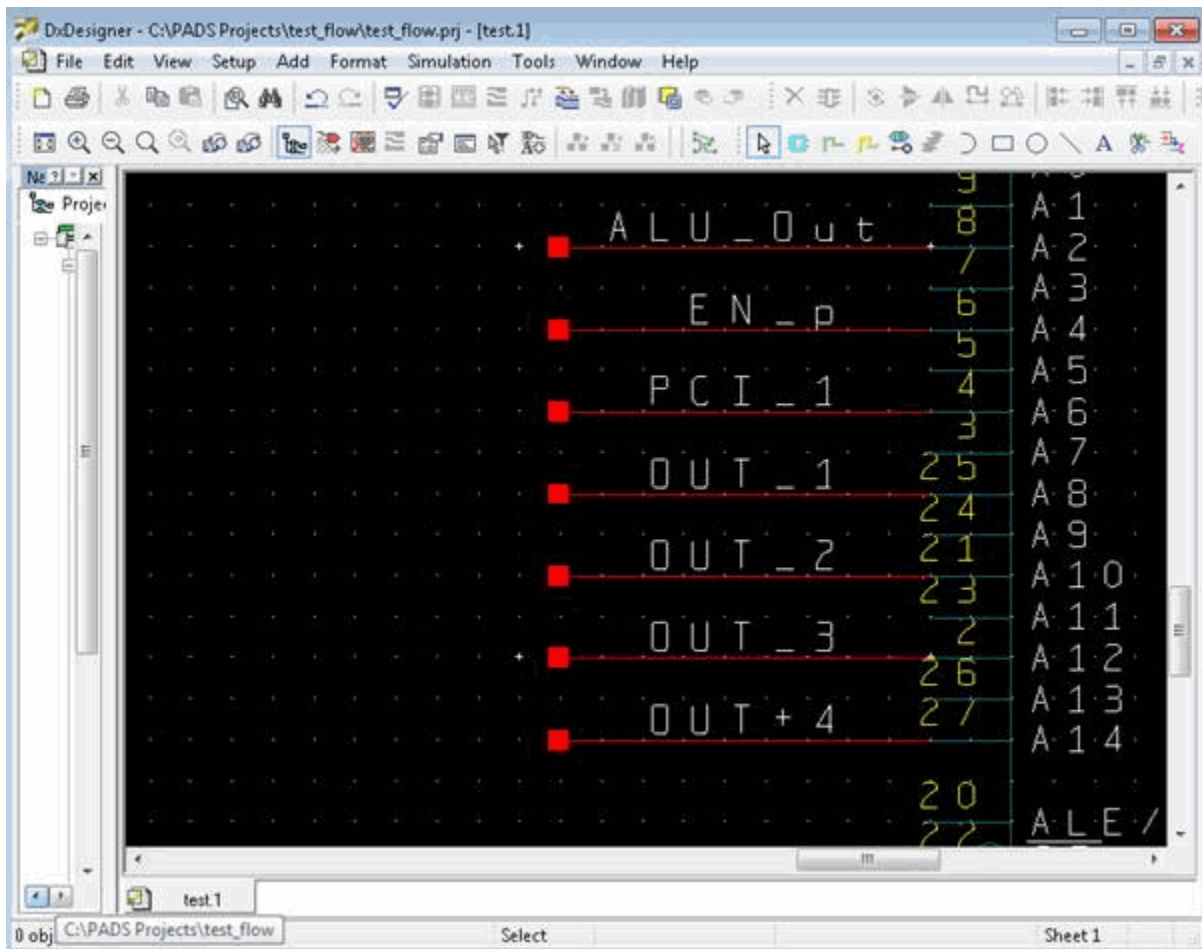
1. In PADS Designer, activate the **PADS Databook Window** and choose the **Symbol View** tab. Symbols can be selected from the **local symbols** partition. Symbols from the Central Library can be selected from the **Part View** tab.
2. Click **Place Symbol** to add the selected symbol to the schematic.

Figure 9-7. Placing Symbols in PADS Designer



3. Once the symbol has been placed on the schematic it can be wired to the desired nets. It is recommended that you use the “Net Stub” technique to wire symbols. This technique consists of placing only a small wire segment from the pins, and adding the appropriate Net Label text property to the wire segment.

Figure 9-8. Wiring a Device in PADS Designer Using the “Net Stub” Technique



4. In PADS Designer, select **Tools > PCB Interface**, and select Create Netlist for Layout, and click OK to generate a netlist.

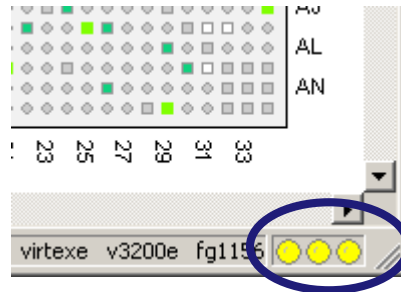
In PADS Designer, to determine the Reference Designator the netlist generation allocates to the symbol, double-click on the symbol to bring up the Properties Window.

Importing Symbols and Schematics

PADS I/O Designer can import symbols and schematics from external files. This allows you to modify symbols and schematics outside PADS I/O Designer using an external tool and then incorporate these changes into the symbol in PADS I/O Designer. The process of updating PADS I/O Designer symbols and schematics based on external tool information is called back annotation. When you make any changes to the symbol and switch back to PADS I/O Designer, the tool will inform you that the symbol needs to be updated by showing a yellow update indicator in the bottom-right corner. If you double-click the indicator, the [Synchronization Wizard](#) will prompt you to update symbols.

Similarly, if you make changes in the tool that require running the export process, PADS I/O Designer will notify you using the yellow synchronization indicator.

Figure 9-9. Yellow Synchronization Indicator



Schematic Update

Schematic update is used to update pin assignment changes made in PADS I/O Designer to the DxDesigner or DA/BA schematic database by selecting **Export > Schematic Update...**

Note You must leave enough space around FPGA symbols in the schematic for net stubs to be added by schematic update. If there is not enough room to add net stubs, schematic update returns an error message stating that some nets could not be added, and directs you to update the schematic manually.

This can be used as an alternative to exporting symbols and/or schematics using **Export > Schematic and Symbols** (see “[Exporting Symbols and Schematics](#)” on page 112). Using schematic update, any symbols which already exist on the schematic are preserved, they are not replaced with I/O Designer-created symbols.

Using this flow, the pin numbers of the symbols are not swapped. Instead, the net names and net stubs that have been placed on the schematic are swapped.

You can use this function with symbols developed outside PADS I/O Designer, as well as with symbols developed inside PADS I/O Designer.

Note Schematic Update does not support Interconnectivity Tables.

Prerequisites

To use this functionality you must initialize PADS I/O Designer and the load the FPGA database, including signals and pin assignments related to the component, from the schematic. You can use the following methods to load the signals and pin assignments:

- Use the PCB Design Wizard option, “Import Signals and assignments only” to import signals and assignments into PADS I/O Designer from the schematic without importing the symbol.
- Use an HDL and/or a Pin Report File for the specific device.
- Use a spreadsheet file with signals and assignments.

Note

In BA/DA, when running Schematic Update, you must close the sheet you are updating or remove the lock on the sheet. Otherwise, PADS I/O Designer skips the locked sheets, and the requested instance is not found, resulting in an error for each skipped sheet.

Additional Information

Schematic update allows the exporting of assignment changes to the schematic tool without changing any symbols. All changes are implemented by updating the connectivity between existing symbols. This section provides some additional information with regard to the process.

- Schematic update progress

Schematic update displays a dialog that indicates elapsed time as the process runs.

- Unassigned signals

Schematic update reports all unassigned signals at invocation to let you know that some pins will not be connected on the schematic. If you intentionally left the signals unassigned, you can ignore the warning and proceed to update the schematic.

- Missing pins

If there are missing pins in the schematic, the schematic update process stops, and an error message displays directing you to the UpdateSchematic.log for a list of missing pins.

- Saving Files on completion

When schematic update completes without errors, PADS I/O Designer immediately saves all files.

- Finding symbols for the FPGA component

Schematic update works only on nets connected to symbols describing the FPGA component on the schematic. The component is matched by the RefDes and Part Number attributes if they are specified in PADS I/O Designer. If RefDes is not specified, the part number is used. Either one of these attributes must be specified for schematic update to match any component on the schematic.

- Net name generation

Schematic update converts any signal name to a net name. For plain signals, this conversion does not change anything. For buses, PADS I/O Designer angle brackets are replaced with round or square brackets, or nothing, according to Net Name Delimiter setting in PADS Designer.

- **Updating nets**

When schematic update finds that a net connected to a given pin number has a different name to the one generated from the signal, a net is updated. The segment of the original net, which is connected to FPGA symbol, is cut from this pin. The new stub net is created with the right net. If the original net is already a net stub, it is not cut but its name is updated instead.

- **Supported use cases:**

Updating schematics created by the user: In order for schematic update to work correctly in this use-case, all PADS I/O Designer signals must have the same names as the nets connected to symbols representing the FPGA device on the schematic. This condition can be met by generating a signal list using the import PCB design wizard in PADS I/O Designer, an HDL file, or a spreadsheet file. If the signal source is different, you must validate that the signal names match.

Another requirement is that symbols matching the PADS I/O Designer component should have the same set of pin numbers as the FPGA device. Pins existing only in the symbols will not be updated. If some pins exist only on the FPGA device, no nets for their signals are generated on the schematic, and an error message displays directing you to the UpdateSchematic.log for a list of missing pins.

Updating schematics generated by PADS I/O Designer: Schematic update can be used to update PADS I/O Designer generated schematics. In this case, only the nets connected to PCB symbols are updated, as functional symbols have no pin number attribute.

Chapter 10

Optimizing the I/O Assignments

I/O Optimization Methods

PADS I/O Designer provides two methods of I/O Optimization. The I/O optimization procedure is much the same for the two methods, with the differences described below. The two methods are:

- I/O Optimization - Layout Database — Using FPGA part data and actual layout data to optimize I/O for all of the FPGAs on a board.
- I/O Optimization - PADS I/O Designer FPGA Database — Using FPGA database and layout data to optimize I/O for one database.

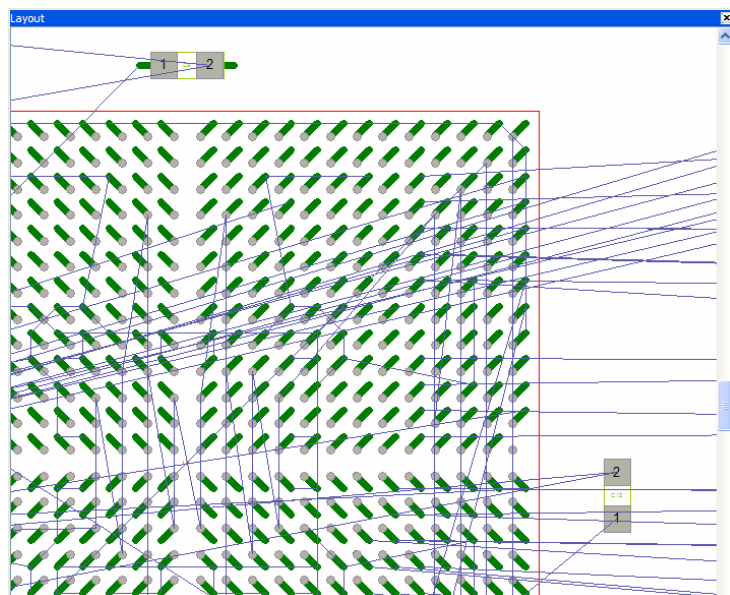
Fan-out Visibility

In the [Layout Window](#), it is possible to view fan-outs imported from the board.

Note

In PADS flow, netlines are drawn from pins or the ends of escape hangers or breakouts.

Figure 10-1. Fan-out Visibility



Layout Database

The layout database is an I/O optimization environment that allows you to optimize the I/O assignments based on a component orientation. Data is imported into the layout database from the schematic and the PCB layout tool or simply created from a netlist and PADS I/O Designer components. The editor within the database allows devices to be moved and new solution scenarios to be created. The best results can then be selected and propagated to the FPGA database.

Note



For each board in the database, there can be only one Layout database.

In order to work with a layout, add a layout database to a board within the project by selecting **File > Add to Board > New Layout**. You can only have one layout database per design.

When working within a layout database, the [Layout Window](#), [Pins List](#), [Connectivity List Window](#) and [Layout Scenarios Window](#) are used to *optimize* I/O assignments between components.

I/O Optimization

I/O Optimization is defined as utilizing the actual layout data to optimize the I/O. The step requires a PCB layout.

In PADS I/O Designer, the layout database can be populated from the PADS physical layout using **Import > Layout**.

Now, the I/O can be further optimized, new layout scenarios can be created and applied.

When loading a board from PADS, you can view the netlines from a pin-to-pin view or a pin to partial route view. This can be valuable when considering I/O alignment with a large bus. A pin-to-pin view may leave unwanted cross-overs based on routes that are already in place.

Note



Partial route traces and breakouts are taken into account during unraveling.

In Allegro flow, PADS I/O Designer does not show traces on the layout, so only unravel from pins is performed.

Synchronizing the Layout Database with a Design

Use the following procedure to import a layout from a PADS layout.

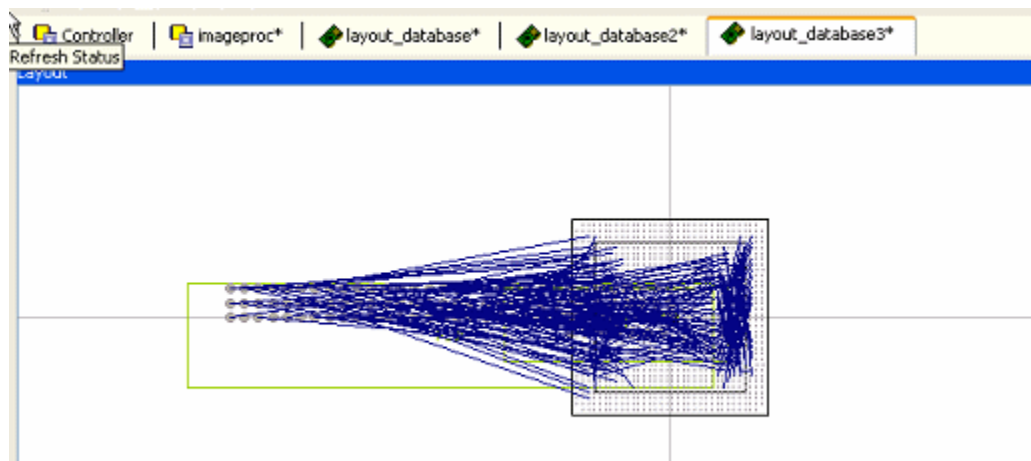
Prerequisite

After exporting the FPGA from PADS I/O Designer to the schematic; generate a netlist in PADS Designer, forward changes to PADS layout, and save layout in PADS.

1. Select **File > Database Properties** and, if necessary, click the **Netlist Source** page.
2. Select Schematic as the **Netlist source** and click **OK**.
3. In PADS I/O Designer, select **Import > Layout** to populate the layout database from the schematic.

The [Layout Window](#) shows the netlines in the design and the [Layout Scenarios Window](#) opens with an “Initial Scenario”, as shown in [Figure 10-2](#).

Figure 10-2. Components Imported Into Layout



4. With the layout database open, select **Tools > IOD Components**. The devices available for I/O optimization are listed here.

Visibility of each component can be turned off and on using the **Visibility** control, or mirrored by checking the **Mirror** control.

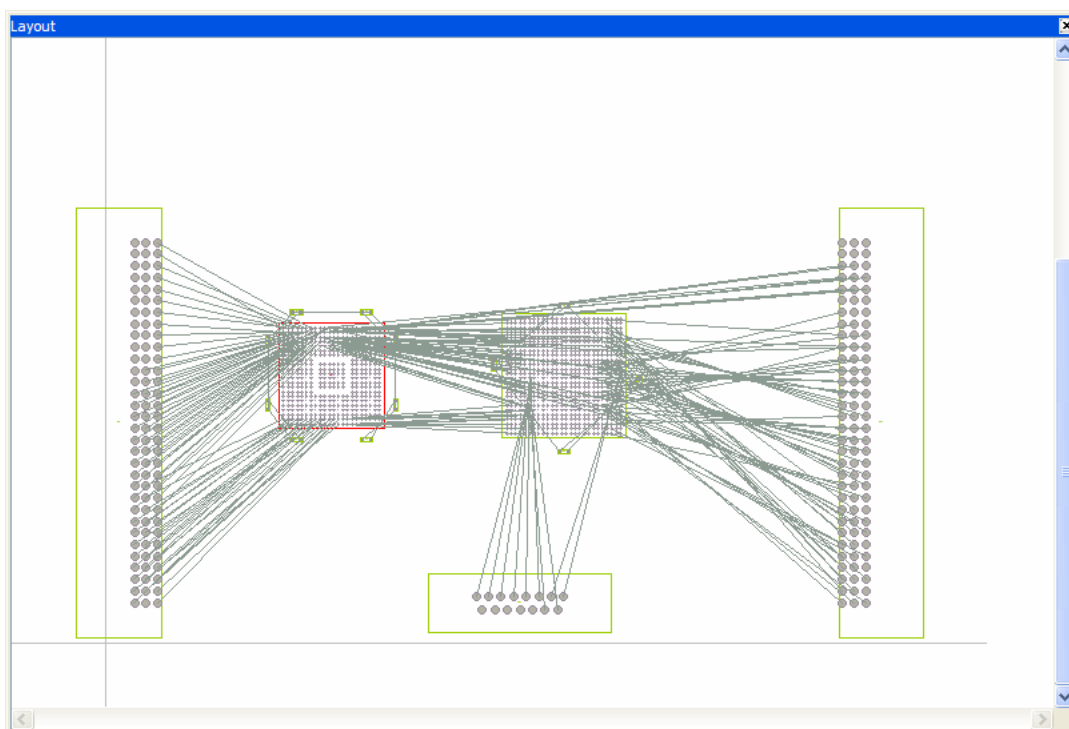
5. (Optional Step) You can filter the components, such that only those you want to use in the PADS I/O Designer optimization process are imported. Select **File > Database Properties** and click the **Component Filter** page. Use regular expressions to specify the reference designators of the unwanted components.

Components can now be moved around freely inside the [Layout Window](#). Next, optimization can be performed using [Unravel Nets](#) and different scenarios can be created until the best solution is found and applied (see “[Creating and Applying a Layout Scenario](#)” on page 129).


Layout Window

The Layout Window provides a graphical representation of the board component layout. Connectivity between components is shown using a logical representation of connectivity known as “rubber bands”. Rubber bands only represent the I/O connectivity between the components on the layout, not their physical implementation.

Figure 10-3. Layout Window



Select Pin/Net Mode

This mode is used to select multiple pins and nets in the Layout Window in order to run appropriate actions on the whole selection; for example, to unravel selected nets for the selected component. To activate this mode, click the  icon in the toolbar. Click the required component and then:

- Use the left mouse button to select individual pins and nets.
- CTRL + click to select multiple pins and nets.
- Drag a rectangle around the required pins and nets.

Show Traces


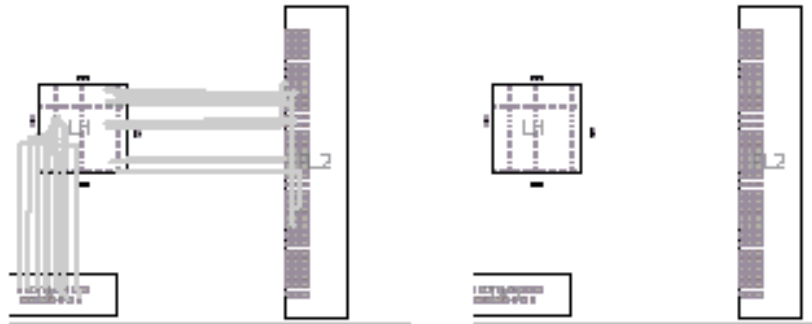

This mode allows you to show or hide physical traces between objects imported directly from the PCB layout. To activate this mode, click the  icon in the toolbar.

Figure 10-4. Show/Hide Traces




Show Netlines

This mode allows you to show or hide netlines. Netlines order is imported from PCB layout. MST and custom orders are indicated by different net colors. To activate this mode, click the  icon in the toolbar.

Connectivity List Window

Displays connectivity between components in a board or package layout when working within a [Layout Database](#).

The Connectivity List Window may be closed and opened at any time; to do this, either select the menu item **View > Windows > Connectivity List**, or click the  button on the view toolbar.

The Connectivity List contains the following information for each net or pin on the device:

Table 10-1. Connectivity List Window Contents

Column	Description
Net/Signal Name	Displays the net name and all pin names connected to the net.
Component	Displays the component Reference Designator to which the pin belongs.
Pin Number	Displays the number of the pin connected to the net.

When a net is selected in the Connectivity List Window, it is automatically highlighted in the [Layout Window](#).

Adjusting Component Orientation

In the [Layout Window](#), you can move components around, rotate them and mirror them in order to achieve the best orientation for the design.

To **move** a component, select it and drag it to a different position.

To **rotate** a component, select it, right-click and select **Rotate 90**, **Rotate 180** or **Rotate 270**.

To **mirror** a component, select it, right-click and select **Horizontal Mirror**.

Once you begin moving components, a new layout scenario is created and the changes you make are listed in the [Layout Scenarios Window](#).

Unravel Nets

Unravel Nets optimizes signal to pin assignments by shortening nets and eliminating crossovers on the PCB or layout. This allows much more efficient routing of nets in the physical design tool. The following sections describe unraveling nets:

- [Unraveling Procedure](#)
- [Unravel All FPGAs](#)

Note



I/O Designer does not perform automated pin swapping on Multi-Gigabit Transceivers (MGT) signal types due to implementation specific pin placement considerations.

Unraveling Procedure

Use the following procedure to perform unraveling:


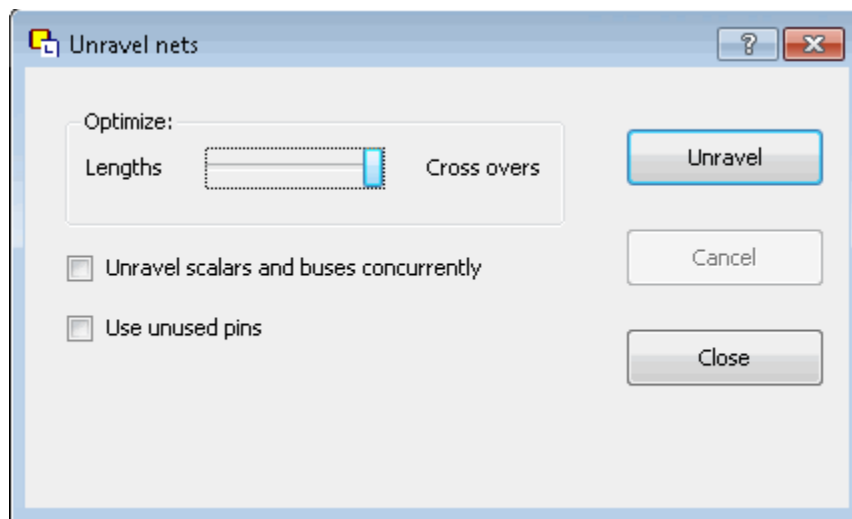
1. In the [Layout Window](#), select the component and / or nets, and either right-click and select **Unravel Nets**, or click the unravel button  to open the Unravel Nets Dialog.

Figure 10-5. Unravel Nets Dialog



2. Check **Unravel scalars and buses concurrently** in order to use all signals selected for unravel as one package of nets and pins (*optional*).

When this option is checked, all signals selected for unravel are used as one package of nets and pins, allowing, for example, a scalar net to be swapped with one of the bits of a bus. When this option is not checked, bus members are unraveled separately within the bus, and scalar nets are unraveled with other scalar nets.

3. Check **Use unused pins** to include unoccupied pins in the unraveling process (*optional*).

When this option is not checked, the unravel process uses only the currently occupied pins, resulting in a set of swaps made on signals. When this option is checked, the algorithm uses both the currently used pins and the unassigned pins, allowing much more flexibility in finding optimal results.

4. Click **Unravel** to begin the unraveling process with the current options.

Any user-defined rules set for the device are taken into account during unraveling. All signals marked as **Locked** in the [Signals List](#) are not moved during unravel process

Select the **Cancel** button at any time during the process to stop unraveling.

5. Click **Close** to save the current unravel settings and exit the dialog.

[Figure 10-6](#) and [Figure 10-7](#) show examples of nets before and after unraveling.

Figure 10-6. FPGA Nets Prior to Unraveling

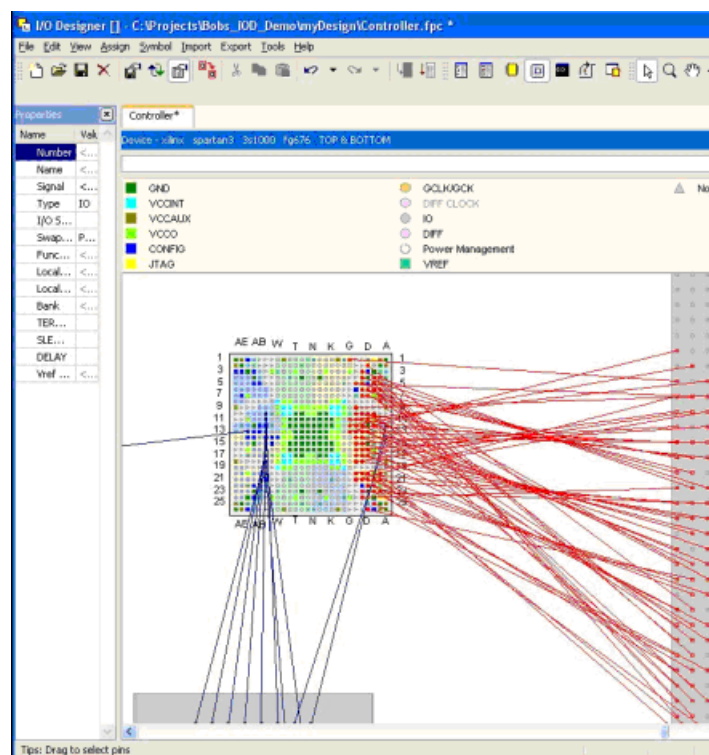
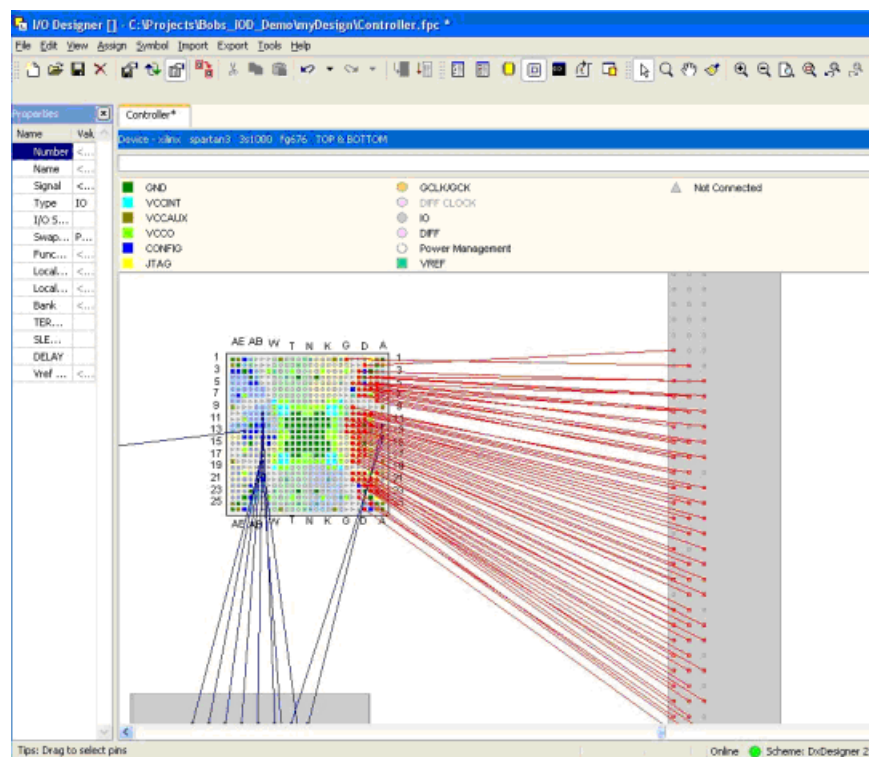


Figure 10-7. FPGA Nets After Unraveling



Unravel All FPGAs

The **Edit > Unravel Nets > Unravel All FPGAs** menu selection unravels all FPGAs in one step. A progress bar tracks progress of the unraveling.

Layout Scenarios Window

A scenario is a group of component placement commands and signal assignment commands within a layout database. A number of different scenarios can be created, and you can switch between them to explore different variants. Each scenario is independent and does not take effect to the design until it is applied.

Scenarios allow the exploration of different layout solutions which may consist of different component orientations (e.g. connector located on the left or right side of the device) and different applications of unraveling. The I/O assignment can be optimized based on different component orientations and have the best solution propagated to the PCB layout process.

The first entry in the Layout Scenario window is the initial scenario - the initial layout of the design as it has been imported. When changes are made to component placement or signal assignment in a layout database, a new scenario is created, and will appear in the Layout Scenarios window. A green arrow next to the scenario name indicates the currently active scenario.

Creating and Applying a Layout Scenario

Use the following procedure to work with a layout, create and apply layout scenarios:

1. Manipulate the components in the layout. A new scenario will appear in the [Layout Scenarios Window](#), named Scenario 1. The changes made are listed in the **Changes** column.
2. Select **Initial Scenario** in the [Layout Scenarios Window](#), and create another scenario in the same way. This scenario will be named Scenario 2.

You now have an initial scenario, which has the initial component placements and signal assignments, and 2 scenarios, each with different modifications.

3. To switch between scenarios in the [Layout Window](#), right-click on a scenario and select **Activate**.

This will not apply the scenario, it just allows you to view and manipulate it. You can continue to make changes to each scenario.

4. Upon creation of a suitable scenario, right-click on the scenario in the [Layout Scenarios Window](#) and select **Apply Scenario**.

The changes recorded in the scenario are applied to the layout, and the FPGA database for the device is updated with the new pin assignments.

Multi-chip PCB Optimization

Note



If you do not hold a multi-device license, then when applying optimization scenarios in a layout database, you can work with only one device at a time.

No Multi-chip License Held

1. Manipulate the components within the layout and perform unraveling on the selected device.
2. Once you are satisfied with the component orientation and the I/O assignments, apply the scenario. See [“Creating and Applying a Layout Scenario”](#) on page 129 for more information.

The FPGA database will be updated with the new pin assignments. After the scenario is applied, the layout editor is returned to the “Initial Scenario”. The Initial Scenario does NOT reflect the changes on the previously selected device. All other scenarios are deleted.

3. Go to the FPGA database and create or update its symbols. See [“Creating, Editing and Updating Symbols and Schematics”](#) on page 95 for more information.
4. Export symbols and schematics for the device by selecting **Export > Schematic and Symbols**, and forward annotate the changes to the PCB tool.

You can now import the PCB layout into the layout database in order to optimize another device with the results of the first device optimization.

Multi-chip License Held

1. Manipulate the components within the layout and perform unraveling. See [“Unravel Nets”](#) on page 126 for more information. You can also use [Unravel All FPGAs](#) to unravel.
2. Once you are satisfied with the component orientation and the I/O assignments on the selected device(s), apply the scenario. See [“Creating and Applying a Layout Scenario”](#) on page 129 for more information.

All FPGA databases are updated with the new pin assignments. After the scenario is applied, the layout editor is returned to the “Initial Scenario”. The Initial Scenario does NOT reflect the changes on the previously selected device. All other scenarios are deleted.

3. Go to the FPGA database and create or update its symbols. See [“Creating, Editing and Updating Symbols and Schematics”](#) on page 95 for more information.
4. Export symbols and schematics for the device by selecting **Export > Schematic and Symbols**, and forward annotate the changes to the PCB tool.

You can now import the PCB layout into the layout database in order to optimize another device with the results of the first device optimization.

Chapter 11

Data Exchange

PADS I/O Designer allows you to exchange a wide variety of design data with external systems such as HDL Entry tools, Synthesis tools, Place & Route tools and Design Definition tools. This chapter describes this data exchange in detail.

- [“HDL and EDIF/XML Files”](#) on page 133
- [“Synthesis Constraints Files”](#) on page 134
- [“Place and Route Constraints Files”](#) on page 135
- [“FPGA Xchange Files”](#) on page 136
- [“Integration with PADS Designer”](#) on page 138
- [“Integration with Pads Layout”](#) on page 147
- [“Constraints Bus Members Generation”](#) on page 149

Note



Before generating a file, its name must be entered in the Database Properties dialog.

Whenever possible, PADS I/O Designer does not overwrite the contents of an existing file. For instance, if a constraints file is generated, and the file already exists, the file is read, modified to reflect the current pin assignments, and then written back. In other words, PADS I/O Designer preserves constraints created by external tools.

HDL and EDIF/XML Files

PADS I/O Designer allows you to work with HDL files in the following way:

- Read signals from an HDL/Netlist file, see [“Importing I/O Signals Using HDL”](#) on page 70.
- Create signals and let PADS I/O Designer generate an HDL/Netlist file
- Update signals from an HDL/Netlist file

To generate an initial VHDL entity or Verilog module from PADS I/O Designer, select **Export > VHDL Entity** or **Export > Verilog Module**. Only ports in entity/module declaration are generated, without any underlying architecture. This allows manual addition of signals within PADS I/O Designer (see the section called Adding, removing, and renaming signals).

Subsequently an HDL file is generated as a starting point for further design, that is, writing architecture/module contents.

PADS I/O Designer monitors HDL files using the [Synchronization Wizard](#). Whenever the HDL file is changed and the ports of the unit displayed in the [Signals List](#) are modified, the Signal List is updated to reflect the actual unit declaration.

Recognizing Differential Signals

PADS I/O Designer recognizes differential signals in an HDL source file. To enable this, select **Setup > Settings + Advanced** and under **HDL source**, select the **Recognize differential signals** option.

PADS I/O Designer recognizes differential signals in an HDL design if the differential buffers are instantiated in the top-level design unit. A successfully recognized differential signal is displayed as a single signal in the [Signals List](#). The signal may then be assigned to a pair of differential pins. If the **by regular expressions** option is selected PADS I/O Designer will try to recognize differential signals using regular expressions set in **Recognition of differential signal with regular expressions**.

Synthesis Constraints Files

PADS I/O Designer allows you to read and generate synthesis constraints files. To generate synthesis constraints file from PADS I/O Designer, choose **Export > Synthesis Constraints File** from the main menu. PADS I/O Designer supports reading and writing constraints files for the following synthesis tools:

- Leonardo Spectrum (.ctr)
- Precision (.sdc)
- Synopsys (.sdc)
- Synplify Pro (.sdc)
- Xilinx XST

Place and Route Tools

PADS I/O Designer allows you to read in a Place & Route constraints file or Pin Report file, change pin assignments, and regenerate the constraints file to be read by the Place & Route software. PADS I/O Designer supports reading and writing constraints files for the following Place & Route tools:

- Libero 11.1 / Actel Designer 10.0 / Actel Designer 9.1 / Actel Designer 9.0 SP2

- Altera Quartus II 13.0 / Altera Quartus II 12.1 / Altera Quartus II 9.0 SP2
- Lattice Diamond 2.2 / Lattice ispLEVER 8.1
- Libero 11.1 / Designer 10.0
- Xilinx ISE 14.4 / Xilinx ISE 14.6 / Xilinx ISE 14.7 / Xilinx Vivado 2012.4 / Xilinx Vivado 2013.2 / Xilinx Vivado 2013.3 / Xilinx ISE 10.1 SP3

Note

PADS I/O Designer supports older FPGA tool versions but these are not included in the standard installation:

Actel Designer 8.6 SP1 / 8.5 SP2 / 8.4 SP2 / 8.3 SP1 / 8.2 SP2 / 8.1 SP2 / 8.0 SP2 / 7.3 SP2 / 7.2 SP2 / 7.1 / 7.0 SP1 / 6.3 / 6.2 SP2 / 6.1 SP1 / 6.0 SP2 / 5.2 SP1 / 5.0 SP1

Altera Quartus II 10.1 / 10.0 SP1 / 9.1 SP2 / 8.0 SP1 / 7.2 SP3 / 7.1 SP1 / 7.0 / 6.1 / 6.0 SP1 / 5.1 SP2 / 5.0 SP2 / 4.2 SP1 / 4.1 SP2 / 4.0 SP1 / 3.0 SP2, MaxPlus II 10

Lattice ispLEVER 8.0 / 7.2 SP2 / 7.2 SP2 / 7.1 SP1 / 7.0 SP2 / 6.1 SP2 / 6.0 SP1 / 5.1 SP2 / 5.0 SP1

Xilinx ISE 14.3 / 14.2 / 14.1 / 13.1 / 12.4 / 12.3 / 12.2 / 12.1 / 11.5 / 11.4 / 11.3 / 11.2 / 11.1 / 9.2 SP4 / 9.1 SP3 / 8.2 SP3 / 8.1 SP3 / 7.1 SP1 / 6.3 SP3 / 6.2 SP3 / 6.1 SP3

These libraries can be provided upon request.

Place and Route Constraints Files

Constraints files pass data such as pin assignments from PADS I/O Designer to the Place & Route tool. To generate a Place & Route constraints file from PADS I/O Designer, select **Export > P&R Constraints File**. Constraints file formats vary between vendors. Below is the list of Vendor - Constraints file formats.

- Actel (*.pin*, *.gcf* or *.pdc*)
- Altera (*.csf* or *.qsf*)

Caution

If you use the XSTL_INPUT_ALLOW_SE_BUFFER option in a QSF file, the QSF file must be imported before the PIN file so that all assignments are imported properly.

- Xilinx (*.ucf*)
- Lattice (*.lpf*)

PADS I/O Designer monitors Place & Route constraints files using the [Synchronization Wizard](#), but they are not selected for update by default. During the constraints file generation process, PADS I/O Designer writes necessary information such as pin assignments, I/O Standards and the target device, to the constraints file. It does not overwrite any other information present in the file. Therefore, it is safe to have a constraints file created in an external program, load the file into PADS I/O Designer, modify it, and then save your changes by selecting **Export > P&R Constraints File**.

Note

It is important to save the FPGA database following an import of constraints files (*.lpf*) or pin report files (*.pad*) from Lattice. This ensures that the correct bus syntax is applied to all bus members.

Pin Report Files

Pin report files format is vendor-dependent:

- Actel (*.rpt*)
- Altera (*.pin*)
- Xilinx (*.pad* or *.csv*)
- Lattice (*.pad*)

These files are generated by Place & Route software. PADS I/O Designer uses these files to update pin assignments, signal data and other settings based on the output of the Place & Route tool. PADS I/O Designer does not generate pin report files.

Note

It is important to save the FPGA database following an import of constraints files (*.lpf*) or pin report files (*.pad*) from Lattice. This ensures that the correct bus syntax is applied to all bus members.

FPGA Xchange Files

FPGA Xchange files contain various design information that may be exchanged between FPGA and PCB tools including:

- Design name
- Device information (vendor, package, speed)
- Pin information (names, numbers, IO standards, assignments)
- Differential pins information

- Signal information (name, direction)
- Swap Groups information

To generate an FPGA Xchange File, go to **Export > FPGA Xchange File**.

FPGA Device Library Path

You can load a device library update outside the install tree. Update the FPGA location library variable by using **Setup > Settings + Paths** and double-clicking the FPGA library location row. You can browse to your path by clicking the ellipsis located just to the right-side of the Path column.

Timing Report Files

Timing report files contain timing information generated by Place & Route software. The information read by PADS I/O Designer from these files include timing constraints, timing actuals, signals, etc.

Timing report file format is vendor-dependent:

Altera: RPT

Actel: Not Available

Lattice: TWR

Xilinx: TWR

Design Rule Check for Altera Devices

PADS I/O Designer enables you to run the Design Rule Check for Altera devices. Go to **Tools > Design Rule Check**.

A grayed-out DRC option in PADS I/O Designer indicates that the option is not available for the current vendor or part.

SSO Check

PADS I/O Designer enables you to run the Xilinx SSO (Simultaneous Switching Output) Check.

PADS I/O Designer supports many, but not all, older families. A grayed-out SSO option in PADS I/O Designer indicates that the option is not available for the current vendor or part family.



Note

For in-depth information on Xilinx SSO, refer to the Xilinx documentation available on the Xilinx web page.

The settings for the SSO Check can be found in Database Properties, on the Advanced tab. These are:

- SSO Package Allowance
- SSO Bank Threshold

To run the SSO Check, go to **Tools > Simultaneous Switching Output Check**.

Automatic SSO Check

You can set up the SSO Check to run automatically when an assignment has changed, by selecting **Setup > Settings**, clicking **Advanced**, and checking the box for **Automatic SSO Check**.

Generating Device List

PADS I/O Designer allows you to generate the full device list available by executing the *generate_all_devices* command in the Console Window.

Exporting Symbols to a Library

You can use the following steps to export symbols created in PADS I/O Designer to a library:

1. Export the symbol to the local symbol partition of your PADS Designer project.
2. Export the symbol to PADS Databook or Symbol Editor.

Integration with PADS Designer

Before you start working with PADS I/O Designer and PADS Designer, ensure that the following conditions are satisfied:

- PADS Designer installed and path to its executables is set in PADS I/O Designer under **Setup > Settings + Paths**.
- In PADS I/O Designer, the **Export type** is set to PADS Designer under **Setup > Settings + Symbol Editor + Export**.

Note



PADS I/O Designer does not support multiple instances of the same PADS I/O Designer component.

Importing an Existing PADS Designer Project

1. Go to **Setup > Settings** and make sure the scheme is set to PADS Designer.
2. Go to **Import > PCB Design Wizard**.
3. In the **Project Path** field, enter the path to the required PADS Designer project or click **Browse** to navigate to it.
4. Click **Next**.
5. Select the symbols that you want to import. If you want PADS I/O Designer to generate a functional symbol for each PCB symbol, select the **Generate Functional Blocks for PCB Symbols** option. If you want to change the symbol type, you can do it by double-clicking the Type column in a symbol entry.

Note



By default, PADS I/O Designer lists only symbols with the following attributes:

DEVICE (for netlist flow) and PKG_TYPE (for PADS Designer)

To browse all available symbols, set the Show All Symbols option.

6. Click **Next**.
7. Set the device that you want to use for the imported design. PADS I/O Designer tries to recognize the correct device to use, but still gives you the ability to choose a different one.
8. Click **Finish** to complete the import process.

Importing Symbols from PADS Designer

PADS I/O Designer imports symbols from the PADS Designer design. When importing symbols that were not generated from PADS I/O Designer, the symbol files are searched for the following port attributes:

- **Pin Label:** sets the Port Label property in PADS I/O Designer
- **# attribute:** sets the Pin Name property in PADS I/O Designer if the value matches Pin Name in the selected device
- **PINTYPE attribute:** sets the Pin Type property in PADS I/O Designer

If some of the pin names in the imported symbol do not match pins in the currently selected device, PADS I/O Designer displays a warning. If more than 50% of all pins do not match, PADS I/O Designer informs you that the device does not match the symbol. If the imported symbol has been generated from PADS I/O Designer, the following information is read from PADS Designer symbol files:

- | | |
|-------------------|--------------|
| • Port Label | • Pin Type |
| • Signal Name | • Port Type |
| • Pin Number | • Port Shape |
| • Pin Function | • Length |
| • PCB Signal Name | • Attributes |

To import symbol from PADS Designer, follow these steps:

1. Choose **Import > Symbols from Board** from the main menu.
2. Choose the *.prj* file that points to the PADS Designer project that you want to import symbols from.
3. Choose the desired symbols from the list, select whether they are PCB symbols and if so, to which functional symbol each of them should belong. If you don't want a particular symbol to belong to any of the existing functional symbols, put the symbol under the top-level entry.
4. To complete the import operation, click **Import**.

Exporting Symbols/Schematics to PADS Designer

Before you start exporting symbols/schematics from PADS Designer to PADS I/O Designer, make sure that you have set your environment correctly.

All symbols created during symbols / schematics export from PADS I/O Designer to PADS Designer are saved as local symbols.

An error can occur if you attempt to export a PADS Designer project created in PADS I/O Designer and do not have Special Components set correctly. The error message that displays includes instructions on how to fix the error:

```
Cannot export schematic: 'Special Component' is not set: PORT_IN  
To change offpage connector settings : run PADS Designer with current  
project, choose 'Setup' from main menu than choose 'Settings', go to the  
'Project' category and open 'Special Components' page.
```

What is Exported to PADS Designer?

- Hierarchy - In case of schematics export, all dependencies between symbols are exported.

Note



During symbol and schematics export, PADS I/O Designer does not generate nets for CONFIG pins. If you want the connection to be created, you have to use the Assign PCB Signal option.

- Symbol properties - During export, symbol properties information is saved to the symbol files in the sym directory. Below is the list of symbol properties available in PADS I/O Designer and their corresponding properties in PADS Designer:
 - Port Label: PIN
 - Signal Name: SIGNALNAME
 - Pin Number: PINNUMBER, PORTID
 - Pin Function: PINFUNCTION
 - PCB Signal Name: IMPLSIGNALNAME
 - Pin Type: PINTYPE
 - Dir: ignored
 - Port Type: exported
 - Port Shape: exported
 - Inverted: exported
 - Length: exported
 - Attributes: exported
- Graphical attributes - Graphical attributes are saved to the symbol file. They include:
 - Colors
 - Fonts: font family, font color, and font size are all exported. However, since there are differences in fonts between PADS I/O Designer and PADS Designer, some fonts may be changed during export. The tool provides special font for use with PADS Designer called SCHEMATIC_DX. It is strongly recommended that you use this font while working with PADS I/O Designer and PADS Designer symbols / schematics.
 - Graphics: graphical items are all exported to PADS Designer.

Updating Symbols/Schematics from PADS Designer

During the update, PADS I/O Designer looks into the PADS Designer iCDB database. PCB symbols mappings are read from the symbols.

What is Updated from PADS Designer?

- **Hierarchy** - Only those schematics / symbols that were generated by PADS I/O Designer are updated. If you have added a symbol to the PADS Designer project and want it to appear in PADS I/O Designer, you have to import this symbol manually by using the Import Symbol command from the menu **Import > Symbol**.
- **Symbol Properties** - the following symbol properties are updated from PADS Designer:
 - Port Label
 - Signal Name
 - Pin Number
 - Pin Function
 - PCB Signal Name
 - Pin Type
 - Port Type
 - Port Shape
 - Length
 - Attributes
- **Graphics Attributes** - All graphical attributes like colors, fonts and graphical items are updated from symbol files. For details, see the section on exporting graphical attributes.

PADS Designer Usage Scenarios

This section details the use of PADS I/O Designer in a PADS Designer flow. The section covers three possible usage models:

- **PADS I/O Designer normal mode (schematic export)**
Full flexibility mode where PADS I/O Designer is used to generate symbols/fractures, schematics and the PCB pin-swap data for layout. The advantage of using this flow is the unconstrained PCB layout optimization ability.
- **PADS I/O Designer with existing symbols (schematic export)**

This is the most restricted mode. It uses existing schematic symbols and the associated PCB mapping data from the symbol library. PADS I/O Designer is used to generate the schematics but there is no swap information exported to the symbols during the export process. In this case, the data must match the PCB mapping data from the library. Using this method limits the ability to optimize the PCB layout to PADS I/O Designer capabilities. The only way to do it is by using PADS I/O Designer's Multi-Component View feature.

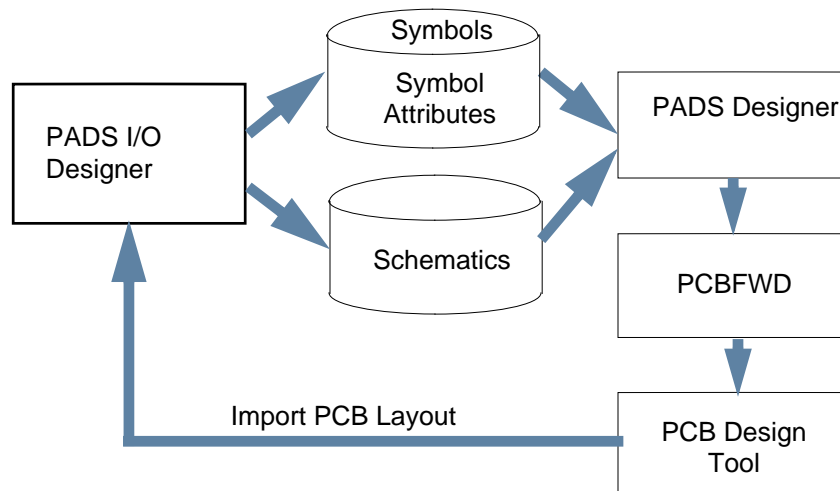
- PADS I/O Designer with existing symbols (schematic update)


This usage model provides the most design structure flexibility. It supports both flat and hierarchical design structures, and divides the design process along traditional team boundaries (librarian, PCB functional designer, PCB physical designer) in alignment with historical design practices. It has the advantage that the FPGA is seen by the PCB as a normal component in your library.

Scenario 1: PADS I/O Designer Normal Mode - PADS Designer

In this mode of operation PADS I/O Designer creates both the symbol data and the PCB mapping data for the design. This allows the greatest level of flexibility when optimizing I/O and allows for fast iteration through the design process. The schematics and symbols are generated by PADS I/O Designer and are simply forward annotated to the PCB design. The process is shown in the flowchart below followed by the steps for this flow.

Figure 11-1. PADS I/O Designer Normal Mode - PADS Designer



After mapping the signal names to the appropriate pins in the device, the Symbols Generator can be used to quickly generate the PCB level symbols and PCB data ready for layout. To launch the Symbols Generator, click the  icon on the toolbar or choose **Symbol > Symbols Generator** from the menu.

For a detailed description of the Symbols Generator, see the section called “[Symbols Generator](#)” on page 96.

The symbol may be fractured according to one of the schemes supported by PADS I/O Designer or created as a custom fracture set.

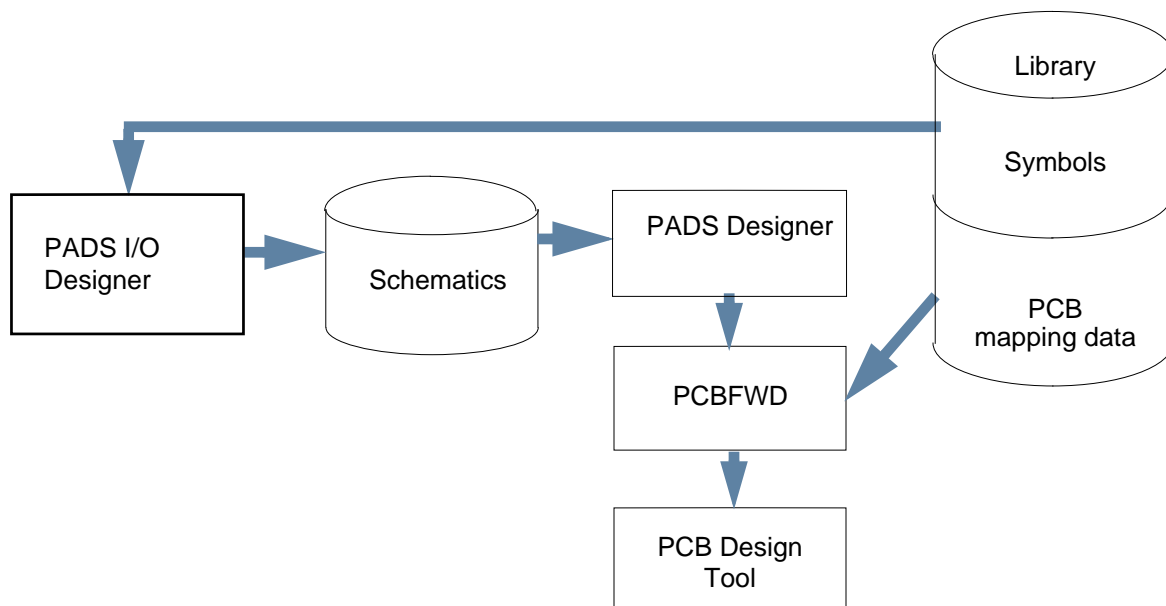
PADS I/O Designer will then generate the required symbols and schematics during the export process, along with a design specific set of pinswap attributes. To export schematics and symbols, use the **Export > Schematic and Symbols** menu entry.

The design is now ready to forward annotate to your PCB design tool.

Scenario 2: PADS I/O Designer with Existing Symbols with PCB Mapping Data - PADS Designer

In this mode of operation PADS I/O Designer uses a set of library data from the symbol library including the PCB mapping data for the existing schematic set. This limits the capability to perform optimization during layout using the PCB tools as the PDB entry in the symbol library will probably not contain any pin-swap information. The process is shown in the flowchart below followed by the steps for this flow.

Figure 11-2. PADS I/O Designer with Existing Symbols and PCB Mapping Data



PADS I/O Designer can import a set of existing symbols and use it to map the pin to signal assignment of the device to the generated schematics. PADS I/O Designer can import from a project file, schematic file or library using the **Import > Symbols from Board** menu item.

Once the symbols are available they can be imported into PADS I/O Designer. In order to associate the symbols with the correct schematic it is necessary to have the top-level functional symbol created in PADS I/O Designer; If it does not exist, then the imported symbols will not have an associated functional block. The functional block can be generated as soon as the HDL definition has been loaded into PADS I/O Designer.

The PCB Symbol check box should be set. Make also sure that the **Readonly** symbol option is checked this time since this will fix the swapping ability that is read from library symbols, and therefore fix the swapping to the way it is defined in the symbol library. Setting the Read-only symbol option also gives the DEVICE attribute a correct value that matches the part number in the symbol library, which causes the pin numbers and mapping data to be taken from the symbol library PDB during netlist generation. Once the symbols are imported, PADS I/O Designer will map signal associations to the symbols.

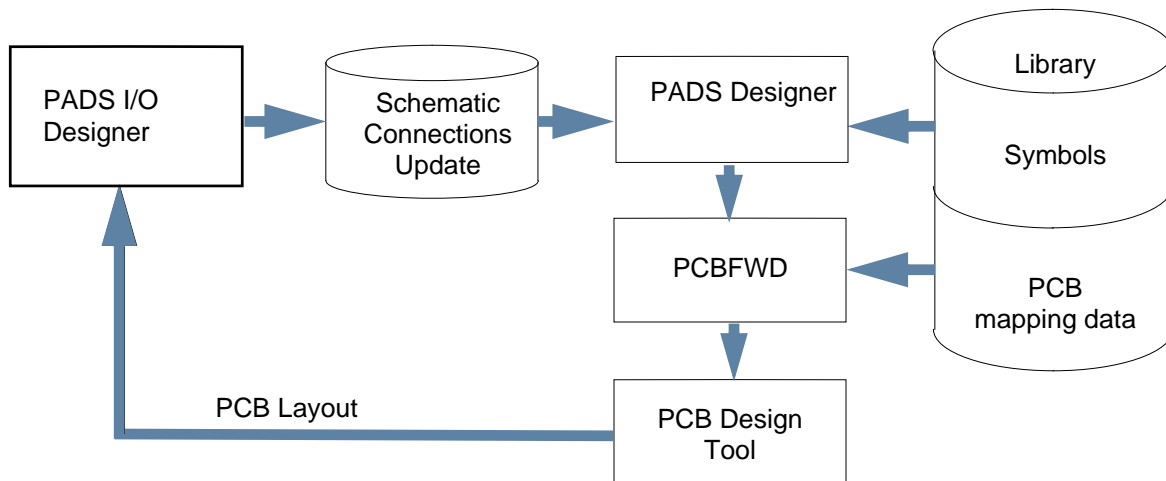
After the symbols have been imported and the assignment completed the schematics must be generated using the **Export > Schematic and Symbols** menu. PADS I/O Designer generates the schematic for the device complete with the correct connectivity.

The design then forward annotates to layout using the entries in the symbol library. Pin swapping will not be available in the PCB design tool, but you can use the device window and layout view in PADS I/O Designer to achieve optimization.

Scenario 3: PADS I/O Designer with Existing Symbols (Schematic Update)

In this mode of operation, supporting both flat and hierarchical design structures, PADS I/O Designer simply updates the connectivity of the schematic. This flow style divides the design process along traditional team boundaries: Component Librarian, PCB Functional Designer, PCB Physical designer aligning with traditional design practices, so PADS I/O Designer usage in this mode is simpler from both a learning perspective and a command set perspective. The process is shown in the flowchart below followed by a description of this flow.

Figure 11-3. PADS I/O Designer with Existing Symbols (Schematic Update)



FPGA Symbols and the PDB are normally placed in the symbol library and re-used across multiple PCB projects. Symbols and the PDB can also be used locally, within the project. The FPGA (as seen by the PCB) is a normal component in your library, just like any other component you use on your PCB. xDM Library supports FPGA components, since they are seen as normal PCB components.

FPGA Symbols are manually instantiated into the design, just like any other component in the design and can be mixed with any other components on the same schematic sheet. The update process updates the connectivity but does not re-generate the schematic, so your edits are never lost.

Using the Interconnectivity Table Editor

You can use PADS Designer to create and manage a design from the Interconnect Table Editor. PADS I/O Designer can create an Interconnect Table (ICT) in addition to a schematic.

Within an PADS Designer design, if the top level is a schematic, PADS I/O Designer can create a functional block with either a schematic or ICT below. If the top level of the design is an ICT, PADS I/O Designer can only create a functional component (analogous to a functional block) and an ICT below.

To export to a connectivity table, choose **Export > Interconnectivity Table**.

Note



Before running the ICT export, a PADS Designer project path should be set and a functional block must be created. All PCB fractures must be present.

Integration with Pads Layout

PADS I/O Designer reads the PCB layout information contained in the Pads Layout layout file. This allows you to view the connections between the current device and the surrounding elements on the PCB schematic.

Figure 11-4. PADS Designer + PADS

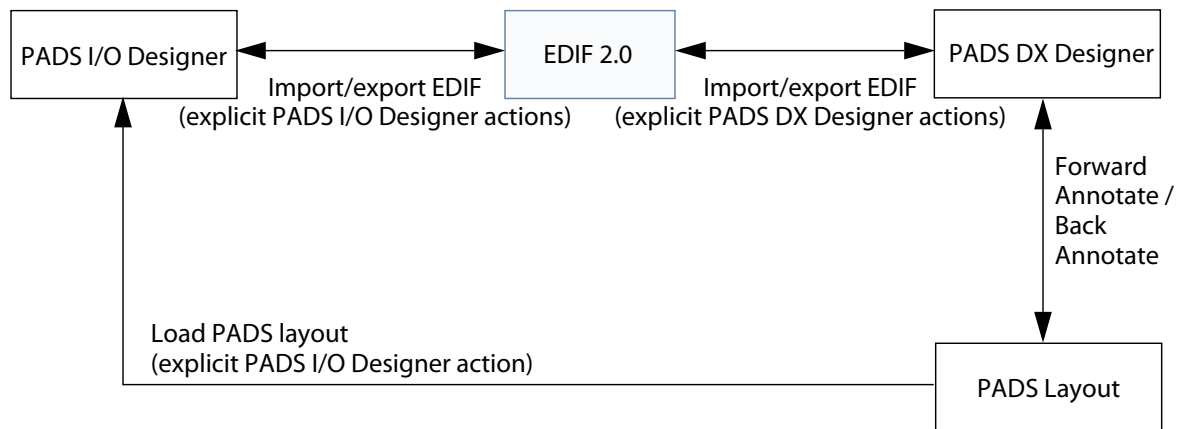
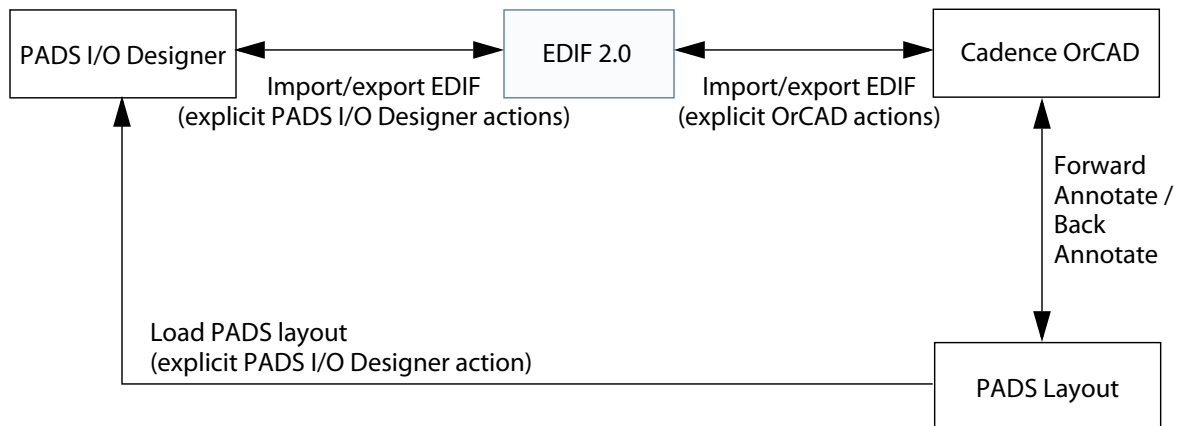


Figure 11-5. Cadence OrCAD + PADS



Note



PADS I/O Designer does not export default I/O standards to Constraint Manager. You must explicitly set signal I/O standards to export the standards to Constraint Manager.

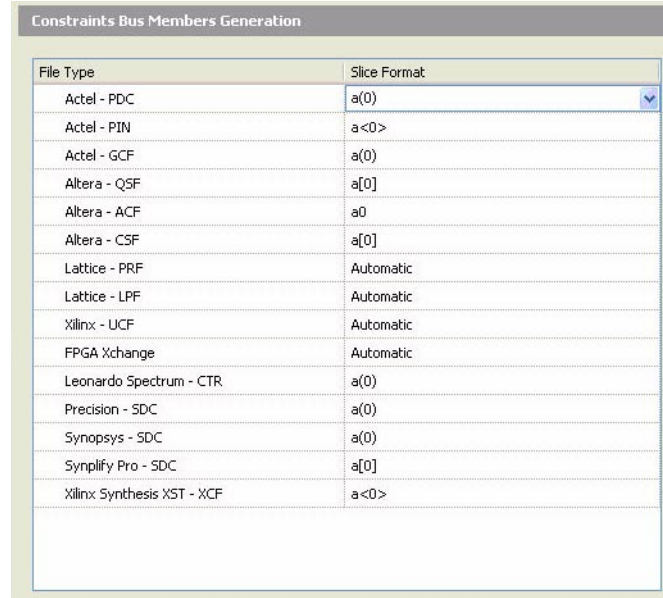
During schematic export, PADS I/O Designer recreates the schematic with related FPGA components so that the PADS I/O Designer connectivity changes are transferred to PADS Designer. To preserve Constraint Manager data:

- When optimization is performed in layout context (layout imported to PADS I/O Designer) you must first import the schematic changes to the FPGA database (RefDes) as prompted by the synchronization wizard. You must synchronize the FPGA database with PADS Designer before performing assignment changes, or Constraint Manager data for the FPGA related nets can be lost.
- When assignments are changed in PADS I/O Designer without layout context (for instance, import of FPGA constraints file, pin report file, and so on), you must first synchronize the PADS I/O Designer FPGA database with the PADS Designer schematic (RefDes imported), as prompted by the synchronization wizard.

Constraints Bus Members Generation

You can choose a constraints delimiter format for a specified constraints type. To display the Constraints Bus dialog, click **Setup > Settings + Constraints Bus Members Generation** (the dialog is shown in [Figure 11-6](#)).

Figure 11-6. Constraints Bus Dialog: Default Values



File Type	Slice Format
Actel - PDC	a(0)
Actel - PIN	a<0>
Actel - GCF	a(0)
Altera - QSF	a[0]
Altera - ACF	a0
Altera - CSF	a[0]
Lattice - PRF	Automatic
Lattice - LPF	Automatic
Xilinx - UCF	Automatic
FPGA Xchange	Automatic
Leonardo Spectrum - CTR	a(0)
Precision - SDC	a(0)
Synopsys - SDC	a(0)
Synplify Pro - SDC	a[0]
Xilinx Synthesis XST - XCF	a<0>

The **File Type** column lists the type of constraints file. The **Slice Format** column shows the current format associated with the file in the adjacent field. Slice format determines the type of delimiters to be used for presenting constraints (for example, writing them to a file).

Initially, default settings populate the dialog table. A row highlighted in bold, indicates the format in the **Slice Format** column has been changed. When changes are applied, only changed settings are written to the local settings file (the IODesigner.xml file). Unchanged rows (that is, default rows) are always read from global settings.

To change a value, highlight the row, click the down arrow to display a drop-down list, and select the appropriate value. Click **Apply** to apply the new value and leave the dialog open or **OK** to save the new value and close the dialog. [Figure 11-7](#) shows a modified entry (Lattice - PRF).

Figure 11-7. Constraints Bus Dialog: Modified Defaults

Constraints Bus Members Generation	
File Type	Slice Format
Actel - PDC	a(0)
Actel - PIN	a[0]
Actel - GCF	a0
Altera - QSF	a[0]
Altera - ACF	a0
Altera - CSF	a[0]
Lattice - PRF	a(0)
Lattice - LPF	Automatic
Xilinx - UCF	Automatic
FPGA Xchange	Automatic
Leonardo Spectrum - CTR	a(0)
Precision - SDC	a_0_
Synopsys - SDC	a(0)
Synplify Pro - SDC	a[0]
Xilinx Synthesis XST - XCF	a<0>

The default values can be restored selectively by choosing the appropriate value for a specific row. To restore all slice formats to their default values, click **Default**.

Some constraints delimiter formats can be set to *Automatic*. This option tells the system to determine the slice format from the appropriate database settings (for example, for a specific vendor or language option).

FPGA Updater Introduction

PADS I/O Designer FPGA Updater allows you to automatically check for, download, and install PADS I/O Designer FPGA Library updates. If you do not want to be notified of updates, or if your design group does not have internet access, you can switch FPGA Updater off.

Note



A design group that does not have routine internet access can assign an administrator to maintain an updated library to share with all members of the group. See [Updating FPGA Libraries in a High Security Design Environment](#).

You can configure FPGA Updater to launch automatically whenever you start PADS I/O Designer, or you can launch it manually.

Manually Launching FPGA Updater

To launch FPGA Updater in manual mode:

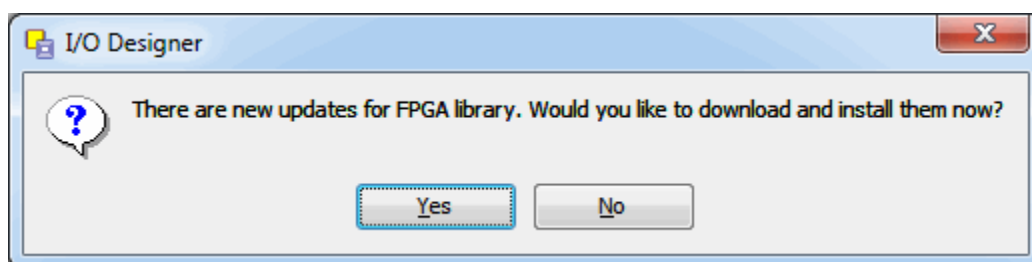
- Select **Help > Check for FPGA Updates** from the PADS I/O Designer menu bar.
or
- Click the **Check for FPGA Updates** button in the Database Properties dialog.

If FPGA Updater is running already, PADS I/O Designer displays the message, “PADS I/O Designer Library Updater is already working.”

Automatically Launching FPGA Updater

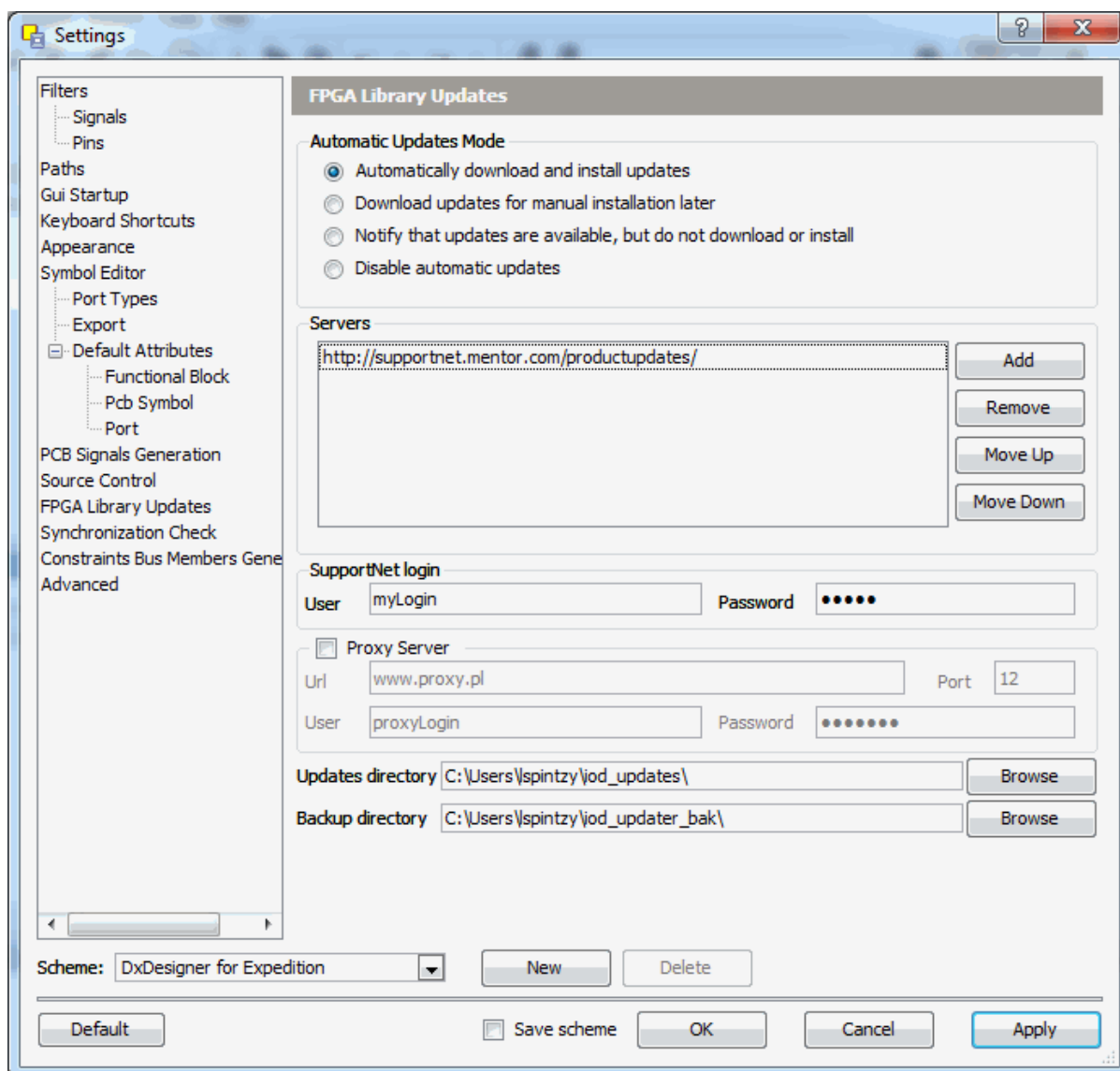
You can configure FPGA Updater to automatically download and install updates, to download updates but not install them, or to notify you of new updates and let you decide whether to download and install them.

If FPGA Updater notifications are on, a message appears when new updates are available:



Configuration

Use the PADS I/O Designer Settings dialog, FPGA Library Updates page, to configure FPGA Updater.



FPGA Library Updates settings:

- Updates Mode - defines actions to take in automatic mode when PADS I/O Designer is starting.
 - Automatically download and install updates
 - Download updates for manual installation later
 - Notify that updates are available, but do not download or install
 - Disable automatic updates
- Servers - Defines the list of servers with updates. FPGA Updater tries to use servers in the specified order. When a server fails, FPGA Updater tries to connect to the next server in the list. The default first server in the list is the SupportNet server. You can add your own local server to the list.
- SupportNet login - FPGA Updater needs your login and password to download updates from SupportNet. You can check to see if new updates are available without entering a login and password.
- Proxy server - If FPGA Updater does not have direct access to the SupportNet server because your network has a gateway server (proxy server), put the proxy server IP address in the Url field. The proxy server must be configured for updates related traffic. Enter the opened port on which proxy server is listening in the Port field. You must also enter a login and password in the appropriate fields, if access to the proxy server is restricted.
- Updates Directory - The directory to which updates download. In the default configuration, you can use the \$TEMP variable to point to a temporary directory.
- Backup Directory - The directory to which FPGA Updater backs up libraries before updating them.

Log file

When FPGA Updater starts, it creates a log file in the log subfolder of the updates directory. Each log file has a unique name consisting of the date and time the file was created, with a “.log” file extension.

Example log file:

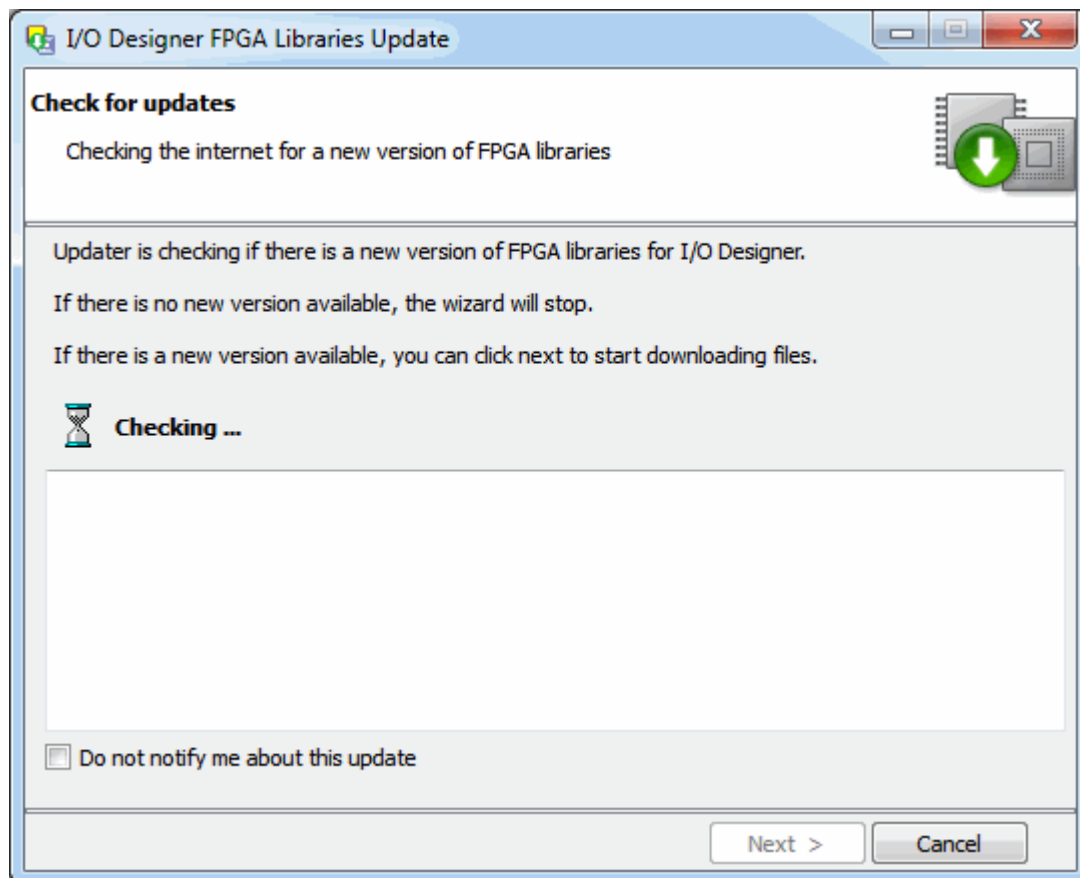
```
Current version of I/O Designer IOD9.1
Current version of library 10_00
Using server http://137.202.162.255/
Downloading manifest file
http://137.202.162.255/iod/updates/libraries/IOD9.1/update.xml
Warning: Could not download manifest file :
http://137.202.162.255/iod/updates/libraries/IOD9.1/update.xml
Using server http://localhostt/
```

```
Downloading manifest file
http://localhostt/iod/updates/libraries/IOD9.1/update.xml
Warning: Could not download manifest file :
http://localhostt/iod/updates/libraries/IOD9.1/update.xml
Using server http://137.202.162.198/
Downloading manifest file
http://137.202.162.198/iod/updates/libraries/IOD9.1/update.xml
Manifest file :
http://137.202.162.198/iod/updates/libraries/IOD9.1/update.xml was
downloaded successfully to path
C:\DOCUME~1\smatysik\LOCALS~1\Temp\iod_updates/update.xml
Loading manifest file
C:\DOCUME~1\smatysik\LOCALS~1\Temp\iod_updates/update.xml
Manifest file C:\DOCUME~1\smatysik\LOCALS~1\Temp\iod_updates/update.xml
loaded successfully.
Comparing update version 10_01 with current version 10_00
Update 10_01 is necessary.
Update 10_01 depends on version 10_00
Downloading manifest file
http://137.202.162.198/iod/updates/libraries/IOD9.1/update_10_00.xml
Manifest file :
http://137.202.162.198/iod/updates/libraries/IOD9.1/update_10_00.xml was
downloaded successfully to path
C:\DOCUME~1\smatysik\LOCALS~1\Temp\iod_updates/update_10_00.xml
Loading manifest file
C:\DOCUME~1\smatysik\LOCALS~1\Temp\iod_updates/update_10_00.xml
Manifest file
C:\DOCUME~1\smatysik\LOCALS~1\Temp\iod_updates/update_10_00.xml loaded
successfully.
Comparing update version 10_00 with current version 10_00
Update 10_00 is not necessary.
Start downloading update 1 of 1
Downloading file
http://137.202.162.198/iod/updates/libraries/IOD9.1/update_10_01/librarie
s_10_01.zip to
C:\DOCUME~1\smatysik\LOCALS~1\Temp\iod_updates/update_10_01/libraries_10_
01.zip
Downloading completed successfully.
Total files to install: 2221
Installing update 1 of 1
Installing file devices/
Installing file devices/altera/
Installing file devices/altera/quartus_92/
Installing file devices/altera/quartus_92/cyclone_iv_gx/
Installing file
devices/altera/quartus_92/cyclone_iv_gx/cyclone_iv_gx.bdev
Installing file devices/altera/quartus_92/cyclone_iv_gx/cyclone_iv_gx.dev
Installing file devices/altera/quartus_92/cyclone_iv_gx/devices.bdev
Installing file devices/altera/quartus_92/cyclone_iv_gx/devices.dev
...
Installation completed successfully.
```

Update Process

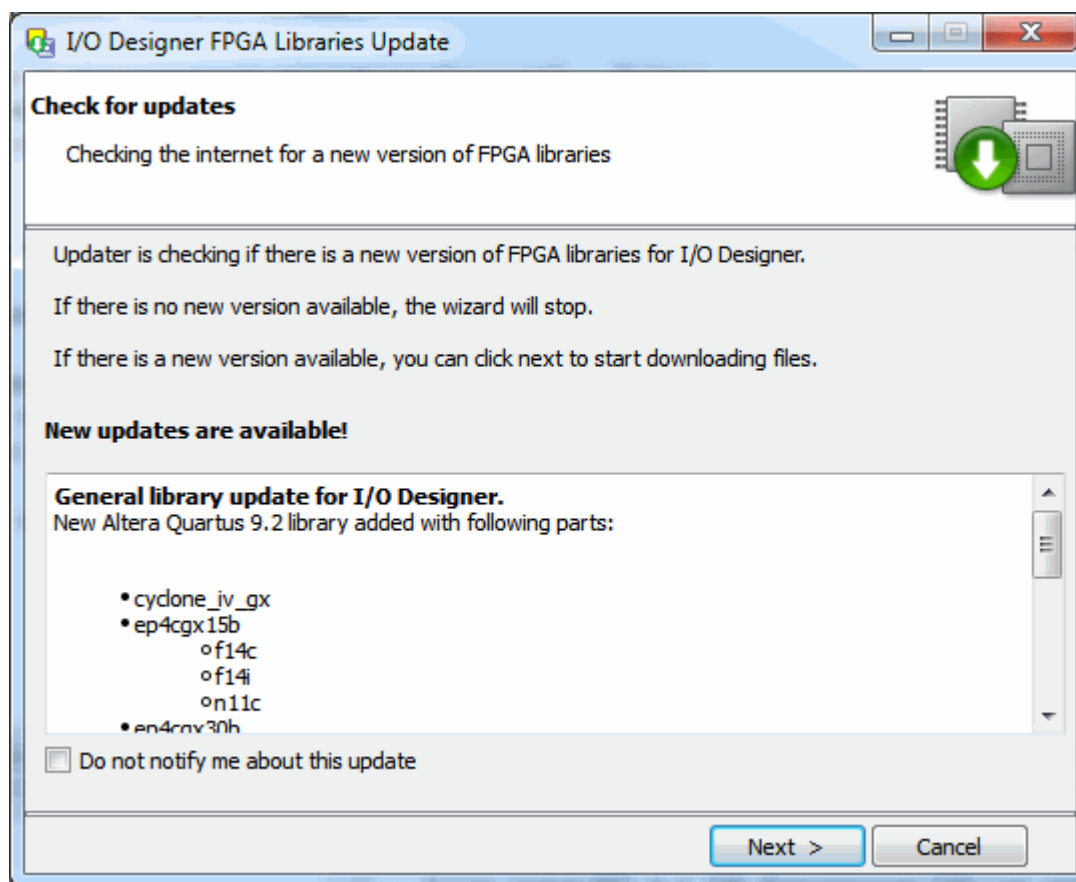
When FPGA Updater is launched, it first loads the configuration and creates a log file, then starts the update process. In order to check for new updates, FPGA Updater first downloads the newest manifest file.

Checking for Updates

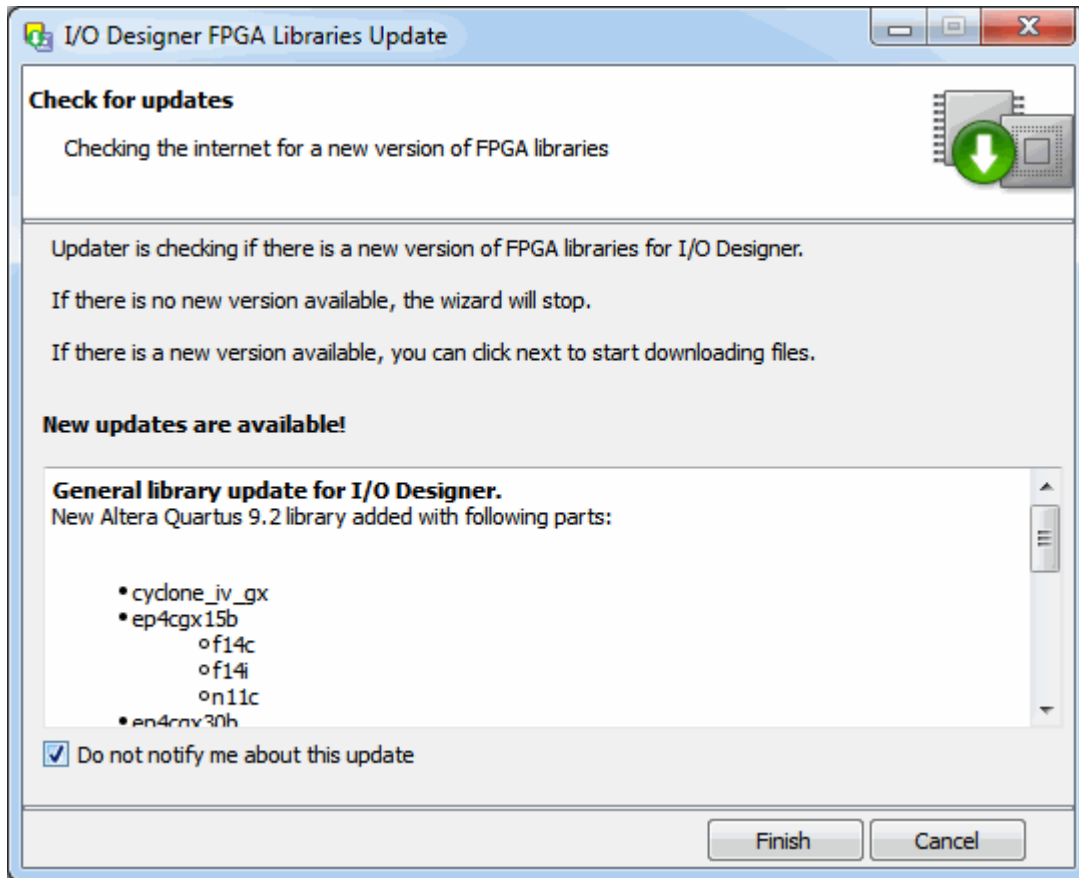


If there are no updates, FPGA Updater displays the message “There is no new version available”.

If there is a new version, FPGA Updater displays information from the manifest file in the first page of FPGA Updater wizard:



Click the checkbox “Do not notify me about this update” to prevent FPGA Updater from notifying you about this update the next time it runs automatically.

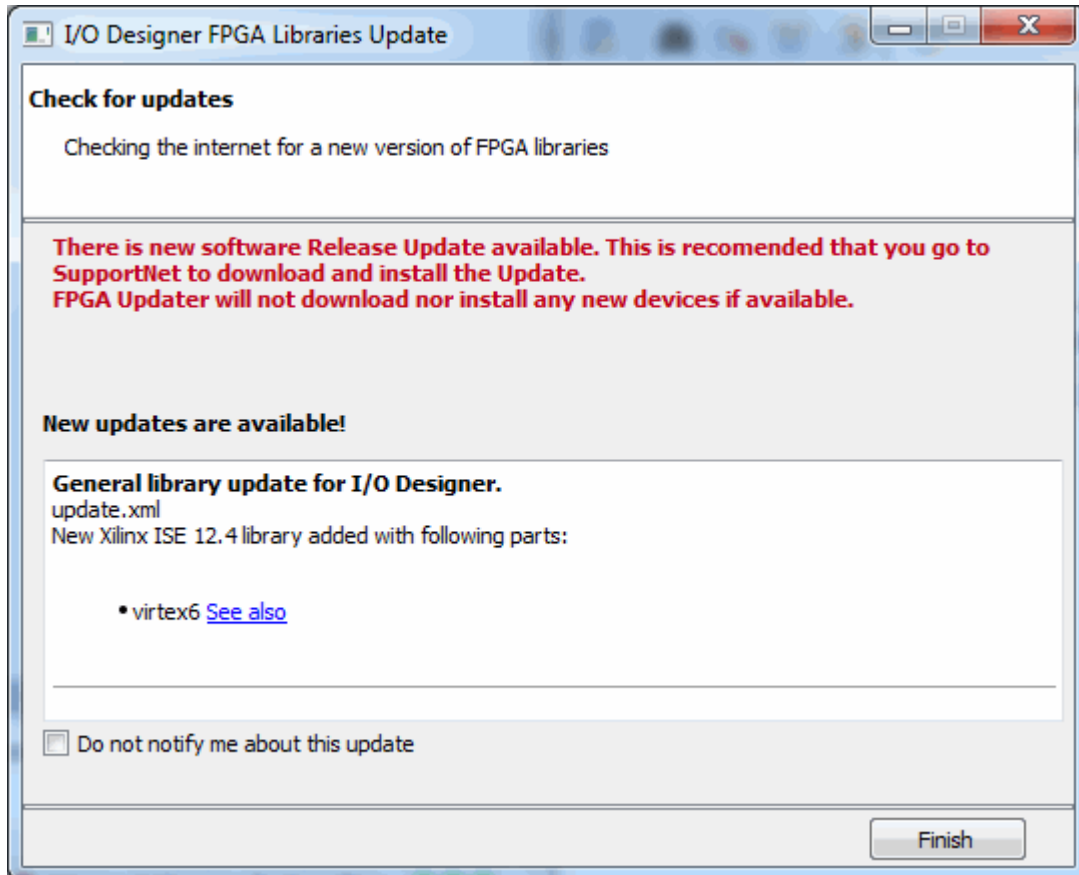


When you check the “Do not notify me about this update” checkbox, the “Next” button is hidden and a “Finish” button is displayed. If you click “Finish” PADS I/O Designer saves the information that you do not want to be notified about that particular update. If you press the “Cancel” button, FPGA Updater closes with no changes.

PADS I/O Designer is always notified when new updates are available. FPGA Library updates are delivered in sequence, and therefore earlier updates must be installed when a new update is installed. If you check the “Do not notify me about this update” checkbox again for the new update then neither update will trigger update notifications in PADS I/O Designer.

Critical Release Update

If new a FPGA update is available that requires an update to PADS I/O Designer before it can function, FPGA Updater displays a message directing you to download and install the software release from SupportNet before attempting to install the FPGA update:



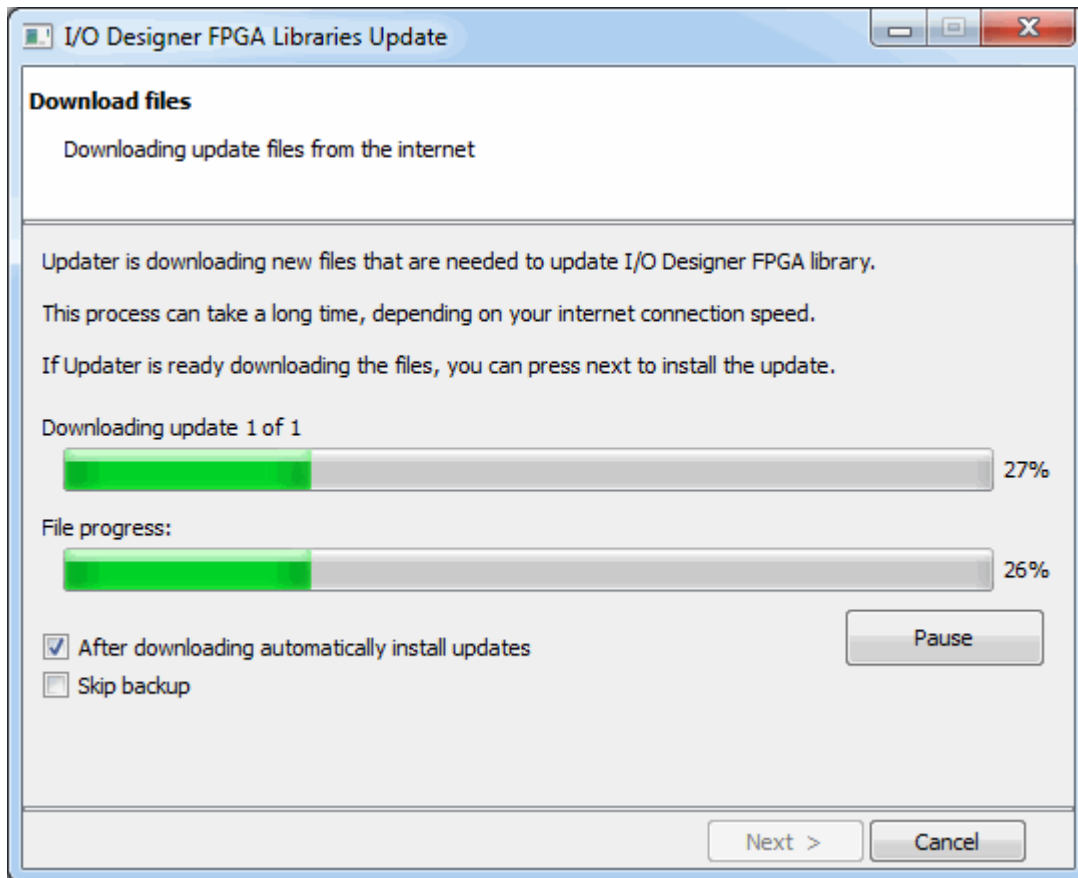
Downloading Updates

To download updates, FPGA Updater uses the first server from the list of servers in the configuration. If there are problems with downloading the manifest file from the first server, FPGA Updater switches to the next server in the list.

After downloading all necessary manifest files FPGA Updater starts downloading the updates files.

If several updates are available, they are downloaded starting with the oldest version. FPGA Updater will continue to download all necessary updates. For example: the newest version is 092_010_000_001. For user1, who has version 092_010_000_000 installed, only one update is downloaded. But for user2 who has 092_009_000_000 installed, two updates are downloaded.

FPGA Updater downloads all files to the updates directory defined in configuration.



You can use the Pause button to pause and resume downloading.

If “After downloading automatically install updates” is checked, FPGA Updater automatically starts the installation process after the update download completes. If this check box is not checked, you must click the “Next” button after downloading to install the updates.

You can check the “Skip backup” checkbox to skip the backup process. This checkbox is unchecked by default, and FPGA Updater performs a backup after downloading updates and before installation. The recommended method is to leave the option unchecked so that backups are always performed, and PADS I/O Designer can always recover your FPGA library.

When downloading updates from the SupportNet server you are prompted for your username and password. You can save login and password information in the configuration through the **Setup > Settings + FPGA Library Updates** page.

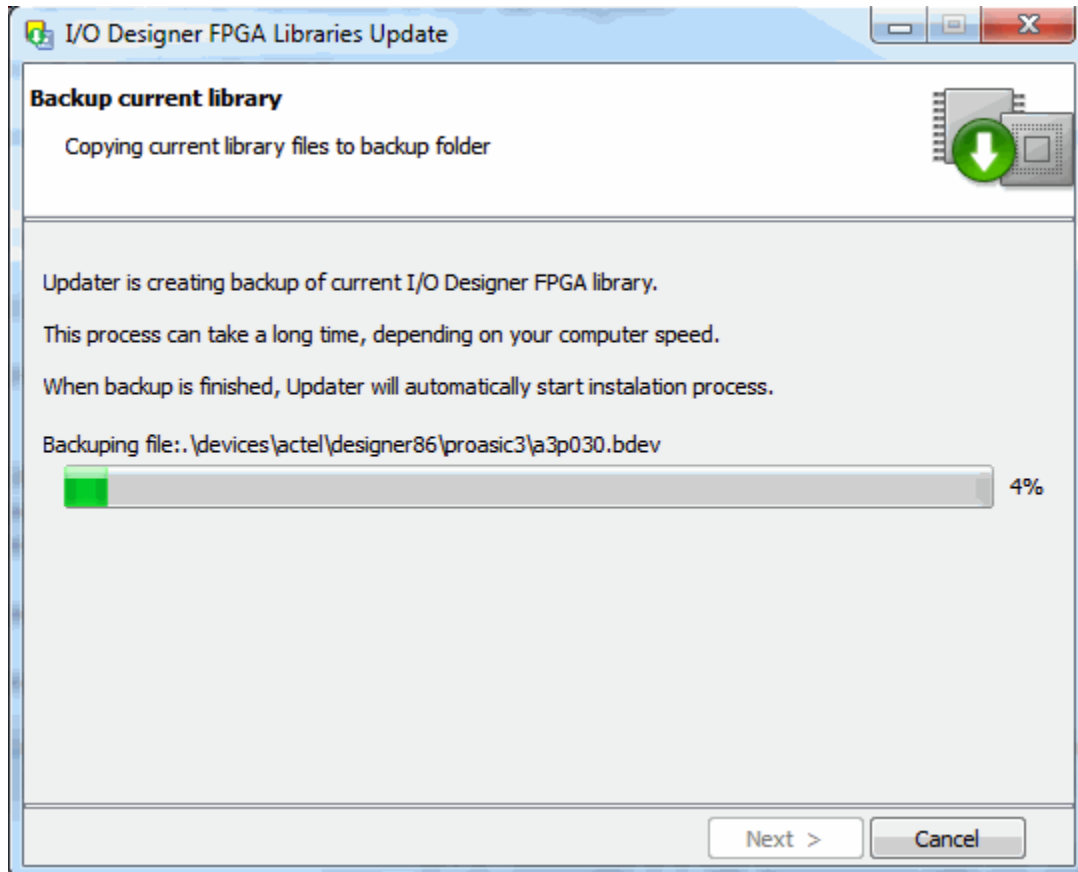
Backup of Current FPGA Library

Before installation process begins, FPGA Updater backs up the current FPGA library to `<BackupDir>/iod_updater_bak_<backup timestamp>`. For example:

c:\iod_updater_bak\iod_updater_bak_2010-11-19-10-45-54\

You can set the location of the backup directory <backupDir> in the Setup > Settings dialog.

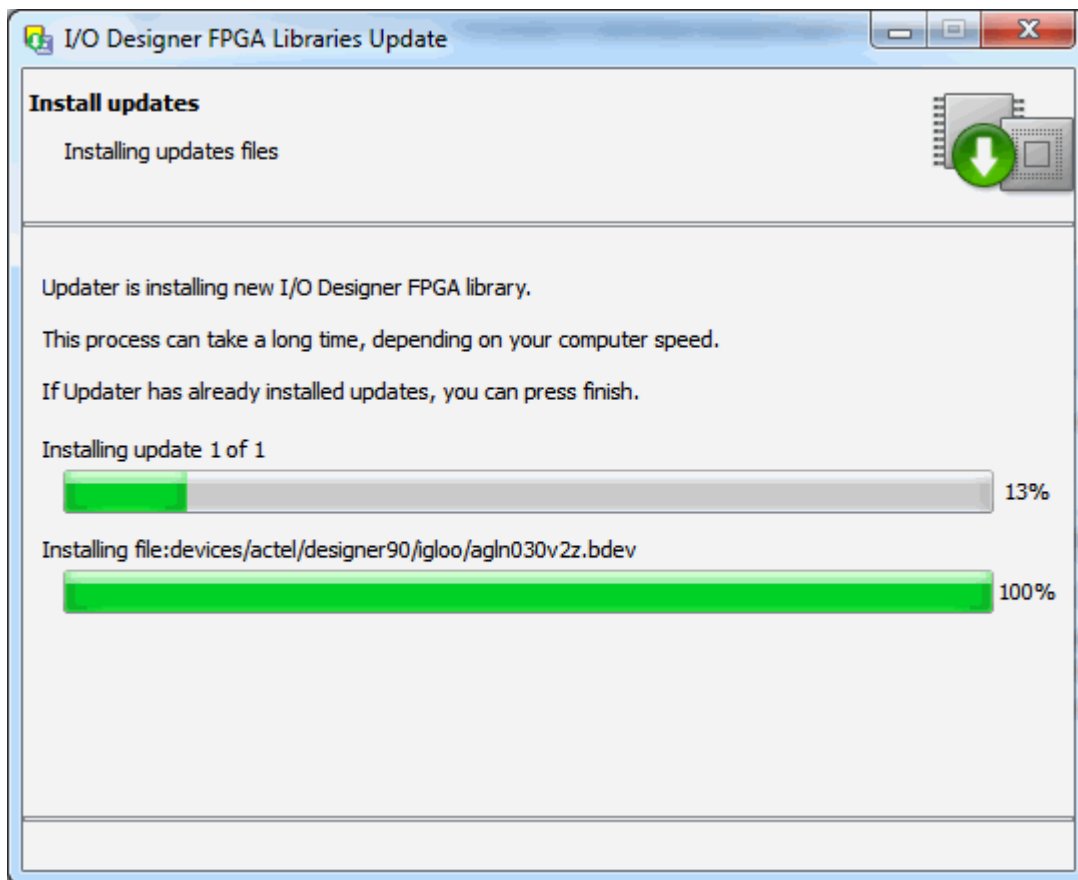
If you press Cancel, the update process stops and the backup is deleted. The backup process can take several minutes, depending on your system.



Installation

Before installation, FPGA Updater checks to see if PADS I/O Designer is running, and if so, displays a message asking you to close it.

FPGA Updater starts the installation with the oldest update. All files are unzipped to the destination folder defined in update.xml, which is the default folder of the FPGA Library pointed to by the \$IOD_LIBRARY_PATH variable.

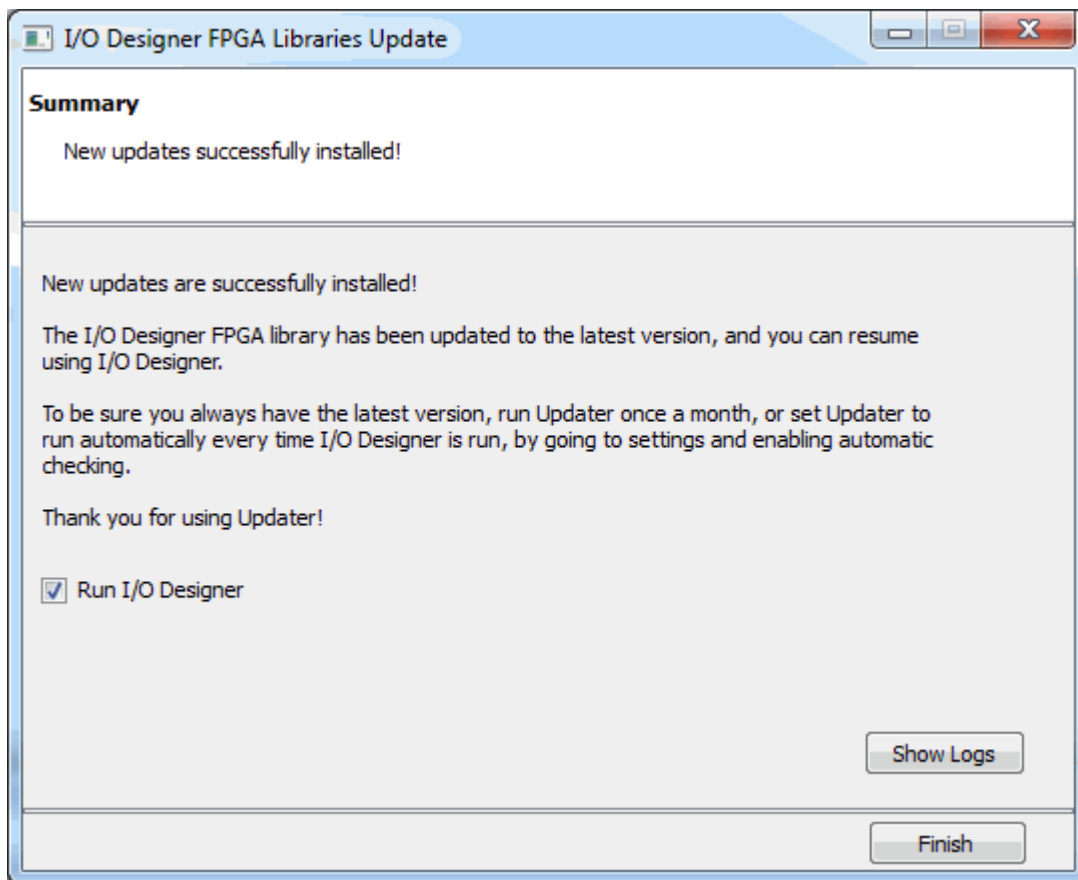


Before unzipping the files, FPGA Updater checks to see if the checksum of the update file matches the checksum in the manifest file. If the checksum is incorrect, FPGA Updater displays an error message and closes.

If during installation any error appears, the rollback Page opens and the backup is copied back into the library directory. If there is no backup, the library may be corrupted.

Summary Page

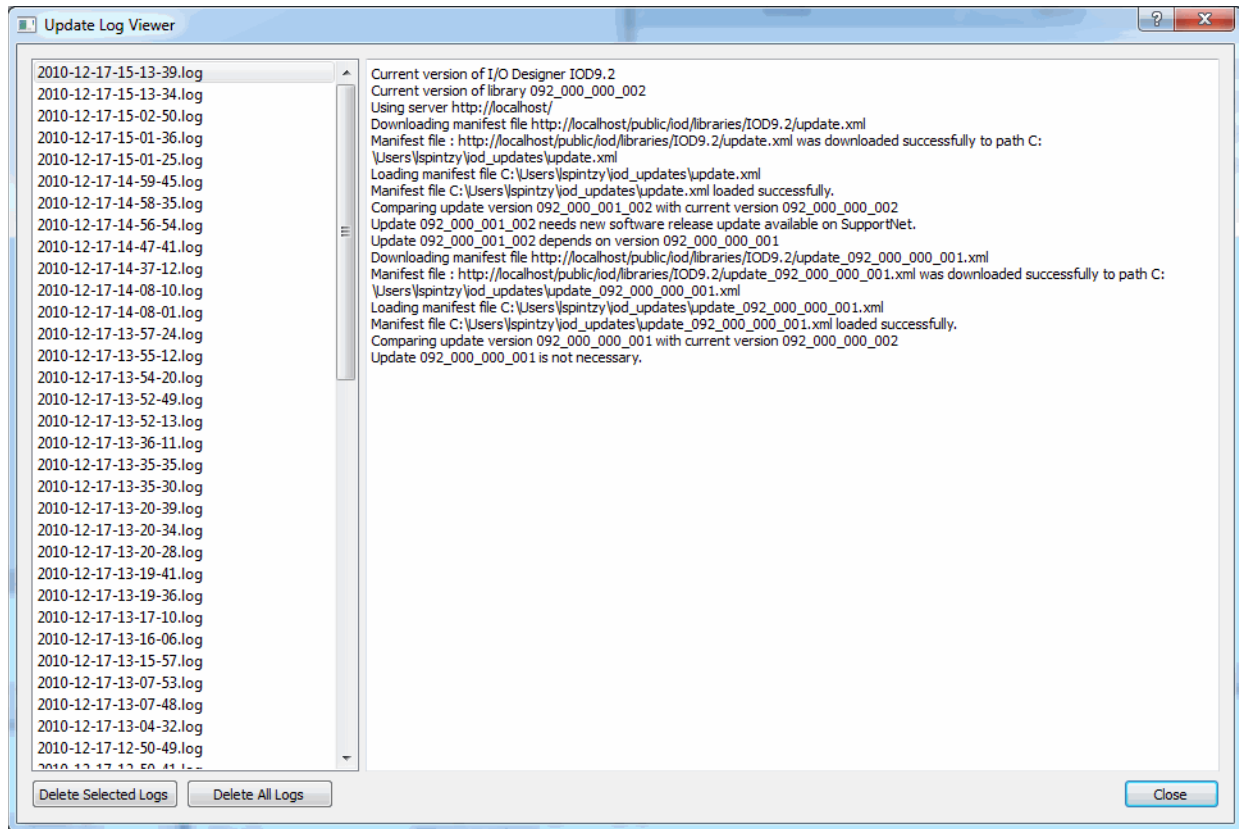
After a successful installation, the Summary page is displayed:



Run PADS I/O Designer checkbox is checked by default. If you do not want to run PADS I/O Designer, clear the checkbox before clicking Finish.

FPGA Updater Log Viewer

Press the “Show Logs” button to open the log viewer when you want to examine the log files:

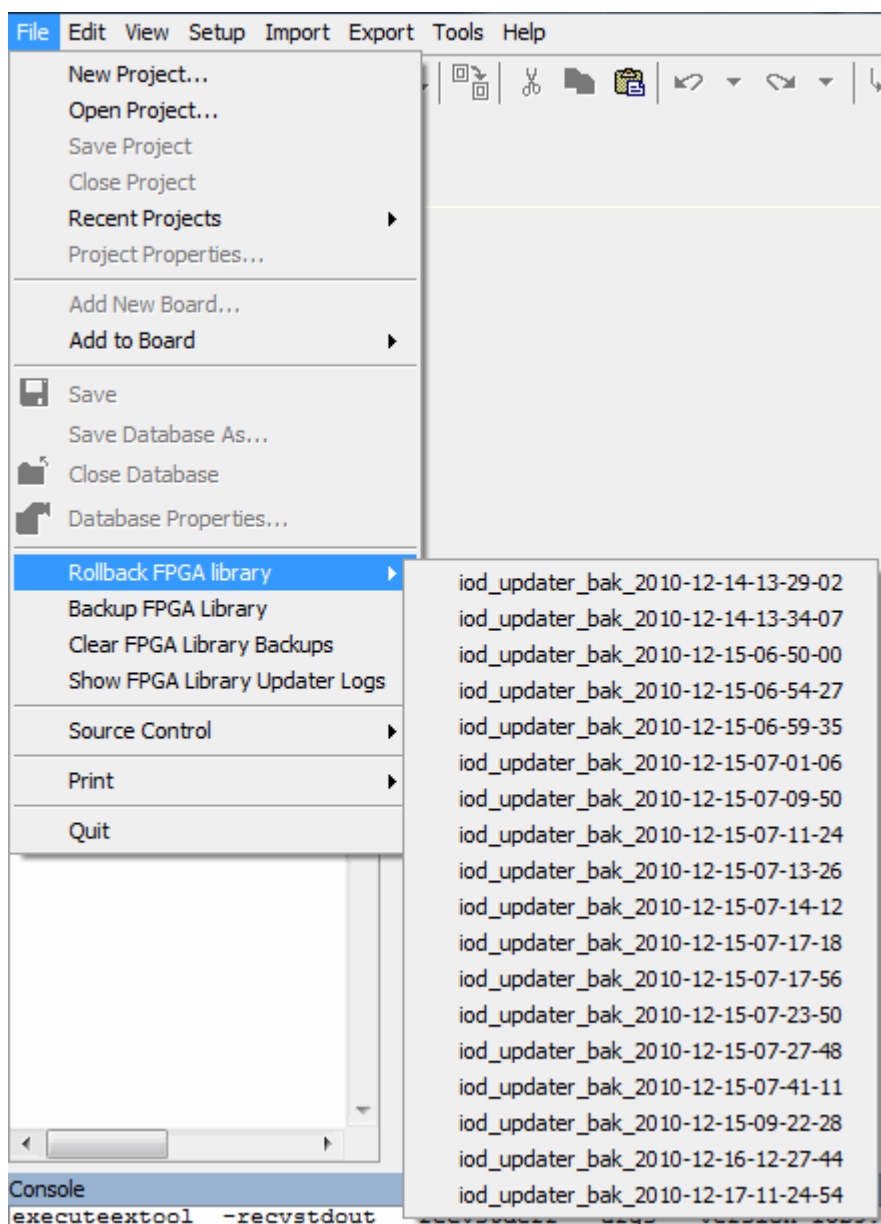


The newest log is first on the list. You can also open LogViewer from I/O Designer; **File > Show FPGA Library Updater Logs**.

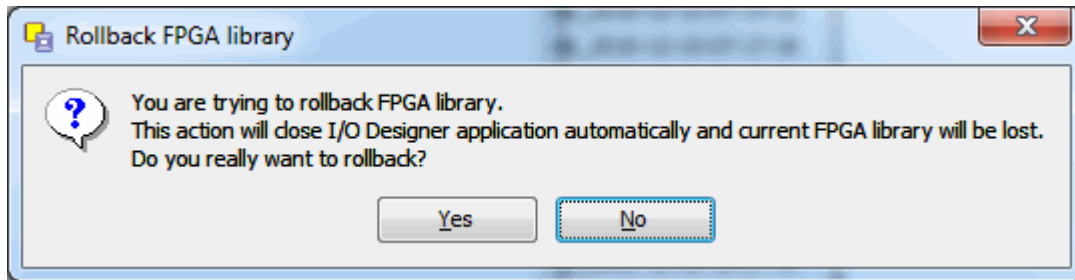
Backup and Rollback of FPGA Library.

The PADS I/O Designer File menu includes selections to help you manage FPGA library backups.

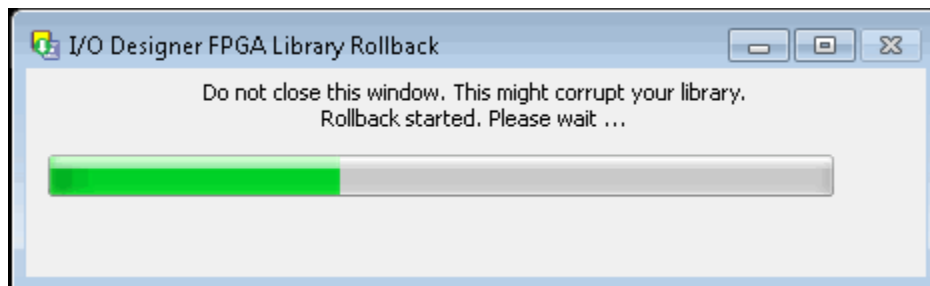
File > Rollback FPGA library allows you to rollback library changes to the selected backup. Selecting one of the menu options (e.g. `iod_updater_bak_2010-12-14-13-29-02`) starts the rollback process.



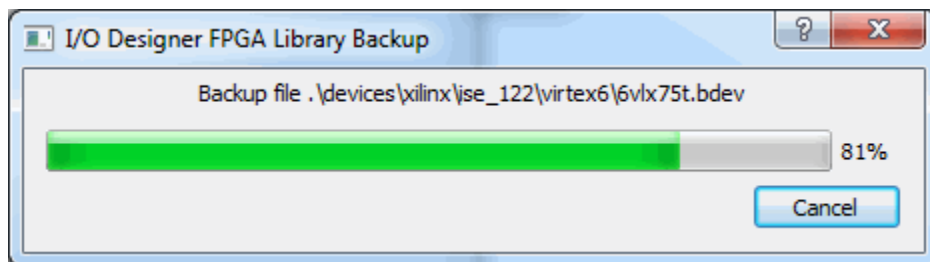
Before beginning the rollback process, PADS I/O Designer displays a message informing you that it will automatically close PADS I/O Designer.



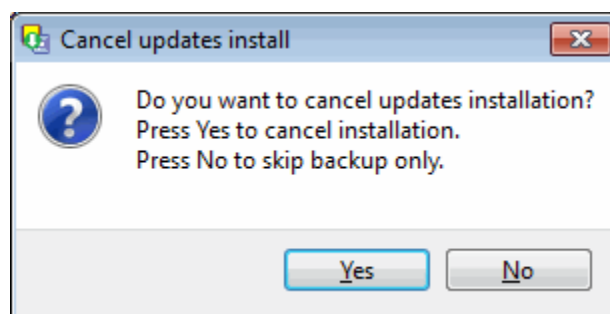
If you confirm the rollback, FPGA Updater starts in rollback mode, displaying only the progress of the rollback operation.



File > Backup FPGA library starts FPGA Updater, and performs a backup. Again progress of the operation is displayed.



Pressing "Cancel" opens the Cancel updates install dialog box, where you are given the option to either cancel the update entirely, or continue with the installation, canceling only the backup.



File > Clear FPGA Library Backups removes all backups from the disk.

File > Show FPGA Library Updater logs opens the LogViewer:

Updating FPGA Libraries in a High Security Design Environment

You can appoint an administrator to update FPGA libraries in an environment where the design group does not generally have internet access.

The administrator uses the following procedure to update FPGA libraries for all users in the design group.

Procedure

1. Choose **Setup > Settings > Paths > <FPGA library location>**, select Set up a shared library folder for all users, and enter the path to a shared library folder for the design group.

The default path is *<install_location>\<release>\SDD_HOME\IODEsigner\library*.

Example: A common network server location mapped to S, with Mentor software version <VX> is: *S:\MentorGraphics\<VX>\SDD_HOME\IODEsigner\library*.

2. Choose one of the following methods to update the FPGA libraries:

If you want to...	Do the following...
Run FPGA Updater from a temporary internet connection	<ol style="list-style-type: none">1. Connect to the internet.2. Run FPGA Updater to update the files in the directory designated in Setup > Settings > FPGA Library Updates > Updates directory.3. When the update is complete, disconnect from the internet.
Manually update FPGA files	<ol style="list-style-type: none">1. Contact Mentor Graphics Customer Support to obtain the updates in zip files. Customer Support can provide the files by mail or as a standard file transfer (such as FTP).2. Extract the zip files sequentially into the shared library folder. <p>Note: The zip files include all updates since the last major release, and you must apply them sequentially.</p>

Results

The FPGA library is updated to include the latest revisions.

Related Topics

[Configuration](#)

[Update Process](#)

Chapter 13

PADS I/O Designer Tool Reference

The following sections give detailed information on the following:

- “[Navigating List Windows](#)” on page 170
- “[Keyboard Shortcuts](#)” on page 177
- “[Managing Keyboard Shortcuts](#)” on page 181
- “[Toolbars](#)” on page 184

Status Bar



A status bar at the bottom of the screen displays information related to the current selection. In addition, tips are continuously displayed here. These are short texts, presenting additional features available in the current context.


Pop-up Menus

The right mouse button is used exclusively to display pop-up menus, giving access to commands specific to a given context. In this manual, the word **click** always refers to the left mouse button click.

You may use operating system dependent ways to display pop-up menus. For details, please refer to the operating system documentation.

Undo and Redo

PADS I/O Designer provides unlimited undo/redo possibilities. Any operation may be undone, and any undone operation may be redone again. There is no limit to the amount of times an undo step is reversed. To undo an operation, use the **Edit > Undo** command, or the  button on the [Main Toolbar](#). To redo an undone operation, use the **Edit > Redo**, or the  button on the [Main Toolbar](#).

It is possible to perform more than one undo or redo action at a time. Click the down arrow  to the right of the Undo and Redo buttons to display a list of operations that may be undone or redone. Moving the mouse pointer over the list highlights the operations, and displays the text below them, such as Undo 5 actions. Clicking on an operation executes the Undo or Redo for that operation and all operations performed since.

The Undo/Redo history will be cleared, as the result of operations that change the entire database, such as **File > New**, **File > Open**, and those Source Control system interface commands that result in reloading the database.

Navigating List Windows

The following sections describe how to navigate the [Signals List](#), [Pins List](#) and [Connectivity List Window](#).

Operations on Columns

Columns in the List Windows are resizable. To shrink or enlarge a column, click the line between the two columns on the header, and drag the separating line to the desired position.

Columns in the List Windows may be freely hidden and displayed again. Right-click on a column header to display the pop-up menu. Each column is listed on the pop-up menu with a check mark to indicate that they are shown. Select the required column from the list to toggle its show/hide state.

Row Selection

Click on a row to select it. The row will be highlighted and any previously selected row will be deselected. To select multiple rows, use **CTRL+click**, which toggles the selection state of a row.

To select a range of adjacent rows, click the first row in the range, and then **SHIFT+click** on the last row in the range.

Whenever a list window is active, only one of the rows is current. The current row does not have to be selected. The current row is outlined with a slightly wider frame. For some operations, especially while using the keyboard, it is important to know which row is current. Use the Space key to toggle the selection state of the current row.

Sorting Rows

Rows in the List Windows can be sorted alphabetically, based on any single column, and also on multiple columns.

To sort the rows by a single column, do one of the following:

- Click on the required column header. A second click on the same column header reverses the sort order.

The triangle shown in the header acts as a sort column indicator.

- Right-click on the required column header and select **Sort Ascending** or **Sort Descending**.

To sort rows based on multiple columns

1. Right-click on any column header and select **Multicolumn Sort...**

This displays the **Multicolumn sort properties** dialog, which contains one row named **sort by**.

2. In the **Column** field, select the first column to start sorting by, using the drop down list.
A second row appears named **then sort by**.
3. Select a second column in the Column field for the **then sort by** row. The drop down list will contain only the remaining columns.
4. Repeat this step to add further columns to define sorting properties.
5. In the Direction field, select **Ascending**, or **Descending**, to set the sort order for each column separately.

Filtering Rows

In order to view only relevant entries in a List Window, filters can be set for each column by entering a value in the filter field at the top of the window.

Filters are maintained for each column separately, and are equally taken into account. Therefore, it is possible to display results that match more than one column filter. For example, in the [Signals List](#), setting a filter of A on the Name column, and a filter of 'In' on the **Dir** column will display signals with names beginning with the letter A and are of the direction 'In'.

Procedure

Use the following procedure to set filters on one or more columns:

1. In the List Window, right-click on the required column header and ensure **Filter Column** is selected. Or use the keyboard shortcut **CTRL+click** on the column header to set this column as the filter column.

The column header is marked with bold face.

2. Enter a value into the **filter field** and press **Enter**.

Only rows matching the value entered are displayed in the List Window. In addition to the filter field for each column, the value of the filter appears in brackets after the column header and in the status bar.

3. Repeat steps 1 and 2 on subsequent columns to filter the results further.

All filters are displayed in the header, after column names, and additionally in the status bar.

To remove all column filters, right-click on any column header in the List Window and select **Reset Filters**.

Filtering to Selected Rows

Select the required rows in the List Window and select **View > Show Only Selected Rows**. An appropriate filter is constructed in such a way that only the selected rows match it.

Filter Rules

Filters operate only on the level of individual signals or pins. Bus signals or bus pins are never taken into account while filtering. For example: the **Add Signal** dialog contains a bus named *ABUS*, with elements *BIN* and *BIN1*. If a filter *A* is set for signal names, the bus is not displayed. On the other hand, if a filter *B* is set, the bus is displayed.

Filters are not case sensitive by default, but case sensitivity may be selected in the **Setup > Settings + Filters** dialog, by checking the **Case sensitive** option in the Filter Settings group.

Filters may contain special characters, which, for instance, allow construction of a filter for displaying only pins which Function contains the text *VREF*. The following sections describe [Wildcard Filters](#), and [Regular Expression Filters](#) in detail.

Wildcard Filters

Wildcard filters use special characters such as *** in a similar way to filename wildcards. Wildcard filters are treated as if they had *** character appended at the end, for example, filter *A* is the same as *A**, and matches strings beginning with the letter *A*.

The following characters are treated special in wildcard filters:

Table 13-1. Wildcard Filters

?	Matches any single character.
*	Matches any number (including zero) of any characters.
[]	Characters inside square brackets are called <i>character classes</i> . Character class matches any single character included in the class. For example, [abc] matches any of the letters a, b, c.
-	This character is treated specially in character classes. It is used to denote ranges. For example, [a-z] matches any letter. A character class may contain several ranges, so e.g., [a-z0-9] matches any letter or digit.

Table 13-1. Wildcard Filters

^	This character used immediately after opening square bracket negates the character classes, e.g., [^a-z] matches any character except letters.
---	--

Regular Expression Filters

Regular expression filters use the popular regular expression syntax known in many programs such as grep, Perl, etc. providing a more powerful filtering method than wildcards.

In order to use regular expressions in filters, they must be enabled. Select **Setup > Settings + Filters** and check the **Regular expressions** option.

The following characters are treated as special in regular expression filters:

Table 13-2. Regular Expression Filters

.		Matches any character.
^		Matches the beginning of the string.
\$		Matches the end of the string.
\		The meaning depends on the character that follows \ . See Table below.
[]		Characters inside square brackets are called character classes. Character class matches any single character included in the class. For example, [abc] matches any of the letters a, b, c.
-		The character - is treated specially in character classes. It is used to denote ranges. For example, [a-z] matches any letter. A character class may contain several ranges, so [a-z0-9] matches any letter or digit.
^		The character ^ used immediately after opening square bracket negates the character class, so [^a-z] matches any character except for letters.
()		Parentheses are used to group elements. Grouping elements of a regular expression is useful for using ? not for a single letter, but a longer expression instead.
?		Matches zero or one occurrence of the preceding expression (i.e., single character, character class, or grouped elements).
+		Matches one or more occurrences of the preceding expression.
*		Matches zero or more occurrences of the preceding expression.

Table 13-2. Regular Expression Filters (cont.)

{ }		Notation {n,m} matches at least <i>n</i> , and at most <i>m</i> occurrences of the preceding expression. There are shortcut forms: {n} means {n,n}; {n,} means {n,max}; {,m} means {0,m}.
-----	--	--

The backslash \ character is used to construct further special constructs, which are listed below.

Table 13-3. Backslash Character Filters

\d	Matches a single digit, so it is the same as [0-9]
\D	Matches non-digits, so it is the same as [^0-9]
\s	Matches whitespace.
\S	Matches non-whitespace.
\w	Matches a word character.
\W	Matches a non-word character.

Regular expression filters do not need to match the entire string to succeed. For instance, while wildcard filter *A** matches any string *beginning* with the letter A, the regular expression filter *A.** matches any string *containing* the letter A. This feature does not limit expressiveness of regular expression filters, since the two special characters ^ \$ may be used to force matching of the whole string. In short, the regular expression equivalent to wildcard filter *A** is *^A.*\$*.

Custom Filters

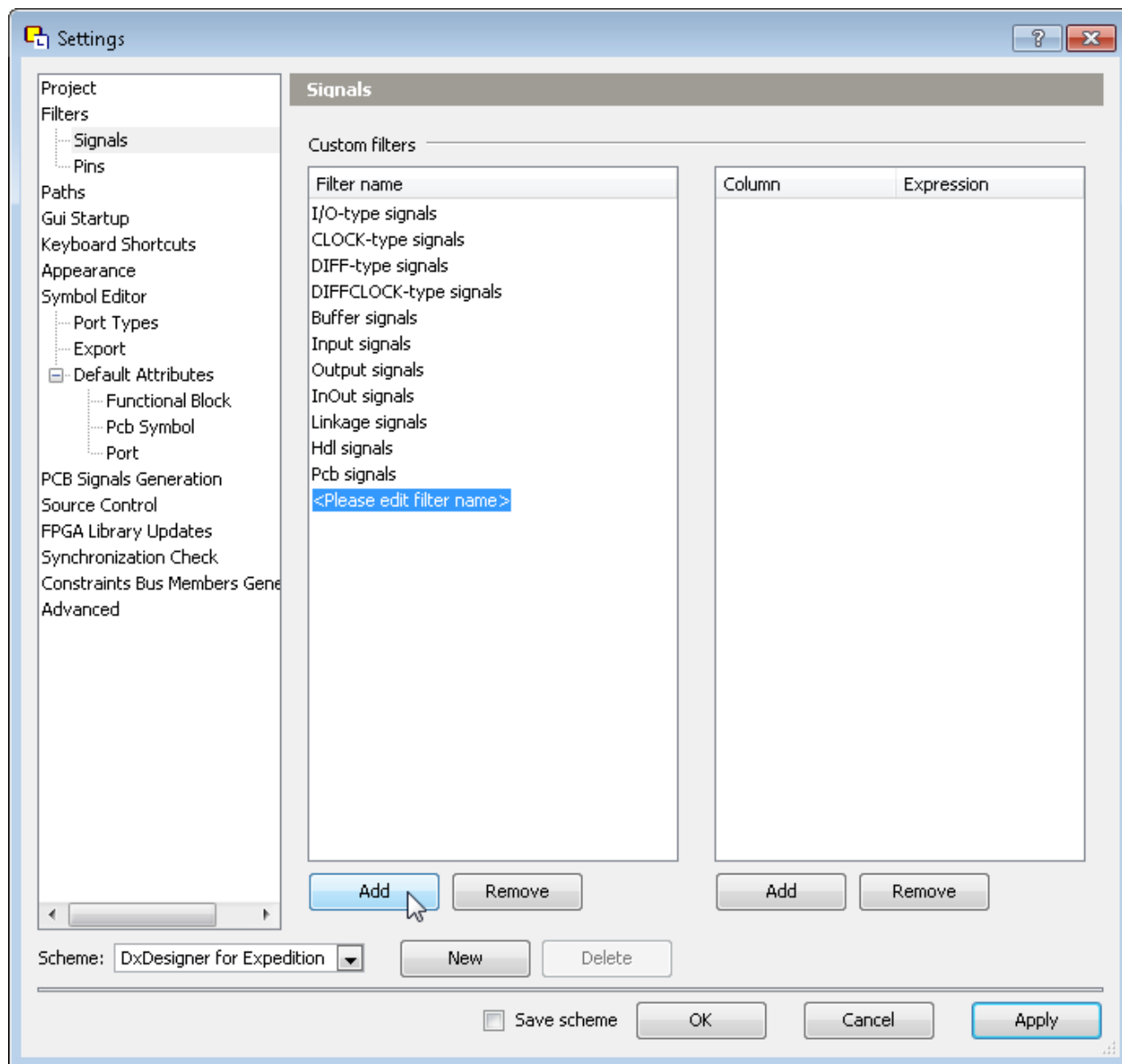
In **Setup > Settings + Filters**, you can alter default filter settings for named groups of signals or pins such as I/O, clock, and differential. You can also define new custom filters that suit your specific needs. You can use a custom pin filter with a custom signal filter to display only those pins on a device that have that signal.

Creating Custom Filters

To create a custom filter:

1. Select **Setup > Settings**.
2. In the Settings Dialog, under **Filters**, select **Signals** or **Pins** to open the Signals or Pins section of the Settings Dialog.
3. In the Signals or Pins section, click **Add** below the Filter subsection to create a new entry. The new entry is highlighted, and reads <Please edit filter name>, as shown in [Figure 13-1](#).

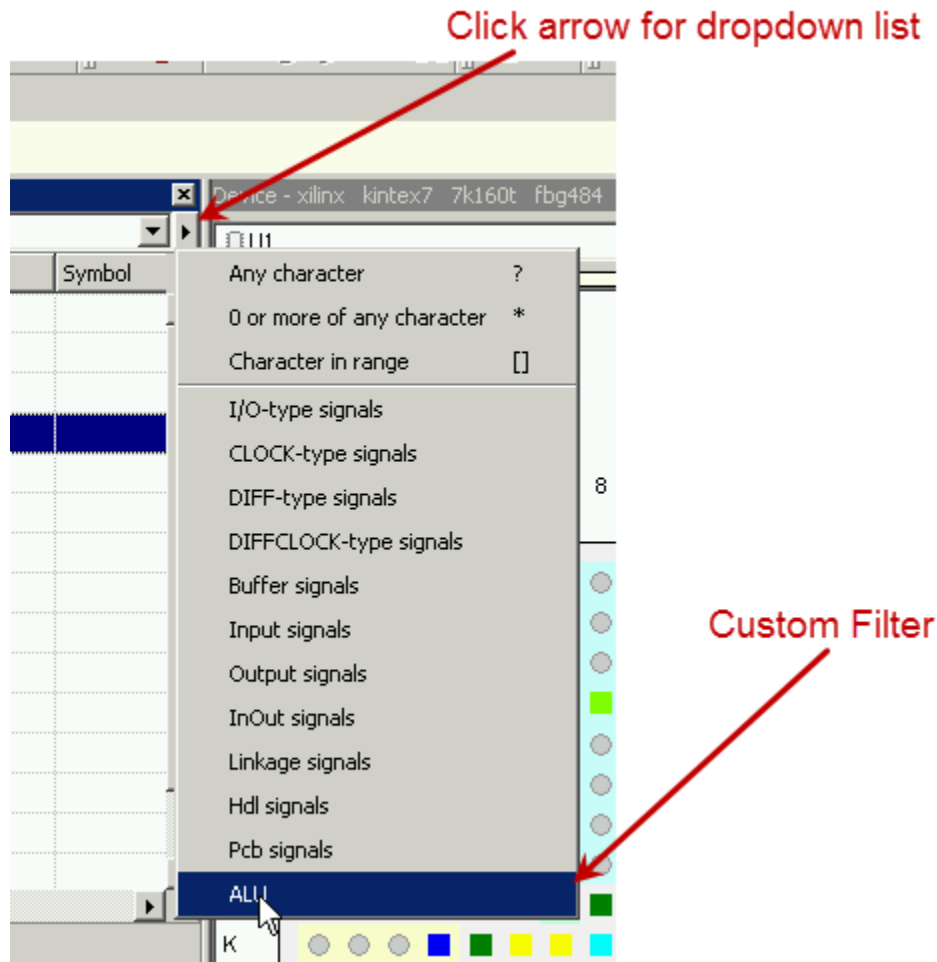
Figure 13-1. Adding Custom Filter



4. Enter an appropriate name for the filter in place of <Please edit filter name>. For example, if you want to filter the ALU signals you could name the filter ALU. (If you inadvertently click elsewhere before entering a name, double- or triple-click **<Please edit filter name>** to re-highlight it, then enter the name of the signals or pins to filter.)
5. Click **Add** below the Column | Expression subsection. A default entry, PCB Name for a signal filter, or Signal for a pin filter, appears under Column.
6. Double click on the default entry under Column to open a dropdown list of available Signals or Pins table columns, and select the column you want to sort on.

7. Double click in the Expression column, just to the right of the Column entry, and enter the filter string. For example, ALU.
8. Click **Apply** and then **OK** at the bottom of the Settings Dialog.
9. Click the arrow at the right side of the Signals or Pins entry boxes to select the custom filters you created from the dropdown list at the top of the Signals or Pins table, as shown in [Figure 13-2](#).

Figure 13-2. Selecting Custom Filter from Dropdown List



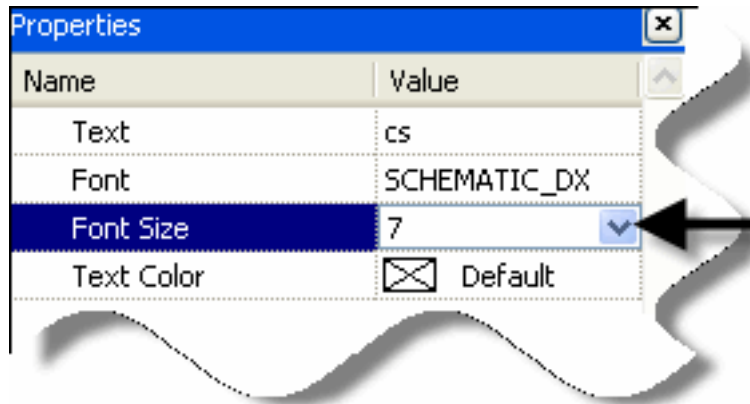
Changing the Pin Name Font Size

If the font size on a pin name is too large in PADS I/O Designer, you can change this setting in the Properties window.

1. Select the pin name to be changed by right-clicking on it.
2. Choose Label Properties to display the Properties window.

3. Check the Font Size. The default is 7. If your pin name is showing a larger font size, click the drop-down box and choose 7.

Figure 13-3. Change Pin Name Font Size



Keyboard Shortcuts

Many operations available in PADS I/O Designer can be performed by using shortcut keys. Most shortcut keys are always available. However, the function of some keys depends on the active window.

- “[Global Shortcuts](#)” on page 178
- “[Shortcuts in Signal and Pins Lists](#)” on page 179
- “[Shortcuts in the Symbol Window](#)” on page 180

Global Shortcuts

Table 13-4. Global Keyboard Shortcuts

Shortcut	Action	Menu Item
CTRL + O	Open Project	File > Open Project
CTRL + S	Saves current database	File > Save Database
ALT + X	Quit PADS I/O Designer	File > Quit
CTRL + Z, ALT + Backspace	Undo	Edit > Undo
CTRL + Y, CTRL + R	Redo	Edit > Redo
CTRL + W	Close Database	File > Close Database
F2	Edit text on symbol	Symbol > Edit Selection > Edit Text
F4	Assign Pins	Assign > Assign Pins
SHIFT + F4	Assign Pins with Overwrite	Assign > Assign Pins with Overwrite
ALT + 1	Show/Hide Signals List	View > Windows > Signal List
ALT + 2	Show/Hide Pins List	View > Windows > Pins List
ALT + 3	Show/Hide Symbol Window	View > Windows > Symbol
ALT + 4	Show/Hide Device Window	View > Windows > Device
ALT + 5	Show/Hide Console Window	View > Windows > Console
ALT + 6	Show/Hide Timings Window	View > Windows > Timings List
ALT + 7	Show/Hide Project Window	View > Windows > Project
ALT + 9	Show/Hide Layout Window	View > Windows > Layout
ALT + Enter	Show Properties Window	Edit > ... Properties (depends on current selection)
CTRL + ALT + 2	Show/Hide Connectivity List Window	View > Windows > Connectivity List
CTRL + ALT + 3	Show/Hide Layout Scenarios Window	View > Windows > Layout Scenarios List

Shortcuts in Signal and Pins Lists

The following keys are recognized in the [Signals List](#) and the [Pins List](#):

Table 13-5. Shortcuts in Signal and Pin Lists

Space	Toggles the selection state of the current row.
Left Arrow	If the current row is bus element, it turns the bus into the current element. If the current row is a bus, and it is expanded, the key collapses the bus. Otherwise, it scrolls the entire list to the left.
Right Arrow	If the current row is a collapsed bus, the right arrow will expand it. If the current row is an expanded bus, the key turns its first element into the current one. Otherwise, it scrolls the entire list to the right.
Up Arrow	Shifts the current row up.
Down Arrow	Shifts the current row down.
Home	Makes the first row current.
End	Makes the last row current.
Page Up	Makes the row above the first visible row current.
Page Down	Makes the row below the last visible row current.
CTRL + Page Up	Switches to the next tab.
CTRL + Page Down	Switches to the previous tab.
CTRL + A	Selects all rows.
Esc	Removes marks, see Mark to Assign.

All keys that move the current row may be used together with these modifier keys:

Table 13-6. Modifier Keys

None	Makes the current row selected, and deselects all other rows.
CTRL	Does not change the selection.
SHIFT	Adds all rows between the selection and the current row to the existing selection.
ALT	Marks the selected rows for assigning.

Additionally, the letter keys are used for incremental search. By typing letters contained in the filter column, you turn the row that starts with those letters into the current row.

Shortcuts in the Symbol Window

The following keys are recognized in the Symbol Window:

Table 13-7. Symbol Window Shortcuts

Esc	Select Mode.
+	Zoom In
-	Zoom Out
Home	Zoom to Fit
CTRL + X, SHIFT + DELETE	Cut.
CTRL + C, CTRL + Insert	Copy
CTRL + V, SHIFT + Insert	Paste
CTRL + A	Select all elements.
F2	Edit the selected text.

Shortcuts in the Device Window

The following keys are recognized in the [Device Window](#):

Table 13-8. Device Window Shortcuts

+	Zoom In
-	Zoom Out
Home	Zoom to Fit
CTRL + A	Selects all pins
Esc	Remove marks, see Mark to Assign

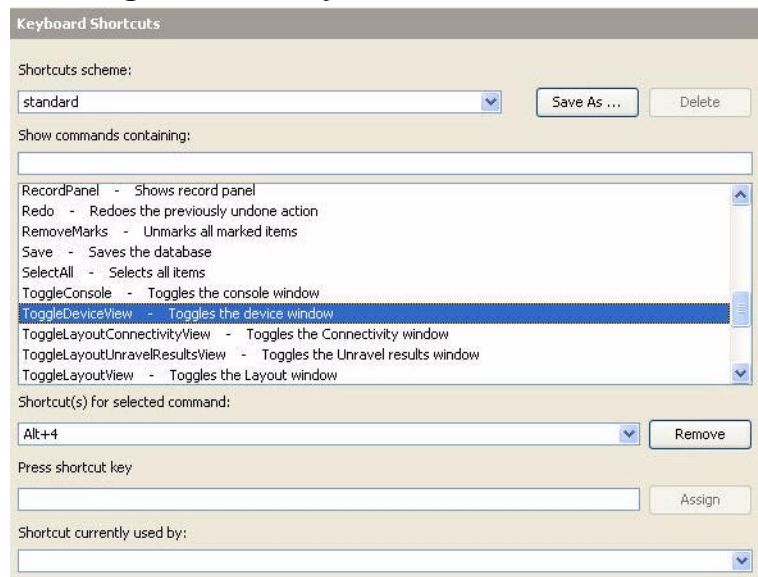
Managing Keyboard Shortcuts

The Keyboard Shortcuts dialog is where you modify the default keyboard-shortcuts scheme and then save it under a new name. You can then load this new shortcut scheme into the GUI.

Modifying Shortcuts

To display the Keyboard Shortcuts dialog shown in [Figure 13-4](#), click **Setup > Settings + Keyboard Shortcuts**.

Figure 13-4. Keyboard Shortcuts Editor



The commands and options available in the Keyboard Shortcuts dialog include:

- **Shortcut scheme:** The drop-down lists lets you select a previously-saved scheme. Click **OK** to accept the changes.
- **Save As ...:** Creates a new shortcut scheme by copying the contents of the currently-selected scheme. After clicking **Save As ...**, you are asked to specify a unique name for the new shortcut scheme.
- **Delete:** Deletes user-defined shortcut schemes. If the button is not enabled, the scheme cannot be deleted.

The lower part of the dialog is where you change the shortcut mappings:

- **Show commands containing:** This is a filter enabling you to search the listed commands for a specific word. All commands containing that word are listed in the pane below. For example, type in `pins` to display all commands containing the word `pins`.

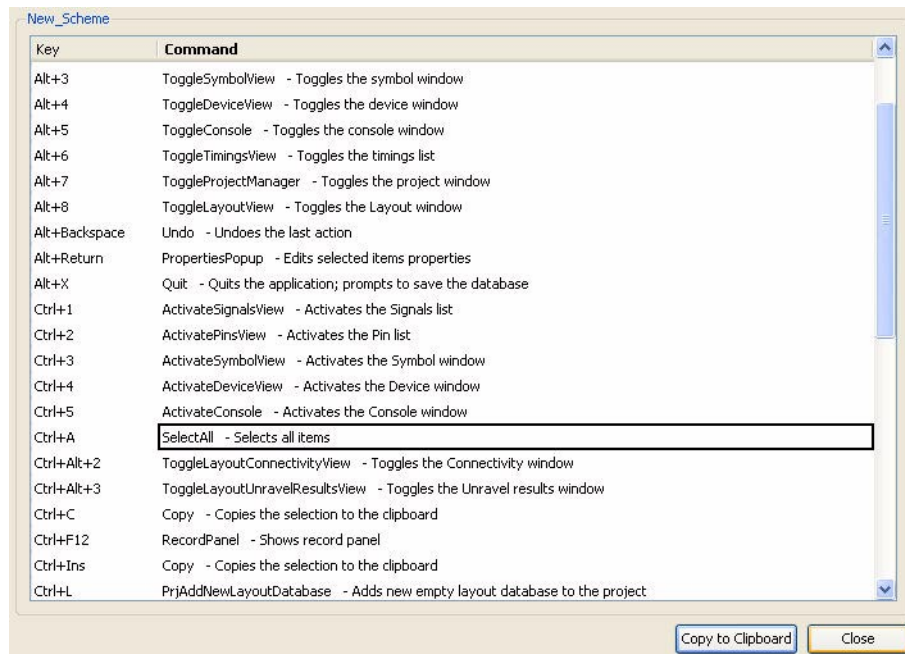
- **Shortcut(s) for selected command:** Displays the shortcut currently defined for the selected command. If blank, no shortcut is defined. A command can be assigned more than one shortcut.
- **Remove:** Deletes the command's keyboard shortcut mapping.
- **Press shortcut key:** Displays the pressed key combination. If the key combination is already assigned to an action (for example, Copy), this action is displayed under **Shortcuts currently used by**.
- **Assign:** Assigns the selected shortcut to the selected command. The button is enabled when you press a unique key combination. In other words, you cannot assign an already-used shortcut.

If you click **Assign** with the default scheme selected, you are prompted to specify a new name for the scheme. This is because the default scheme cannot be modified (it is read-only).

Active Keyboard Shortcuts View

The current keyboard assignments can be displayed from the Help menu as shown in [Figure 13-5](#). To display the assignments, click **Help** > **View Keyboard Shortcuts**.

Figure 13-5. Shortcut Assignments



Click the column header to sort the columns by command or by the shortcut key.

The **Copy to Clipboard** command copies the content of the dialog to the clipboard. If required, it can then be pasted into, for example, a *.txt* file.

Toolbars

Toolbars provide quick access to most frequently used menu options. Toolbars can be docked, undocked and switched on/off. This section provides detailed information on the functions available on each toolbar:

- “[Main Toolbar](#)” on page 184
- “[View Toolbar](#)” on page 185
- “[Mode Toolbar](#)” on page 186
- “[Layout View Toolbar](#)” on page 186
- “[Zoom Toolbar](#)” on page 187
- “[Symbol Toolbar](#)” on page 188
- “[Source Control Toolbar](#)” on page 189

Main Toolbar

The main toolbar contains:

Table 13-9. Main Toolbar

















	New FPGA database
	New layout database
	Open existing project
	Save current database file
	Save all database files and project file
	Close current database file
	Database properties
	Synchronize databases
	Cross Probing

Table 13-9. Main Toolbar (cont.)

	Cut
	Copy
	Paste
	Undo previous operation
	Redo previous operation
	Assign pins
	Assign pins with overwrite

View Toolbar

The View toolbar is used to display or hide different windows in the PADS I/O Designer workspace.

Table 13-10. View Toolbar












	Show/Hide Signals List
	Show/Hide Pins List
	Show/Hide Symbol Window
	Show/Hide Device Window
	Show/Hide Timings Window
	Show/Hide Connectivity List Window (Layout database only)
	Show/Hide Layout Scenarios Window (Layout database only)
	Show/Hide Layout Window (Layout database only)

Table 13-10. View Toolbar

	Show/Hide Project Window
	Show/Hide Console Window
	Show/Hide Properties Window

Mode Toolbar

Table 13-11. Mode Toolbar

	Select Mode
	Zoom Mode
	Pan Mode
	Assign Mode
	Select Pin/Net Mode



Related Topics

- “[Select Pin/Net Mode](#)” on page 124

Layout View Toolbar

When working within a layout database, the Layout View toolbar is available.

Table 13-12. Layout View Toolbar

	Show/Hide Traces
	Show/Hide Netlines



Related Topics

- “[Show Traces](#)” on page 124
- “[Show Netlines](#)” on page 125

Unravel Toolbar







When working within a layout database, the Unravel toolbar is available.

Table 13-13. Unravel Toolbar

	Unravel Nets
	Unravel all FPGAs

Zoom Toolbar

Table 13-14. Zoom Toolbar








	Zoom In
	Zoom Out
	Zoom to Fit
	Zoom to Selection
	Undo Zoom Operation
	Redo Zoom Operation

Related Topics

- [“Device View - Show Layout”](#) on page 33

Symbol Toolbar

Table 13-15. Symbol Toolbar












	Create New Symbol
	Symbols Generator
	Draw Arc
	Draw Circle
	Draw Line
	Draw Rectangle
	Insert Text

Related Topics

- [“Symbol Window”](#) on page 25

Source Control Toolbar

Table 13-16. Source Control Toolbar

	Get Project from Source Control System (Get Project)
	Create New Project (Create Project)
	Add New File (Add New)
	Get Latest Version (Get Latest Version)
	Check Out File (Check Out)
	Check In File (Check In)
	Undo Checkout (Undo Checkout)
	Set Label (Set Label)
	Revision History (Show History)
	Show Differences (Show Diff)
	Refresh Files Status (Refresh Status)

Related Topics

- [“Source Control Settings”](#) on page 51

The following sections are included in this chapter:

- “[Creating and Running TCL Scripts](#)” on page 191
- “[PADS I/O Designer TCL Commands](#)” on page 191
- “[PADS I/O Designer Defined Scalar TCL Variables](#)” on page 217
- “[PADS I/O Designer Defined TCL Array Variables](#)” on page 219

Creating and Running TCL Scripts

In PADS I/O Designer the built-in TCL interpreter is the base for all commands and operations. PADS I/O Designer performs the following steps to execute an operation:

- Constructs a TCL command line
- Displays the command line in the [Console Window](#)
- Executes the command

Note



Commands can be executed directly in GUI mode by entering the command into the [Console Window](#).

PADS I/O Designer includes a TCL interpreter with PADS I/O Designer-specific extensions. All standard TCL commands are available. All TCL scripts to be executed by PADS I/O Designer have to be written using the standard TCL syntax. This manual does not contain any introduction to the TCL language.

What follows is the list of all TCL extensions specific to PADS I/O Designer. Each of listed commands can be invoked with the *-help* parameter and information about the command's parameters will be displayed. Parameter ? has the same meaning as *-help*.

PADS I/O Designer TCL Commands

addandedittext

```
addandedittext [-parentId <value>] [-string | -signalname |  
-pinnumber | -pinname | -pinfunction | -implsignalname | -custom]
```

```
[-text <value>] [-left | -center | -right] [-top | -vcenter |  
-bottom] [-angle <0|90|180|270>] [-font <value>] [-color <value>]  
-pos <value> -symbol <value> -view <value>
```

Adds a graphical text to a symbol and starts-up its edition.

addarc

```
addarc -symbol <value> [-pencolor <value>] [-penwidth <value>]  
[-penstyle <value>] [-factor <value>] start middle end
```

Adds an arc to a symbol.

addbezier

```
addbezier -symbol <value> [-pencolor <value>] [-penwidth <value>]  
[-penstyle <value>] bezierpoints
```

Adds a Bezier curve to a symbol.

addbitmap

```
addbitmap -symbol <value> [-rect <coordinates relative to left top symbol  
corner>] [-file <path to the image file>]
```

Adds an image file to a symbol. This is available for PADS Designer for Xpedition and PADS Designer for non-Xpedition flows only.

addcircle

```
addcircle -symbol <value> [-pencolor <value>] [-penwidth <value>]  
[-penstyle <value>] [-fillstyle <value>] [-fillcolor <value>] rect
```

Adds an circle to a symbol.

addexternalfile

```
addexternalfile [-filepath <path to the external file>] [-filetype  
<value>]
```

Adds information about an external file of any type to a database. This information is used by synchronization commands. The `-filetype` value is a text identifying the type of the file. This text should be exactly the same as the one supported by synchronization command.

addignorelengthnet

```
addignorelengthnet
```

Defines the name of the net whose length will not be minimized by automatic unraveling. This is useful for power and ground nets. Command has only one parameter specifying the name of the net.

addline

```
addline -symbol <value> [-pencolor <value>] [-penwidth <value>]  
[-penstyle <value>] points
```

Adds a line to a symbol.

addportwithlabel

```
addportwithlabel -symbol <value> [-signal | -standalonesignal | -pin  
| -standalonepin | -buspin | -signalpn | -signalpcb | -busslice |  
-unknown] [-extttype <value>] [-shape <value>] [-rangeText <value>]  
[-range <value>] [-angle <0|90|180|270>] [-len <value>] <-signalname  
| -pinname | -pinnumber | -pinfunction | -implsignalname |  
-custom> -pos <value> [-slot <value>] [-diff <value>] [-customtext  
<value>] [-inverted <value>] [-noupdate <value>] [-nocheck <value>]  
[-loaddefaultattr <value>] identifier
```

Adds a port to a symbol.

addportwithlabelandedit

```
addportwithlabelandedit -symbol <value> [-signal | -standalonesignal  
| -pin | -standalonepin | -buspin | -signalpn | -signalpcb  
| -busslice | -unknown] [-extttype <value>] [-shape <value>]  
[-rangeText <value>] [-range <value>] [-angle <0|90|180|270>]  
[-len <value>] <-signalname | -pinname | -pinnumber | -pinfunction  
| -implsignalname | -custom> -pos <value> -view <value> [-slot  
<value>] [-diff <value>] [-customtext <value>] [-noupdate <value>]  
identifier
```

Adds a port to a symbol and enters label editing mode.

addrect

```
addrect -symbol <value> [-pencolor <value>] [-penwidth <value>]  
[-penstyle <value>] [-fillstyle <value>] [-fillcolor <value>] rect
```

Adds a rectangle to a symbol.

addscenario

```
addscenario scenarioName
```

Adds new layout scenario. The new scenario becomes active. Command has only one parameter specifying the name of new scenario.

addtext

```
addtext [-parentId <value>] [-string | -signalname | -pinnumber |  
-pinname | -pinfunction | -portlabel | -implsignalname | -custom]  
[-text <value>] [-left | -center | -right] [-top | -vcenter |  
-bottom] [-angle <0|90|180|270>] [-font <value>] [-deffont] [-color
```

```
<value>] -pos <value> -symbol <value>
```

Adds a graphical text to a symbol.

bottomalign

```
bottomalign symbol items
```

Aligns items for a given symbol.

ces_set_db

```
ces_set_db database_file
```

Command sets the CES database file.

changeattribvisibility

```
changeattribvisibility [-port <value>] [-index <value>] [-owner  
<value>] [-none | -name | -value | -both] symbol attrib
```

Command changes symbol attributes' visibility.

changebuspinrange

```
changebuspinrange name from to
```

Command changes bus pin range. from and to are integer values representing bounds for the range. The width of the bus must remain unchanged for the command to succeed.

changebusrange

```
changebusrange name from to
```

Command changes bus signal range. from and to are integer values representing bounds for the range. The width of the bus must remain unchanged for the command to succeed.

changeexttype

```
changeexttype <symbol> <port> <type>
```

Command changes the port type.

changelabelvisibility

```
changelabelvisibility symbol label visible
```

Command changes the symbol label's visibility.

changeshape

```
changeshape <symbol> <port> <shape>
```

Command changes the shape of a port.

check_sw

```
check_sw [-all] [-SYMBOL_WIZARD_GENERATION] [-HDL_FILE]  
[-SPREADSHEET_FILE] [-NETS_TEXT_IN_FILE] [-FPGA_XCHANGE_FILE]  
[-CONSTRAINTS_FILE] [-PIN_REPORT_FILE] [-TIMING_REPORT_FILE]  
[-SYNTHESIS_CONSTRAINTS_FILE] [-EXPEDITION_LAYOUT] [-EXPEDITION_NETPROPS]  
[-ICDB_CONSTRAINTS] [-LAYOUT_ASSIGNMENTS] [-IDCDV_LOGIC] [-DDP_LOGIC]  
[-IDX_LOGIC] [-EDIF_LOGIC] [-IDX_SIGNALS]
```

Command returns synchronization wizard information about the selected synchronization item. The **-all** switch dumps output for all synchronization wizard items.

Example:

```
check_sw [-CONSTRAINTS_FILE]
```

returns one of the following:

```
# {SW_CONSTRAINTS_FILE_CHECK = Match}  
when tracking of particular synchronization item proceeds smoothly.  
# {SW_CONSTRAINTS_FILE_CHECK = FileNotExist}  
when tracking of a particular synchronization item is disabled because  
file which it tracks does not exist.  
# {SW_CONSTRAINTS_FILE_CHECK = ExportNeeded}  
when synchronization wizard needs to export item data pointed to by a  
particular synchronization item.  
# {SW_CONSTRAINTS_FILE_CHECK = ImportNeeded}  
when synchronization wizard needs to import item data pointed to by a  
particular synchronization item.  
# {SW_CONSTRAINTS_FILE_CHECK = OFF}  
when tracking of particular synchronization wizard item is disabled.
```

copy

```
copy symbol items
```

Command copies symbol items.

creatediffpair

```
creatediffpair diffpair_name pos_name neg_name
```

Command creates a differential pair out of *pos_name* and *neg_name* signals.

createsymbol

```
createsymbol [-pcb] [-background <value>] [-parent <value>] [-rect
```

```
<value>] [-nouupdate <value>] name
```

Command creates a new symbol.

delattrib

```
delattrib [-port <value>] [-index <value>] [-owner <value>] symbol  
attrib
```

Command removes attributes from symbols.

delete

```
delete symbolname items
```

Command performs the Delete operation in a symbol.

deletesymbol

```
deletesymbol <symbol>
```

Command removes the specified symbol.

delinstanceattrib

```
delinstanceattrib [-index <value>] [-owner <value>] symbol attrib
```

Command deletes the specified symbol's attribute.

editbackground

```
editbackground [-on | -off] symbol
```

Command starts/stops the edition of the background in the specified symbol.

eval

```
eval arg[arg]
```

Command evaluates a Tcl script. eval takes one or more arguments, which together comprise a Tcl script containing one or more commands.

executeextool

```
executeextool [-blocking] [-recvstdout] [-recvstderr] [-initdir <value>]
```

Command invokes an external tool.

export_all_schematics

```
export_all_schematics [-force] [-noviewer] [-nopowercheck]
```

Arguments:

[-force] — Forces export/update of schematic means in spite of any synchronization wizard notifications that might appear.

[-noviewer] — Prevents schematic tool from opening after export.

[-nopowercheck] — Prevents update power signals from running before the export. When using the GUI, the update of power signals can be set to run automatically during export by selecting Setup > Settings, PCB Signals Generation, and checking **Enable Automatic PCB Signals Update**.

exportbackground

```
exportbackground fileName symbolName
```

Command exports the symbol's background.

exportschematic

```
exportschematic <-dc | -da | -dx> [-lmc <value>] [-partition <value>]  
[-pdb <value ...>] [-notimestamp] [-noviewer] [-nounattphy]  
[-warningsaserrors] [-overwrite] [-skipcolors] [-portdirection]  
[-multipages] [-danglingnets] [-swapgroups] [-locallmssymbols]  
[-hkp] [-diffpairattr] fileName
```

Command exports schematics.

exportsymbol

```
exportsymbol <-dc | -da | -dx> [-lmc <value>] [-partition <value>]  
[-pdb <value ...>] [-notimestamp] [-noviewer] [-skipcolors]  
[-portdirection] [-swapgroups] [-locallmssymbols] [-hkp] fileName  
symbolName [symbolName]
```

Command exports the specified symbol to the specified file.

generate_constraints_file

Command generates Place & Route constraints file according to the constraints_ file read-only variable set by the set_constraints_file command.

generate_fpga_xchange_file

```
generate_fpga_xchange_file
```

Command generates FPGA Xchange file according to `fpga_xchange_file` read-only variable which is set by `set_fpga_xchange_file` command.

generatesymbols

```
generatesymbols [-create | -update] [-file <filepath and filename>]  
symbolname
```

Command generates symbols. Its arguments correspond to the GUI options available in the Symbols Generator.

Note



The `generatesymbols` command replaces the `symbolwizard` command, but I/O Designer still supports the `symbolwizard` command to allow you to update symbols from older PADS I/O Designer versions.

Examples:

```
generatesymbols  
generatesymbols -create  
generatesymbols -create -file C:/symbols_generator.cfg
```

where `symbol_generator.cfg` is a settings file. It can be created by copying an existing ‘*fpc’ file from the SymbolsGenerator section. This section is automatically updated after each successful symbol generation. You can also get the IOD default file from `/resource/symbols_generator.sgs`. Although the settings file can be edited manually, it is not recommended.

generate_synthesis_constraints_file

Command generates constraints file according to read-only `constraints_file` variable, which is set by `set_synthesis_constraints_file` command.

gettracksyncstatus

```
gettracksyncstatus <-SYMBOL_WIZARD_GENERATION | -HDL_FILE |  
-SPREADSHEET_FILE | -NETS_TEXT_IN_FILE | -FPGA_XCHANGE_FILE |  
-CONSTRAINTS_FILE | -PIN_REPORT_FILE | -TIMING_REPORT_FILE |  
-SYNTHESIS_CONSTRAINTS_FILE | -EXPEDITION_LAYOUT | -EXPEDITION_NETPROPS |  
-ICDB_CONSTRAINTS | -LAYOUT_ASSIGNMENTS | -IDCDV_LOGIC | -DDP_LOGIC |  
-IDX_LOGIC | -EDIF_LOGIC | -IDX_SIGNALS>
```

Command returns 1 if tracking of the particular Synchronization wizard item is enabled, or 0 if tracking is disabled. Command returns an empty string if the particular item is not listed in the synchronization wizard.

help

```
help [<command>]
```

Command displays the list of all PADS I/O Designer-specific commands. If used with argument, displays the usage of the specified command.

hide

```
hide symbolname items
```

Command hides symbol items.

horzmirror

```
horzmirror <symbol> <item1> [<item2> [...]]
```

Command performs the Horizontal Mirror operation. See also vertmirror command.

hsrename

```
hsrename oldName newName
```

Command changes the signal's HDL Name to newName.

importdesign

```
importdesign [-dc | -dx | -da | -edif] [-mapfile <value>] [-viewpoint  
<value>] [-oatfile <value>] [-cddbfile <value>] [-vendor <value>]  
[-tool <value>] [-family <value>] [-device <value>] [-package  
<value>] [-signalsmapping <value ...>] [-create_functional_symbols]  
[-wizard] [-showpartpage] fileName symbols
```

Command imports design information from external design files. See also the Import PCB Design Wizard.

importlmcsymbol

```
importlmcsymbol [-all] [-pcb] [-pcbonly] [-readonly] [-wizard]  
[-nogui] [-parent <value>] [-lmc <value>] [-partition <value>] [-dc  
| -dx | -da | -edif] [-vendor <value>] [-tool <value>] [-family  
<value>] [-device <value>] [-package <value>] [-properties <value  
...>] symbols
```

Command imports symbols from the Central Library.

importsymbol

```
importsymbol [-all] [-pcb] [-pcbonly] [-readonly] [-nogui]  
[-checkportexttype] [-parent <value>] [-mapfile <value>] [-lmc  
<value>] [-partition <value>] [-dc | -dx | -da | -edif] [-mapfonts]  
[-properties <value ...>] [-mgc_comps <value ...>] fileName symbols
```

Command imports specified symbols from the specified file. The switch *-all* imports all symbols, while the *-pcb* switch imports PCB symbols only, adding them to the specified functional block.

invertbuspins

```
invertbuspins pin_name [pin_name ...]
```

Command inverts bus pins.

invertbussignals

```
invertbussignals signal_name [signal_name ...]
```

Command inverts bus signals.

leftalign

```
leftalign symbol items
```

Command aligns items for a given symbol.

mergepins

```
mergepins bus_name {pin_name ...}
```

Command combines the specified pins to a bus pin.

mergesignals

```
mergesignals bus_name {signal_name ...}
```

Command combines the specified signals to a bus.

paste

```
paste symbol data
```

Command pastes the data into the symbol.

pasteat

```
pasteat symbol posx posy items portspos
```

Command pastes the data into the symbol at the specified position.

prjaddexistingdb

```
prjaddnewdb <filename>
```


Adds an existing FPGA or layout database to the current design (same as **File > Add to Board > Existing Database**).

prjaddnewdb

```
prjaddnewdb [-layout]
```

Adds a new FPGA or layout database to the current design (same as **File > Add to Board > New FPGA or New Layout**). Use the argument **layout** to add a layout database to the design.

prjopen

```
prjopen <filename>
```

Opens an existing project (same as **File > Open Project**).

prjclose

```
prjclose [-savealldb] [-saveprj]
```

Closes the currently open project (same as **File > Close Project**).

prjopendb

```
prjopendb name -loadbprefs <Untitled> <my_fpga>
```

Opens the database.

redo

```
redo [ count ]
```

Loads a list of databases specified by the argument.

removefill

```
removefill symbol items
```

Command switches the fill of elements of a symbol to Transparent.

renameattrib

```
renameattrib [-port <value>] symbol old_name new_name owner
```

Command renames symbol's attribute.

renamebuspin

```
renamebuspin <oldname> <newname>
```

Command renames a bus pin.

renamesymbol

```
renamesymbol <oldname> <newname>
```

Command renames a symbol.

reshapearc

```
reshapearc -symbol <value> -id <value> [-factor <value>] start  
middle end
```

Command changes the shape of an arc.

reshapebezier

```
reshapebezier -symbol <value> -id <value> bezierpoints
```

Command changes the shape of a Bezier curve.

reshapecircle

```
reshapecircle -symbol <value> -id <value> newrect
```

Command changes the shape of a circle.

reshapeline

```
reshapeline -symbol <value> -id <value> points
```

Command changes the shape of a line.

reshapeoutline

```
reshapeoutline symbol rect
```

Command changes the shape of the outline of a symbol.

reshaperect

```
reshaperect -symbol <value> -id <value> newrect
```

Command changes the shape of a rectangle.

rightalign

```
rightalign symbol items
```

Command aligns items for a given symbol.

rotate

```
rotate symbol angle items
```

Command rotates elements of a symbol.

rundrc

```
rundrc [-cf <value>] [-src <value>] [-log <value>]
```

Command runs the Design Rule Check for Altera and Xilinx devices. The command can be invoked with the following command line arguments:

```
-cf <value>: allows you to specify a constraint file  
-src <value>: allows you to specify an hdl source file  
-log <value>: allows you to specify a log file
```

sadd

```
sadd name(s) dir type [iostd]
```

Command adds signals with the name and direction specified to the database. If the *name* parameter is a string, the scalar signal will be added. If the *name* parameter is a list, the bus signal will be added. The first element of the list will be the bus' name. The remaining elements of the list will be the names of the elements of the bus. The directions available are: *in*, *out*, *inout*, *buffer*, and *linkage*.

savedb

```
savedb <filename>
```

Command saves the current database to the specified file.

scaddnew

```
scaddnew [-binary] [-description <description>] [-comment <comment>]  
<filename>
```

Command performs the Add New File operation.

scalebackground

```
scalebackground symbolName
```

Command scales the symbol's background so that it fits the outline.

sccheckin

```
sccheckin [-keepcheckedout] [-label <label>] [-comment <comment>]
```

Command performs the Check-In operation.

sccheckout

```
sccheckout [-force]
```

Command performs the Check-Out operation. The switch *-force* executes the operation unconditionally, without this switch the confirmation dialog is displayed, whenever some changes are about to be lost.

sccreateproject

```
sccreateproject [-comment comment] <name>
```

Command performs the Create Project operation.

scdiff

```
scdiff [-short] [-rev <revision>] [-rev2 <revision2>] [-log  
<filename>]
```

Command performs the Diff operation. If the filename argument is given, the differences report is saved to the file, instead of being displayed in the dialog. The *-short* lists only categories of differences, without details. With arguments *-rev*, and *-rev2* you can display diff between any versions of the database.

scgetlatestversion

```
scgetlatestversion [-force]
```

Command performs the Get Latest Version operation. The switch *-force* executes the operation unconditionally, without this switch the confirmation dialog is displayed, whenever some changes are about to be lost.

scgetproject

```
scgetproject <name> <directory>
```

Command performs the Get Project operation.

scgetrev

```
scgetrev [-force] [-rev <revision>] <filename>
```

Command performs the Get operation. The selected revision is stored to the given filename. The switch *-force* executes the operation unconditionally. Without this switch the confirmation dialog is displayed, whenever some changes are about to be lost, and for read-only files.

schematicupdate_all

```
schematicupdate_all [-force] [-nopowercheck]
```

Arguments:

[-force] — Forces export/update of schematic means in spite of any synchronization wizard notifications that might appear.

[-nopowercheck] — Prevents update power signals from running before the export. When using the GUI, the update of power signals can be set to run automatically during export by selecting Setup > Settings, PCB Signals Generation, and checking **Enable Automatic PCB Signals Update**.

schistory

```
schistory [-log <filename> [-notime]]
```

Command performs the Show History operation. If the filename argument is given, the history of versions is saved to the file, instead of being displayed in the dialog. The switch **-notime** disables saving of date/time information.

sclabel

```
sclabel [-comment <comment>] [-rev <revision>] <label>
```

Command performs the Label operation.

screfreshstatus

Command performs the Refresh Status operation.

scundocheckout

```
scundocheckout [-force]
```

Command performs the Undo Check-Out operation. The switch *-force* executes the operation unconditionally, without this switch the confirmation dialog is displayed, whenever some changes are about to be lost.

set_cdb_flow

```
set_cdb_flow [-hkp] [-cdbflow <value>] [-lmc <value>] [-partition  
<value>] [-pdb <value ...>]
```

Command sets the Central Library export options.

set_constraints_file

```
set_constraints_file [-maxplus_acf | -quartus_csf | -ise_ucf |  
-designer_pdc | -designer_pin | -designer_gcf | -quartus_qsf |  
-lattice_prf | -lattice_lpf] constraints_file_name
```

Command changes constraints file.

set_design_architect_project

```
set_design_architect_project [-path <value>] [-viewpoint <value>]  
[-mapfile <value>] [-catalog <value>]
```

Command sets the Design Architect project path and viewport file path.

set_dx_designer_project

```
set_dx_designer_project [-path <value>] [-top <value>] [-oat <value>]
```

Command sets the PADS Designer project path and OAT file path.

set_expedition_layout

```
set_expedition_layout [-refdes <value>] layout_file
```

Command sets the Xpedition or Pads Layout layout file.

set_fpga_xchange_file

```
set_fpga_xchange_file <filename>
```

Command changes FPGA Xchange file.

setpagesize

```
setpagesize <-A0 | -A1 | -A2 | -A3 | -A4 | -A | -B | -C | -D | -E> <-landscape | -portrait>
```

Example:

```
setpagesize -A4 -landscape
```

Command changes size of exported sheets. This command also updates border settings used during export sheet to PADS Designer.

set_pin_report_file

```
set_pin_report_file [-ise_pad | -ise_csv | -quartus_pin | -maxplus_fit  
| -designer_rpt | -isp_pad] pin_report_file_name
```

Command changes the pin report file.

set_synthesis_constraints_file

```
set_synthesis_constraints_file [-synplicity_sdc | -synopsys_sdc |  
-leonardo_ctr | -precision_sdc | -xst_xcf] <filename>
```

Command changes synthesis constraints file.

set_timing_report_file

```
set_timing_report_file [-ise_twr | -ise_rpt] timing report file name
```

Command changes the timing report file.

setalign

```
setalign [-left | -center | -right] [-top | -vcenter | -bottom]  
symbol item
```

Command changes the alignment of texts in a symbol.

setattrib

```
setattrib [-port <value>] [-index <value>] [-owner <value>] [-locked  
<value>] symbol attrib [values]
```

Command changes the symbol attributes.

setbackground

```
setbackground <symbol> <background>
```

Command changes the background in the specified symbol.

setbackgroundproperty

```
setbackgroundproperty symbol background items
```

Sets the background properties for particular items.

setdefaultlinestyle

```
setdefaultlinestyle symbol items
```

The command sets default line style for items in a symbol.

setdefaultoutline

Command changes the outline color of the symbol elements to default.

setdesignator

```
setdesignator symbol designator
```

Command sets the symbol designator.

setfillcolor

```
setfillcolor symbol color items
```

Command changes the fill color of the symbol elements.

setFont

```
setFont -symbol <value> [-font <value>] [-deffont] items
```

Command changes the font of texts in a symbol.

setfontsize

```
setfontsize symbol size items
```

Command changes the font size of texts in a symbol.

setgeom

```
setgeom geom
```

Command sets the symbol property: PCB Geometry (Design Architect) or PKG_TYPE (PADS Designer).

setfunctionalblock

```
setfunctionalblock <pcbsymbol> <functionalblock>
```

Command changes functional block to which the specified PCB symbol belongs.

setinstanceattrib

```
setinstanceattrib [-index <value>] [-locked <value>] [-owner  
<value>] symbol attrib value
```

Command sets the specified symbol's attribute.

setinvert

```
setinvert [-on | -off] symbol port
```

Command changes the Inverted attribute of a port.

setlabeltype

```
setlabeltype -symbol <value> -port <value> <-signalname | -pinname |  
-pinnumber | -pinfunction | -implsignalname | -custom>
```

Command changes the text which is used as port label.

setlinestyle

```
setlinestyle symbol style items
```

Command sets the line style for symbol's items.

setoutlinecolor

```
setoutlinecolor symbol color items
```

Command changes the outline color of elements of a symbol.

setpart

```
setpart -vendor <vendor> -tool <tool> -family <family> -device  
<device> -package <package> [-premove]
```

Command changes the selected physical device.

Note



The `-premove` switch is for changing the numbering convention from, for example, P1, P2, P3... to 1, 2, 3...

setpartno

```
setpartno partno
```

Command sets the part number.

setpinlabel

```
setpinlabel -symbol <value> -port <value> -recreate <value>  
-labeltype <value> label
```

Command changes the port label properties.

setportlength

```
setportlength symbol port len
```

Command changes the port length.

setreadonly

```
setreadonly symbol readonly
```

Command makes the symbol read-only.

setsigprop

```
setsigprop signal prop_name prop_value
```

Command sets the additional signal properties.

setsimilardevices

```
setsimilardevices part
```

Command performs the **View > Pins from Other Devices** operation.

setsourcefile

```
setsourcefile [-vhdl | -verilog | -edif_xml] [-unit <value>]  
[-include <value>] [-noload] file [additional_files ...]
```

Command changes the selected HDL file. The language of the file may be also selected, as well as the name of an entity/module to be used. Additional HDL files may be appended as next arguments.

setstyle

```
setstyle [-color <value>] [-automaticcolor] [-linestyle <value>]  
[-linewidth <value>] [-fillstyle <value>] [-automaticstyle]  
[-items <value ...>] [-symbol <value>]
```

Command sets the color, line style, fill style, and line width.

settext

```
settext symbol text items
```

Command changes the text in a symbol.

settextcolor

```
settextcolor symbol color items
```

Command changes the color of a text in a symbol.

settracksyncstatus

```
settracksyncstatus <-SYMBOL_WIZARD_GENERATION|-HDL_FILE|  
-SPREADSHEET_FILE|-NETS_TEXT_IN_FILE|-FPGA_XCHANGE_FILE|  
-CONSTRAINTS_FILE|-PIN_REPORT_FILE|-TIMING_REPORT_FILE|  
-SYNTHESIS_CONSTRAINTS_FILE|-EXPEDITION_LAYOUT|-EXPEDITION_NETPROPS|  
-ICDB_CONSTRAINTS|-LAYOUT_ASSIGNMENTS|-IDCDV_LOGIC|-DDP_LOGIC|  
-IDX_LOGIC|-EDIF_LOGIC|-IDX_SIGNALS> <-on|-off>
```

Command Enables (**-on** switch) or disables (**-off** switch) a particular Synchronization wizard item.

setvendor

```
setvendor vendor
```

Sets the current vendor.

show

```
show symbolname items
```

Command shows symbol items.

source

```
source scriptfile
```

Command invokes a script file in the PADS I/O Designer environment.

splitdiffpair

```
splitdiffpair diffpair_name
```

Command splits a differential pair created through *creatediffpair* command.

splitpins

```
splitpins <buspin> ...
```

Command splits the specified bus pins.

splitsignals

```
splitsignals <bus> ...
```

Command splits the specified bus signals.

sremove

```
sremove [-norec] name ...
```

Command removes specified signals from the database.

srename

```
srename <oldname> <newname>
```

Command renames a signal.

swappins

```
swappins [-pins | -bus | -swapgroup | -bank] pins ...
```

The command swaps pins according to a rule specified by one of the *-pins* / *-bus* / *-swapgroup* / *-bank* options.

symbolwizard

```
symbolwizard [-pcb | -functional | -both] [<-single | -splitByPowerBanks  
| -separateDataAndControl | -splitByComponents> [-reuseexistingsymbols]  
[-splitfunctional] [-splitfunctionalbypcb] [-powerpins] [-configpins]  
[-splitbypagesize] [-threshold <value>] [-powerpinsbybank <value>]  
[-splitpowerpinsbybank] [-powerpinsbyimplsignal <value>] [-  
splitpowerpinsbyimplsignal]  
[-addpowers <value>] [-addpowers2pcb] [-addconfigs <value>]  
[-addconfigs2pcb] [-vrefpins] [-dcipins] [-nomgt | -symbolpermgtchannel  
| -symbolforallmgtchannels | -symbolpermgtblock | -symbolforallmgtblocks]  
[-fullpcb] [-addsymbolname] [-description <value>] [-clktop |  
-clkbottom | -clkleft | -clkright] [-vcctop | -vccbottom | -vccleft  
| -vccright] [-gndtop | -gndbottom | -gndleft | -gndright] [-pinname  
| -signalname | -pinnumber | -pinfunction | -implsignalname |  
-custom] [-len <value>] [-spacing <value>] [-background <value>]  
[-additionallabel] [-addpinname | -addsignalname | -addpinnumber |  
-addpinfunction | -addimplsignalname | -addcustom] [-addlabelover  
| -addlabelside | -addlabelunder] [-pcbilen <value>] [-pcbspacing  
<value>] [-pcbbackground <value>] [-pcbpinname | -pcbsignalname |  
-pcbpinnumber | -pcbpinfunction | -pcbimplsignalname | -pcbcustom]  
[-pcbadditionallabel] [-pcbaddpinname | -pcbaddsignalname |  
-pcbaddpinnumber | -pcbaddpinfunction | -pcbaddimplsignalname |  
-pcbaddcustom] [-pcbaddlabelover | -pcbaddlabelside | -pcbaddlabelunder]  
[-powerpinname | -powersignalname | -powerpinnumber | -powerpinfunction  
| -powerimplsignalname | -powercustom] [-poweradditionallabel]  
[-poweraddpinname | -poweraddsignalname | -poweraddpinnumber |  
-poweraddpinfunction | -poweraddimplsignalname | -poweraddcustom]  
[-poweraddlabelover | -poweraddlabelside | -poweraddlabelunder]  
[-configpinname | -configsignalname | -configpinnumber | -  
configpinfunction  
| -configimplsignalname | -configcustom] [-configadditionallabel]  
[-configaddpinname | -configaddsignalname | -configaddpinnumber |  
-configaddpinfunction | -configaddimplsignalname | -configaddcustom]  
[-configaddlabelover | -configaddlabelside | -configaddlabelunder]  
[-pcbpartno <value>] [-pcbgeom <value>] symbolname
```

Note

The `generatesymbols` command replaces the `symbolwizard` command, but PADS I/O Designer still supports `symbolwizard` to allow you to update symbols from older PADS I/O Designer versions.

taddclockcon

```
taddclockcon -clock <value> -reqTime <value ...> -dutyCycle <value ...> [-actTime <value ...>] [-hl <value>]
```

The command adds a Clock timing constraint.

taddtcocon

```
taddtcocon -signal <value> -clock <value> -reqTime <value ...> [-actTime <value ...>] [-min <value ...>] [-ba <value>] [-hl <value>]
```

The command adds a Clock to Pad timing constraint.

taddtpdcon

```
taddtpdcon -signalIn <value> -signalOut <value> -reqTime <value ...> [-actTime <value ...>]
```

The command adds a Pad to Pad timing constraint.

taddtsucon

```
taddtsucon -signal <value> -clock <value> -reqTime <value ...> [-actTime <value ...>] [-hold <value ...>] [-ba <value>] [-hl <value>]
```

The command adds a Pad to Setup timing constraint.

tdelclockcon

```
tdelclockcon -clock <value>
```

The command removes a Clock timing constraint.

tdeltcocon

```
tdeltcocon -signal <value> -clock <value>
```

The command removes a Clock to Pad timing constraint.

tdeltpdcon

```
tdeltpdcon -signalIn <value> -signalOut <value>
```

The command removes a Pad to Pad timing constraint.

tdeltsucon

```
tdeltsucon -signal <value> -clock <value>
```

The command removes a Pad to Setup timing constraint.

topalign

```
topalign symbol items
```

Aligns items for a given symbol.

typescompatibility

```
typescompatibility [-set|-rem] signal_type pin_type
```

Makes or removes an association between the specified signal and pin types such that assignments between signals and pins of those types can be made. By default, this command sets types compatibility using the `-set` switch. When removing types compatibility it is necessary to add the switch `-rem`. It's obligatory to specify signal and pin types.

unassign

```
unassign signals ...
```

Used to unassign signals.

unassignall

```
unassignall
```

Removes all assignments.

unassignpins

```
unassignpins pins ...
```

The command unassigns pins.

undo

```
undo [<levels>]
```

Command performs an Undo operation.

unravel

Alias for the unravelnets command. See [unravelnets](#).

unravelnets

```
unravelnets [-p] [-ignoreBuses] [-unused] [-useLayers] [-ignoreTraces]  
[-offConcurrency] [-crossLevel <value>] [-components <value ...>] [-items  
<value ...>]
```

Unravels connections between components (used for both FPGA and layout databases). By default (without switches) optimization minimizes number of crossings. It does not use unused pins and does not unravel buses concurrently. It does not optimize on different layers separately. It optimizes for all signals. When optimizing the layout database, it optimizes all components if possible.

Arguments:

- p — signifies that *items* provided are pins not signals.
- ignoreBuses — unravels scalars and busses concurrently.
- unused — use unused pins for unraveling.
- useLayers — unravel on different layers separately, used only in Xpedition (supported only in layout database).
- ignoreTraces — ignores connections through traces not finished with “Route Targets” (supported only in layout database).
- offConcurrency — limits unraveling to only a single selected FPGA, so as to not consume a multi-chip license.
- crossLevel — No longer used - left for backward compatibility.
- components — refDes’ of components to be unraveled (not important for FPGA database because there's only one component) - if not provided, IOD unravels all FPGA components on the layout.
- items — signal names or pin numbers connected to the nets that need to be unraveled. Not used when unraveling more than 1 component.

updatehdl

Command performs the update operation from the HDL file.

update_from_constraints_file

Command performs the update operation from the constraints file defined by `constraints_file` variable. If the `constraints_file` variable has not been set, no update is performed.

update_from_fpga_xchange_file

Command performs the update operation from the FPGA Xchange file defined by `fpga_xchange_file` variable. If the `fpga_xchange_file` variable has not been set, no update is performed.

update_from_pin_report_file

Command performs the update operation from the pin report file defined by `pin_report_file` variable. If the `pin_report_file` variable has not been set, no update is performed.

update_from_synthesis_constraints_file

Command performs the update operation from the synthesis constraints file defined by `synthesis_constraints_file` variable. If the `synthesis_constraints_file` variable has not been set, no update is performed.

updatepowersignals

```
updatepowersignals
```

Command creates and assigns (and if necessary, reassigns) signals that should be assigned to ensure correct device operation.

update_symbols_attributes

```
update_symbols_attributes
```

The command updates symbol's attributes.

updatesymbols

```
updatesymbols <-dc | -da | -dx> [-graphics] [-attributes] [-assignments]  
[-designator] [-verifyIODattribs] [-skip_signals <value ...>]  
[fileName]
```

Command performs the update operation from the file containing symbols.

Arguments:

```
-assignments
```

Update pin assignments

```
-graphics
```

Update graphical data

```
-attributes
```

Update attributes

<filename>

Use the file <filename> to perform the update instead of current file.

vertmirror

```
vertmirror <symbol> <item1> [<item2> [...]]
```

Command performs the Vertical Mirror operation. See also horzmirror command.

PADS I/O Designer Defined Scalar TCL Variables

constraints_file

TCL read-only variable containing the name of the constraints file.

constraints_file_format

TCL read-only variable containing the name of the constraints file format.

database_file

TCL variable containing the file name of the current database.

ddp_project_path

TCL variable containing the directory name of the selected Design Architect project.

device

TCL variable containing the name of the selected device. This variable is read-only; to change device use the setpart command.

dx_project_path

TCL variable containing the directory name of the selected DxDesigner project.

family

TCL variable containing the name of the selected family. This variable is read-only; to change family use the setpart command.

fpga_xchange_file

TCL read-only variable containing the name of the FPGA Xchange file.

hdl_file

TCL variable containing the name of the selected HDL file. This variable is read-only; to change an HDL file name use the `setsourcefile` command.

package

TCL variable containing the name of the selected package. This variable is read-only; to change the package use the `setpart` command.

pin_report_file

TCL read-only variable containing the name of the pin report file.

pin_report_file_format

TCL read-only variable containing the name of the pin report file format.

speed

TCL variable containing the selected speed.

sso_bank_threshold

TCL variable containing the SSO Bank Threshold.

sso_package_allowance

TCL variable containing the SSO Package Allowance.

synthesis_constraints_file

TCL read-only variable containing the name of the synthesis constraints file.

synthesis_constraints_file_format

TCL read-only variable containing the name of the synthesis constraints file format.

synthesis_tool

TCL variable containing the synthesis tool name.

timing_report_file

TCL variable containing path to the timing report file.

timing_report_file_format

TCL variable containing the timing report file format string.

vendor

TCL variable containing the name of the selected vendor. This variable is read-only; to change the vendor name use the setpart command.

verilog_search_path

TCL variable containing the Verilog search path.

PADS I/O Designer Defined TCL Array Variables

hdlsigname

TCL array containing HDL names of signals.

sdiffnames

TCL array mapping signal names to differential pairs.

sdir

TCL array mapping signal names to signal directions.

slock

TCL array containing the Locked By strings. All signals from the current database are stored in the slock array. The values of the slock array are the Locked By strings displayed in the Signal List.

sswapgroup

TCL array mapping signals to swap groups.

stypes

TCL array mapping signals to types.

pinbank

TCL array mapping pin numbers to power banks. This array is read-only.

pinfunction

TCL array mapping pin numbers to pin functions. This array is read-only.

pinname

TCL array mapping pin numbers to pin names. This array is read-only.

pinnumber

TCL array containing the pin assignments. All signals from the current database are stored in the pinnumber array. The values of the pinnumber array are the assigned pin numbers. Signals without assigned pins have an empty value in the array.

pinsignal

TCL array mapping pin numbers to signals. The values of the pinsignal array are the signals to which pins are assigned. Pins not assigned to signals have an empty value in the array.

pinswapgroup

TCL array mapping pins to swap groups.

pintypes

TCL array mapping pin numbers to pin types such as clock, diff, etc.

plock

TCL array containing the Locked By strings. All pins from the current database are stored in the plock array. The values of the plock array are the Locked By strings displayed in the Pin List.

Source Control Configuration

PADS I/O Designer supports several source control systems, such as:

- CVS
- RCS
- Source Safe

To choose a source control system, open the **Setup > Settings** dialog. On the Source Control page a system may be chosen. Additional configuration settings are available there, depending on the chosen system.


The source control interface is implemented in PADS I/O Designer using TCL interface. It is possible to add the support for another source control system by editing a TCL script. The list in the Settings dialog contains the custom item. Selecting this item turns on the interface described in the *scc_custom.tcl* file, located in the *tcl* subdirectory of the PADS I/O Designer installation directory. The comments on the *scc_custom.tcl* file contain the details on customizing.

The main advantage of using source control is to have a repository, which stores the base version of the database. Designers do not modify the database directly in the repository, but rather work on the local copy. The usual way to utilize Source Control is to get the database from the repository, and store it in the local directory. The database will be read-only and may be reviewed. If a designer wants to make some changes in the database, the database should be checked-out, which means that the database is marked in the repository as being modified by the designer. Only one designer may modify the database at a time. After completing changes, the database may be checked-in to the repository, creating a new version of the stored database. PADS I/O Designer integrates support to check-in and check-out databases, review database history, and so on.

Source Control Usage

All Source Control options can be found in two places: in the **File > Source Control** menu, and in the Source Control toolbar.

Getting Existing Databases

To start using a database that is already stored in the repository (e.g., added there by some other designer) use the **File > Source Control > Get Project** command  button. This button displays a dialog for entering the Source Control project name, and the local directory, where the database is to be stored.

Adding Databases to the Repository

To add a database to the repository:

1. Select **File > Source Control > Create Project**.
2. Enter the Source Control project name
3. Enter a comment which will be stored in the repository with the first version of the database.

Additional Files in the Repository

To add an additional file, such as an HDL file, to the repository:

1. Select **File > Source Control > Add New**.
2. Specify the path to the required file.
3. Enter a comment which will be stored in the repository.

Since binary files may in some circumstances be explicitly stated as such in the repository, the **Add New** command contains the Binary option.

Getting Latest Database Version

To get the latest version of the database from the repository, and store it in the local directory, select **File > Source Control > Get Latest Version**.

Checking Out the Database

To check out the database, select **File > Source Control > Check Out**. If the database is not checked out, each time you try to modify it the software asks you whether or not the database should be checked out.

Checking In the Database

To check-in the database:

1. Select **File > Source Control > Check In**.
2. Enter a comment which will be stored in the repository to summarize all changes made to the database since it was checked out.

Undo Check Out

To cancel all the changes made after the database was checked out, and resign from checking out, select **File > Source Control > Undo Check Out**.

Setting Labels

To set the label (or tag) to the last version of the database, select **File > Source Control > Set Label**. Labels are displayed in the [Browse Changes History](#) dialog and may be used when source control software is used not within PADS I/O Designer. For instance, many source control systems contain commands to go to a version with a given label.

Browse Changes History

To review the history of changes made to the database, select **File > Source Control > Show History**. The dialog displays all database versions, with labels, including the user that created the version, the time of creation, and comments. All this information may be displayed in more convenient way after clicking on the Details button.

The Diff button in the History dialog allows comparing two version of the database. In case two versions are selected, the button will display the differences between the selected versions. If a single version is selected, the differences between the selected version and local version will be displayed. The dialog displayed after clicking Diff is described in details below, in the section about the Show Diff command.

The Get button in the History dialog allows retrieval of any version from the repository, and save it in the local directory.

Showing Differences

To demonstrate the differences between the local version, and the latest version stored in the repository, select **File > Source Control > Show Diff**. This button displays a dialog presenting information in two panes. The left one summarizes the differences, and the right one presents details about the category selected on the left side.

Refreshing Source Control Status

It is possible that the state of the database file is changed outside of PADS I/O Designer. To synchronize Source Control, select **File > Source Control > Refresh Status**.

Chapter 16

Dialog Boxes and Field Reference

This chapter contains reference information on the following dialog boxes in PADS I/O Designer.

Table 16-1. Dialog Boxes Summary

Dialog Box	Description
Settings Dialog Box	Use this dialog box to view and edit PADS I/O Designer settings.
Database Properties Dialog Box	Use this dialog box to set information for the database.
Layout Database Properties Dialog Box	Use this dialog box to specify layout database properties.
Add Signal Dialog Box	Use this dialog box to create a new signal and add it to the database.
Edit Primitive Value(s) Dialog Box	Used to specify the values of the selected rule primitive when creating or editing rules using the Rules Wizard.
Import Design Wizard Dialog Box	Use this dialog box to import symbols with signals and pins into an FPGA database directly from the design.
xDX IOD Components Dialog Box	Use this dialog box, within a layout database, to specify components to import into the layout either from a design, or added manually.
Library Components Dialog Box	Use this dialog box, within a layout database, to specify components to import into the layout either from a design, or added manually.
Rule Editor Dialog Box	Use this dialog box to manage user-defined rules for an FPGA database and give access to the Rules Wizard in order to create new rules.
Rules Wizard	Use this dialog box to define I/O placement rules for a database.

Table 16-1. Dialog Boxes Summary

Dialog Box	Description
Unravel Nets Dialog Box	Use this dialog box to specify settings for unraveling nets.
External Tools Dialog	Use this dialog box to define external programs (such as the layout tool), tcl scripts, and tcl commands that can be invoked directly from the Tools menu.

Settings Dialog Box

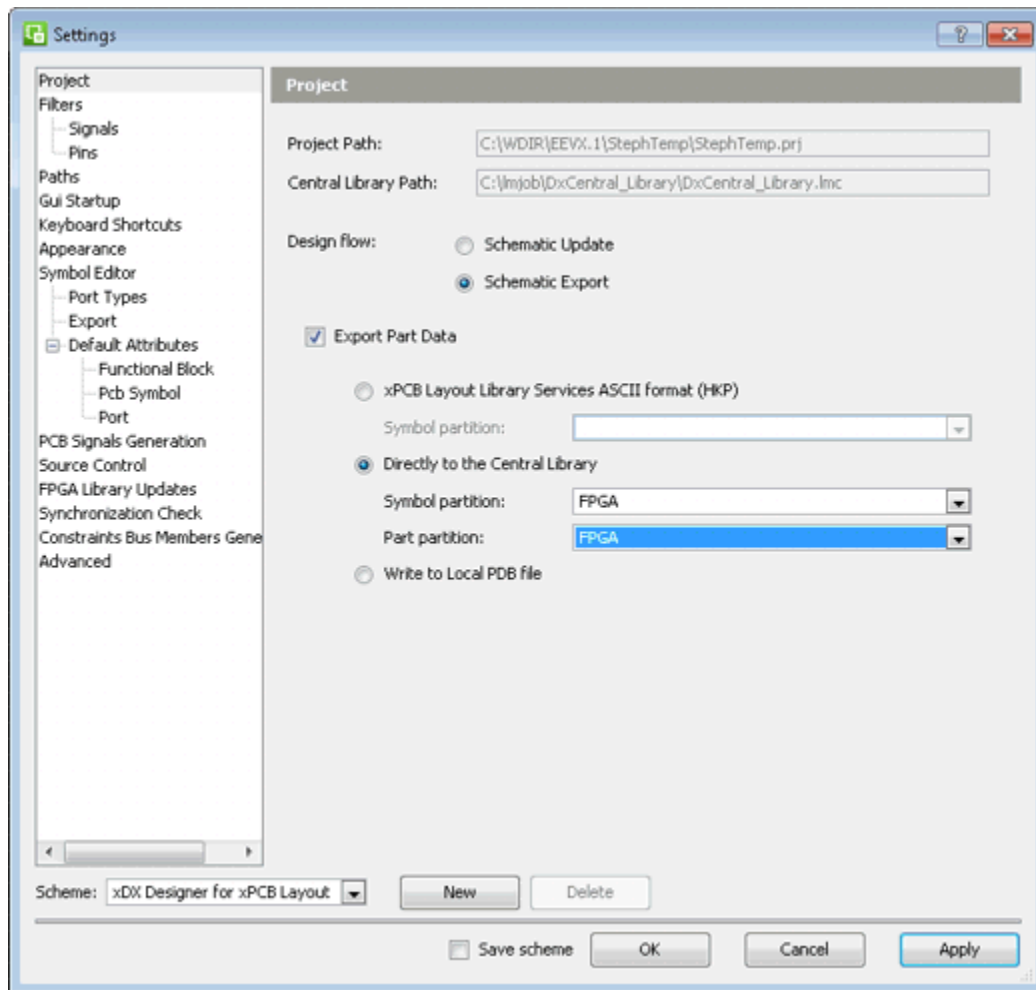
To access:

- **Setup > Settings**
- **File > Project Properties**

Use this dialog box to view and edit PADS I/O Designer settings.

Highlight an item in the left pane to display the settings in the right pane (in [Figure 16-1](#)).

Figure 16-1. Settings Dialog Box - Project



Fields

Table 16-2. Settings Dialog Box Contents

Field	Description
Project	
Project Path	The path to the current project.

Database Properties Dialog Box

To access: **File > Database Properties**

Use this dialog box to set information for the database.

The available contents differ depending on the type of database: FPGA (*.fpc*) or layout (*.lpc*).

To access the FPGA properties dialog box, click **File > Database Properties**, and click the appropriate page from the list. (You can change the active FPGA by double-clicking it from the Project Window.)

Table 16-3. FPGA Properties Dialog Box

Field	Description
Vendor and Device	
Vendor	Specifies the device Vendor: <ul style="list-style-type: none">• Actel• Altera• Lattice• Xilinx
Tool	Specifies the tool or library from which the device originates
Family	Specifies the family of devices
Device	Specifies the device itself
Package	Specifies the package for the device
Speed	Specifies the speed at which the device runs
Default single	Specifies the default I/O standard for single signals.
Default differential	Specifies the default I/O standard for differential signals.
Signals Source	
Signals source	Specifies the format of the source signals file: VHDL, Verilog, or Spreadsheet.
File Path	Specifies the path to the source data file.
Browse	Opens a Browse window to navigate to the source file.
More	Opens the Settings dialog to show related signals settings.
VHDL definition (active when the signals source is VHDL)	
VHDL File	Specifies the path to the VHDL file.
Entity	Specifies the entity from the source data to use.
Additional VHDL files	Specifies additional VHDL files to use.

Table 16-3. FPGA Properties Dialog Box

Field	Description
Verilog definition (active when the signals source is Verilog)	
Verilog File	Specifies the path to the Verilog file.
Module	Specifies the module from the source data to use.
Verilog search paths	Specifies additional Verilog files to use.
Spreadsheet definition (active when the signals source is Spreadsheet)	
Spreadsheet File	Specifies the path to the spreadsheet file.
Delimiter	Specifies the delimiter used in the imported data.
Attributes	Specifies the signal attribute order to use in the spreadsheet.
File preview	Displays a preview of the data.
Place and Route	
Constraints file name	Specifies a Place and Route constraints file.
Pin report file name	Specifies a Place and Route pin report file.
Timing report file name	Specifies a Place and Route timing report file.
FPGAX file name	Specifies an FPGAX file.
Synthesis	
Tool name	Specifies the Synthesis tool used: <ul style="list-style-type: none">• Leonardo Spectrum• Precision• Synopsys• Synplify Pro• Xilinx Synthesis
Constraints file	Specifies a Synthesis constraints file.
PCB Flow	
Part number	Allows you to enter a PCB Part Number.
Cell name	Allows you to enter a PCB cell name.
Ref Des	Allows you to enter a Reference Designator for the device.
Advanced	
SSO Package Allowance	Sets an allowance value for the SSO check.
SSO Bank Threshold	Sets a threshold value for the SSO check.

Related Topics

[FPGA Device Setup](#)

FPGA Xchange Files

Importing FPGA Vendor Files

Place and Route Constraints Files

Synthesis Constraints Files

Layout Database Properties Dialog Box

To access: Make sure the layout is highlighted (double-click it) in the Project window, then select **File > Database Properties**.

Use this dialog box to specify layout database properties.

Table 16-4. Layout Database Properties Dialog Box

Field	Description
Netlist Source	
Netlist source	Specifies the format of the source data for the Netlist file: Schematic, or Spreadsheet.
Spreadsheet	
Spreadsheet file	Specifies the path to the source data file.
Delimiter	Specifies the character which defines the start of each entry.
Attributes	Filters components in the schematic to import only those you want to use in the I/O Designer optimization process.
Component Filter	
Import Layout RefDes Filter - Regular Expression	Filters components in the layout whose RefDes satisfies the regular expression. Example: Enter <i>R/C</i> to disable the display of all components with a RefDes containing R or C.

Add Signal Dialog Box

To access:

- Right-click in the Signals List and select **Add Signal...**
- Select the menu item **Edit > Add Signal**

Use this dialog box to create a new signal and add it to the database.

The new signal appears in the **Add Signal** dialog.

Table 16-5. Add Signal Dialog Box Contents

Field	Description	Allowed Values
Name	Specifies a name for the signal	--
Direction	Specifies the signal's direction	In, Out, Inout, Buffer, Linkage
Type	Specifies the signal's type	--
I/O Standard	Specifies an I/O Standard for the signal	--
Range	Specifies a range of values for the signal	

Related Topics

[Signals List](#)

[Assigning Signals to Pins](#)

Edit Primitive Value(s) Dialog Box

To access: Move a rule primitive from the **Available Primitives:** list to the **Rule Primitives:** list on the **I/O cell selection** page of the [Rules Wizard](#).

Used to specify the values of the selected rule primitive when creating or editing rules using the [Rules Wizard](#).

Table 16-6. Edit Primitive Value(s) Dialog Box Contents

Field	Description
Operator	Specifies an operator for the primitive such that when the primitive is, is not, contains, does not contain, matches or does not match the value(s) specified, the rule is executed.
Operand	Specifies the value(s) or range of values for the condition.
List of available values	Allows selection of values from those available. The presence of this field is dependent upon the type of operator selected.

Related Topics

[User-defined Rules](#)
[Rules Wizard](#)

Import Design Wizard Dialog Box

To access: **Import > PCB Design Wizard**

Use this dialog box to import symbols with signals and pins into an FPGA database directly from the design.

Table 16-7. Import Design Wizard Dialog Box Contents

Field	Description
Pcb Flow	
Project Path	Specifies the path to the project containing the required design.
Design Name	Selects the top-level design within the specified project, containing the required component.
Ref Des	Specifies the Reference Designator of the required component.
Choose Symbols	
Filter	Allows the list of available symbols to be filtered using regular expressions.
...	Accesses a list of popular filter options.
Apply	Applies the filter to the list of symbols in the design.
Import signals and assignments only	The wizard will import only the signals and assignment data from the selected symbol in the list, the symbol itself will not be imported.
Generate Functional Blocks for PCB Symbols	Functional block symbols will be generated in PADS I/O Designer for the imported PCB symbol.
FPGA Flow	
Vendor	Specifies the device Vendor: <ul style="list-style-type: none"> • Actel • Altera • Lattice • Xilinx
Tool/Library	Specifies the tool or library from which the device originates
Family	Specifies the family of devices
Device	Specifies the device itself
Package	Specifies the package for the device
Speed	Specifies the speed at which the device runs
Choose signal mapping	

Table 16-7. Import Design Wizard Dialog Box Contents

Field	Description
Imported PCB signal	Lists the signals found in the PCB symbol specified for import.
Existing HDL signal	Lists the HDL signals currently held in the FPGA database.

Related Topics

[PCB Design Wizard](#)

xDX IOD Components Dialog Box

To access: **Tools > xDX IOD Components**

Use this dialog box, within a layout database, to specify components to import into the layout either from a design, or added manually.

Table 16-8. xDX IOD Components Dialog Box Contents

Field	Description
Instance name	Allows a name to be entered for the component instance.
Database File	Allows selection of loaded database files from opened PADS I/O Designer projects.
Database Type	FPGA.
Position X	X coordinate of the component's position.
Position Y	Y coordinate of the component's position.
Angle	Specifies the component's rotational position.
Mirror	Mirrors the component.
Visibility	Turns visibility of component on and off in layout.
(Right-click) Insert	Adds a new component to the list.
(Right-click) Delete	Removes the currently selected component from the list.

Library Components Dialog Box

To access: **Tools > Library Components**

Use this dialog box within a layout database to specify components to import into the layout either from a design, or added manually.

Table 16-9. Library Components Dialog Box Contents

Field	Description
Instance name	The name of the component instance.
Partition	The library component's partition name.
Cell	The library component's cell name.
Position X	X coordinate of the component's position.
Position Y	Y coordinate of the component's position.
Angle	Specifies the component's rotational position
Mirror	Mirrors the component
Visibility	Turns visibility of component on and off in layout.
(Right-click) Delete	Removes the currently selected component from the list





Rule Editor Dialog Box

To access: **Tools > Rule Editor**

Use this dialog box to manage user-defined rules for an FPGA database and give access to the [Rules Wizard](#) in order to create new rules.

This dialog box allows rules databases to be imported and exported from/to other projects.


Table 16-10. Rule Editor Dialog Box Contents

Field	Description
Import	Allows a previously created rules engine database file to be imported.
Export	Allows a previously created rules engine database file to be imported.
Rules:	Lists the name and description of all the rules applied to the current database.
	Opens the Rules Wizard to begin creation of a new rule.
	Deletes the rule currently selected in the Rules: list.
	Moves the selected rule up one place in the list.
	Moves the selected rule down one place in the list.
Object Description	Gives details on the rule currently selected in the Rules: list. Underlined values in this window can be selected for editing; this opens the Edit Primitive Value(s) Dialog Box .

Related Topics

[User-defined Rules](#)
[Rules Wizard](#)
[Edit Primitive Value\(s\) Dialog Box](#)

Rules Wizard

To access: **Tools > Rule Editor**, then click the New Rule  icon

Use this dialog box to define I/O placement rules for a database.

Table 16-11. Rules Wizard





Field	Description
Set rule name and description	
Name:	Specifies a name for the rule
Description:	Allows description for the rule to be entered
Failure answer:	Allows a failure answer to be entered. This text will be returned if the attempted assignment breaks this rule.
Next	Moves the wizard to the next page
Rule Scope	
Available Primitives	Lists the device primitives available for the rule.
Rule Primitives	Lists the device primitives selected for this rule.
	Moves the selected primitive from the Available Primitives: list to the Rule Primitives: list, adding it to the rule definition. Opens the Edit Primitive Value(s) Dialog Box to allow conditions to be set.
	Moves the selected primitive from the Rule Primitives: list back to the Available Primitives list, removing it from the rule definition.
Back	Moves the wizard to the previous page.
Next	Moves the wizard to the next page.
Set allowance for rule	
Allow assignments	Specifies that the rule conditions will <i>allow</i> assignments when met.
Forbid assignments	Specifies that the rule conditions will <i>forbid</i> assignments when met.
Pins Conditions /Signal Conditions	
Available Primitives	Lists the available primitives for the rule.
Rule Primitives	Lists the primitives selected for this rule.
	Moves the selected primitive from the Available Primitives: list to the Rule Primitives: list, adding it to the rule definition. Opens the Edit Primitive Value(s) Dialog Box to allow conditions to be set.

Table 16-11. Rules Wizard

Field	Description
	Moves the selected primitive from the Rule Primitives: list back to the Available Primitives list, removing it from the rule definition.
Add Term	Allows another term to be added to the rule.
Back	Moves the wizard to the previous page.
Next	Moves the wizard to the next page.
Finish	Completes the rule definition and exits the dialog.

Related Topics

[User-defined Rules](#)

[Rule Editor Dialog Box](#)

[Edit Primitive Value\(s\) Dialog Box](#)

Unravel Nets Dialog Box

To access: **Edit > Unravel Nets**

Use this dialog box to specify settings for unraveling nets.

Table 16-12. Unravel Nets Dialog Box Contents

Field	Description
Optimize	Adjusts the unravelling algorithm such that it focuses on: <ul style="list-style-type: none">• Shortest possible net length• Least number of crossovers
Unravel scalars and buses concurrently	All signals selected for unravel are used as one package of nets and pins.
Use unused pins	The unravel process will use unoccupied pins in addition to swapping those already assigned.
Unravel on each layer separately	Nets are unravelled on each layer separately, and nets that cross between layers are ignored.

Related Topics

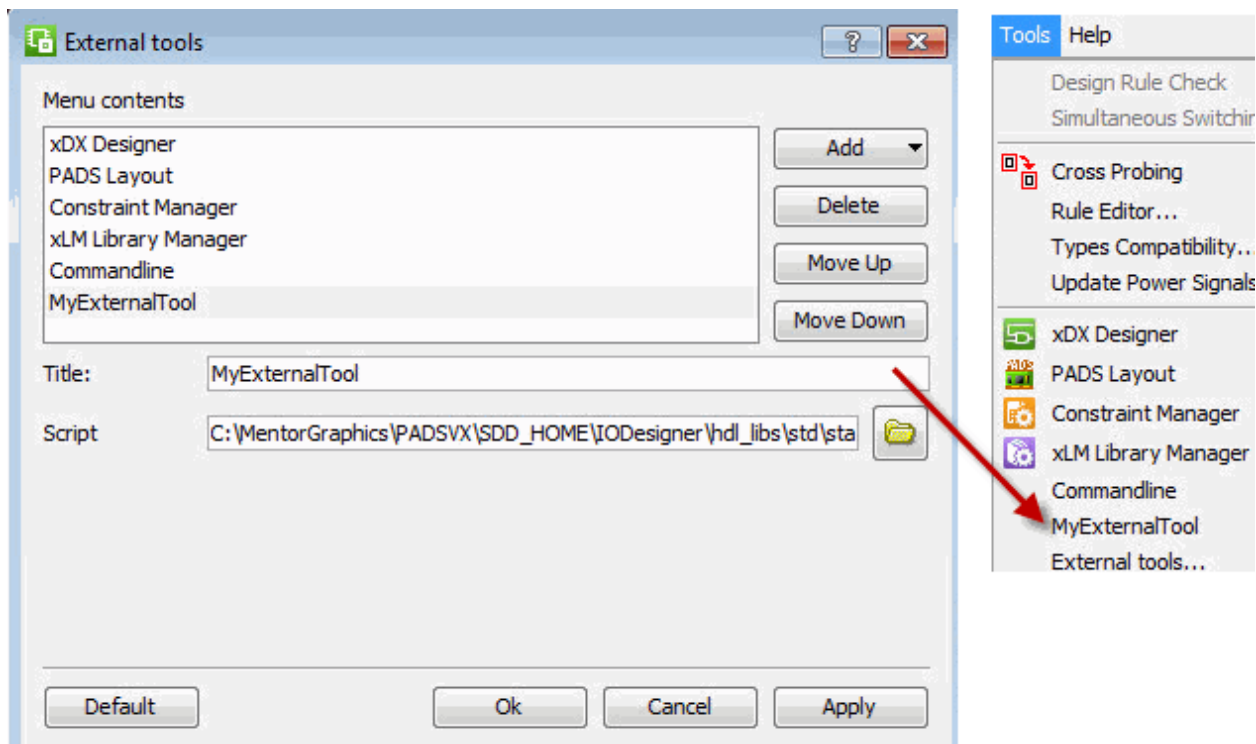
[Unravel Nets](#)

External Tools Dialog

To access: **Tools > External Tools**.

Use this dialog box to define external programs (such as the layout tool), tcl scripts, and tcl commands that can be invoked directly from the Tools menu.

Figure 16-2. External Tools Dialog Box: Script Tool




The contents of the dialog box depends on the selected item. For example, selecting the external tool PADS Designer includes a **Command** and an **Arguments** field. The commands available in the External Tools dialog are described in [Table 16-13](#).

Table 16-13. External Tools Dialog Box Contents

Field	Description
Add	The drop-down lists lets you select the type of tool to add. The available options are External tool, Tcl script, and Tcl command. Depending on the type of tool you select, the dialog displays additional data-entry fields.
Delete	Deletes the selected tool. You are asked to confirm the operation.
Move Up / Move Down	Moves the selected item up or down the contents list. The order is reflected in the Tools menu.

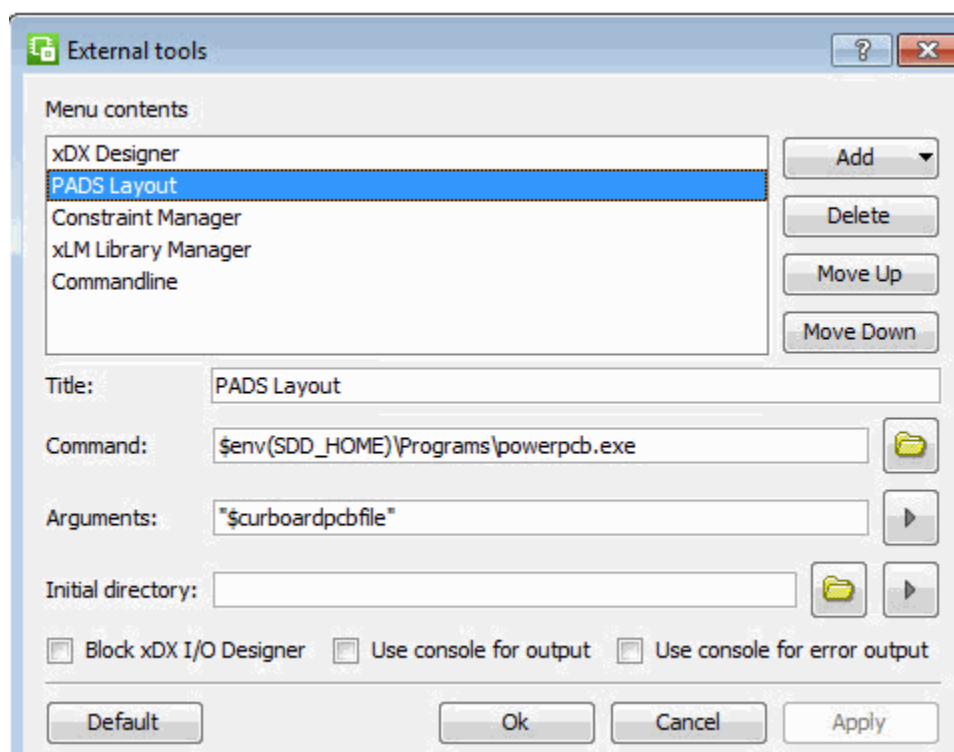
Table 16-13. External Tools Dialog Box Contents

Field	Description
Default	Loads the default tools from the configuration file and adds them to the list.
Title	The name script or tool. This name can be edited.
Script	Lets you specify a tcl script. Click the Browse icon to search for the correct location. Executing tcl scripts affects the state of open PADS I/O Designer documents and is therefore useful for automating frequently-used sequences of commands.
Command	<p>This is where you specify the path to the executable. Click the Browse icon to search for the correct location. If required, this line can also contain <i>tcl</i> variables and expressions. Hold the mouse cursor over the field to display the full contents of the field (type of <i>tooltip</i>). This also applies to the Arguments and Initial directory fields.</p> <p>When adding a tcl command in the Command field, you can provide the name of the tcl command. See “PADS I/O Designer TCL Commands” on page 191 for a list of commands.</p>
Arguments	Lets you specify arguments for the command.
Initial directory	Points to the directory or folder from where the tool is invoked.
	Displays a list of available tcl arrays and variables. Click a tcl variable to insert it into arguments or into the initial directory.

The checkboxes at the bottom of the dialog box, do the following (when checked):

- **Block PADS I/O Designer:** External programs are invoked in a modal way such that other actions and user events in PADS I/O Designer are suspended until the external tool finishes its execution.
- **Use console for output:** All standard output from the executed program is printed to the PADS I/O Designer console window.
- **Use console for error output:** All error output from the executed program is printed to the PADS I/O Designer console window.

Figure 16-3. External Tools Dialog Box: PADS Layout



Appendix A

Preferred Devices List

PADS I/O Designer allows you to limit the range of devices available for selection by using the preferred devices list. This enables only the selected devices to be displayed in the Database Properties dialog.

The preferred devices list is stored in the file named *componentlist.ini*. The algorithm for locating this file is the same as the one for locating the configuration file with the exception that the COMPONENTLIST_INI_DIR variable is used instead of MGC_IO_DESIGNER_HOME. See “[Configuration File \(IODESIGNER.xml\)](#)” on page 16. The preferred devices list is an ASCII file, which can be edited with any text editor.

The preferred devices list consists of sections of the following syntax:

```
[section]
subsection
subsection
...
```

Here is the sample *componentlist.ini* file:

```
[Vendors]
xilinx

[xilinx]
virtex
virtex2
xbr
xc4000e

[virtex2]
2v1000
2v2000
2v8000
```

Start from the section *Vendors*, where all available vendors should be listed. All available families are then listed in the section named after the vendor (Xilinx in the example). We can further limit devices and packages available within a family. The idea is that if a section exists in the preferred devices list, then only listed subsections are available. In the example, four Xilinx families: virtex virtex2 xbr xc4000e are available, and in the family virtex2 only three listed devices are available. In the remaining families all devices are available.

Appendix B

Vendor Support Information

Following is a list of vendor-dependent features in PADS I/O Designer:

Table B-1. Vendor Features

Description	Xilinx	Altera	Lattice	Actel
Pin numbers, pin names, pin function description for all device pins	Yes	Yes	Yes	Yes
Pin I/O bank number for assignable I/O pins and for Vcco, Vref, and other I/O bank dedicated power or ground pins	Yes	Yes	Yes	Yes
IO Banking Rules: Output standards with the same output VCCO requirement can be combined in the same bank	Yes	Yes	Yes	Yes
IO Banking Rules: Input standards with the same input VCCO and input VREF requirements can be combined in the same bank	Yes	Yes	Yes	Yes
IO Banking Rules: Input standards and output standards with the same input VCCO requirements can be combined in the same bank	Yes	Yes	Yes	Yes
IO Banking Rules: When combining bi-directional I/O with other standards, make sure the bi-directional standard can meet rules 1 through 3 above	Yes	Yes	Yes	Yes
IO Banking Rules: No more than one single termination type (input or output) is allowed in the same bank.	Yes	N/A	N/A	N/A
IO Banking Rules: No more than one split termination type (input or output) is allowed in the same bank.	Yes	N/A	N/A	N/A

Table B-1. Vendor Features (cont.)

Description	Xilinx	Altera	Lattice	Actel
IO Banking Rules: The placement of single-ended I/O pins with respect to LVDS I/O pins is restricted as detailed in “Design for Multiple Devices in a Common Package” on page 77.	N/A	Yes	N/A	N/A
Check swap group names. It is not allowed to assign signal to a pin if swap groups are different.	Yes	Yes	Yes	Yes
Check if type of signal and pin are the same. It is possible to force assignment in this case.	Yes	Yes	Yes	Yes
Recognize differential pin pairs; differential pins that belong together	Yes	Yes	Yes	Yes
Recognize differential signals, set a diff type for them and assign to diff pin pairs	Yes	Yes	Yes	Yes
Pin properties: available sink/source current levels for each I/O pin based upon selected I/O standard	Yes	Yes	Yes	Yes
SSO (WASSO) check for some families - based on vendor specific data and rules for SSO (WASSO)	Yes	No	No	No
Multiple sets of IO standards per family. If some pins have a smaller set of IO standards available, we are not able to prevent users from choosing a wrong IO standard for that pin. Therefore, we do not cover Low Capacitance (LC) check for pins.	Yes	Yes	Yes	No
Check direction when signal is assigned.	Yes	Yes	Yes	Yes
Special or additional types for Local Clock pins.	Yes	N/A	N/A	N/A
Ability to parse HDL code to recognize MGT channel pins. MGT types must be set for signals manually, before automatic assignment.	No	No	N/A	N/A
Interior cell information. Position of pads internally to a device.	No	Yes	No	No
VREF regions.	No	No	No	No

Table B-1. Vendor Features (cont.)

Description	Xilinx	Altera	Lattice	Actel
Checks that open drain is turned off for all pins with a differential I/O standard	?	No	?	?
Checks to see if the drive strength assignments are within the specifications of the I/O standard	Yes	Yes	Yes	Yes
Checks to see if the pin location supports the assigned drive path	N/A	No	N/A	N/A
Checks if the pin location supports BUSHOLD (dedicated clock pins do not support BUSHOLD)	N/A	Yes	N/A	N/A
Checks if the pin location supports WEAK_PULLUP (dedicated clock pins do not support WEAK_PULLUP)	N/A	Yes	N/A	N/A
Checks if the combined drive strength of consecutive pads does not exceed a certain limit	N/A	No	N/A	N/A
Checks if the pin location along with the I/O standard assigned support PCI_IO clamp diODE	N/A	No	N/A	N/A
Checks if pins connected to PLL are assigned to the dedicated PLL pin locations	N/A	No	N/A	N/A
Checks that no single-ended I/O pin exists in the same bank as a DPA	N/A	No	N/A	N/A
Checks if single-ended output pins are a certain distance away from a differential I/O pin	N/A	Yes	N/A	N/A
Checks if single-ended output pins are a certain distance away from a VREF pad	N/A	Yes	N/A	N/A
Checks if single-ended input pins are a certain distance away from a differential I/O pin.	N/A	Yes	N/A	N/A
Checks that there are no more than a certain number of outputs or bidirectional pins in a VREFGROUP when a VREF is used	N/A	Yes	N/A	N/A
Checks if too many outputs are in a VREFGROUP	N/A	Yes	N/A	N/A

Table B-1. Vendor Features (cont.)

Description	Xilinx	Altera	Lattice	Actel
Set any IO pin as a VREF for > 128 macrocells Coolrunner II devices	No	N/A	N/A	N/A
Support for PROHIBIT constraint	No	N/A	N/A	N/A
Timing closure support (see the next 4 points)	No	Yes	Yes	Yes
a) Supported sources of actual times	*.twr	*.rpt	*.twr	N/A
b) Support constraint files as a source/destination	*.ucf	*.qsf	*.lcf, *.prf	N/A
c) Support for Constraint Manager	No	No	No	N/A
d) Support for synthesis constraint files	No	No	No	N/A

Appendix C

Device-dependant Assignment Rules

The following assignment rules are implemented in PADS I/O Designer beginning with release IOD7.4.

Actel Assignment Rules

- Banking rules: I/O banking restrictions based on VCCIO VREF for I/O standards.
- Check if the type of signal and pin are the same.
- Check to see if the drive strength assignment is within the specifications of the I/O standard.
- Multiple sets of I/O standards per family. Some pins have a smaller set of I/O standards available.

Altera Assignment Rules

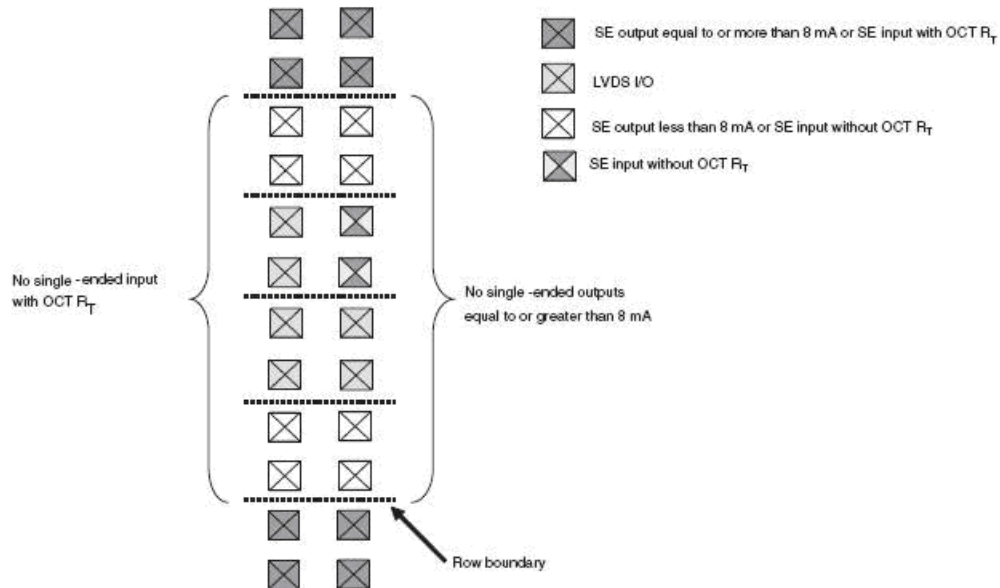
- Banking rules: I/O banking restrictions based on VCCIO VREF for I/O standards.
- I/O bank capacity. Checks the number of pins assigned to a bank against the number of pins allowed in the bank.
- I/O bank VCCIO voltage compatibility. Checks that no more than one VCCIO is required from the pins assigned to the I/O bank.
- I/O bank VREF voltage compatibility. Checks that no more than one VREF is required from the pins assigned to the I/O bank.
- I/O standard and location conflicts. Checks if the pin location supports the assigned I/O standard.
- I/O standard and signal direction conflicts. Checks if the pin location supports the I/O standard assigned and the direction. For example, certain I/O standards on a particular pin location can only support output pins.
- I/O standard and drive strength conflicts. Checks to see if the drive strength assignments are within the specifications of the I/O standard.
- BUSHOLD and location conflicts. Checks if the pin location supports BUSHOLD (dedicated clock pins do not support BUSHOLD).

- WEAK_PULLUP and location conflicts. Checks if the pin location supports WEAK_PULLUP (dedicated clock pins do not support WEAK_PULLUP).
- A PLL IO bank does not support both a single-ended I/O and a differential signal simultaneously. Checks that there are no single-ended I/O pins present in the PLL I/O bank when a differential signal exists.
- Single-ended output is required to be a certain distance away from a differential I/O pin. Checks if single-ended output pins are a certain distance away from a differential I/O pin.
- Single-ended output has to be a certain distance away from a VREF pad. Checks if single-ended output pins are a certain distance away from a VREF pad.
- Single-ended input is required to be a certain distance away from a differential I/O pin. Checks if single-ended input pins are a certain distance away from a differential I/O pin.
- Too many outputs in a VREFGROUP. Checks if too many outputs are in a VREFGROUP.
- Too many inputs in a VREFGROUP. Checks if too many inputs are in a VREFGROUP.
- Check if the type of signal and pin are the same.
- Interior cell information. Position of pads internally in a device - used internally.
- VREF regions - used internally.
- Give a warning if WEAK_PULLUP and BUSHOLD are used at the same time.
- Some double function pins (RX, TX, etc.) can be used only as differential inputs or differential outputs

I/O Pin Placement with Respect to LVDS I/O Pins

For the Altera Stratix III family, the placement of single-ended I/O pins with respect to LVDS I/O pins is restricted. As shown in [Figure C-1](#), row I/O single-ended outputs with driving strength equal to or greater than 8 mA must be placed at least one row away from the LVDS I/O. The same restriction applies to single-ended inputs with OCT RT. You can place single-ended outputs with driving strength less than 8 mA in the rows adjacent to the LVDS I/O. The restriction does not apply when you use the LVDS input buffer for differential HSTL/SSTL input. Single-ended inputs without OCT RT have no placement restriction. When DPA is enabled, the constraint on single-ended I/O is the same as that on regular LVDS I/O.

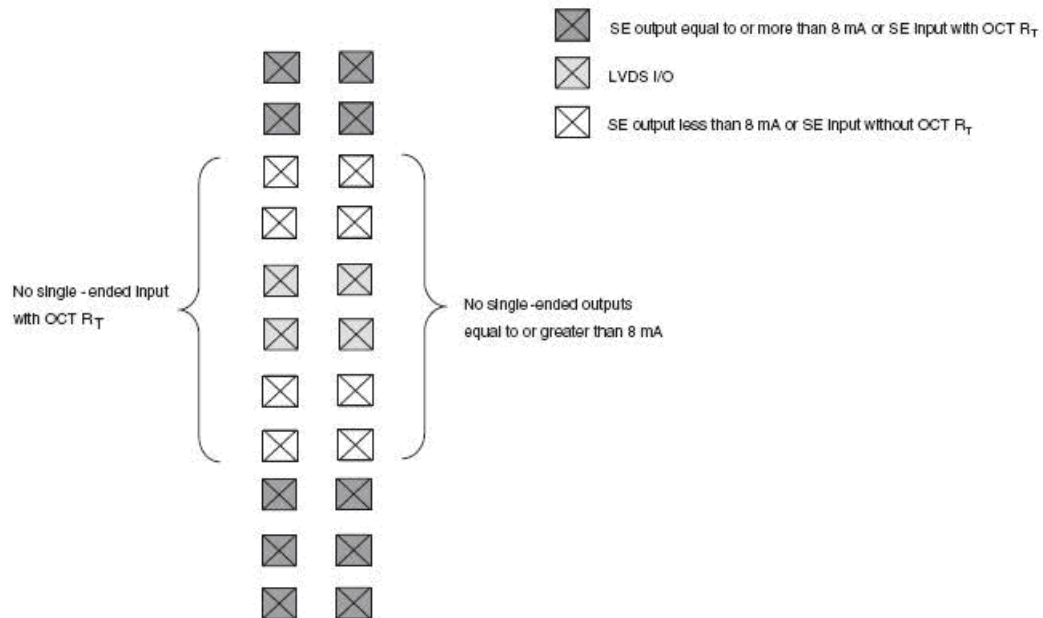
Figure C-1. Single-Ended Row I/O Pin Placement for LVDS I/O Pins



The restriction on placing single-ended column I/O is similar to that on row I/O. Single-ended outputs with drive strength equal to or greater than 8 mA must be placed at least four I/Os away from the LVDS I/O. The same rule applies to single-ended input with OCT R_T . The restriction does not apply when the LVDS input buffer is used for differential HSTL/SSTL inputs. Single-ended outputs with a driving strength less than 8 mA and single-ended inputs without OCT R_T have no restriction. [Figure C-2](#) shows the single-ended I/O placement rules for column I/O.

Figure C-2. Single-Ended Column I/O Pin Placement for LVDS I/O Pins

Single-Ended Column I/O Pin Placement with Respect to LVDS I/O Pins



Xilinx Assignment Rules

- Banking rules: I/O banking restrictions based on VCCIO VREF for I/O standards.
- Check if type of signal and pin are the same.
- Check to see if the drive strength assignment is within the specifications of the I/O standard.
- Multiple sets of I/O standards per family. Some pins have a smaller set of I/O standards available.

Example C-1. Xilinx Assignment Rule

1. The following packages do not support DC1 in Banks 1 and 2: SF363, FF668, FF676, FF672, and FF1152.
 2. The following devices do not support DC1 in Banks 1 and 2: XC4VLX15, XC4VLX25, XC4VSX25, XC4VSX35, XC4VFX12, XC4VFX20, XC4VFX40, AND XC4VFX60
- When the clock-capable I/Os are driven by single-ended clocks, then the clock must be connected to the positive (P) side of the differential “clock capable” pin pair. The negative (N) side can be used as a general purpose I/O or left unconnected (Virtex4 Virtex5).
 - Virtex-4 has I/Os which cannot support differential output IO standards (identified in package files - *_LC_*).
 - Some pins of devices can be used only as inputs (e.g., pin name:”IP_x”) - Spartan3A.
 - An IOB cannot be configured to be both an input and an output if BIDIR ALLOWED is FALSE. The BIDIR_ALLOWED attribute is already available for IO standards and should be used in this case to block assignments of bidirections signals, if needed.

Note



The Xilinx Zynq family uses two types of pins: normal I/O pins that you can configure in PADS I/O Designer, and processor pins connected to an MIO interface that are dedicated and are not configurable in PADS I/O Designer. You can use the processor pins, but if processor pin assignments are exported to the constraint file (.ucf) and then imported to Xilinx ISE, an error occurs. To avoid the error, you should delete the processor pin entries in the constraint file manually, before importing the constraint file to Xilinx ISE.

Lattice Assignment Rules

- Banking rules: I/O banking restrictions.
- Check if type of signal and pin are the same.

- Check to see if the drive strength assignment is within the specifications of the I/O standard.
- Multiple sets of I/O standards per family. Some pins have a smaller set of I/O standards available.

Appendix D

Generic IO Standards

Table D-1. Generic IO Standards

I/O Standards	Actel	Altera	Lattice	Xilinx
TTL	TTL	TTL	NA	TTL
AGP	NA	AGP 2X	AGP2X33	AGP
CTT	NA	CTT	NA	CTT
GTL	GTL33	GTL	NA	GTL
GTL25	GTL25	NA	NA	NA
GTL_DCI	NA	NA	NA	GTL_DCI
GTLP	GTLP33	GTLP+	NA	GTLP
GTLP25	GTLP25	NA	NA	NA
GTLP_DCI	NA	NA	NA	GTLP_DCI
GTL12	NA	NA	GTL12	NA
GTLPLUS15	NA	NA	GTLPLUS15	NA
HSLVDCI_15	NA	NA	NA	HSLVDCI_15
HSLVDCI_18	NA	NA	NA	HSLVDCI_18
HSLVDCI_25	NA	NA	NA	HSLVDCI_25
HSLVDCI_33	NA	NA	NA	HSLVDCI_33
HSTL_I	HSTLI	HSTL CLASS I	HSTL15_I	HSTL_I
HSTL_I_DCI	NA	NA	NA	HSTL_I_DCI
HSTL_I_DCI_18	NA	NA	NA	HSTL_I_DCI_18
HSTL_II	HSTLII	HSTL CLASS II	HSTL15_II	HSTL_II
HSTL_II_DCI	NA	NA	NA	HSTL_II_DCI
HSTL_II_T_DC I	NA	NA	NA	HSTL_II_T_DC I

Table D-1. Generic IO Standards (cont.)

I/O Standards	Actel	Altera	Lattice	Xilinx
HSTL_II_DCI_18	NA	NA	NA	HSTL_II_DCI_18
HSTL_II_T_DCI_18	NA	NA	NA	HSTL_II_T_DCI_18
HSTL_III	NA	HSTL CLASS III	HSTL15_III	HSTL_III
HSTL_III_DCI	NA	NA	NA	HSTL_III_DCI
HSTL_III_DCI_18	NA	NA	NA	HSTL_III_DCI_18
HSTL_IV	NA	HSTL CLASS IV	HSTL15_IV	HSTL_IV
HSTL_IV_18	NA	NA	HSTL18_IV	HSTL_IV_18
HSTL_IV_DCI	NA	NA	NA	HSTL_IV_DCI
HSTL_IV_DCI_18	NA	NA	NA	HSTL_IV_DCI_18
HSTL_I_18	NA	1.8-V HSTL CLASS I	HSTL18_I	HSTL_I_18
HSTL_II_18	NA	1.8-V HSTL CLASS II	HSTL18_II	HSTL_II_18
HSTL_III_18	NA	NA	HSTL18_III	HSTL_III_18
CMOS	CMOS	NA	NA	NA
LVC MOS2	NA	NA	NA	LVC MOS2
LVC MOS12	LVC MOS12	1.2 V	LVC MOS12	LVC MOS12
LVC MOS15	LVC MOS15	1.5 V	LVC MOS15	LVC MOS15
LVC MOS18	LVC MOS18	1.8 V	LVC MOS18	LVC MOS18
LVC MOS25	LVC MOS25	2.5 V	LVC MOS25	LVC MOS25
LVC MOS_30	NA	3.0-V LVC MOS	NA	NA
LVC MOS33	LVC MOS33	LVC MOS	LVC MOS33	LVC MOS33
LVC MOS50	LVC MOS25_50	NA	NA	NA
LVTTL33D	NA	NA	LVTTL33D	NA
LVC MOS33D	NA	NA	LVC MOS33D	NA
LVC MOS25D	NA	NA	LVC MOS25D	NA

Table D-1. Generic IO Standards (cont.)

I/O Standards	Actel	Altera	Lattice	Xilinx
LVC MOS18D	NA	NA	LVC MOS18D	NA
LVC MOS15D	NA	NA	LVC MOS15D	NA
LVC MOS12D	NA	NA	LVC MOS12D	NA
LVDCI_15	NA	NA	NA	LVDCI_15
LVDCI_18	NA	NA	NA	LVDCI_18
LVDCI_25	NA	NA	NA	LVDCI_25
LVDCI_33	NA	NA	NA	LVDCI_33
LVDCI_DV2_1 5	NA	NA	NA	LVDCI_DV2_1 5
LVDCI_DV2_1 8	NA	NA	NA	LVDCI_DV2_1 8
LVDCI_DV2_2 5	NA	NA	NA	LVDCI_DV2_2 5
LVDCI_DV2_3 3	NA	NA	NA	LVDCI_DV2_3 3
LVDS	LVDS	LVDS	LVDS	LVDS
LVDS_25	NA	NA	LVDS25	LVDS_25
LVDS25E	NA	NA	LVDS25E	NA
LVDS_1R	NA	LVDS_1R	NA	NA
LVDS_3R	NA	LVDS_3R	NA	NA
MLVDS	NA	NA	MLVDS	NA
MLVDS25	NA	NA	MLVDS25	NA
MINI_LVDS	NA	mini-LVDS	MINILVDS	MINI_LVDS_2 5
mini-LVDS_E_1R	NA	mini-LVDS_E_1R	NA	NA
mini-LVDS_E_3R	NA	mini-LVDS_E_3R	NA	NA
mini-LVDS_1R	NA	mini-LVDS_1R	NA	NA
mini-LVDS_3R	NA	mini-LVDS_3R	NA	NA
LVDS_25_DCI	NA	NA	NA	LVDS_25_DCI
LVDS_25_DT	NA	NA	NA	LVDS_25_DT

Table D-1. Generic IO Standards (cont.)

I/O Standards	Actel	Altera	Lattice	Xilinx
LVDS_33	NA	NA	NA	LVDS_33
LVDS_33_DCI	NA	NA	NA	LVDS_33_DCI
LVDS_E_1R	NA	LVDS_E_1R	NA	NA
LVDS_E_3R	NA	LVDS_E_3R	NA	NA
LVPECL	LVPECL	NA	NA	LVPECL
LVPECL_33	NA	NA	LVPECL33	LVPECL_33
LVPECL_25	NA	DIFFERENTIAL LVPECL	NA	LVPECL_25
BLVDS_25	NA	BLVDS	BLVDS25	BLVDS_25
LDT_25	NA	NA	NA	LDT_25
LDT_25_DT	NA	NA	NA	LDT_25_DT
LVDSEXT_25	NA	NA	NA	LVDSEXT_25
LVDSEXT_25_DCI	NA	NA	NA	LVDSEXT_25_DCI
LVDSEXT_25_DT	NA	NA	NA	LVDSEXT_25_DT
LVDSEXT_33	NA	NA	NA	LVDSEXT_33
LVDSEXT_33_DCI	NA	NA	NA	LVDSEXT_33_DCI
ULVDS_25	NA	NA	NA	ULVDS_25
ULVDS_25_DT	NA	NA	NA	ULVDS_25_DT
LVTTL	LVTTL	LVTTL	LVTTL33	LVTTL
LVTTL	NA	3.3-V LVTTL	NA	NA
LVTTL_30	NA	3.0-V LVTTL	NA	NA
PCI30	NA	3.0-V PCI	NA	NA
PCI33_3	NA	NA	PCI33	PCI33_3
PCI33_5	NA	NA	NA	PCI33_5
PCI	PCI	3.3-V PCI	NA	PCI66_3
PCIX30	NA	3.0-V PCI-X	NA	NA
PCIX	PCIX	3.3-V PCI-X	PCIX33	PCIX
PCIX66_3	NA	NA	NA	PCIX66_3

Table D-1. Generic IO Standards (cont.)

I/O Standards	Actel	Altera	Lattice	Xilinx
PCIX15	NA	NA	PCIX15	NA
SSTL15_I	NA	SSTL-15 CLASS I	SSTL15	NA
SSTL15_II	NA	SSTL-15 CLASS II	NA	NA
SSTL18_I	NA	SSTL-18 CLASS I	SSTL18_I	SSTL18_I
SSTL18_II	NA	SSTL-18 CLASS II	SSTL18_II	SSTL18_II
SSTL18_II_DCI	NA	NA	NA	SSTL18_II_DCI
SSTL18_II_T_D CI	NA	NA	NA	SSTL18_II_T_D CI
SSTL18_I_DCI	NA	NA	NA	SSTL18_I_DCI
SSTL2_I	SSTL2I	SSTL-2 CLASS I	SSTL25_I	SSTL2_I
SSTL2_I_DCI	NA	NA	NA	SSTL2_I_DCI
SSTL2_II	SSTL2II	SSTL-2 CLASS II	SSTL25_II	SSTL2_II
SSTL2_II_DCI	NA	NA	NA	SSTL2_II_DCI
SSTL2_II_T_D CI	NA	NA	NA	SSTL2_II_T_D CI
SSTL3_I	SSTL3I	SSTL-3 CLASS I	SSTL33_I	SSTL3_I
SSTL3_I_DCI	NA	NA	NA	SSTL3_I_DCI
SSTL3_II	SSTL3II	SSTL-3 CLASS II	SSTL33_II	SSTL3_II
SSTL3_II_DCI	NA	NA	NA	SSTL3_II_DCI
RSDS	NA	RSDS	RSDS	RSDS_25
DIFF_HSTL_I	NA	DIFFERENTIAL HSTL	HSTL15D_I	DIFF_HSTL_I
DIFF_HSTL_II	NA	DIFFERENTIAL HSTL CLASS II	HSTL15D_II	DIFF_HSTL_II
DIFF_HSTL_III	NA	NA	HSTL15D_III	DIFF_HSTL_III

Table D-1. Generic IO Standards (cont.)

I/O Standards	Actel	Altera	Lattice	Xilinx
DIFF_HSTL_I_18	NA	DIFFERENTIAL 1.8-V HSTL CLASS I	HSTL18D_I	DIFF_HSTL_I_18
DIFF_HSTL_II_18	NA	DIFFERENTIAL 1.8-V HSTL CLASS II	HSTL18D_II	DIFF_HSTL_II_18
DIFF_HSTL_III_18	NA	NA	HSTL18D_III	DIFF_HSTL_III_18
DIFF_HSTL_I_DCI	NA	NA	NA	DIFF_HSTL_I_DCI
DIFF_HSTL_II_DCI	NA	NA	NA	DIFF_HSTL_II_DCI
DIFF_HSTL_I_DCI_18	NA	NA	NA	DIFF_HSTL_I_DCI_18
DIFF_HSTL_II_DCI_18	NA	NA	NA	DIFF_HSTL_II_DCI_18
DIFF_SSTL15_I	NA	DIFFERENTIAL 1.5-V SSTL CLASS I	NA	NA
DIFF_SSTL15_II	NA	DIFFERENTIAL 1.5-V SSTL CLASS II	NA	NA
DIFF_SSTL18_I	NA	DIFFERENTIAL 1.8-V SSTL CLASS I	SSTL18D_I	DIFF_SSTL18_I
DIFF_SSTL18_II	NA	DIFFERENTIAL 1.8-V SSTL CLASS II	SSTL18D_II	DIFF_SSTL18_II
DIFF_SSTL18_I_DCI	NA	NA	NA	DIFF_SSTL18_I_DCI
DIFF_SSTL18_II_DCI	NA	NA	NA	DIFF_SSTL18_II_DCI
DIFF_SSTL2_I	NA	DIFFERENTIAL SSTL-2 CLASS I	SSTL25D_I	DIFF_SSTL2_I
DIFF_SSTL2_II	NA	DIFFERENTIAL SSTL-2 CLASS II	SSTL25D_II	DIFF_SSTL2_II

Table D-1. Generic IO Standards (cont.)

I/O Standards	Actel	Altera	Lattice	Xilinx
DIFF_SSTL2_I	NA	DIFFERENTIAL 2.5-V SSTL CLASS I	NA	NA
DIFF_SSTL2_II	NA	DIFFERENTIAL 2.5-V SSTL CLASS II	NA	NA
DIFF_SSTL2_I_DCI	NA	NA	NA	DIFF_SSTL2_I_DCI
DIFF_SSTL2_II_DCI	NA	NA	NA	DIFF_SSTL2_II_DCI
DIFF_SSTL3_I	NA	NA	SSTL33D_I	DIFF_SSTL3_I
DIFF_SSTL3_II	NA	NA	SSTL33D_II	DIFF_SSTL3_II
HSTL_I_12	NA	1.2-V HSTL	NA	HSTL_I_12
HSTL_II_12	NA	1.2-V HSTL CLASS II	NA	NA
CML	NA	CML	NA	NA
COMPACT_PCI	NA	COMPACT PCI	NA	NA
AGP_1X	NA	AGP 1X	AGP1X33	NA
STI33V	NA	3.3-V SCHMITT TRIGGER INPUT	NA	NA
STI25V	NA	2.5-V SCHMITT TRIGGER INPUT	NA	NA
PCML33	NA	3.3-V PCML	NA	NA
PCML25	NA	2.5-V PCML	NA	NA
PCML15	NA	1.5-V PCML	NA	NA
PCML14	NA	1.4-V PCML	NA	NA
PCML12	NA	1.2-V PCML	NA	NA
HYPERTRANSPORT	NA	HYPERTRANSPORT	HYPT	HT_25
SIMPLE_RSDS	NA	SIMPLE RSDS	NA	NA

Table D-1. Generic IO Standards (cont.)

I/O Standards	Actel	Altera	Lattice	Xilinx
DIFF_HSTL_12	NA	DIFFERENTIAL 1.2-V HSTL	NA	NA
DIFF_HSTL_12_II	NA	DIFFERENTIAL 1.2-V HSTL CLASS II	NA	NA
HCSL	NA	HCSL	NA	NA
X25TO18	NA	NA	NA	X25TO18
TMDS_33	NA	NA	NA	TMDS_33
MINI_LVDS_33	NA	NA	NA	MINI_LVDS_33
RSDS_33	NA	NA	NA	RSDS_33
RSDS_E_1R	NA	RSDS_E_1R	NA	NA
RSDS_E_3R	NA	RSDS_E_3R	NA	NA
RSDS_1R	NA	RSDS_1R	NA	NA
RSDS_3R	NA	RSDS_3R	NA	NA
PPDS_33	NA	NA	NA	PPDS_33
PPDS_25	NA	PPDS	NA	PPDS_25
PPDS_E_3R	NA	PPDS_E_3R	NA	NA
TRLVDS	NA	NA	TRLVDS	NA
SSTL15D	NA	NA	SSTL15D	NA
RSDSE	NA	NA	RSDSE	NA
PPLVDS	NA	NA	PPLVDS	NA
BUS_LVDS	NA	BUS LVDS	NA	NA

— A —

[addandedittext](#), 191
[addarc](#), 192
[addbezier](#), 192
[addellipse](#), 192
[Adding Graphics](#), 105
[addline](#), 193
[addportwithlabel](#), 193
[addportwithlabelandedit](#), 193
[addrect](#), 193
[addtext](#), 193
[Arcs](#), 106
[Assign Commands](#), 84
[Assign Mode](#), 84
[Assign Pins](#), 84
[Assignment rules](#)
 [Actel](#), 251
 [Altera](#), 251
 [Lattice](#), 254
 [Xilinx](#), 254
[Attribute](#)
 [IOD_GENERATED](#), 113
 [IOD_RESERVEDSHEET](#), 113

— B —

[Bank Display](#), 30
[bottomalign](#), 194
[Bus Pin](#)
 [Definition](#), 85
[Bus Signal](#)
 [Definition](#), 73

— C —

[cell name](#)
 [PCB](#), 229
[ces_set_db](#), 194
[Change pin name font size](#), 176
[changeattribvisibility](#), 194
[changebuspinrange](#), 194
[changebusrange](#), 194

[changeexttype](#), 194
[changelabelvisibility](#), 194
[changeshape](#), 195
[check power signals](#), 103
[Circles](#), 106
[Clock Assignments](#), 88
[Clock Display](#), 31
[componentlist.ini](#), 245
[Connectivity List window](#), 125
[Console Window overview](#), 35
[Console Window Shortcuts](#), 36
[Constraints Bus](#)
 [dialog](#), 149
[constraints_file](#), 217
[constraints_file_format](#), 217
[copy](#), 195
[creatediffpair](#), 195
[createsymbol](#), 195

— D —

[Database](#)
 [Wizard](#), 65
[Database Repository](#), 221
[database_file](#), 217
[ddp_project_path](#), 217
[delattrib](#), 196
[delete](#), 196
[Delete Elements](#), 108
[deletesymbol](#), 196
[delimiter,constraints bus](#), 149
[delinstanceattrib](#), 196
[device](#), 217
[Device Window](#)
 [Bank Display](#), 30
 [Clock Display](#), 31
 [Modes](#), 34
 [opening the](#), 27
 [Pin Display](#), 29
[Device Window Shortcuts](#), 180
[dx_project_path](#), 217

DxDesigner
 Setting Up Environment, [138](#)

— E —

Edit Symbol Elements, [107](#)
 editbackground, [196](#)
 Ellipses
 See Circles, [106](#)
 Environment
 DxDesigner, [138](#)
 Environment Variables
 IOD_PATH_FILE_VD_DX, [16](#)
 setting, [15](#)
 exportbackground, [197](#)
 exportschematic, [197](#)
 exportsymbol, [197](#)

— F —

family, [217](#)
 Filtering, [171](#)
 Filters
 regular expression, [173](#)
 wildcard, [172](#)
 Font size
 change, [176](#)
 fpga_xchange_file, [217](#)

— G —

generate_constraints_file, [197](#)
 generate_fpga_xchange_file, [197](#)
 generate_synthesis_constraints_file, [198](#)
 Get Existing Database
 Existing Databases
 Get, [222](#)
 Graphics
 Adding, [105](#)
 Arcs, [106](#)
 Circles, [106](#)
 Lines, [106](#)
 Rectangles, [106](#)
 Text, [106](#)
 Graphics Properties, [48](#)

— H —

hdl_file, [218](#)
 hdlsignature, [219](#)
 help, [198](#)

hide, [199](#)
 horzmirror, [199](#)
 hsrename, [199](#)

— I —

ICE, [146](#)
 importdesign, [199](#)
 importlmcsymbol, [199](#)
 importsymbol, [199](#)
 interconnect table editor, [146](#)
 invertbuspins, [200](#)
 invertbussignals, [200](#)
 IOD_GENERATED, [113](#)
 IOD_PATH_FILE_VD_DX, [16](#)
 IOD_RESERVEDSHEET, [113](#)

— K —

keyboard shortcuts
 editing, [181](#)

— L —

Layout database
 optimizing assignments, [122](#)
 properties dialog, [231](#)
 leftalign, [200](#)
 Lines, [106](#)

— M —

mergepins, [200](#)
 mergesignals, [200](#)
 MGT Pins, [90](#)

— N —

Netlist source, layout database, [231](#)

— P —

package, [218](#)
 Pads Layout, [147](#)
 paste, [200](#)
 pasteat, [200](#)
 PCB cell name, [229](#)
 PCB layout
 Device Window, [28](#), [29](#)
 layout database, [122](#)
 loading a, [54](#)
 Pin Display, [29](#)
 Pin name font size, [176](#)

Pin Properties, [43, 48](#)
 Pin Report Formats, [136](#)
 Pin Types, [86](#)
 pin_report_file, [218](#)
 pin_report_file_format, [218](#)
 pinbank, [219](#)
 pinfunction, [220](#)
 pinname, [220](#)
 pinnumber, [220](#)
 Pins
 printing, [24](#)
 pinsignal, [220](#)
 pinswapgroup, [220](#)
 pintypes, [220](#)
 Place & Route Constraint Formats, [135](#)
 plock, [220](#)
 Port Properties, [45](#)
 Preferred Devices, [245](#)
 Printing
 signals and pins
 Signals
 printing, [24](#)
 Properties
 Graphic, [48](#)
 Pin, [43, 48](#)
 Port, [45](#)
 Symbol, [48](#)
 Text, [48](#)
 Properties Window
 Overview, [40](#)

— R —

redo, [201](#)
 Redo command, [169](#)
 removefill, [201](#)
 renameattrib, [201](#)
 renamebuspin, [201](#)
 renamesymbol, [202](#)
 reshapearc, [202](#)
 reshapebezier, [202](#)
 reshapeellipse, [202](#)
 reshapepline, [202](#)
 reshapeoutline, [202](#)
 reshapeirect, [202](#)
 rightalign, [202](#)
 rotate, [203](#)

Rows
 Filtering, [171](#)
 Selecting, [170](#)
 rundrc, [203](#)

 — S —
 sadd, [203](#)
 savedb, [203](#)
 scaddnew, [203](#)
 scalebackground, [203](#)
 sccheckin, [203](#)
 sccheckout, [204](#)
 sccreateproject, [204](#)
 scdiff, [204](#)
 scenarios, applying a layout, [129](#)
 scgetlatestversion, [204](#)
 scgetproject, [204](#)
 scgetrev, [204](#)
 Schematic Update, [118](#)
 schistory, [205](#)
 sclabel, [205](#)
 screfreshstatus, [205](#)
 scundocheckout, [205](#)
 sdifffnames, [219](#)
 sdir, [219](#)
 Selecting rows, [170](#)
 set_cdb_flow, [205](#)
 set_constraints_file, [206](#)
 set_design_architect_project, [206](#)
 set_dx_designer_project, [206](#)
 set_expedition_layout, [206](#)
 set_fpga_xchange, [206](#)
 set_pin_report_file, [206](#)
 set_synthesis_constraints_file, [207](#)
 set_timing_report_file, [207](#)
 setalign, [207](#)
 setattrib, [207](#)
 setbackground, [207](#)
 setbackgroundproperty, [207](#)
 setdefaultlinestyle, [207](#)
 setdefaultoutline, [207](#)
 setdesignator, [208](#)
 setfillcolor, [208](#)
 setfont, [208](#)
 setfontsize, [208](#)
 setfunctionalblock, [208](#)

- setgeom, [208](#)
- setinstanceattrib, [208](#)
- setinvert, [208](#)
- setlabletype, [209](#)
- setlinestyle, [209](#)
- setoutlinecolor, [209](#)
- setpart, [209](#)
- setpartno, [209](#)
- setportlabel, [209](#)
- setportlength, [209](#)
- setreadonly, [210](#)
- setsigprop, [210](#)
- setsimilardevices command, [210](#)
- setstyle command, [210](#)
- settext, [210](#)
- settextcolor, [210](#)
- Settings dialog, viewing, [61](#)
- setvendor, [211](#)
- Shortcut Editor
 - viewing assigned keys, [183](#)
- show, [211](#)
- Signal Types, [86](#)
- Signals List, [20](#)
- Slice Format, constraints bus, [149](#)
- slock, [219](#)
- Sorting Rows, [170](#)
- Source Control Systems, [221](#)
- speed, [218](#)
- splitdiffpair, [211](#)
- splitpins, [211](#)
- splitsignals, [211](#)
- sremove, [211](#)
- srename, [212](#)
- SSO Check, [137](#)
- sso_bank_threshold, [218](#)
- sso_package_allowance, [218](#)
- sswapgroup, [219](#)
- stypes, [219](#)
- Swap, [85](#)
- swappins, [212](#)
- Symbol Editor settings, [27](#)
- Symbol Generator, [96](#)
- Symbol Properties, [48](#)
- Symbol Window
 - Overview, [176](#)

- Symbol Window Shortcuts, [180](#)
- Symbol Windows
 - Settings, [27](#)
- Symbols
 - Backgrounds, [108](#)
 - Edit Elements, [107](#)
 - Functional, [95](#)
 - Import from DxDesigner, [139](#)
 - PCB, [95](#)
- symbolwizard, [212](#)
- Synchronization Wizard, [49](#)
- synthesis_constraints_file, [218](#)
- synthesis_constraints_file_format, [218](#)
- synthesis_tool, [218](#)

— T —

- taddclockcon, [213](#)
- taddtcocon, [213](#)
- taddtpdcon, [213](#)
- taddtsucon, [213](#)
- TCL Command
 - addandedittext, [191](#)
 - addarc, [192](#)
 - addbezier, [192](#)
 - addellipse, [192](#)
 - addline, [193](#)
 - addportwithlabel, [193](#)
 - addportwithlabelandedit, [193](#)
 - address, [193](#)
 - addtext, [193](#)
 - bottomalign, [194](#)
 - ces_set_db, [194](#)
 - changeattribvisibility, [194](#)
 - changebuspinrange, [194](#)
 - changebusrange, [194](#)
 - changeexttype, [194](#)
 - changelabelvisibility, [194](#)
 - changeshape, [195](#)
 - copy, [195](#)
 - creatediffpair, [195](#)
 - createsymbol, [195](#)
 - delattrib, [196](#)
 - delete, [196](#)
 - deletesymbol, [196](#)
 - delinstanceattrib, [196](#)
 - editbackground, [196](#)

exportbackground, [197](#)
 exportschematic, [197](#)
 exportsymbol, [197](#)
 generate_constraints_file, [197](#)
 generate_fpga_xchange_file, [197](#)
 generate_synthesis_constraints_file, [198](#)
 generatesymbols, [198](#)
 help, [198](#)
 hide, [199](#)
 horzmirror, [199](#)
 hsrename, [199](#)
 importdesign, [199](#)
 importlmcsymbol, [199](#)
 importsymbol, [199](#)
 invertbuspins, [200](#)
 invertbussignals, [200](#)
 leftalign, [200](#)
 mergepins, [200](#)
 mergesignals, [200](#)
 paste, [200](#)
 pasteat, [200](#)
 redo, [201](#)
 removefill, [201](#)
 renameattrib, [201](#)
 renamebuspin, [201](#)
 renamesymbol, [202](#)
 reshapearc, [202](#)
 reshapebezier, [202](#)
 reshapeellipse, [202](#)
 reshapeline, [202](#)
 reshapeoutline, [202](#)
 reshaperect, [202](#)
 rightalign, [202](#)
 rotate, [203](#)
 rundrc, [203](#)
 sadd, [203](#)
 savedb, [203](#)
 scaddnew, [203](#)
 scalebackground, [203](#)
 sccheckin, [203](#)
 sccheckout, [204](#)
 sccreateproject, [204](#)
 scdiff, [204](#)
 scgetlatestversion, [204](#)
 scgetproject, [204](#)
 scgetrev, [204](#)
 schistory, [205](#)
 sclabel, [205](#)
 screfreshstatus, [205](#)
 scundocheckout, [205](#)
 set_cdb_flow, [205](#)
 set_constraints_file, [206](#)
 set_design_architect_project, [206](#)
 set_dx_designer_project, [206](#)
 set_expedition_layout, [206](#)
 set_fpga_xchange_file, [206](#)
 set_pin_report_file, [206](#)
 set_synthesis_constraints_file, [207](#)
 set_timing_report_file, [207](#)
 setalign, [207](#)
 setattrib, [207](#)
 setbackground, [207](#)
 setbackgroundproperty, [207](#)
 setdefaultlinestyle, [207](#)
 setdefaultoutline, [207](#)
 setdesignator, [208](#)
 setfillcolor, [208](#)
 setfont, [208](#)
 setfontsize, [208](#)
 setfunctionalblock, [208](#)
 setgeom, [208](#)
 setinstanceattrib, [208](#)
 setinvert, [208](#)
 setlabeltype, [209](#)
 setlinestyle, [209](#)
 setoutlinecolor, [209](#)
 setpart, [209](#)
 setpartno, [209](#)
 setportlabel, [209](#)
 setportlength, [209](#)
 setreadonly, [210](#)
 setsigprop, [210](#)
 setsimilardevices, [210](#)
 setstyle, [210](#)
 settext, [210](#)
 settextcolor, [210](#)
 setvendor, [211](#)
 show, [211](#)
 splitdiffpair, [211](#)
 splitpins, [211](#)

- splitsignals, 211
- sremove, 211
- srename, 212
- swappins, 212
- symbolwizard, 212
- taddclockcon, 213
- taddtcocon, 213
- taddtpdcon, 213
- taddtsucon, 213
- tdelclockcon, 213
- tdeltcocon, 213
- tdeltpdcon, 214
- tdeltsucon, 214
- topalign, 214
- unassign, 214
- unassignall, 214
- unassignpins, 214
- undo, 214
- unravelnets, 215
- update_from_constraints_file, 215
- update_from_fpga_xchange_file, 216
- update_from_pin_report_file, 216
- update_from_synthesis_constraints_file, 216
- update_symbol_attributes, 216
- updatehdl, 215
- updatepowersignals, 216
- updatesymbols, 216
- vertmirror, 217
- TCL Variable
 - constraints_file, 217
 - constraints_file_format, 217
 - database_file, 217
 - ddp_project_path, 217
 - device, 217
 - dx_project_path, 217
 - family, 217
 - fpga_xchange_file, 217
 - hdl_file, 218
 - hdlsignature, 219
 - package, 218
 - pin_report_file, 218
 - pin_report_file_format, 218
 - pinbank, 219
 - pinfunction, 220
 - pinname, 220
 - pinnumber, 220
 - pinsignal, 220
 - pinswapgroup, 220
 - pintypes, 220
 - plock, 220
 - sdiffnames, 219
 - sdir, 219
 - slock, 219
 - speed, 218
 - sso_bank_threshold, 218
 - sso_package_allowance, 218
 - sswapgroup, 219
 - stypes, 219
 - synthesis_constraints_file, 218
 - synthesis_constraints_file_format, 218
 - synthesis_tool, 218
 - timing_report_file, 218
 - timing_report_file_format, 219
 - vendor, 219
 - verilog_search_path, 219
- tdelclockcon, 213
- tdeltcocon, 213
- tdeltpdcon, 214
- tdeltsucon, 214
- Text, 106
- Text Properties, 48
- TextEditor Window
 - opening the, 27
- timing_report_file, 218
- timing_report_file_format, 219
- Timings Window
 - Overview, 36
- Toolbars
 - Main, 184
 - Overview, 184
 - Source Control, 189
 - View, 185
- tools
 - external, 242
- topalign, 214
- U —
 - unassign, 214
 - unassignall, 214
 - unassignpins, 214

undo, [214](#)
 Undo command, [169](#)
 unravelnets, [215](#)
 update_from_constraints_file, [215](#)
 update_from_fpga_xchange_file, [216](#)
 update_from_pin_report_file, [216](#)
 update_from_synthesis_constraints_file, [216](#)
 update_symbol_attributes, [216](#)
 updatehdl, [215](#)
 updatesymbols, [216](#)

— V —

vendor, [219](#)
 verilog_search_path, [219](#)
 vertmirror, [217](#)
 VREFs, [89](#)

— W —

Wildcard filters, [172](#)

Third-Party Information

For third-party information, refer to [*Third-Party Software*](#)

End-User License Agreement with EDA Software Supplemental Terms

Use of software (including any updates) and/or hardware is subject to the End-User License Agreement together with the Mentor Graphics EDA Software Supplement Terms. You can view and print a copy of this agreement at:

mentor.com/eula