
NCLaunch User Guide

Product Version 3.1 June 2000

© 2000 Cadence Design Systems, Inc. All rights reserved.
Printed in the United States of America.

Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 1-800-862-4522.

All other trademarks are the property of their respective holders.

Restricted Print Permission: This publication is protected by copyright and any unauthorized use of this publication may violate copyright, trademark, and other laws. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. This statement grants you permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used solely for personal, informational, and noncommercial purposes;
2. The publication may not be modified in any way;
3. Any copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement; and
4. Cadence reserves the right to revoke this authorization at any time, and any such use shall be discontinued immediately upon written notice from Cadence.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. The information contained herein is the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only by Cadence's customer in accordance with, a written agreement between Cadence and its customer. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

Preface	4
About This Manual	5
<u>Manual Organization</u>	5
<u>Typographic and Syntax Conventions</u>	5
Platform Support	6
Related Manuals	6
Related Resources	6
<u>Verilog Resources</u>	6
<u>VHDL Resources</u>	7
1	
Overview of NCLaunch	9
Usage Concepts	9
<u>Environment</u>	9
<u>Selection</u>	10
<u>Customization of File Types and Available Commands</u>	11
<u>Tools and Options</u>	11
<u>Saving and Restoring Data</u>	12
Invocation and Command Syntax	12
<u>NCLaunch Components</u>	13
Importing a Design	14
Exporting a Design	15
2	
Using NCLaunch	16
Menu Bar	18
<u>File Menu</u>	19
<u>Edit Menu</u>	25
<u>Tools Menu</u>	32
<u>Utilities Menu</u>	43

NCLaunch User Guide

<u>Help Menu</u>	47
<u>File Browser</u>	49
<u>Library Browser</u>	50
<u>Console Window</u>	52
<u>Status Bar</u>	53
<u>Tool Bar</u>	53

Preface

NCLaunch is a graphical user interface that helps you to manage large design projects and lets you configure and launch your Cadence simulation tools. NC Launch is integrated into the Cadence Interleaved Native Compiled Architecture (INCA) and is a component of the Affirma™ SimVision analysis environment.

This preface discusses the following topics:

- [About This Manual](#)
- [Related Manuals](#)
- [Related Resources](#)

About This Manual

To use NCLaunch most effectively, you should be familiar with performing digital design and analysis using either the Verilog® Hardware Description Language (HDL), or the Very High Speed Integrated Circuit Hardware Description Language (VHDL), and you should be familiar with the Verilog-XL®, Affirma™ NC VHDL, and/or Affirma™ NC simulators.

Manual Organization

This manual is organized into the following sections:

If You Want to Learn About

The use model and startup procedure of NCLaunch

Using NCLaunch

You Should Read

Chapter 1, “Overview of NCLaunch,”

Chapter 2, “Using NCLaunch.”

Typographic and Syntax Conventions

The following syntax conventions are used in this manual:

Font	Meaning	Example
<i>Italic</i>	Titles of books	See the <i>Affirma NC Verilog Simulator Help</i> for more information.
Literal	Program output or text that you type at the command line	> nclaunch
<i>variable</i>	Variables	> nclaunch -cdslib <i>filename</i>
<u>Hypertext Link</u>	Hypertext links that take you to other relevant sections of this book or to other relevant books	See the <u>File Manager</u> section for more information.

Platform Support

NCLaunch supports the same set of platforms that the NC products support for current release. These platforms include: Solaris 2.5.1, Solaris 2.7, HP 10.20, HP 11.0, NT 4.0, Linux Redhat 6.1 and Windows 98.

Related Manuals

You can use NCLaunch with the Affirma™ NC Verilog®, Affirma™ NC VHDL®, and Affirma™ NC simulators. The following documents give you more information about these products:

- If you want to use the Affirma™ NC Verilog simulator to simulate your design, you should read the *Affirma™ NC Verilog Simulator Help*.
- If you want to use the Affirma™ NC VHDL simulator to simulate your design, you should read the *Affirma™ NC VHDL Simulator Help*.
- If you want to simulate a mixed-language Verilog HDL and VHDL design, you should read the appropriate sections in either the *Affirma™ NC VHDL Simulator Help* or the *Affirma™ NC Verilog Simulator Help*.
- If you want to use the Affirma™ SimVision analysis environment, you should read the *Affirma™ SimVision Analysis Environment User Guide*.
- If you want to use the Signalscan Waves waveform viewing tool, you should read the *Signalscan Waves User Guide*.
- If you want to use the Comparescan waveform comparison tool, you should read the *Comparescan User Guide*.

Related Resources

Hardware design requires the knowledge of one or more of several special languages—such as Verilog and VHDL. The following books give you more information about these languages.

Verilog Resources

The following books detail the specifications of the Verilog HDL (Hardware Description Language) and Tcl (Tool Command Language). These include:

- *IEEE Standard Hardware Description Language Based on the Verilog Hardware Description Language* (IEEE Std 1364), IEEE, New York, October, 1996.

NCLaunch User Guide

Preface

- *OVI Verilog Hardware Description Language Reference Manual*, Version 2.0
- Ousterhout, John K., *Tcl and the Tk Toolkit*, Addison-Wesley, Reading, MA, 1994.

There are also several books available that show you how to use the Verilog HDL. These include:

- Donald E. Thomas and Philip Moorby, *The Verilog Hardware Description Language*, Kluwer Academic Publishing, 1996.
- E. Sternheim, R. Singh and Y. Trivedi, *Digital Design With Verilog HDL*, Chapman & Hall, 1991.
- E. Sternheim, *Digital Design and Synthesis With Verilog HDL*, Automata Publishing Company, 1993.
- J. Bhasker, *A Verilog HDL Primer*, Star Galaxy Press, 1997.
- Douglas J. Smith, *HDL Chip Design: A Practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs Using VHDL or Verilog*, Doone Publications, 1996.
- Samir Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis*, Prentice Hall, 1996.
- James M. Lee, *Verilog Quickstart*, Kluwer Academic Publishing, 1997.

VHDL Resources

The following books detail the specifications of the Very High Speed Integrated Circuit Hardware Description Language (VHDL). These include:

- *IEEE Standard VHDL Language Reference Manual* (IEEE Std 1076 - 1987), IEEE, New York, March 1988.
- *IEEE Standard VHDL Language Reference Manual* (IEEE Std 1076 - 1993), IEEE, New York, June 1994.

There are also several books available that show you how to use VHDL. These include:

- Douglas J. Smith, *HDL Chip Design: A Practical Guide for Designing, Synthesizing & Simulating ASICs & FPGAs Using VHDL or Verilog*, Doone Publications, June 1996.
- Douglas L. Perry, *VHDL 3rd Edition*, McGraw Hill Text, June 1998.

NCLaunch User Guide

Preface

- Peter J. Ashenden, *Designer's Guide to VHDL*, Morgan Kaufman Publishers, December 1995.
- Stanley Mazor, Patricia Langstraat, *A Guide to VHDL 2nd edition*, Kluwer Academic Publishers, August 1993.

Overview of NCLaunch

NCLaunch is a graphical user interface that gives you a unified view of the files and libraries in your design. The tool provides you with an easy and consistent way to configure and launch your Cadence simulation tools.

There are several components that need to be considered before, during, and after simulation:

- the HDL files that make up your design
- the libraries into which the files are compiled
- the NC based tools that are available to help with your simulation process.

NCLaunch helps you to manage your files and libraries, and makes you aware of the different NC tools that are available to you.

Usage Concepts

As your HDL designs grow more complex, you need to manage larger design hierarchies and interact with more extended tool flows. This can make it difficult to design, verify, and deliver products with the maximum number of features possible in a minimal amount of time. As a consequence, increasing your design productivity and efficiently managing and verifying your HDL projects becomes an important design goal. NCLaunch can help you get your designs into simulation faster, so that you can find the maximum number of problems in the least amount of time.

The following sections describe concepts with which you should be familiar before running NCLaunch. These concepts focus on the behavior of the tool.

Environment

The `ncvlog` and `ncvhdl` compilers, which parse and analyze your Verilog and VHDL source files, store compiled objects (Verilog modules, macromodules, and UDPs or VHDL entities,

NCLaunch User Guide

Overview of NCLaunch

architectures, packages, package bodies, and configurations) and other derived data in libraries that are organized within a Library.Cell:View (L.C:V) structure.

Three configuration files help you manage your data and control the operation of the various tools and utilities:

- `cds.lib`

Defines your design libraries and associates logical library names with physical library locations.

- `hdl.var`

Defines variables that affect the behavior of tools and utilities.

- `setup.loc`

Specifies the search order that tools and utilities use when searching for the `cds.lib` and `hdl.var` files.

See the chapter on setting up your environment in the [*Affirma NC VHDL Simulator Help*](#) for detailed information about these configuration files.

Selection

NCLaunch presents two different browsers that display data about your files and designs. Both browsers are simply displaying the information that is stored in your directories in a way that makes the information easier to interact with.

The files that make up your design are presented in the [File Browser](#). The elements that are stored within your library are presented in the [Library Browser](#). You can select and manipulate any objects displayed within these browsers that can be used as input to NC based tools, or that were generated as output from such tools.

Within the File Browser, a Verilog file (usually represented by using the `.v` file extension) and a VHDL file (usually represented by using the `.vhd` file extension) are objects that you can use with the appropriate NC compiler. When you select a file with a known extension, the icons that represent these tools become enabled and you can simply press the button on the tool bar to run the appropriate compiler.

For each NC tool that is enabled on the tool bar, there is at least one selectable object displayed in one of the browsers on which you can use the tool. Thus, you can step through simulation by selecting the appropriate input to the tool that you wish to run. By enabling and disabling the tool bar icons, you will quickly become aware of the required input for each tool.

Customization of File Types and Available Commands

NCLaunch provides you with two ways to customize your default operations and bind them to specific file types.

- You can change the extension(s) that is associated with an existing operation by using the *Edit - Setup File Types* menu option. For example, if you use the extension `.vh` instead of `.vhd` to represent a VHDL file, you can use the *Edit - Setup File Types* menu option to change this mapping.
- You can define a new operation and then associate it with an extension and file type. For example, you can easily set up the system to run `make` if you double-click on a Makefile.

By default, NCLaunch has a predefined set of operations for each NC tool, based on the expected input for that tool. These default operations are itemized in [Figure 2-11](#) on page 29.) For the set of known tools, you can submit a selected file with the correct file extension to the tool along with its predefined set of options simply by pressing the appropriate tool bar button. In this case, the command line that you submit is:

```
<toolname> <predefined options> <selected file or files>
```

If you change a file extension, the behavior of the tool remains the same when you submit the run. If you define a new command that is not one of the predefined set of tools, NCLaunch submits the command exactly as specified. You must specify the exact syntax and also specify whether the new command allows multiple selections, and whether it operates on a file or a directory.

For example, if you want to add a customized operation for Makefiles, you can specify `make -f %f` where `%f` specifies the location of the selected file. Or you can state that you simply want to run `make` by specifying: "make" without the `%f`.

Tools and Options

NCLaunch provides a set of predefined options for each of the available NC tools. These tools include `ncvlog`, `ncvhdl`, `ncelab` and `ncsim`. You can change the default options to each tool by selectin it from the *Tools - Advanced Options...* pulldown on the main menu. When you change the options to a tool, those options remain in effect until you change them again. You can specify only one set of options per tool. These options apply to any invocation of the tool, whether you invoke it from the pulldown menu, the tool bar, or a popup menu.

A small set of options is available only on each tool invocation form. These options are the ones that tend to change most frequently, such as method of invocation (batch, interactive or graphical), log filename, and work library. Once you set these options on a tool invocation form, the options remain in effect until you change them.

NCLaunch User Guide

Overview of NCLaunch

NCLaunch saves all option settings to disk upon exit and reuses them upon the next invocation of the tool.

Saving and Restoring Data

When you customize NCLaunch or your design in any way, you need to save your customizations, so that you can always reuse them. NCLaunch automatically saves customized information when you exit the tool.

NCLaunch saves two different types of data when you exit the tool. NCLaunch stores each data type differently, and also restores data differently, depending upon the method you use to invoke the tool.

The first type of data that NCLaunch save is *user data*. This information includes the file setup data that is described in the previous sections, and runtime environment data such as your choice of editor. NCLaunch stores user data in your home directory; the tool loads your user data each time you invoke NCLaunch.

The second type of data is *design data*. This information changes based on the design that you are currently simulating. This information includes tool options. NCLaunch stores design data in your invocation or design directory. You specify your design directory within the application or on the command line. NCLaunch does not load the same design data for different designs because changes that you make to tool options for one design can prohibit a different design from simulating correctly. This is analogous to command files, which are different for each design.

Invocation and Command Syntax

In most cases, you invoke NCLaunch with the following command in a shell window on a UNIX system:

```
nclaunch <options>
```

However, if you are running the desktop simulator on a Windows NT system, you invoke NCLaunch with the following command:

```
ncdesktop <options>
```

The command-line options to `nclaunch` (and `ncdesktop`) are defined as follows:

- `-cdslib filename`

Specifies the name of the `cds.lib` file that you want to use.

NCLaunch User Guide

Overview of NCLaunch

- `-directory directory_name`

Specifies the name of an alternate directory from which you want to run NCLaunch. NCLaunch uses this directory to search for design files, load your library using the `cds.lib` file, and run the simulation.
- `-help`

Displays this list of recognized options and exits NCLaunch.
- `-input tcl_command_file`

Reads the specified TCL command file and executes each command in the file
- `-keyfile tcl_log_file`

Specifies the name of the file to which NCLaunch writes the TCL commands that you enter during a session (default: `nclaunch.key`). You can use a key file as input to the `-input` option.
- `-new`

Specifies that you want to start a new design. NCLaunch opens a form that lets you choose the new design directory from which you want to start.
- `-nocopyright`

Suppresses the printing of copyright banner.
- `-nokey`

Suppresses the generation of the default keyfile.
- `-work library_name`

Specifies the name of the work library.
- `-version`

Writes out the version number and exits NCLaunch.

NCLaunch Components

The main NCLaunch window is divided into the following components to help you set up your designs and manage your simulations:

- Menu Bar and Tool Bar - Let you start various tools and manipulating design elements.
- File Browser - Displays and manipulates files in the file system.

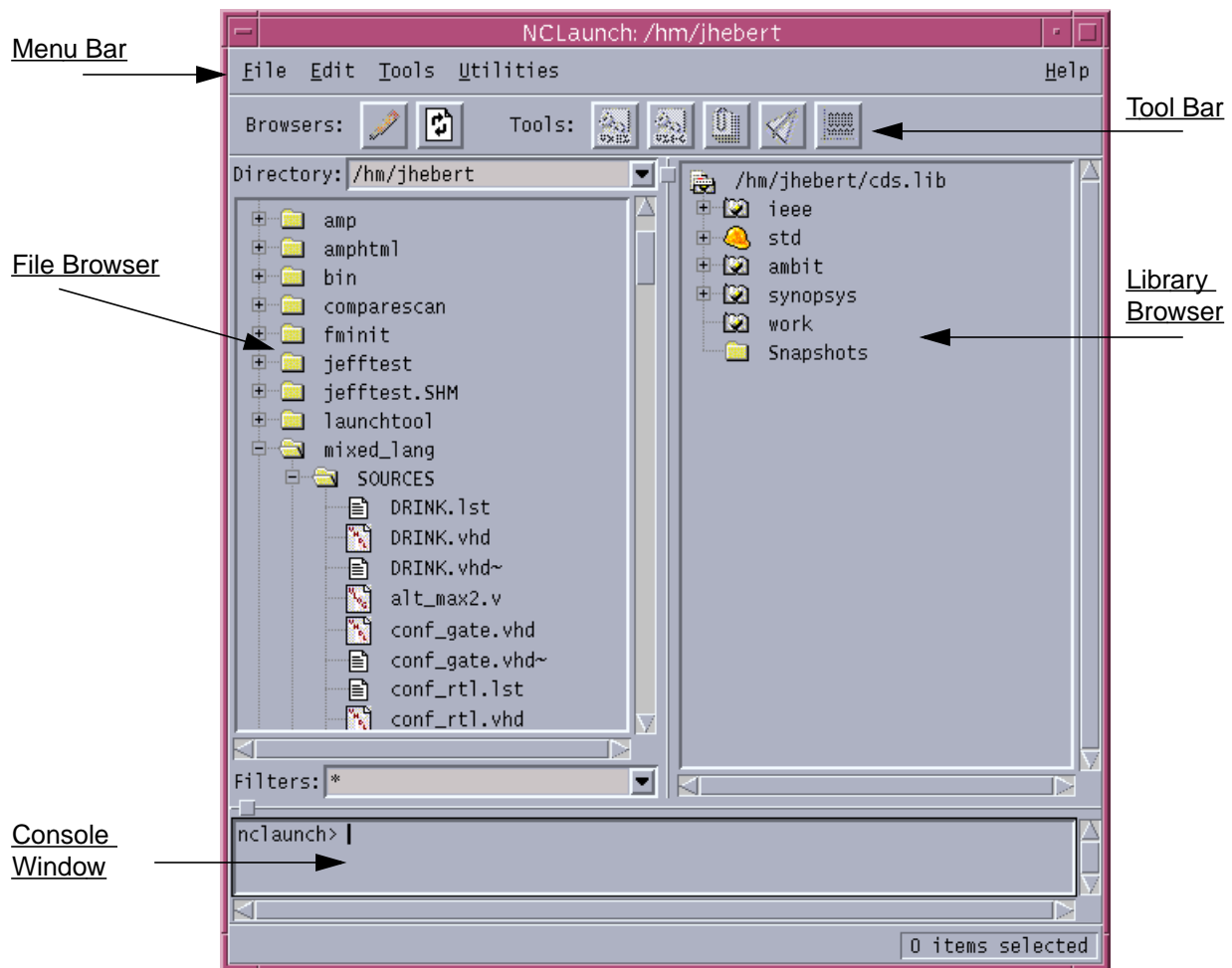
NCLaunch User Guide

Overview of NCLaunch

- Library Browser - Displays and manipulates objects in your library. (Supports NC-packed libraries and 5.X libraries.)
- Console Window - Displays output from tools, and takes user input to generate TCL scripts.

Figure 1-1 on page 14 shows the components of the NCLaunch main window.

Figure 1-1 NCLaunch Components



Importing a Design

There is no need to import a design into NCLaunch. NCLaunch initializes at invocation, based on the contents of the working directory. The File Browser displays the files in the current directory and lets you manipulate them. If you change your working directory, NCLaunch

reinitializes its File Browser to the new directory, reloads the new `cds.lib` file for the Library Browser, and sets your tool options based on the defaults that you have set in that directory.

Exporting a Design

NCLaunch can create a build script based on a snapshot. You can save this script to disk and then build and run a simulation with or without NCLaunch.

Using NCLaunch

The NCLaunch tool consists of a single main window that contains multiple browsers, which are integrated with the suite of NC tools. The integrated tools include the following:

- Compilers: `ncvhdl`, `ncvlog`, `ncsdfc`
- Elaborators: `ncelab`, `ncupdate`
- Simulator: `ncsim`
- Utilities: Signalscan Waves, Comparescan, NCBrowse message browser

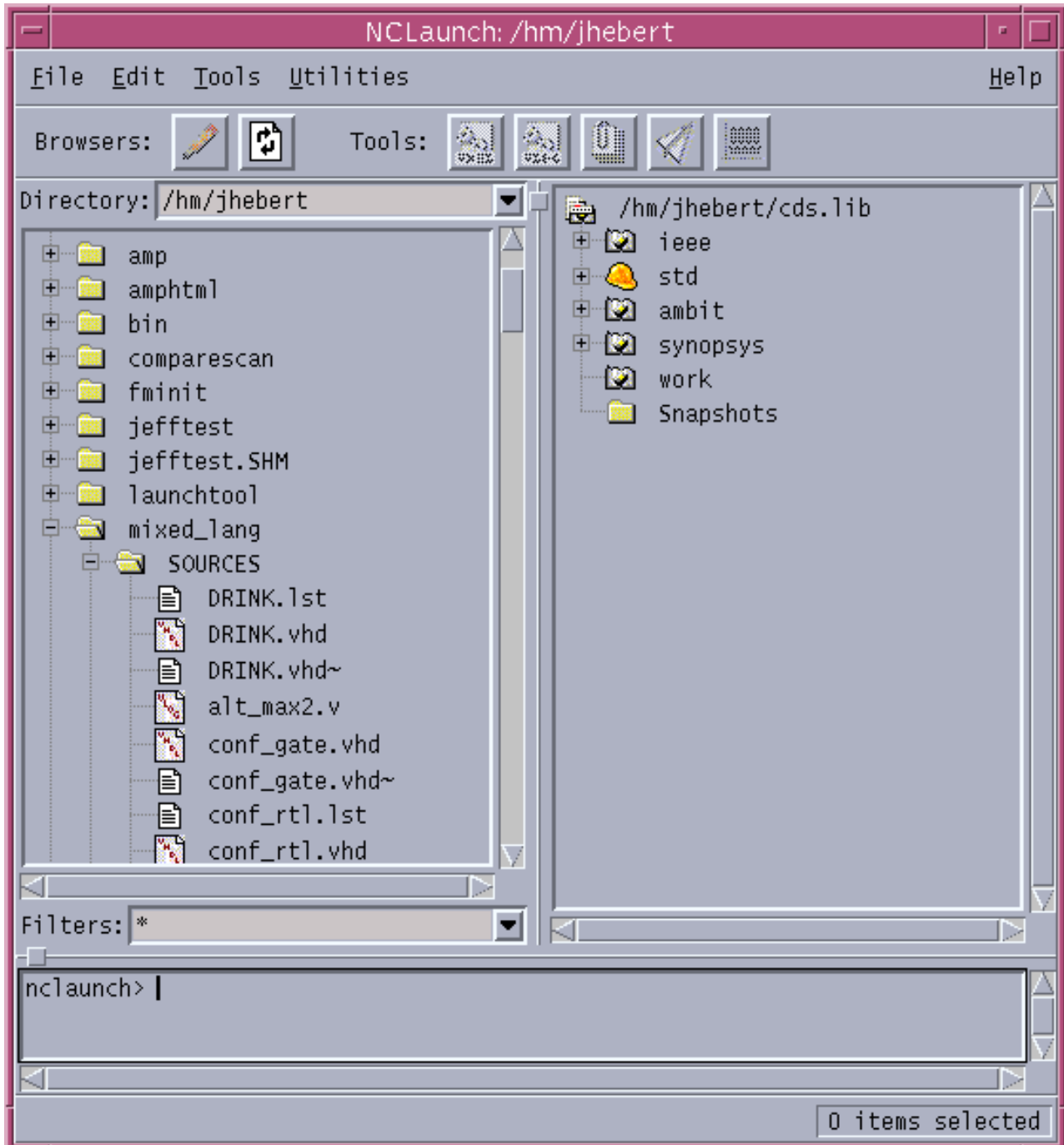
The compilers interact with the files that you have selected from the File Browser, which is located on the left side of the main window. The elaborators and the simulator interact with the objects in the Library Browser, which is located on the right side of the main window, as shown in [Figure 1-1](#) on page 14.

A Tool Bar gives you access to the most commonly used tools specified above, namely `ncvhdl`, `ncvlog`, `ncelab`, `ncsim` and Signalscan Waves. The Tool Bar lets you select an object from either browser and use the appropriate tool on that object. NCLaunch also has an I/O region in a console window at the bottom of the main window, in which you can see the output of all commands. A sample layout of the main window appears in [Figure 2-1](#) on page 17.

NCLaunch User Guide

Using NCLaunch

Figure 2-1 NCLaunch Main Window

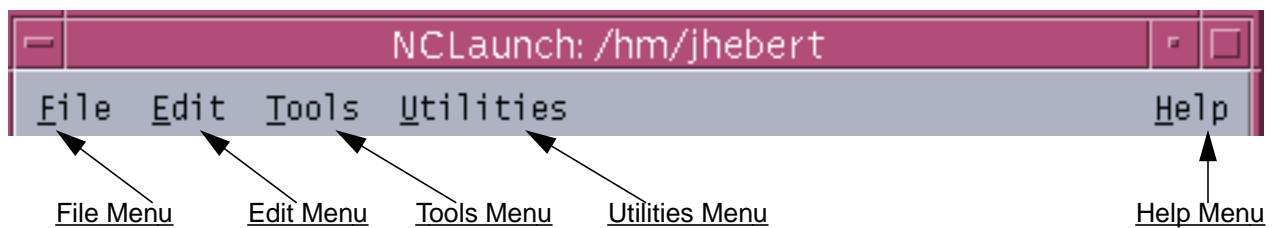


Menu Bar

The following figure represents the menu bar structure at the top of the main window. Some menu items are sensitive to the current selection, as applicable. For example, the *File-Edit* menu item is only enabled if you select a source file (a non-directory file) in the File Browser.

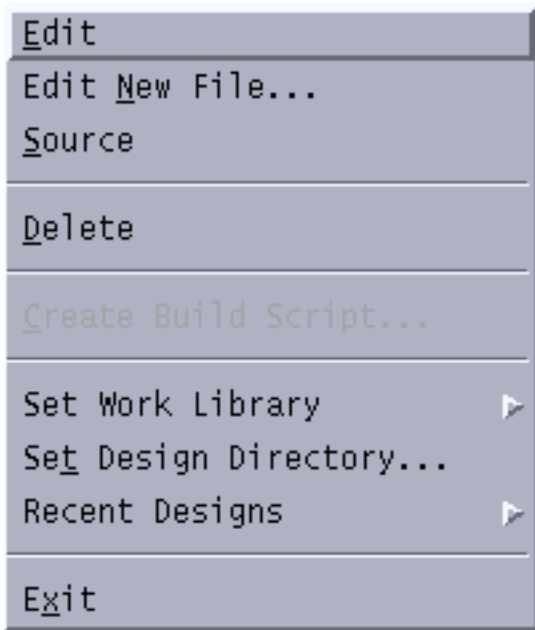
The menu bar contains four pulldown menus, as shown in [Figure 2-2](#) on page 18.

Figure 2-2 Menu Bar



You can also execute some commands using pop-up menus. To display a pop-up menu, position the cursor over an object in the File Browser or Library Browser window and click the right mouse button. A pop-up menu appears, listing some of the most commonly used main menu commands that you can execute on that object.

File Menu



The *File* menu contains commands that let you perform operations on files.

You can:

- Edit a new or existing file.
- Execute the `source` command on any file.
- Delete a file.
- Create a build script.
- Specify your working library.
- Specify your design directory.
- Show your most recent designs.
- Exit.

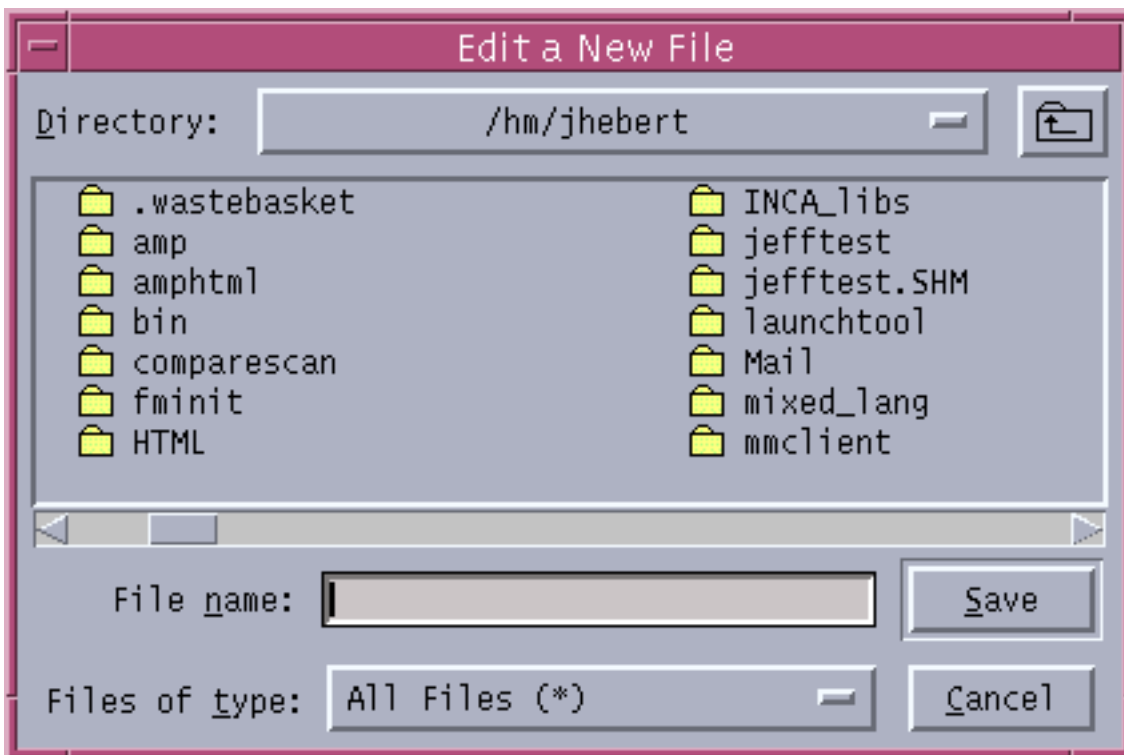
Edit

NCLaunch uses the editor that you have designated in your preferences to edit the selected file.

Edit New File...

This menu option opens the Edit a New File form, shown in [Figure 2-3](#) on page 20. In this form, you double-click on a directory to open it, and then enter a filename and click *Save* to create a new file in the directory that you have opened. NCLaunch uses the editor that you have designated in your preferences.

Figure 2-3 Edit a New File Form



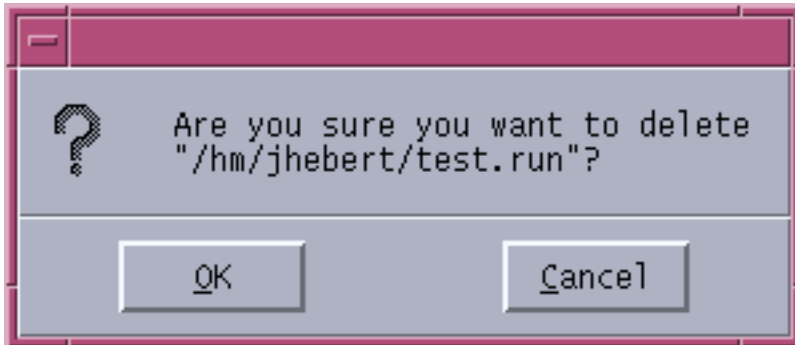
Source

Use this menu option to run the commands in a TCL command file. The Console window shows any output that the commands in your file generate. You can also enter the `source` command directly in the Console window.

Delete

Use this menu option to delete the selected file. A confirmation box asks if you are sure that you want to delete the file, as shown in [Figure 2-4](#) on page 21.

Figure 2-4 Delete File Form

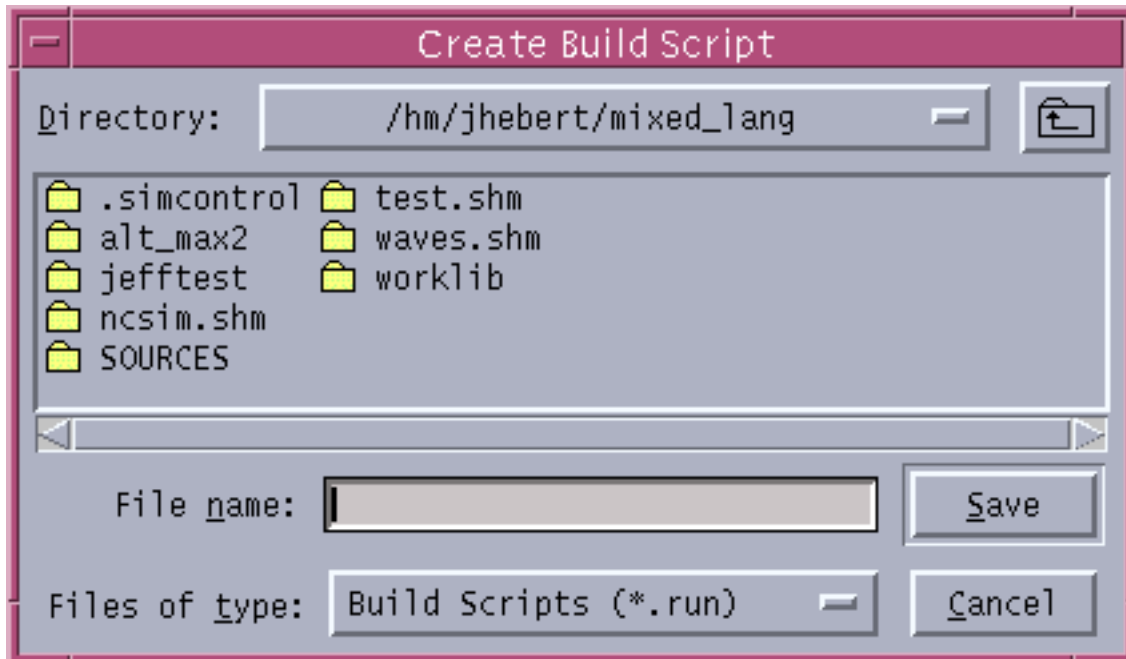


Create Build Script...

This menu option is only available if you select a cell or snapshot. NCLaunch uses the `ncupdate` utility to generate a script that you can use to recreate the selected object. You can then use this script to build the selected object independent of NCLaunch.

NCLaunch opens the Create Build Script form, shown in [Figure 2-5](#) on page 22. Select the directory in which you want the script created, enter a filename for the script, and click *Save* to create a build script. You can use the *Files of type* filter to display other scripts in the current directory, in case you want to overwrite one of them.

Figure 2-5 Create Build Script Form



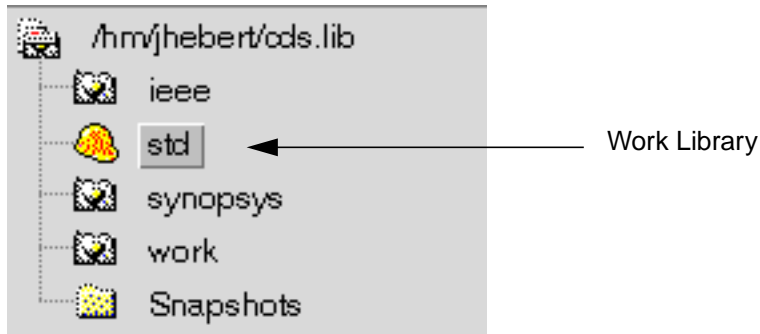
Set Work Library

This submenu displays the list of libraries that are defined in your `cds.lib` file. You can select the library that you want to name as your work library.

You can also change the work library by clicking on a library in the Library Browser portion of the NCLaunch main window, pressing the right mouse button, and selecting *Set as Work Library* in the resulting popup menu.

A hard hat icon in the library browser portion of the NCLaunch main window indicates the current work library, as shown in [Figure 2-6](#) on page 23.

Figure 2-6 Current Work Library

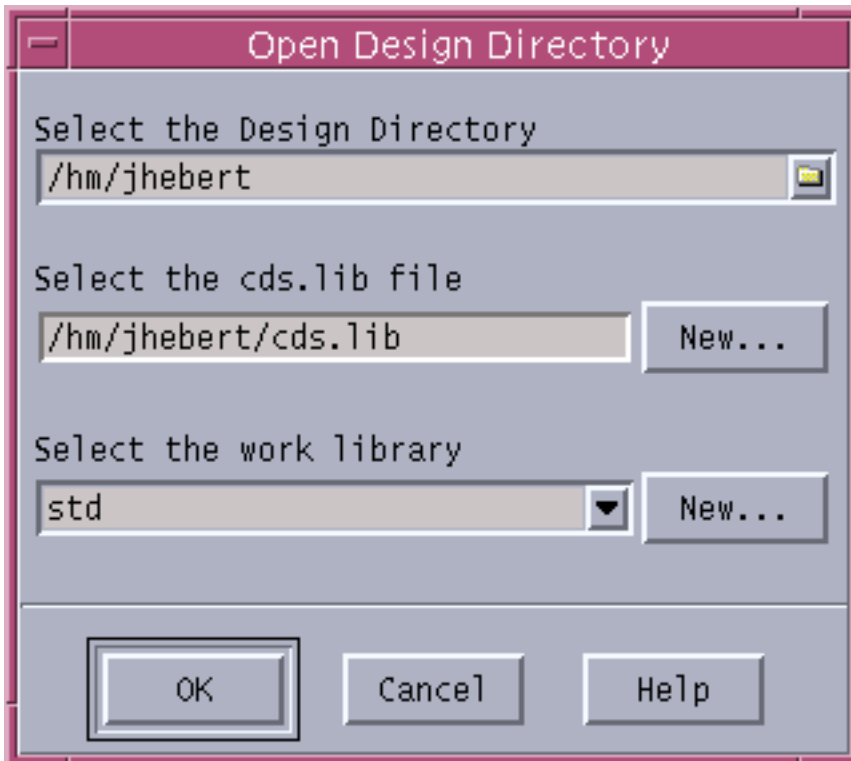


Set Design Directory...

This menu entry displays the Open Design Directory form, shown in [Figure 2-7](#) on page 24, which requests three pieces of information:

- The name of the directory in which you wish to work. This is the directory from which NCLaunch will invoke all tools.
- The name of your `cds.lib` file. By default NCLaunch expects this file to be in the new design directory. If the `cds.lib` file is not in the new directory, you can specify the name of a different file or choose to create a new file. If you request a new file, NCLaunch uses the default name, `cds.lib`, and links it to the default file.
- The name of the library that you want to use as your work library. NCLaunch presents you with a list of the libraries that are contained within the `cds.lib` file. If you create a new `cds.lib` file, you must create a new library and set it as the work library.

Figure 2-7 Open Design Directory Form



Recent Designs

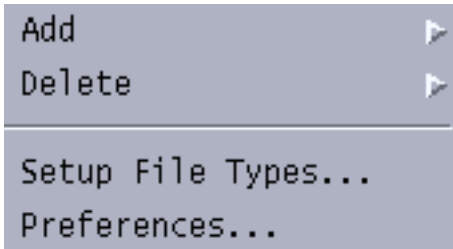
This submenu displays the names of the five most recently opened designs (based on their directory names). By selecting a new design from this submenu you cause, NCLaunch to reinitialize its browsers and change its working directory, based on your selection. NCLaunch then stores the selected design information in your default settings.

Exit

Use this menu option to exit the NCLaunch application. Exiting the application does not terminate any batch jobs that you have already launched. This menu item is always available.

Upon exit, NCLaunch saves your general default settings to your home directory, and saves your design default settings to your current working directory.

Edit Menu



The *Edit* menu contains commands that let you modify your `cds.lib` file.

You can:

- Add a new library, library reference, or `cds.lib` reference.
- Delete the contents of a library, a cell, a view, or a snapshot.
- Associate file types with specific actions.
- Define preferences for your NCLaunch session.

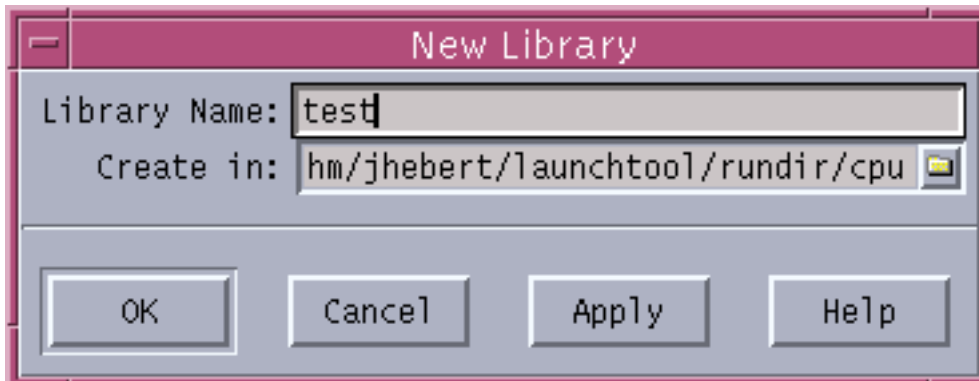
Add

This submenu of the *Edit* menu lets you create the following:

- *New Library...*

This menu item presents a form that lets you specify the name of a new library and map it to a physical location on your disk. NCLaunch writes this entry, the library name, and its location to your `cds.lib` file. The New Library form is shown in [Figure 2-8](#) on page 26.

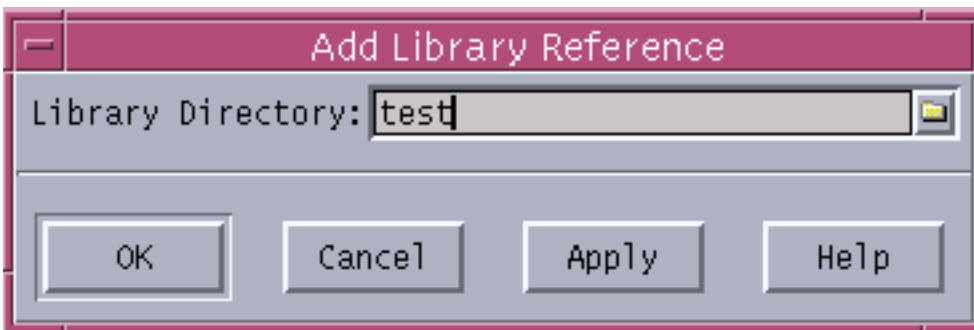
Figure 2-8 Add New Library Form



■ *Library Reference...*

This menu item opens a form that lets you add a new library reference to your `cds.lib` file. The Add Library Reference form is shown in [Figure 2-9](#) on page 26.

Figure 2-9 Add Library Reference Form

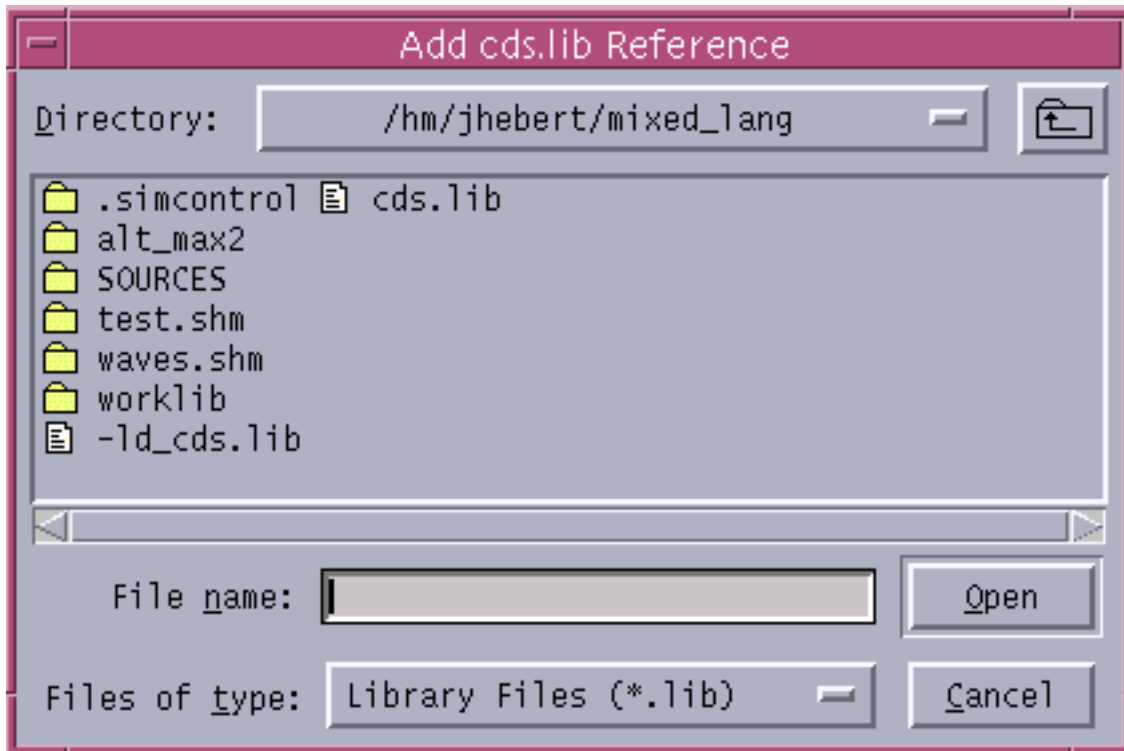


■ *cds.lib Reference...*

This menu item opens a form that lets you include a reference to another `cds.lib` file in your `cds.lib` file.

The Add `cds.lib` Reference form is shown in [Figure 2-10](#) on page 27.

Figure 2-10 Add cds.lib Reference Form



Delete

The *Delete* submenu lets you remove the following:

- *Library Contents*

This submenu is only enabled when you select libraries from the Library Manager. The command runs the `ncrm` utility to delete the library from the disk. This command provides a confirmation form before actually removing the item from the disk. The menu command also removes the selected library from your `cds.lib` file.

- *Cell*

This item is only enabled when you select cells from the Library Manager. This menu command runs the `ncrm` utility to delete the selected item from the library. This command provides a confirmation form before actually removing the item from the library.

- *View*

This item is only available when you select cell views from the Library Manager. This menu command runs the `ncrm` utility to delete the selected item from the library. This

NCLaunch User Guide

Using NCLaunch

command provides a confirmation form before actually removing the item from the library.

■ Snapshot

This item is only enabled when you select a snapshot from the Library Manager. This command runs the `ncrm` utility to delete the snapshot from the library. This command provides a confirmation form before actually removing the item from the library.

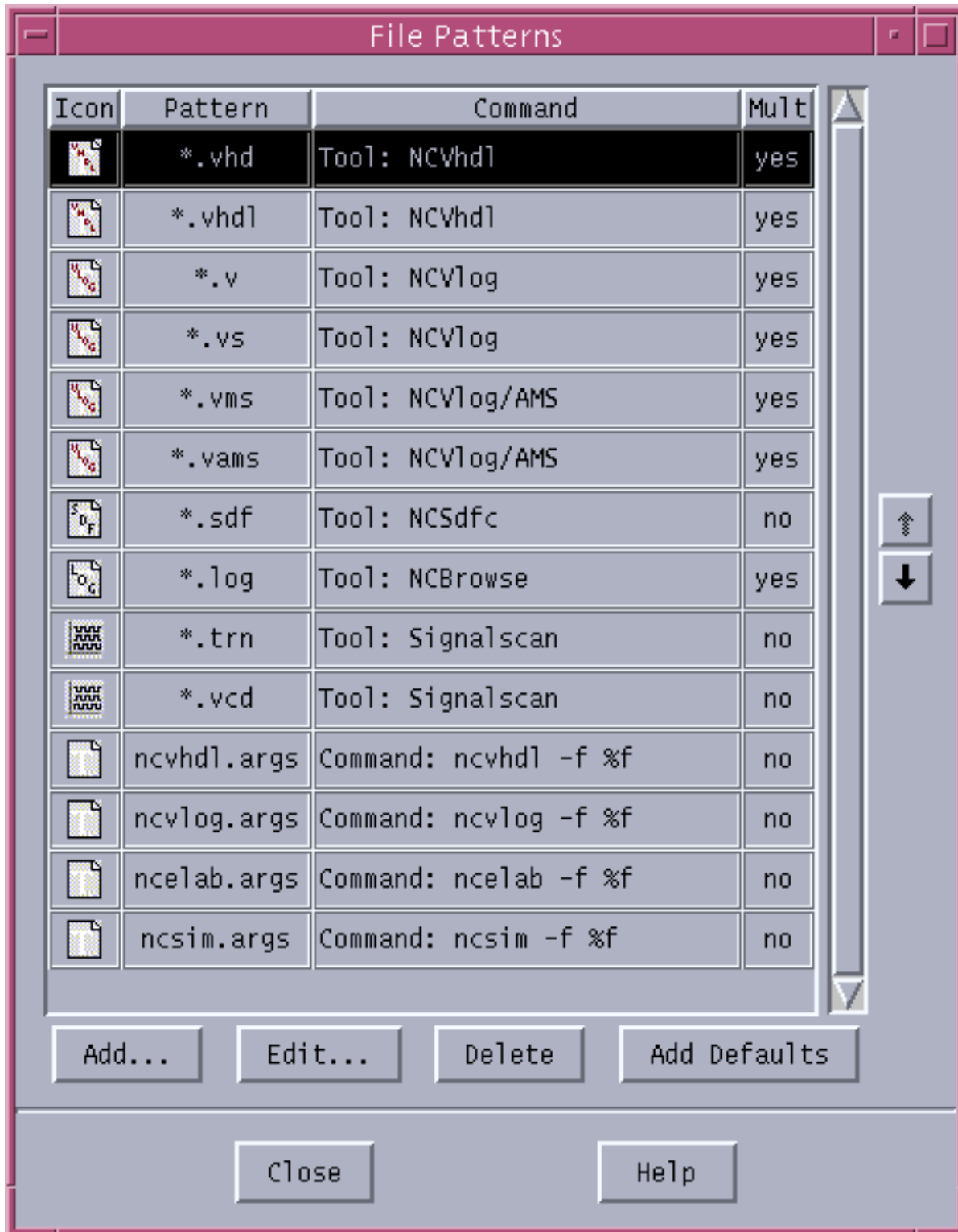
Setup File Types...

This menu item opens a form that lets you associate file types with a tool within the launcher. By default, the system assigns commonly used file extensions to the set of NC-based tools. [Figure 2-11](#) on page 29 shows the default file associations.

NCLaunch User Guide

Using NCLaunch

Figure 2-11 File Patterns



NCLaunch User Guide

Using NCLaunch

The top entries in the table in the File Patterns form define tools that are fully integrated into NCLaunch. These tools respond to selection and they include a set of predefined options, as well as the selected file or files, when you submit these commands.

In the File Browser, you can always double click on a file that has been bound to a tool to run the appropriate command on that file. The File Browser uses a special icon to designate a file as bound to a tool. The popup menu for that file contains an entry for the tool invocation.

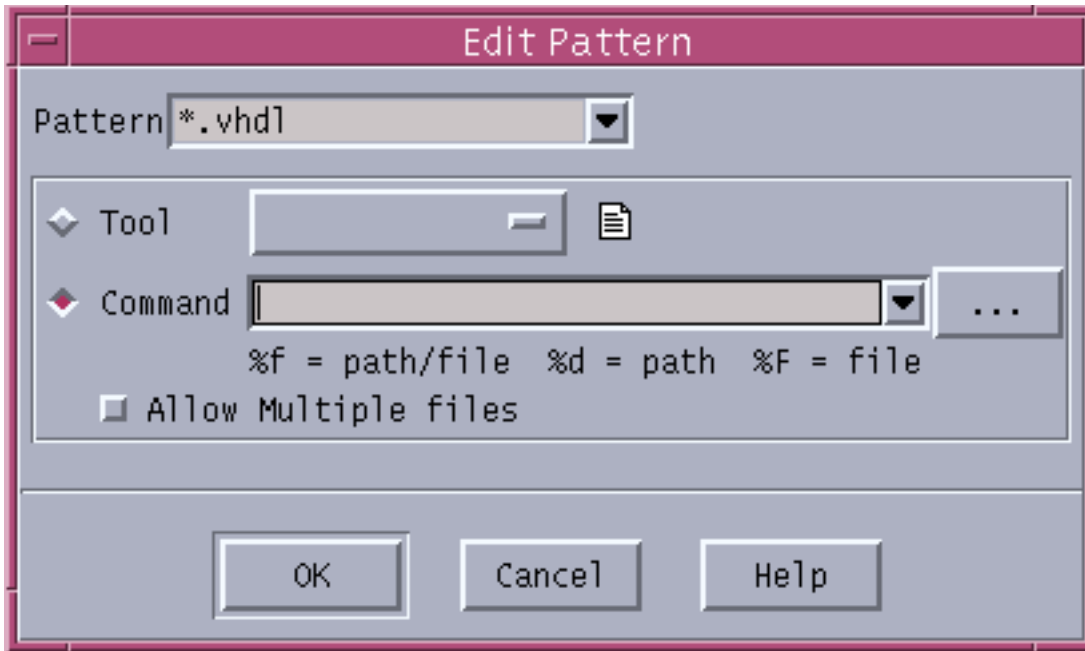
In addition to the predefined file types, you can map your own file extension to a given command string. This lets you customize your environment to run your own commands. Click the *Add* button to add a new pattern, as shown in [Figure 2-12](#) on page 30, or click the *Edit* button to edit an existing pattern, as shown in [Figure 2-13](#) on page 31.

Figure 2-12 Add Pattern Form

The image shows a dialog box titled "Add Pattern". It contains the following elements:

- A "Pattern" label followed by a text input field and a small downward arrow icon.
- A "Tool" label with a diamond icon, followed by a dropdown menu showing "NCVhd1" and a small icon of a document with a red 'X'.
- A "Command" label with a diamond icon, followed by a long text input field.
- A small square checkbox below the command field.
- At the bottom, three buttons: "OK", "Cancel", and "Help".

Figure 2-13 Edit Pattern Form



In both the Add Pattern and Edit Pattern forms, you specify the file type (wildcard with extension or explicit file name) and the command that you want to execute when you select a file of that type. The File Browser displays a special icon for all files that are associated with user-defined commands.

When you specify a user-defined command, you use the following special characters to indicate where the object name appears in the command line:

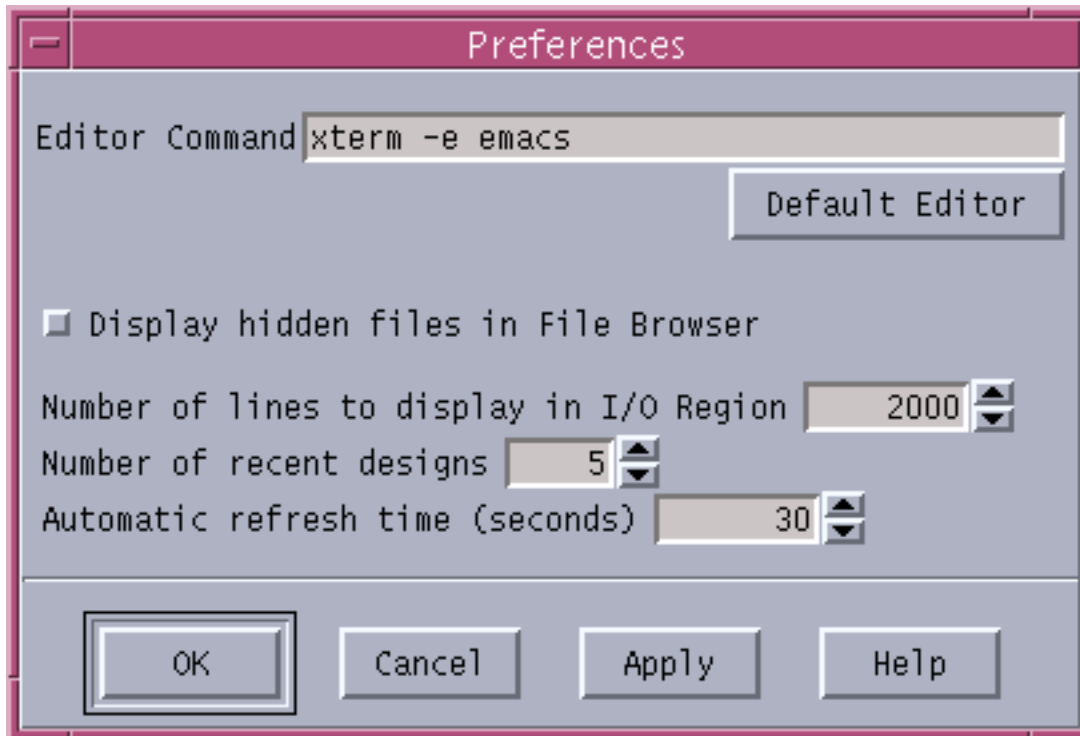
- %f - full pathname of the file
- %F - just the name of the file
- %d - name of the directory in which the file exists

NC Launch stores customized file types with your general options in your home directory (c : if your home is not set on NT), and uses them in all current and future sessions.

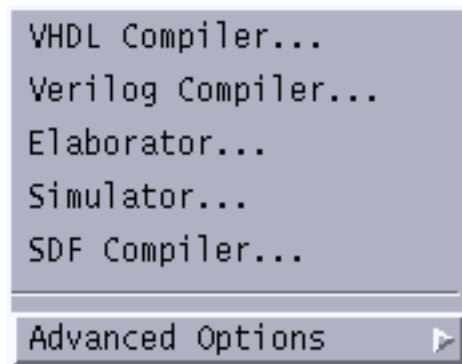
Preferences...

This menu entry opens a form that lets you customize your preferences for NCLaunch. NCLaunch stores the information that you specify on the Preferences form in your home directory (c : if your home is not set on NT) and uses it in all current and future sessions. The Preferences form is shown in [Figure 2-14](#) on page 32.

Figure 2-14 Preferences Form



Tools Menu



The *Tools* menu contains entries that open tool invocation forms that let you invoke simulation tools.

Each tool runs in the Console window region of NCLaunch, with the exception of the simulator. On the simulator invocation form, you specify whether you want to run the simulation in batch, interactive, or graphical mode. If you run the tool in batch or graphically,

NCLaunch User Guide

Using NCLaunch

the simulation executes in the background. If you choose interactive mode, the simulation executes in an xterm or command window, so that you can then interact with the simulation in the other window, leaving the Console window I/O region open for other interactions.

The contents of the tool invocation forms depend on the object selections that you make before opening the forms. The first entry on each of the forms is selection dependent. In some cases, the tool may accommodate multiple inputs or selections. In this case, selection order is very important because it is maintained in the form. If you select any item before you open an invocation form, the items appear in the first field in the order that you selected them. If you make an additional selection while the form is open, NCLaunch appends the the recent selection to the first field. If you make a replacement selection while the form is open, NCLaunch clears the field and only adds the new selection.

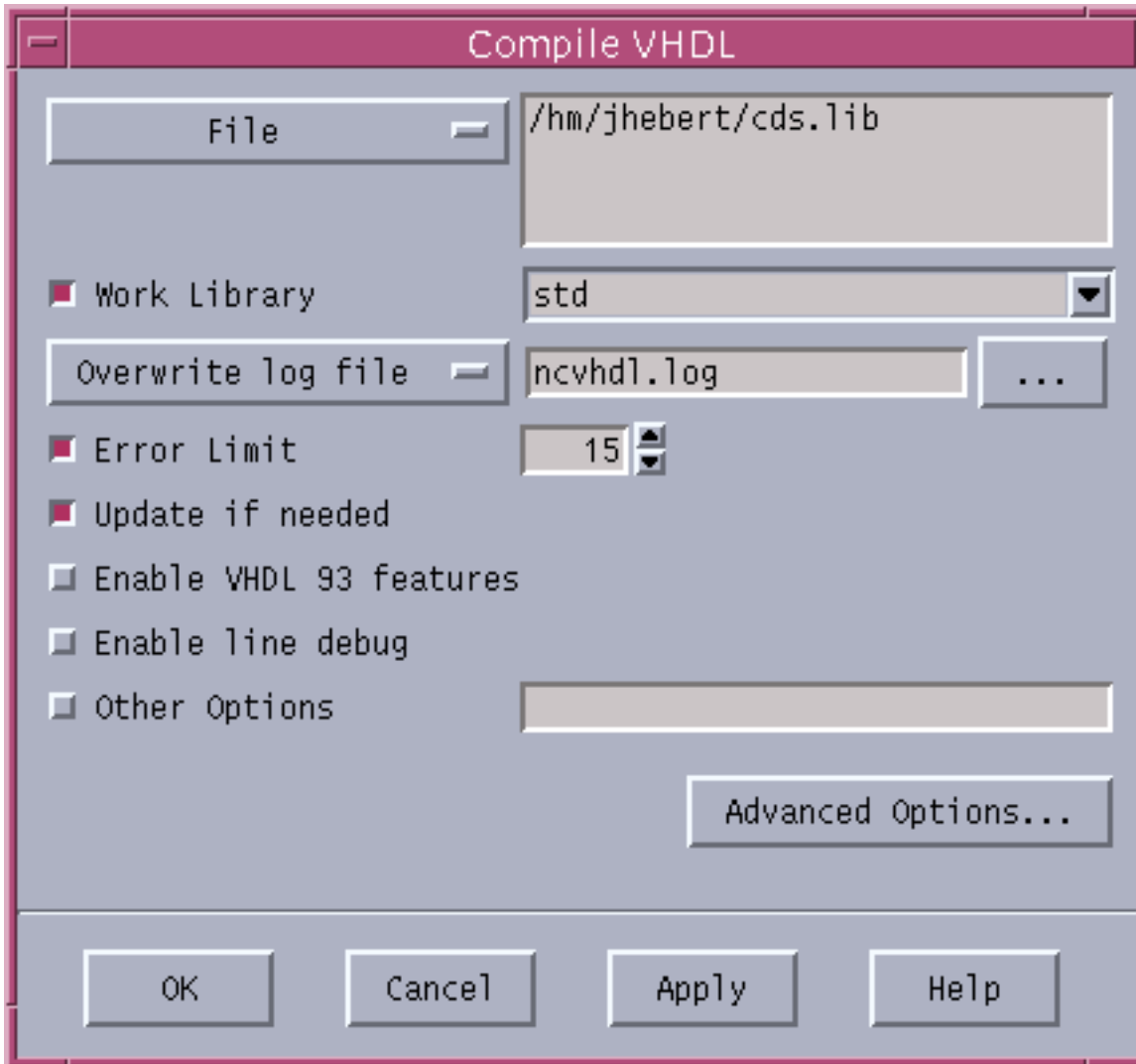
The *Tools* menu contains commands that start the following tools:

- [VHDL Compiler](#)
- [Verilog Compiler](#)
- [Elaborator](#)
- [Simulator](#)
- [SDF Compiler](#)
- You can also [set options](#) for each of the above tools.

VHDL Compiler

The *VHDL Compiler* menu option launches the `ncvhd1` compiler. The `ncvhd1` compiler analyzes and compiles VHDL source files. See the *Affirma NC VHDL Simulator Help* for full details on the `ncvhd1` compiler and on the available options. The Compile VHDL form appears on [Figure 2-15](#) on page 34.

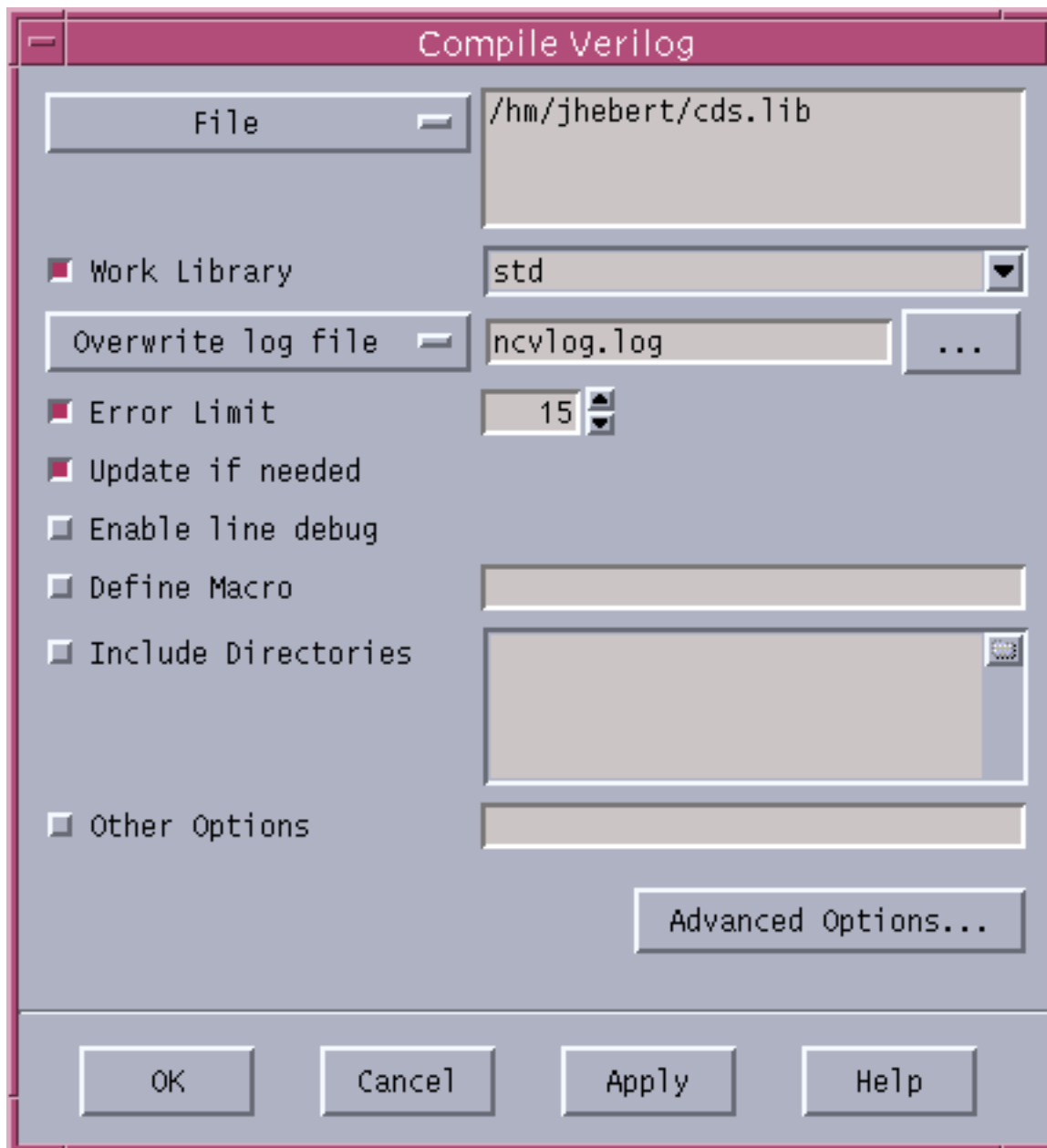
Figure 2-15 Compile VHDL Form



Verilog Compiler

The *Verilog Compiler* menu option launches the `ncvlog` compiler. The `ncvlog` compiler analyzes and compiles Verilog source files. See the *Affirma NC Verilog Simulator Help* for full details on the `ncvlog` compiler and on the available options. The Compile Verilog form appears in [Figure 2-16](#) on page 35.

Figure 2-16 Compile Verilog Form



Elaborator

The *Elaborator* menu option launches the `ncelab` elaborator. The elaborator takes the Library.Cell:View name of the top-level HDL design unit(s) as input. The elaborator then constructs a design hierarchy based on the instantiation and configuration information in your design, establishes the connectivity of the design, and computes the initial values for all of the objects in the design.

See the sections on elaborating your design in either the *Affirma NC VHDL Simulator Help* or the *Affirma NC Verilog Simulator Help* for full details on the elaborator. The Elaborate form is shown in [Figure 2-17](#) on page 36.

Figure 2-17 Elaborate Form

The screenshot shows the 'Elaborate' dialog box with the following fields and options:

- Design Unit:** ([lib.]cell[:view])
- Snapshot Name
- Work Library: std
- Overwrite log file: ncelab.log
- Error Limit: 15
- Access Visibility: Read
- Other Options
- Advanced Options...** button
- OK**, **Cancel**, **Apply**, **Help** buttons

Simulator

The *Simulator* menu option launches either the `ncvlog` or `ncvhdl` simulator, depending on the type of source file that you have selected. The simulators simulate the snapshot file that the elaborator has generated.

See the sections on simulating your design in either the [Affirma NC VHDL Simulator Help](#) or the [Affirma NC Verilog Simulator Help](#) for full details on the simulator. The Simulate form is shown in [Figure 2-18](#) on page 37.

Figure 2-18 Simulate Form

The image shows a dialog box titled "Simulate" with a light blue background. It contains the following elements:

- Snapshot:** A dropdown menu showing "([lib.]cell[:view])".
- Command Filename:** A text input field with a browse button ("...").
- Overwrite log file:** A checked checkbox followed by a text input field containing "ncsim.log" and a browse button ("...").
- Error Limit:** A checked checkbox followed by a spinner control set to "15".
- Update if needed:** An unchecked checkbox followed by a checked checkbox labeled "Update source files".
- Run Mode:** A dropdown menu set to "Graphical UI".
- Other Options:** A text input field.
- Advanced Options...:** A button located below the "Other Options" field.
- Buttons:** "OK", "Cancel", "Apply", and "Help" buttons are arranged horizontally at the bottom of the dialog.

SDF Compiler

The *SDF Compiler* menu option launches the SDF Compiler. You use an SDF file to annotate timing and delay data information into your model. You must compile SDF files to annotate your timing information when you use the Affirma NC VHDL or Affirma NC Verilog simulator.

See the sections in either the *Affirma NC VHDL Simulator Help* or the *Affirma NC Verilog Simulator Help* for full details on compiling SDF.

Figure 2-19 Compile SDF Form

The image shows a dialog box titled "Compile SDF". It has a standard Windows-style title bar with a close button. The main area contains the following elements:

- A text field labeled "SDF File" containing the path "/hm/jhebert/cds.lib".
- A text field labeled "Overwrite log file" containing "ncsdfc.log", followed by a browse button "...".
- A "Compile" button.
- Two checkboxes: "Output file name" (unchecked) and "Other Options" (unchecked).
- A text field for "Output file name" containing "cds.lib.X".
- A text field for "Other Options" which is currently empty.
- An "Advanced Options..." button.
- At the bottom, four buttons: "OK", "Cancel", "Apply", and "Help".

Advanced Options

The following option forms provide a list of the command-line options that are available for use with each specific tool. Once you set and save these options, each invocation of the tool uses them. These options apply when you run the tool from the *Tools* menu, the Tool Bar, or a popup menu. If you type a tool invocation command into the Console window I/O region, then NCLaunch runs that command exactly as specified and does not adhere to the options that you set with the corresponding options form.

NCLaunch User Guide

Using NCLaunch

You can also specify options by clicking the *Advanced Options* button on the invocation form for each tool.

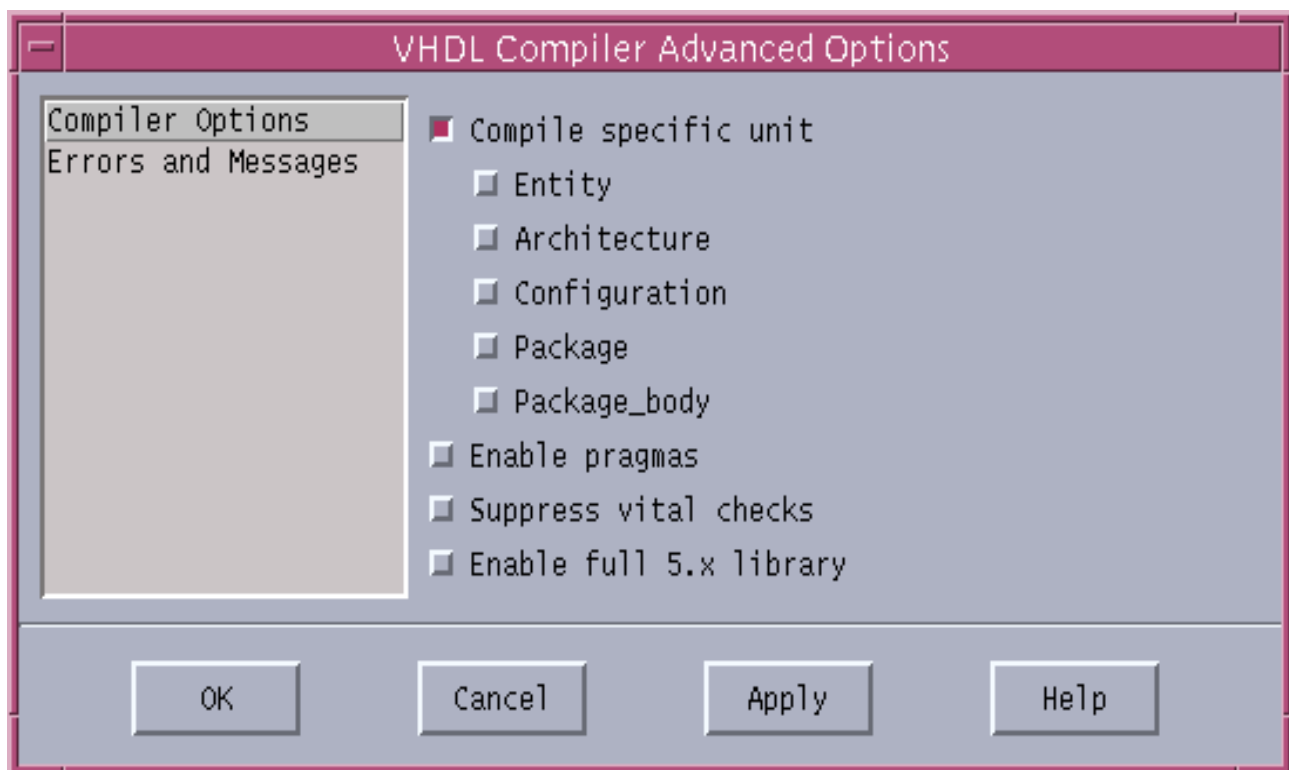
When you exit NCLaunch, it saves your advanced options to your current working directory. This lets you save options and associate them with a specific design.

VHDL Compiler...

This menu item opens the VHDL Compiler Advanced Options form, shown in [Figure 2-20](#) on page 39, which provides a list of the command-line options that are available for use with the `ncvhd1` compiler.

In the VHDL Compiler Advanced Options form, select either *Compiler Options* or *Errors and Messages* to see a set of option buttons specific to that topic.

Figure 2-20 VHDL Compiler Advanced Options Form



NCLaunch User Guide

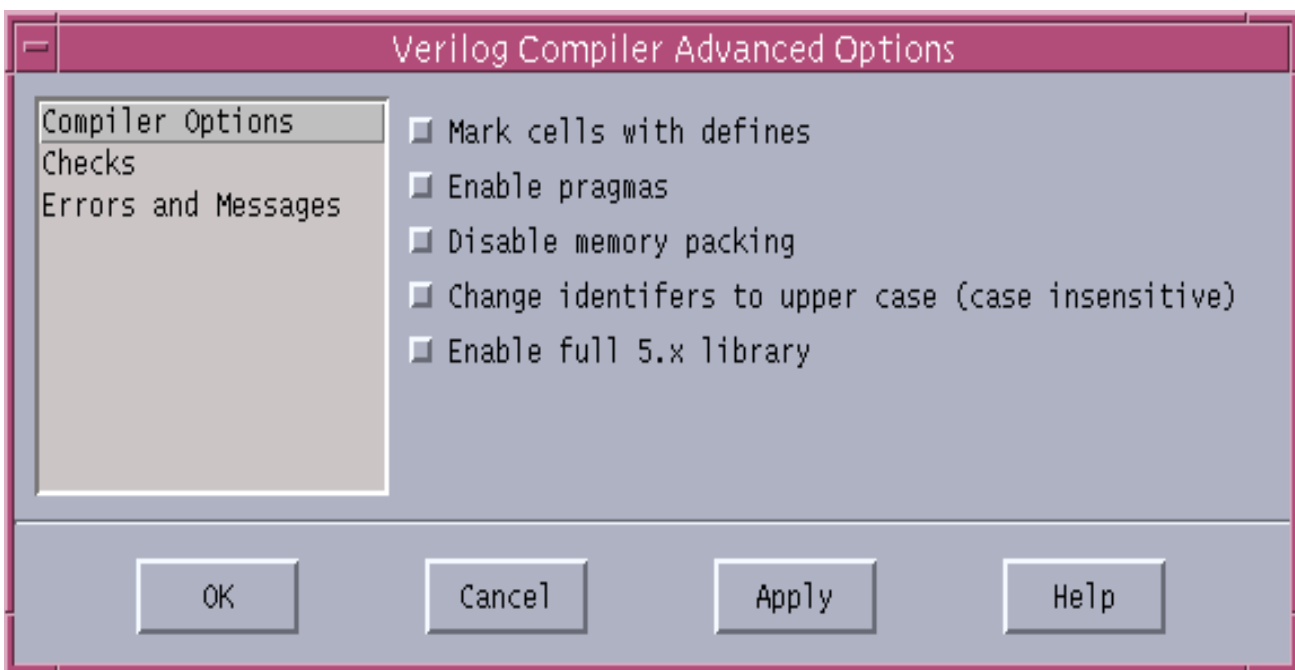
Using NCLaunch

Verilog Compiler...

This menu option opens the Verilog Compiler Advanced Options form, shown in [Figure 2-21](#) on page 40, which provides a list of the command-line options that are available for use with the `ncvlog` compiler.

In the Verilog Compiler Advanced Options form, select *Compiler Options*, *Checks*, or *Errors and Messages* to see a set of option buttons specific to that topic.

Figure 2-21 Verilog Compiler Advanced Options Form

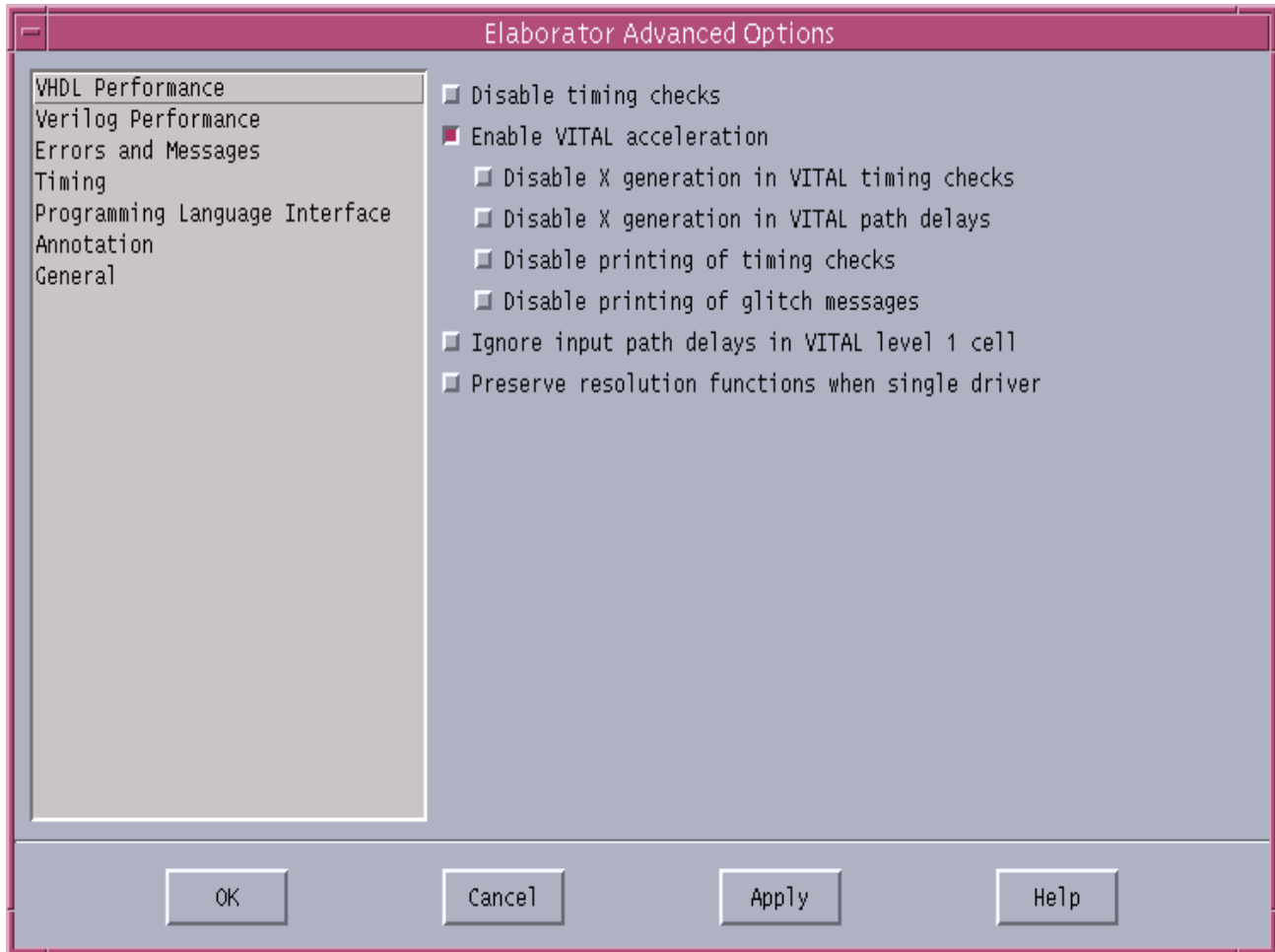


Elaborator...

This menu option opens the Elaborator Advanced Options form, shown in [Figure 2-22](#) on page 41, which provides a list of the command-line options that are available for use with the `ncelab` tool.

In the Elaborator Advanced Options form, select *VHDL Performance*, *Verilog Performance*, *Errors and Messages*, *Timing*, *Programming Language Interface*, *Annotation*, or *General* to see a set of option buttons specific to that topic.

Figure 2-22 Elaborator Advanced Options Form

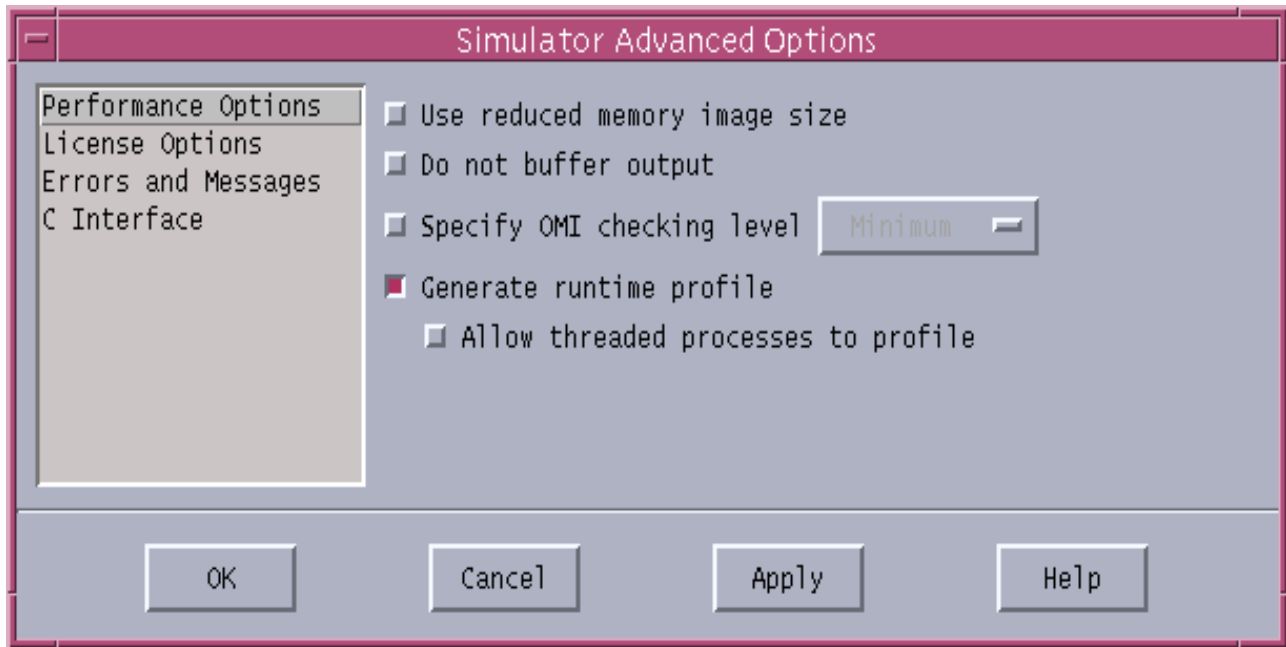


Simulator...

This menu option opens the Simulator Advanced Options form, shown in [Figure 2-23](#) on page 42, which provides a list of the command-line options that are available for use with the `ncsim` tool.

In the Simulator Advanced Options form, select *Performance Options*, *Licence Options*, *Errors and Messages*, or *C Interface* to see a set of option buttons specific to that topic.

Figure 2-23 Simulator Advanced Options Form

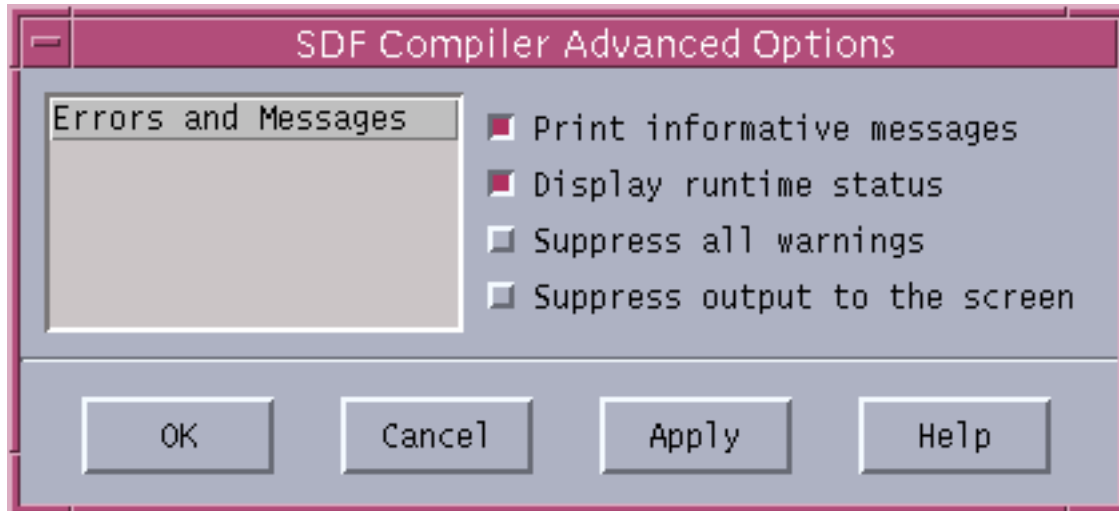


SDF Compiler...

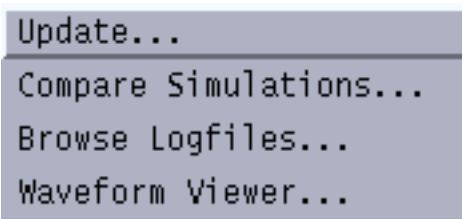
This menu option opens the SDF Compiler Advanced Options form, shown in [Figure 2-24](#) on page 43, which provides a list of the command-line options that are available for use with the SDF compiler tool.

In the SDF Compiler Advanced Options form, press or release the toggle buttons to enable or disable the option.

Figure 2-24 SDF Compiler Advanced Options Form



Utilities Menu



The items under the *Utilities* menu are tools that do not require extensive support for option handling. These tools are simple utilities that aid in the debugging or validation process of your simulation. You can start the following utilities from the *Utilities* menu:

- Update...
- Compare Simulations...
- Browse Logfiles...
- Waveform Viewer...

Update...

This utility operates on the cell or snapshot that you select in the NCLaunch main window. When you choose the *Update...* menu option, NCLaunch presents the Update form, on which you select options that control the update operation. Once you apply these option, this

NCLaunch User Guide

Using NCLaunch

form runs the `ncupdate` utility, which rebuilds any out-of-date dependencies for the selected object. The Update form is shown in [Figure 2-25](#) on page 44.

See the sections in either the *Affirma NC VHDL Simulator Help* or the *Affirma NC Verilog Simulator Help* for full details on using the `ncupdate` utility.

Figure 2-25 Update Form

The image shows a dialog box titled "Update". The "Design Unit" field contains the text "std.s85:module". Below this field is the text "([lib.]cell[:view])". The dialog contains several options, each with a checkbox:

- Do not check for out of date sources
- Force recompilation of all units
- Recompile IEEE library
- Exclude libraries [text field]
- Error Limit [spinner set to 15]
- Print Messages
- Other Options [text field]
- Generate scripts [text field] [browse button]
- Display script
- Suppress recompilation

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

Compare Simulations...

This utility requires you to submit two SHM databases for comparison. When you choose the *Compare Simulations...* menu option, NCLaunch opens the Compare Simulations form, shown in [Figure 2-26](#) on page 45. Once you set and apply the options on this form, NCLaunch runs the Comparescan utility to detect simulation differences. Comparescan is only available on UNIX systems and the menu is inactive for NT users.

See the [Comparescan User Guide](#) for full details on using Comparescan.

Figure 2-26 Compare Simulations Form

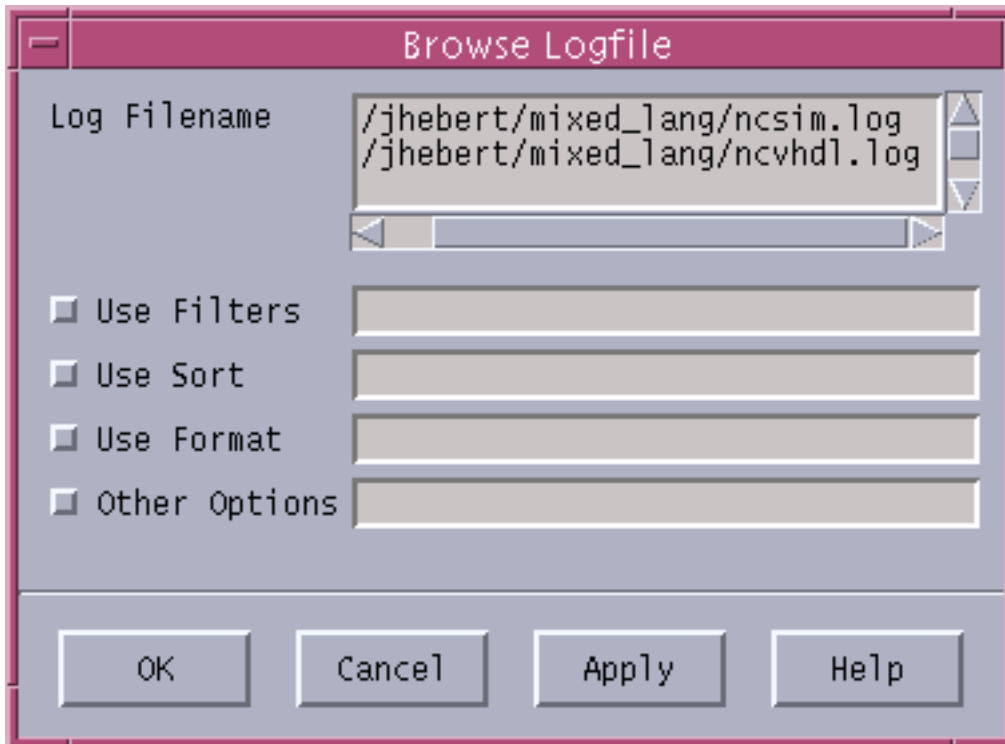
The screenshot shows a dialog box titled "Compare Simulations". It features a list box for "Databases" with two entries: "ebert/comparescan/samples/test.trn" and "ebert/comparescan/samples/golden.trn". Below this is a "Rules File" field containing "an/samples/entire_design.rules" and a browse button "...". There is also a "Generate Error Database" field with a browse button "...". The "Generate Report" field has a dropdown menu set to "Summary". Four checkboxes are present: "Run comparison upon startup", "Run comparison in batch mode", and "Other Options", all of which are unchecked. At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

Browse Logfiles...

This menu option opens the Browse Logfile form, shown in [Figure 2-27](#) on page 46, which lets you run the NCBrowse message browser to graphically display error messages that are

contained in the log file that you select in the NCLaunch main window. NCBrowse is only available on UNIX systems.

Figure 2-27 Browse Logfile Form



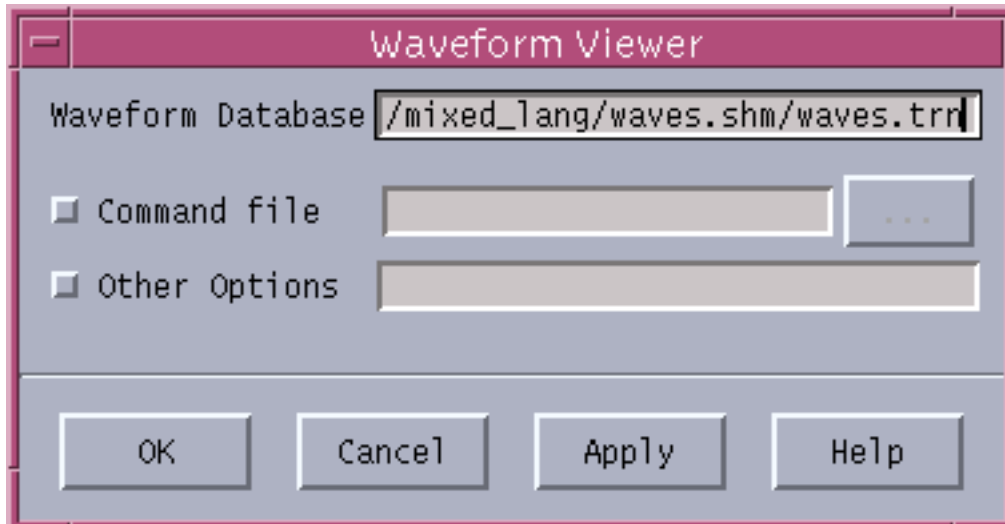
In this form, the *Log Filename* field shows the name of the `.log` file(s) that you select in NCLaunch before opening the form. You can also type the log filename directly into the field. You can specify the filters, sort order, and formats that you want to use in the message browser. Additionally, you can specify any other NCBrowse option.

See the [*NCBrowse Message Browser User Guide*](#) for full details on this utility.

Waveform Viewer...

This menu option opens the Waveform Viewer form, shown in [Figure 2-28](#) on page 47, which lets you run the waveform tool, Signalscan Waves, on the database that you select in NCLaunch before opening the form.

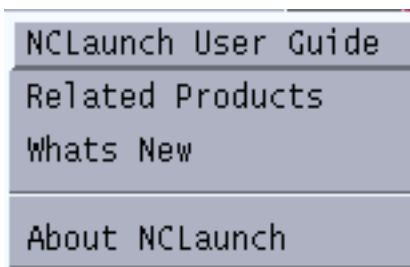
Figure 2-28 Waveform Viewer Form



The *Waveform Database* field shows the `.trn` file(s) that you selected in the NCLaunch main window before opening the form. If you select the *Command file* option, NCLaunch selects to submit the selected file with the `-do` option to the `signalscan` command. The *Other Options* field provides a means for you to submit any additional options to the `signalscan` command. In addition, the `-waves` option is enabled by default.

See the [Signalscan Waves User Guide](#) for full details on using the Signalscan Waves utility.

Help Menu



The items under the *Help* menu display a help message or invoke the document reader to display the appropriate document for that topic.

NCLaunch User Guide

Links to the HTML document display system and displays the *NCLaunch User Guide*.

NCLaunch User Guide

Using NCLaunch

Related Products

- *NC Verilog Help*: Links to the [*Affirma NC Verilog Simulator Help*](#).
- *NC VHDL Help*: Links to the [*Affirma NC VHDL Simulator Help*](#).

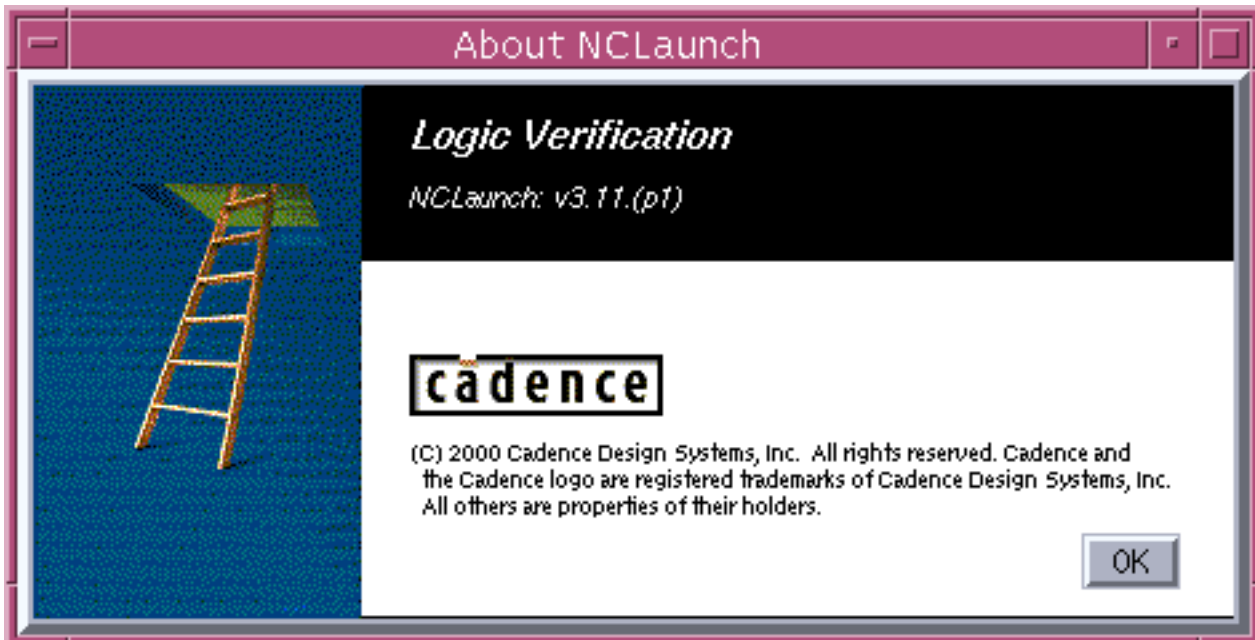
What's New

- *NCLaunch Product Notes*: Links to a document that describes the new features in this release of the product.
- *NCLaunch KPNS*: Links to a document that describes the known problems and suggested solutions for this release of the product.

About NCLaunch

Displays a form that contains the Cadence copyright message and the version of NCLaunch that you are running, as shown in [Figure 2-29](#) on page 48.

Figure 2-29 About NCLaunch



File Browser

The NCLaunch File Browser displays objects in your file system. By default, the browser displays the contents of your working directory. You can easily navigate around your file system, changing the display of the File Browser. This does not impact your current design directory.

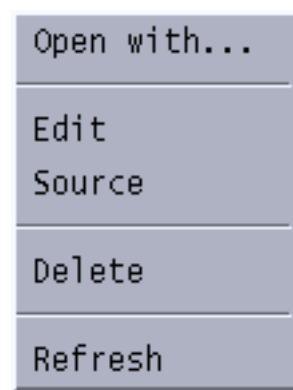
The File Browser lets you select any file, but only recognizes those that are bound to a tool. The popup menu for the File Browser is shown in [Figure 2-30](#) on page 49. You access this popup menu by selecting a file and pressing the right mouse button.

Figure 2-30 File Browser Popup Menus

Bound File Popup Menu
(Verilog Source File)



Unbound File Popup Menu



The first item in either menu is the command to which the file is bound, such as `ncvlog` for `.v` files. If you select this menu choice, NCLaunch provides a form that lets you make any last minute changes before running the tool. This is similar to selecting the tool from the *Tools* menu. However, double-clicking on an object simply issues the command to which the file is bound. This is similar to selecting the tool from the Tool Bar.

The *Edit* command uses your editor choice (as defined in your preferences) to bring up the selected file.

The *Source* command executes the commands stored in the selected file.

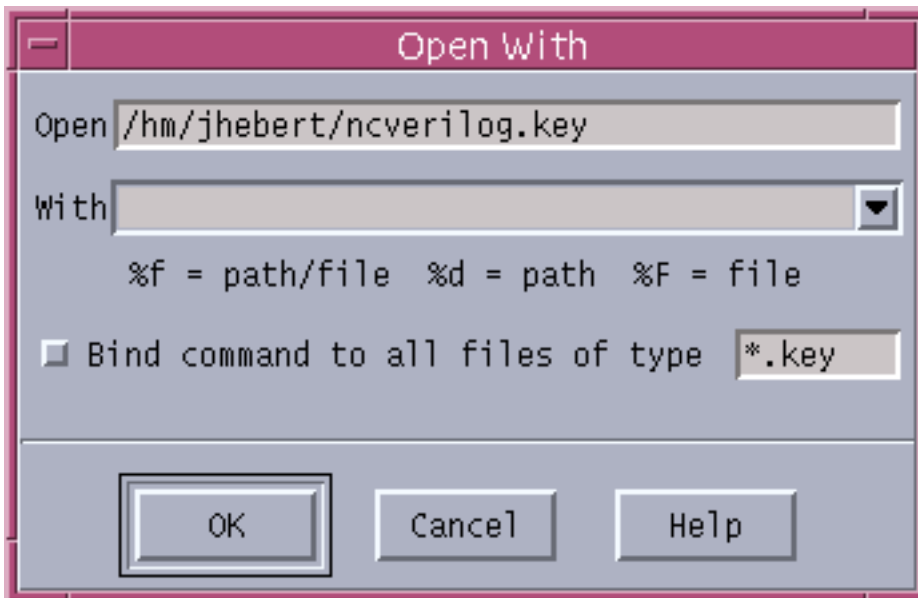
The *Run* command is only available if the file is executable. This command lets you run a program in the Console window I/O region. The *Run with args...* command lets you add arguments to the command.

The *Delete* command deletes the selected file(s) from the disk.

The *Refresh* command redraws the File Browser display. This command is not sensitive to file selection.

For files that are not bound to any command, the *Open With* command opens a form asking you what you want to do with the file, as shown in [Figure 2-31](#) on page 50. This form has a pulldown menu that lets you access the list of known tools; the form also has an area in which you can type in a new command to bind to the type. You can choose to make this new binding a permanent change, and if so, NCLaunch adds the entry to your setup.

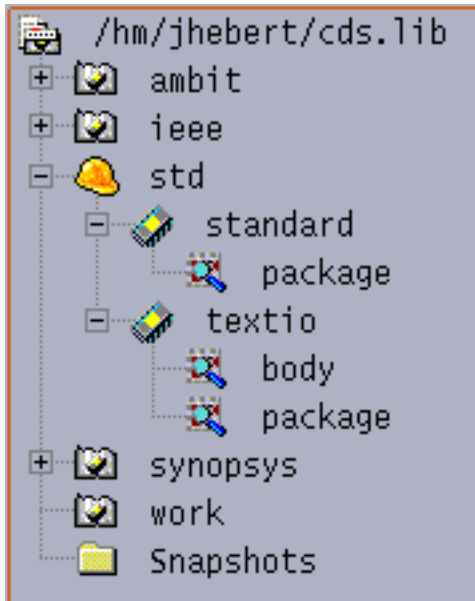
Figure 2-31 Open With Form for Unbound Files



Library Browser

The Library Browser reads and displays the contents of a `cds.lib` file and its corresponding libraries. The `cds.lib` file and its libraries are displayed as a tree to enable you to expand your desired libraries to view and manipulate the cells and design units that make up the library. The Library Browser can display libraries in either packed library or 5.X format. [Figure 2-32](#) on page 51 shows an expanded view of the current work library.

Figure 2-32 Library Browser Hierarchy



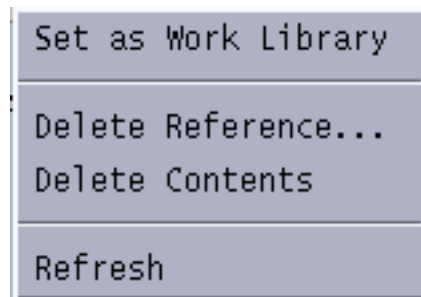
The Library Browser provides access to the following tools and functions for each object. The popup menus contain the entries shown in [Figure 2-33](#) on page 51, based on the type of file that you select.

Figure 2-33 Library Browser Popup Menus

Cell and Design Unit Popup Menu



Library Popup Menu



The popup menu for cell and design elements contains commands that let you launch `ncupdate`, `ncelab`, `ncvhdl`, or `ncvlog` with the selected file(s) as input.

The popup menu for libraries (directories) contains commands to set the selected library as the work library, to delete the library from your `cds.lib` file, and to delete the contents of the selected library.

You can also delete elements from your `cds.lib` file using these popup menus.

Note: There is no confirmation box when deleting elements from the Library Browser popup menus.

Console Window

The Console window lets you submit batch commands to your simulation tools and utilities and view the output of running processes. Standard output messages from running processes are displayed in black, while any error messages (messages that are written to `stderr` by the associated applications) are displayed in red. Any command that NCLaunch issues that requires user interaction is directed to an xterm or command window in which you can interact with the tool. When you issue any command through NCLaunch, it sends the command line through the TCL interpreter and logs the command in a key file. You can later use a key file as a script to run the same set of commands.

The I/O region contains a popup that supports the following commands:

- *Cut* : Cut the selected text from the I/O region; you can only cut user input from the text area.
- *Copy* : Copy the selected text from the I/O region.
- *Paste* : Paste the cut or copied text to the input region.
- *Clear* : Clear the entire I/O region.
- *Select All* : Copy all of the text in the Console window to the paste buffer.

The I/O region is a TCL interpreter, and therefore, it accepts any TCL command. You can also issue operating system commands through this window.

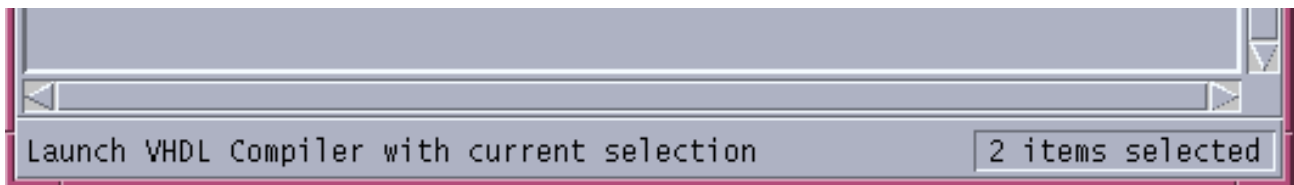
Note: Do not use this functionality to change your current working directory. This does not change the working directory of the tool. Any commands that you issue through the NCLaunch GUI first checks whether the working directory is correct. If it is not correct, NCLaunch changes the directory back to your design directory and then runs the requested tool. NCLaunch keeps the working directory of the I/O region linked to the design directory. The only way to change the working directory of the tool is through the *Set Design Directory* option in the *File* pulldown menu.

Status Bar

The Status Bar area is a small window that runs across the bottom of the screen, in which a single line of text informs you about various UI activities. When you move the cursor over a Tool Bar button or a menu item, the Status Bar area displays an informative message that describes the function of that item. This area also displays the number of selected items.

[Figure 2-34](#) on page 53 shows the Status Bar after you have selected two VHDL files and positioned the cursor over the VHDL tool bar button.

Figure 2-34 Status Bar



Tool Bar

The Tool Bar contains buttons that let you quickly execute commands. Tools that you start from these buttons execute immediately, with the selected file as input. Any output from tools that you run from the Tool Bar appears in the Console window.

You must use the *Tools* or *Utilities* menus to start tools, if you want to specify any options when launching a tool.

[Figure 2-35](#) on page 54 shows the Tool Bar.

NCLaunch User Guide

Using NCLaunch

Figure 2-35 Tool Bar

