cadence®

Cadence Library Path Editor User Guide

Product Version ICADVM20.1
October 2020

Printed in the United States of America.

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

Contents

Preface

The Cadence® Library Path Editor helps you to define the libraries being used in your design. You can set your `cds.lib` file to point to the reference and design libraries you want to use in your design. This user guide describes how to use the Library Path Editor. It assumes that you are familiar with the Virtuoso® design environment.

This user guide is aimed at developers and designers of integrated circuits and assumes that you are familiar with:

- The Virtuoso design environment and application infrastructure mechanisms designed to support consistent operations between all Cadence® tools.
- The applications used to design and develop integrated circuits in the Virtuoso design environment, notably, the Virtuoso Layout Suite, and Virtuoso Schematic Editor.
- The Virtuoso design environment technology file.
- Component description format (CDF), which lets you create and describe your own components for use with Layout XL.

This preface contains the following topics:

Scope

Unless otherwise noted, the functionality described in this guide can be used in both mature node (for example, IC6.1.8) and advanced node and methodologies (for example, ICADVM20.1) releases.

Label	Meaning
(ICADVM20.1 Only)	Features supported only in ICADVM20.1 advanced nodes and advanced methodologies releases.
(IC6.1.8 Only)	Features supported only in mature node releases.

Licensing Requirements

For information on licensing in the Virtuoso design environment, see the *Virtuoso Software Licensing and Configuration Guide*.

Related Documentation

What's New and KPNS

- *Cadence Library Path Editor What's New*
- *Cadence Library Path Editor Known Problems and Solutions*

Installation, Environment, and Infrastructure

- *Cadence Installation Guide*
- *Cadence Application Infrastructure User Guide*
- *Cadence Library Manager User Guide*
- *Virtuoso Design Environment User Guide*

Additional Learning Resources

Video Library

The Video Library on the Cadence Online Support website provides a comprehensive list of videos on various Cadence products.

To view a list of videos related to a specific product, you can use the *Filter Results* feature available in the pane on the left. For example, click the *Virtuoso Layout Suite* product link to view a list of videos available for the product.

You can also save your product preferences in the Product Selection form, which opens when you click the *Edit* icon located next to *My Products*.

Virtuoso Videos Book

You can access certain videos directly from Cadence Help. To learn more about this feature and to access the list of available videos, see Virtuoso Videos.

Rapid Adoption Kits

Cadence provides a number of Rapid Adoption Kits that demonstrate how to use Virtuoso applications in your design flows. These kits contain design databases and instructions on how to run the design flow.

To explore the full range of training courses provided by Cadence in your region, visit Cadence Training or write to training_enroll@cadence.com.

Note: The links in this section open in a separate web browser window when clicked in Cadence Help.

Help and Support Facilities

Virtuoso offers several built-in features to let you access help and support directly from the software.

- The Virtuoso *Help* menu provides consistent help system access across Virtuoso tools and applications. The standard Virtuoso *Help* menu lets you access the most useful help and support resources from the Cadence support and corporate websites directly from the CIW or any Virtuoso application.

- The Virtuoso Welcome Page is a self-help launch pad offering access to a host of useful knowledge resources, including quick links to content available within the Virtuoso installation as well as to other popular online content.

  The Welcome Page is displayed by default when you open Cadence Help in standalone mode from a Virtuoso installation. You can also access it at any time by selecting *Help – Virtuoso Documentation Library* from any application window, or by clicking the *Home* button on the Cadence Help toolbar (provided you have not set a custom home page).

For more information, see Getting Help in *Virtuoso Design Environment User Guide.*

Customer Support

For assistance with Cadence products:

- Contact Cadence Customer Support

  Cadence is committed to keeping your design teams productive by providing answers to technical questions and to any queries about the latest software updates and training needs. For more information, visit https://www.cadence.com/support.

- Log on to Cadence Online Support

  Customers with a maintenance contract with Cadence can obtain the latest information about various tools at https://support.cadence.com.

Feedback about Documentation

You can contact Cadence Customer Support to open a service request if you:

- Find erroneous information in a product manual
- Cannot find in a product manual the information you are looking for
- Face an issue while accessing documentation by using Cadence Help

You can also submit feedback by using the following methods:

- In the Cadence Help window, click the *Feedback* button and follow instructions.
- On the Cadence Online Support Product Manuals page, select the required product and submit your feedback by using the *Provide Feedback* box.

Typographic and Syntax Conventions

The following typographic and syntax conventions are used in this manual.

text	Indicates names of manuals, menu commands, buttons, and fields.
`text`	Indicates text that you must type exactly as presented. Typically used to denote command, function, routine, or argument names that must be typed literally.
`z_argument`	Indicates text that you must replace with an appropriate argument value. The prefix (in this example, *`z_`*) indicates the data type the argument can accept and must not be typed.
|	Separates a choice of options.
`{ }`	Encloses a list of choices, separated by vertical bars, from which you **must** choose one.
`[ ]`	Encloses an optional argument or a list of choices separated by vertical bars, from which you **may** choose one.
`[ ?argName` *`t_arg`* `]`	Denotes a *key argument*. The question mark and argument name must be typed as they appear in the syntax and must be followed by the required value for that argument.
`...`	Indicates that you can repeat the previous argument. Used with brackets to indicate that you can specify zero or more arguments. Used without brackets to indicate that you must specify at least one argument.
`,...`	Indicates that multiple arguments must be separated by commas.
`=>`	Indicates the values returned by a Cadence® SKILL® language function.
`/`	Separates the values that can be returned by a Cadence SKILL language function.

If a command-line or SKILL expression is too long to fit within the paragraph margins of this document, the remainder of the expression is moved to the next line and indented. In code excerpts, a backslash (\ ) indicates that the current line continues on to the next line.

1

Using the Library Path Editor

The following topics are discussed in this chapter:

Read First: Library Definition Files in IC614

From IC614, `lib.defs` library definition files are no longer supported in Virtuoso. The only supported format now being `cds.lib`.

A new CdsLib plugin (release 31.09) allows for OpenAccess applications to read `cds.lib` files, where previously they required to use `lib.defs`.

Note: The plugin will ignore any OpenAccess related attributes that are assigned to libraries in `lib.defs` files.

This *Cadence Library Path Editor User Guide* therefore places focus on the use of `cds.lib` files. It should be noted however that where `lib.defs` are still being used, the commands and instructions documented in this manual will, in most cases, still be applicable. Again though, Cadence actively encourages the use of `cds.lib` files as library definition files.

Note: `lib.defs` files may still be optionally saved for use with other, non-Cadence programs, or OpenAccess tools that do not have the CdsLib plugin.

About the Library Path Editor and cds.lib

The library path editor (cdsLibEditor) enables you to view and edit the information in a `cds.lib` library definition file.

A `cds.lib` file is needed to point to the reference and design libraries you want to use in your design.

For more information about `cds.lib` refer to the *Cadence Application Infrastructure User Guide*.

Opening and Closing the Library Path Editor

Opening the library path editor

The *library path editor* opens from the UNIX command line and Cadence® tools such as the Virtuoso® design environment's Command Interpreter Window (CIW).

To open the library path editor from the UNIX command line,

➤ Type

```
cdsLibEditor [ -namespace namespace ] &
```

where *namespace* is the set of rules defining valid identifier and keyword types for the design tool in use (the default is `CDBA`).

To open a `cds.lib` library definition file in the library path editor, use the following commands:

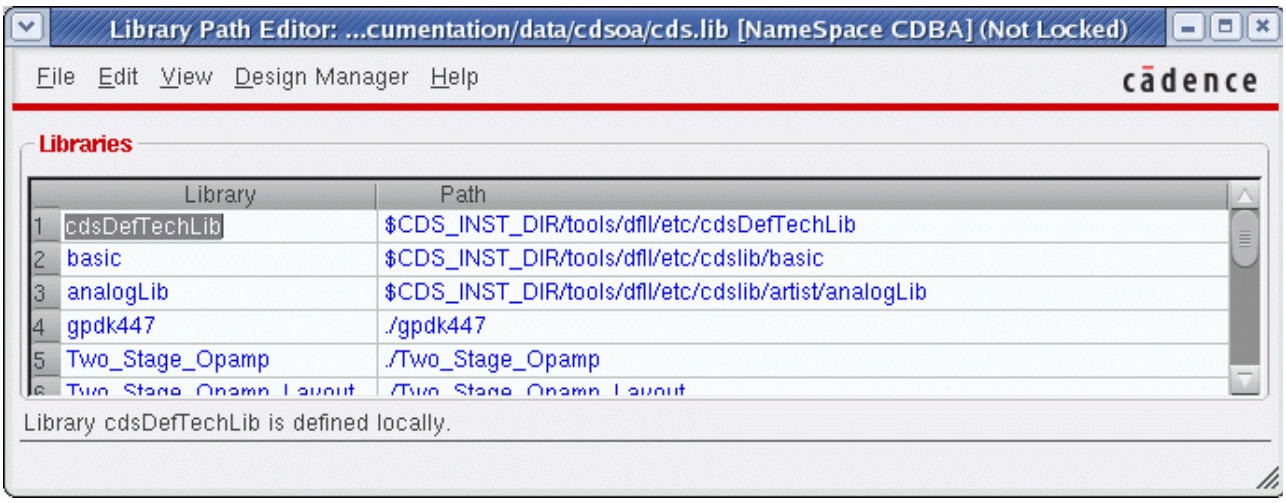
```
cdsLibEditor -cdslib cds.lib
```

To open the library path editor from the CIW,

➤ Choose *Tools – Library Path Editor*.

The *library path editor* window opens and presents library path information in one of the following ways:

■ Displaying the libraries and their paths that are defined in `cds.lib` files.



- Displaying the *Select New File Format* form because a `cds.lib` file does not exist in your design hierarchy. In this situation a library definition file needs to be created.

  See Creating Library Definitions Files on page 16.

Note: The applications, such as Library Manager, Library Selector, and Library Path Editor, will start with the same font as Virtuoso. Though, it may not be noticeable unless the default font is changed using the Set Fonts dialog box. Once you choose the font using the Set Fonts dialog box, the font of these applications will change accordingly.

Closing the library path editor

To close the library path editor,

1. Choose *File – Exit*.

   If you have not saved changes, a dialog box asks you whether you want to save your changes.

2. Do one of the following:

   - Click *Yes* to save your changes and exit.
   - Click *No* to discard your changes and exit.
   - Click *Cancel* to dismiss the form without exiting.

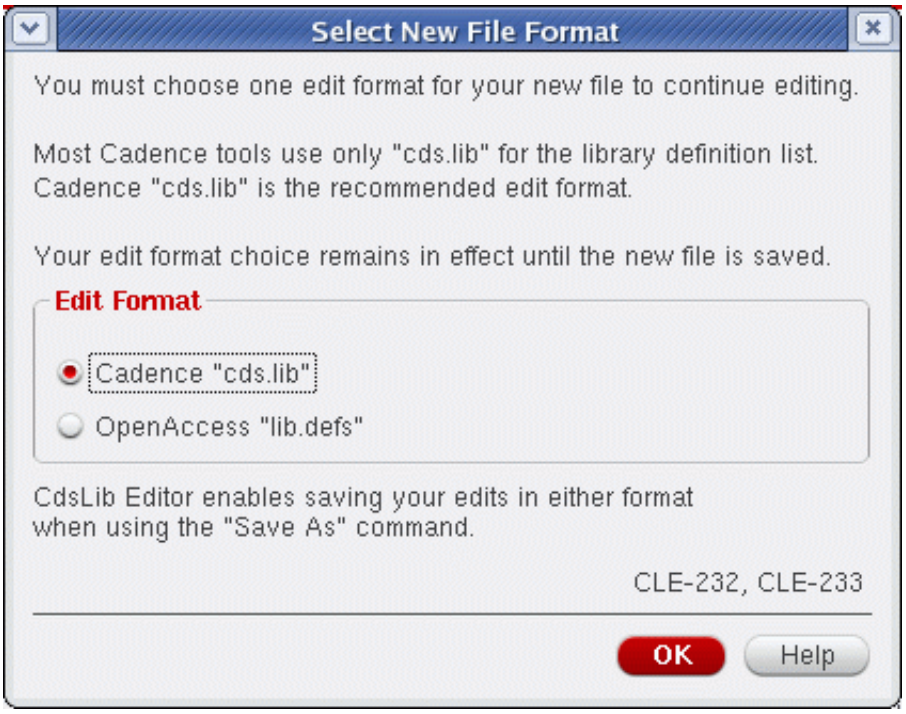
Creating Library Definitions Files

If your design directory does not contain library definitions files, you can create them:

- Using the *File – New* command in the library path editor.

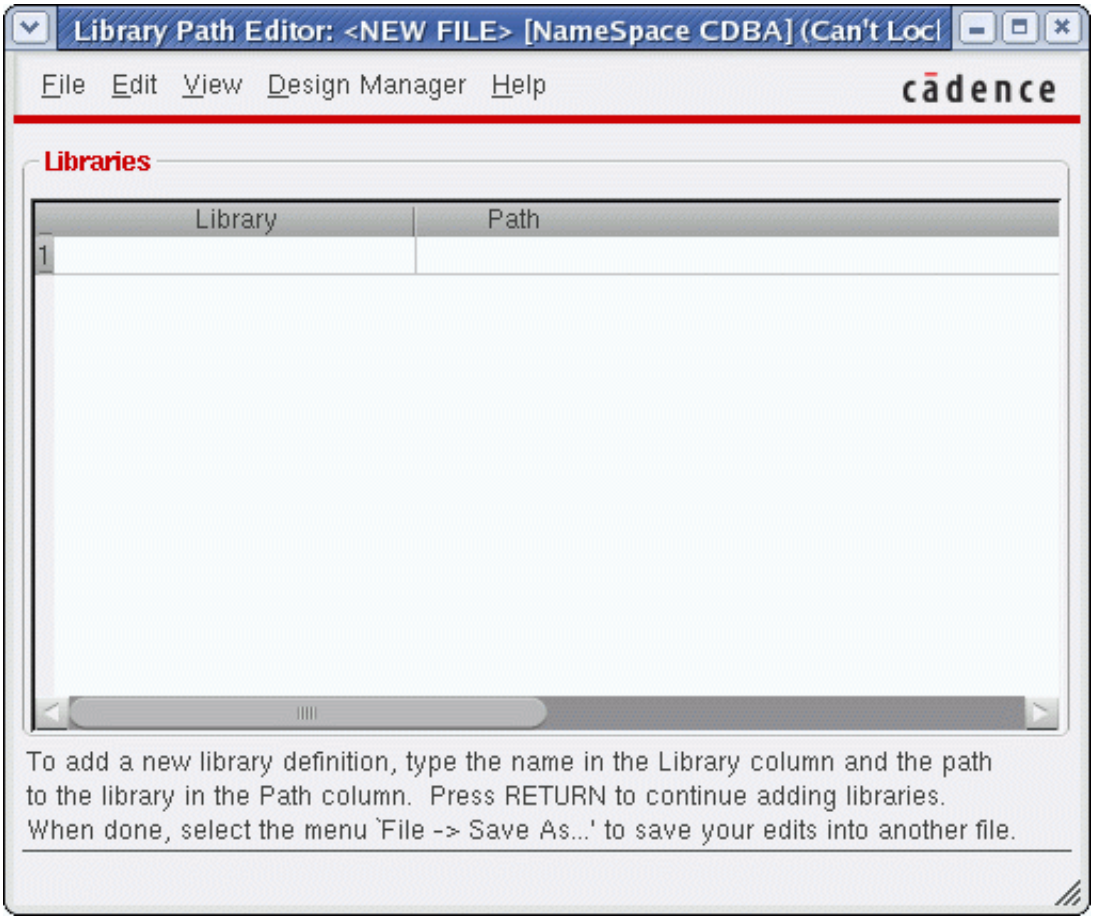
  a. In the *Select New File Format* form, choose one of the following:

  - *Cadence “cds.lib”*
  - *OpenAccess “lib.defs”*

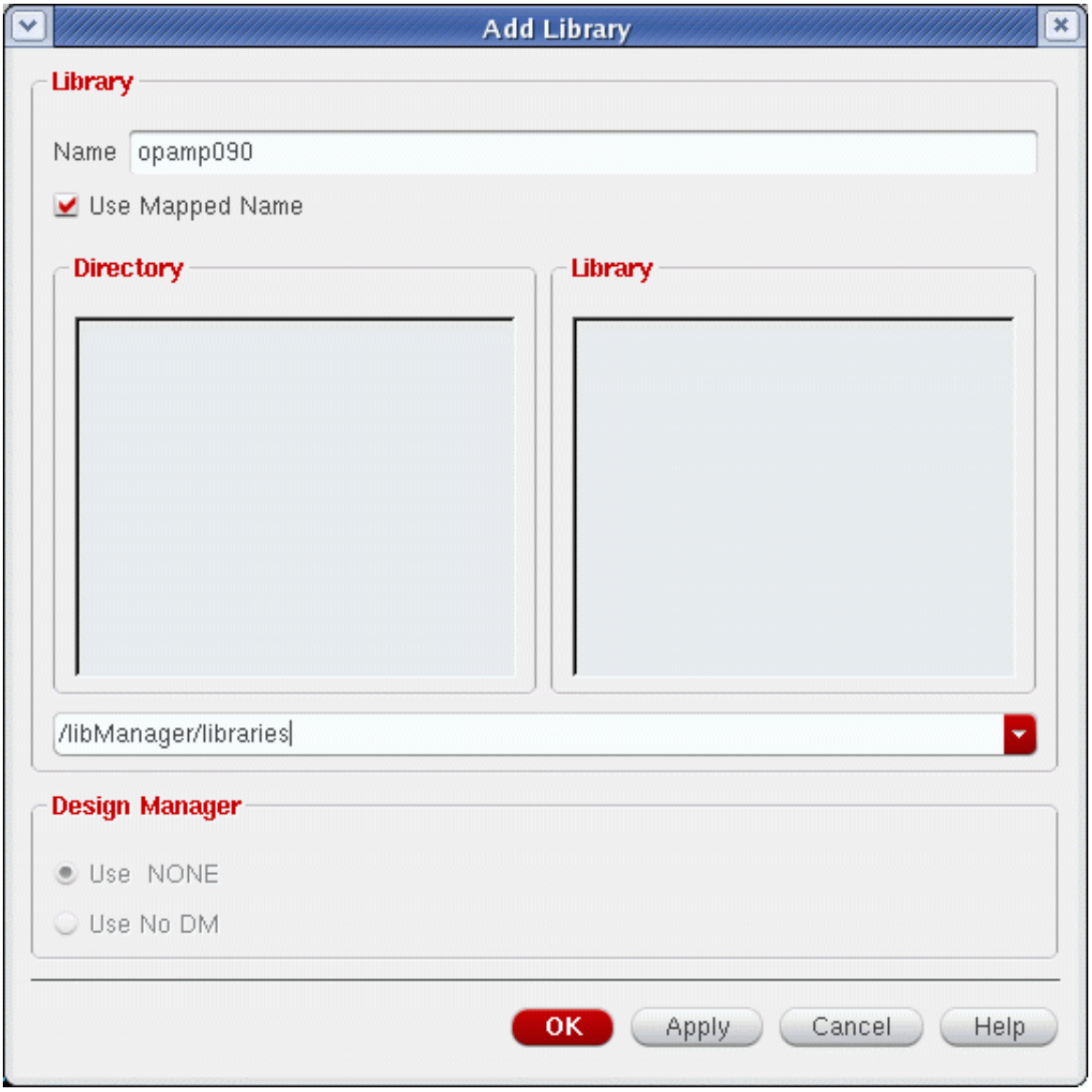


  b. Click *OK*.

The library path editor displays blank *Library* and *Path* fields.

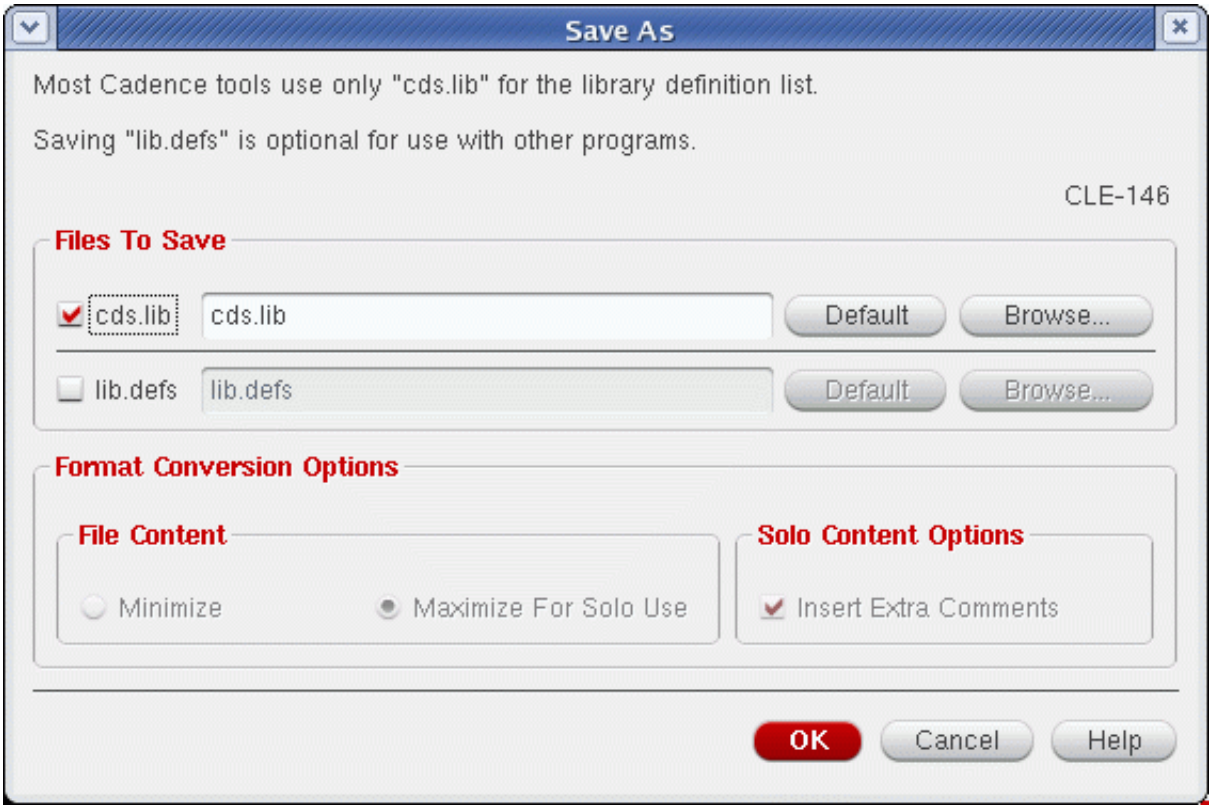


c. Add libraries using the *Edit – Add Library* command.



d. Locate the library to be added and then click *OK*.

e. Create the missing file by choosing *File – Save As*.



f. Click *OK* to create the library.

- Copy an existing `cds.lib` file into your working directory.
  - For example, copy the Cadence-supplied `cds.lib` file found at
    `$CDS_INST_DIR/share/cdssetup/cds.lib`
- Create a `cds.lib` using a text editor.

  See Keywords and Syntax of Library Definitions Files on page 38.

  Example keywords and paths:

  `INCLUDE` *`your_install_dir`*`/share/cdssetup/cds.lib`

  where *`your_install_dir`* is the path to the installation directory.

  `DEFINE` *`yourLibraryName your_install_dir`*`/tools/etc/cdslib/`*`libraryName`*

  where *`yourLibraryName`* is the name you wish to apply to the library, *`your_install_dir`* is the path to your installation directory, and *`libraryName`* is the name of the library you are adding to the new `cds.lib` file.

Caution

Names must follow namespace rules. Refer to the Cadence Application Infrastructure User Guide ***for details.***

Adding Libraries to a Library Definition File

To add a library to a library definition file:

1. Choose *File – Open* in the Library Path Editor window.
2. Select the *cds.lib* file in the File Open form.
3. Click *OK*.

   The library path editor displays the contents of the files.
4. Choose *Edit – Add Library*.

   The Add Library form appears.
5. In the *Name* field, type the name of the existing or new library you want to add.
6. Select the path to the library using the *Directory* and *Library* list boxes or type the path into the type-in field below the list boxes.

   You can click the directory names in the *Directory* list box to move up and down in the directory hierarchy. Only library directories containing a valid `cdsinfo.tag` file appear in the *Library* scrolling list box. (For more information about the `cdsinfo.tag` file, see the *Cadence Application Infrastructure User Guide*.)

   Note: The library added is the library specified in the *Name* field with the path shown in the *Directory* list box. Be sure that the path shown in the field below the *Directory* list box does not include the library name; if it does, the library path editor will create a directory with the same name as the library, plus the library. For example, if you specify the path `/usr1/lib_dir/lib1` where `lib1` is the library name you specify in the *Name* field, the library path editor will create the library `/usr1/lib_dir/lib1/lib1`.
7. Select *Use Mapped Name* if you want to map a directory name that appears in the *Library* list back to the application's name space. The mapped name is displayed in the *Name* field. If this option is not selected, the directory name is added the way it is.

   Note: The directory name must be a valid name for it to be mapped.

   For example, if you created a library `sample.lib` with a Cadence application, it would have been mapped to `sample#2elib` in the file system. When you try to add that library in the Add Library form, the *Library* list box displays the file system name: `sample#2elib`. If you select the *Use Mapped Name* option, the library is mapped back and is added as `sample.lib`; otherwise, the library is added as `sample#2elib`.
8. In the *Design Manager* section, select a design management system or *Use No DM* if you do not want the library to be managed.

Note: If you are not running the software from a design-managed workarea, the *Design Manager* field is grayed-out. If you are running the software from a design-managed workarea, the appropriate design management system is listed as a choice.

9. Click *OK*.

The library name and path appear in the library path editor window.

10. Choose *File – Save As* (see also Locking the File for Editing).

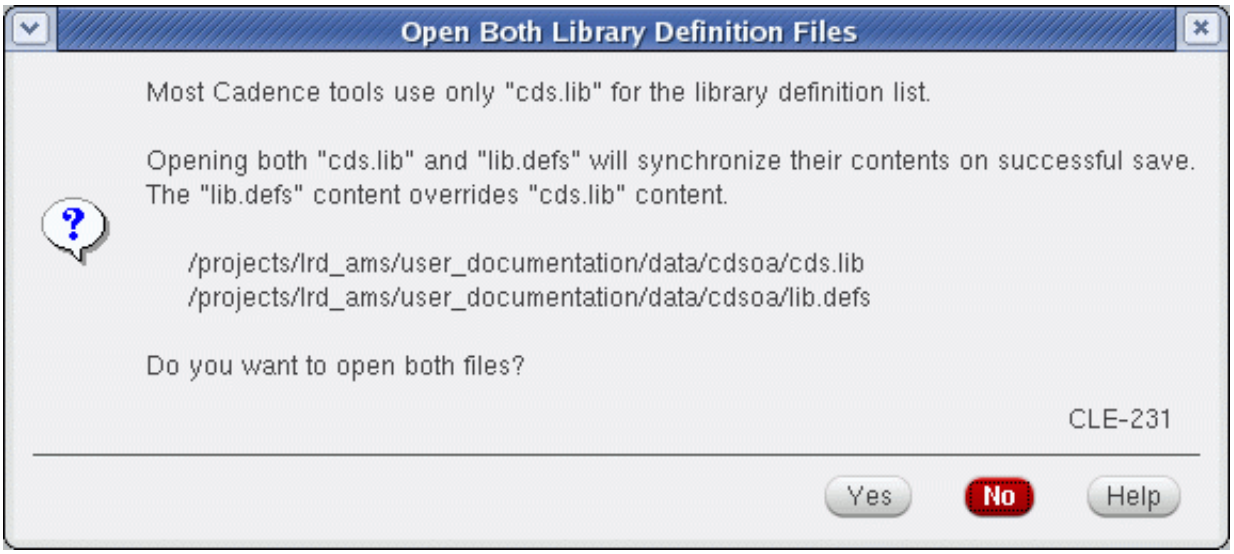
Ensure that the `cds.lib` file has a check mark next to it in the Save As form.

- Click *OK*.
- Click *Yes* in the dialog box to overwrite the existing files.

Open Both the cds.lib and lib.defs Files

If you choose to open both the `cds.lib` and `lib.defs` file in the File Open form, this will display the Open Both Library Definitions Files form. From here you will be asked to confirm that action.

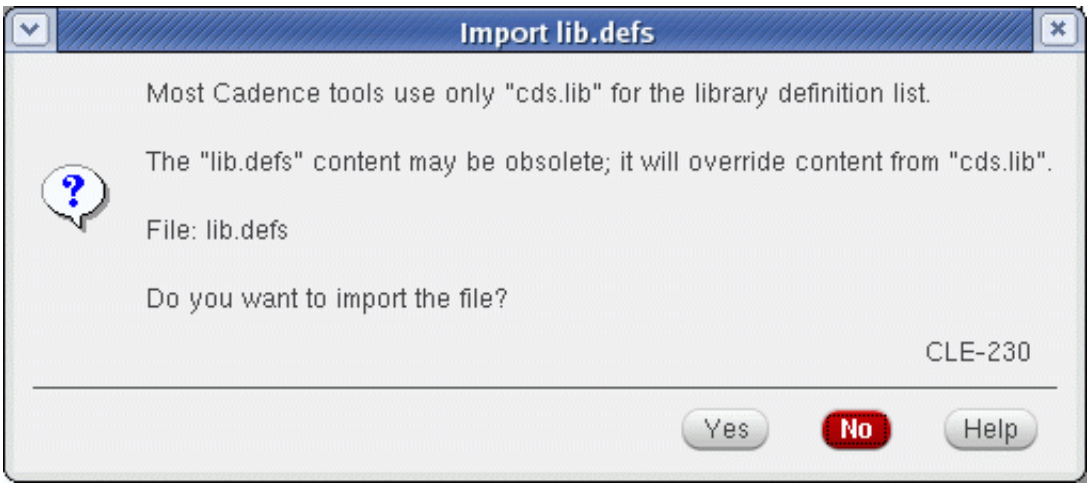
Choosing to open both the `cds.lib` and `lib.defs` files will result in their contents being synchronized on a successful save. Here, the `lib.defs` file content will override that of the `cds.lib` content.



Importing a lib.defs File

To import a `lib.defs` file, select *File – Import "lib.defs"*. This will display the Import lib.defs form.

From here, you can continue to choose to import a `lib.defs` file, but are warned that the `lib.defs` file content may be obsolete (see Read First: Library Definition Files in IC614) and that it would also override any content from the `cds.lib` file.



Deleting Libraries

To delete a library from the Library Path Editor window,

1. Choose *File – Open* in the library path editor form.
2. Select the file from the File Open form.
3. Click *OK*.
4. Click a library name you want to delete.
5. Choose *Edit – Remove Library Definition*.

   The library is deleted from the library path editor window
6. Choose *File – Save As* (see also Locking the File for Editing).

   Ensure that the `cds.lib`file has a check mark next to it in the Save As form.

   - Click *OK*.
   - Click *Yes* in the dialog box to overwrite the existing files.

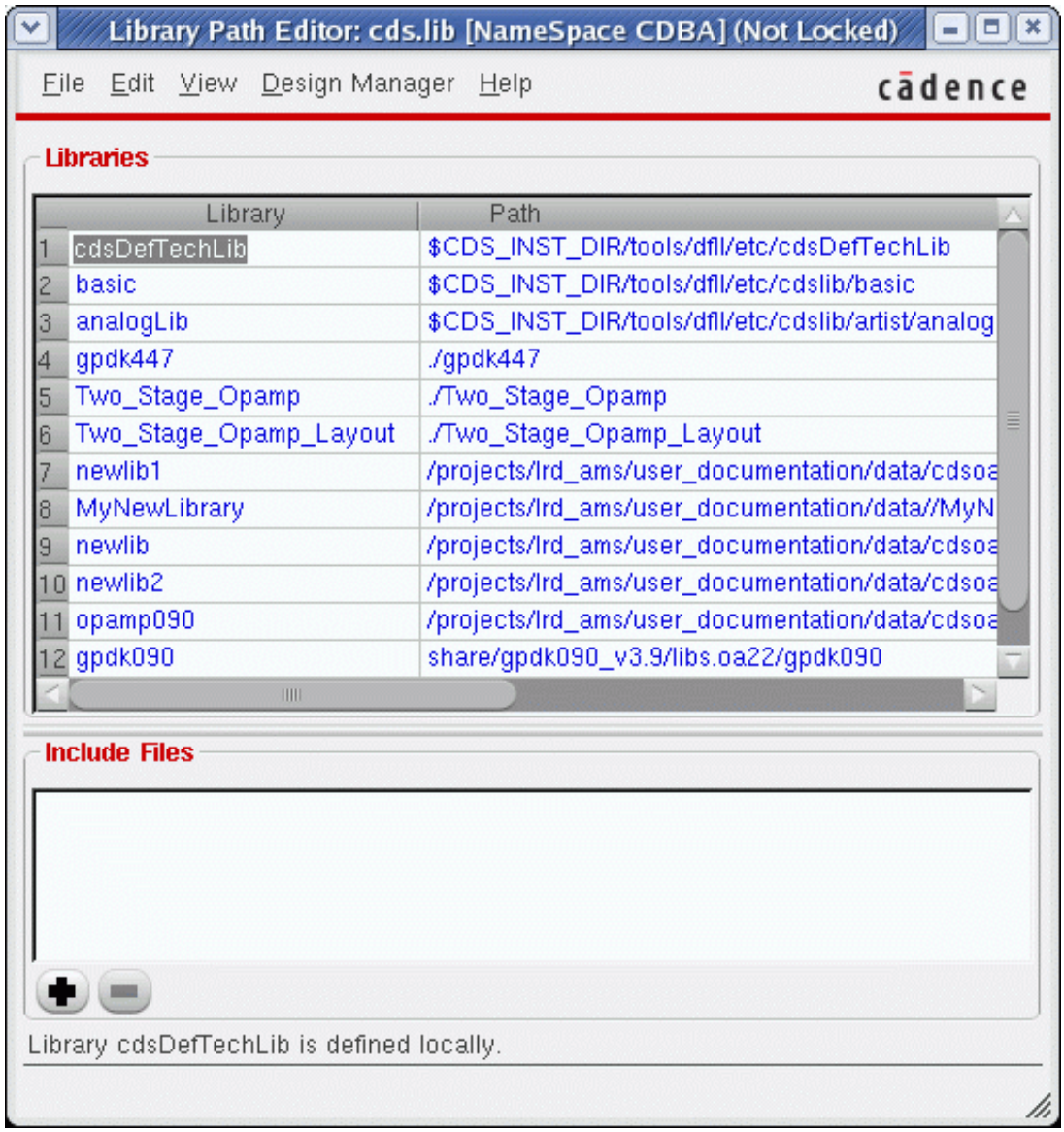
Note: The library data itself is not being removed here (from the file system), rather its entry in the `cds.lib` is being removed. This action will however prevent the library from being accessed from other tools that use the same `cds.lib`. Other tools, or sessions, can continue to use the same library data if they are using a different definitions file, or if the `cds.lib` is re-edited to again have access to the library data.

Including an Existing Library Definitions File

To include another library definitions file in your library definitions file,

1. Choose *View – Include Files*.

   An *Include Files* list box appears at the bottom of the library path editor window, which displays the names of all library definitions files currently referenced by the library path editor.



2. Click the plus (+) sign to open the Include cds.lib File form.

3. In the form, specify the library definitions file that you want to include by selecting it or by typing its path in the *File name* field or by a combination of the two.

   Note: The file you specify must be a regular ASCII file.

4. Click *Open*.

   The name of the file appears in the *Include Files* list box in the library path editor window. The libraries defined in the file appear in the *Libraries* list.

5. Choose *File – Save As* (see also Locking the File for Editing).

   Ensure that the `cds.lib`file has a check mark next to it in the Save As form.

   - Click *OK*.
   - Click *Yes* in the dialog box to overwrite the existing files.

Deleting Included Files

To delete an included file from the library path editor and from your library definition files,

1. Choose *View – Include Files*.
2. In the *Include Files* list box, click a path.
3. Click the minus sign.

   The path to the included file is removed from the *Include Files* list box and the names and paths of libraries in the included file are removed from the *Library* and *Path* list boxes.

4. Choose *File – Save As* (see also Locking the File for Editing).

   Ensure that the `cds.lib`file has a check mark next to it in the Save As form.

   - Click *OK*.
   - Click *Yes* in the dialog box to overwrite the existing files.

     The `include` statement is removed from the library definitions file; the included file is unaffected.

Note: The included file itself is not being removed here (from the file system), rather it is the included entry..

Locking the File for Editing

You can lock the `cds.lib` file that is currently open in the library path editor to prevent other users from modifying it while you are editing it.

To lock the file,

➤ Choose *Edit – Exclusive Lock*.

The file is locked. The window banner indicates that it is locked. Also, the *File – Save* command is now available (this command is usually grayed-out when the file is not locked and you need to use *File – Save As* instead.)

To release the lock,

➤ Choose *Edit – Exclusive Lock* again.

Important

File – Save is not used in the documentation examples included in this manual as its action requires a lock. Locking a `cds.lib` file can however confuse other Cadence applications that have not been designed to handle, or are aware of, any file modifications directly. For example, the creation, deletion or renaming of a library requires re-write access to the `cds.lib`, and the operation will likely fail if the `cds.lib` is locked by an active Library Path Editor session over a long period.

To enable the new transient edit lock mode, add the following environment variable in the .cdsenv file:

```
cdsLibEditor.main       transientSaveLocks       boolean       t
```

Using this environment variable, the file is locked transiently only during the Save process.

However, if the above environment variable is set to nil, the Save command will be disabled until the exclusive lock is obtained.

```
cdsLibEditor.main       transientSaveLocks       boolean       nil
```

Default Value: `t`

To remove the exclusive lock, add the following environment variable in the .cdsenv file:

```
cdsLibEditor.main       autoExclEdit       boolean       nil
```

To set the exclusive lock by default, add the following environment variable in the .cdsenv file:

```
cdsLibEditor.main      autoExclEdit      boolean      t
```

It is not recommended to set the autoExcl lock to t because other Cadence programs malfunction when the cds.lib file is locked.

Default value: `nil`

To turn off the popup messages, add the following environment variable in the .cdsenv file:

```
cdsLibEditor.main      warnExclLock      boolean      nil
```

Working with Library Definitions Files

Displaying Duplicate Library Entries

If you use `include` statements, you might have the same library defined multiple times in your library definition files, which will cause some applications to generate error or warning messages. Because inclusion can be nested, it is simple to accidentally set up the system to define a library more than once. For example:

```
INCLUDE locationA/global.libs
INCLUDE locationB/global.libs
```

Every library will be defined at least twice in the above example.

To display duplicate entries,

➤ Choose *View – Duplicate Entries* in the library path editor.

The duplicate entries are displayed in the library path editor window. You can use this feature to trace multiple entries to their sources.

Note: This feature does not remove the multiple entry. See "Deleting Included Files" on page 27 to remove a library definition.

Listing Locally Defined Libraries

To list only locally defined libraries in the *Library* column in the Library Path Editor,

➤ Choose *View – Local Defines Only*.

The names of libraries listed in any included files are not displayed in the *Library* list box in the library path editor window.

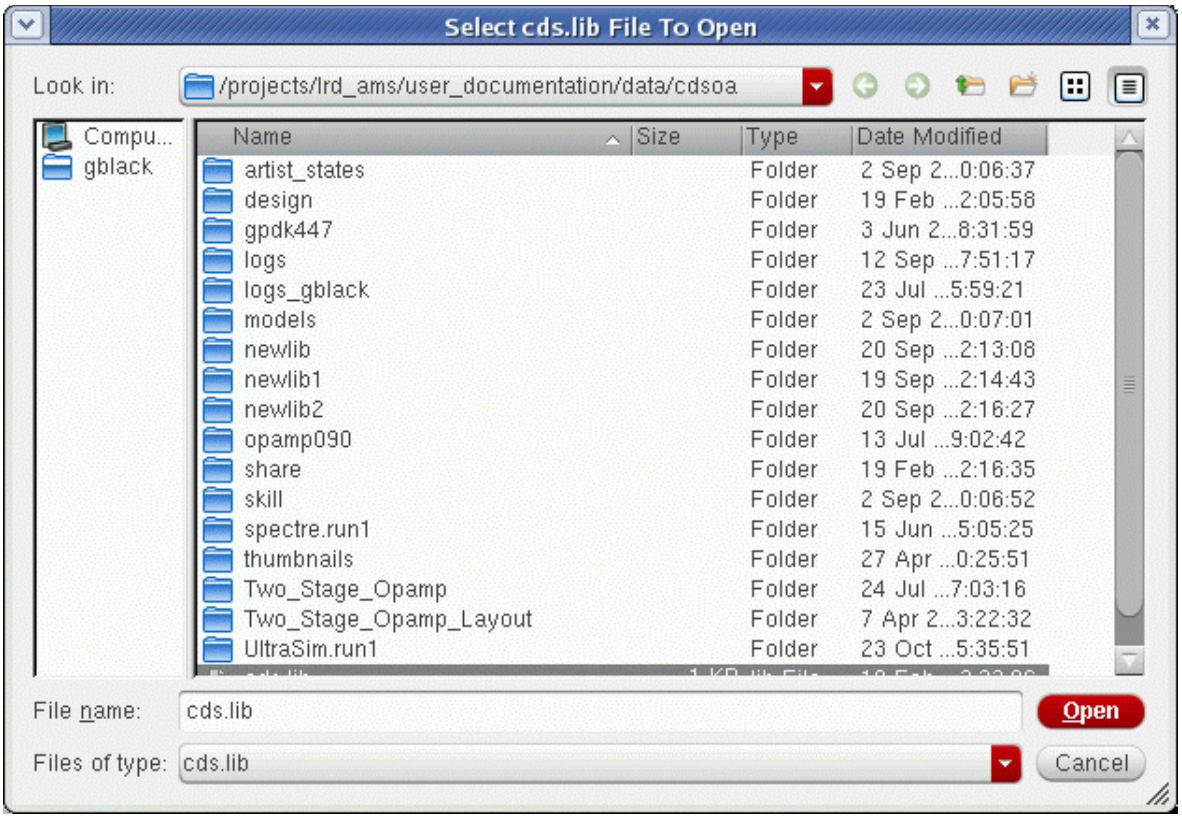
Opening and Using Multiple Library Definitions Files

To display another `cds.lib` file in the library path editor window,

1. Choose *File – Open*.

   The `cds.lib` files in your working directory are displayed. These are the default selections. To look in another directory for a different `cds.lib`, click the *Browse* button.

   The Select cds.lib File To Open form is displayed.



2. In the form, specify the library definitions file that you want to include by selecting it or by typing its path in the *File name* field or by a combination of the two.

   You can use the *Files of type* field to filter files—you can choose to display only library definition files in the selected directory or all files.

3. Click *Open*.

   The file you choose appears in the Library Path Editor window. This file is used in your design session.

If you do not have permission to write to the library definitions file, a message appears, indicating that the file you specified is in read-only mode and you cannot make any changes to it.

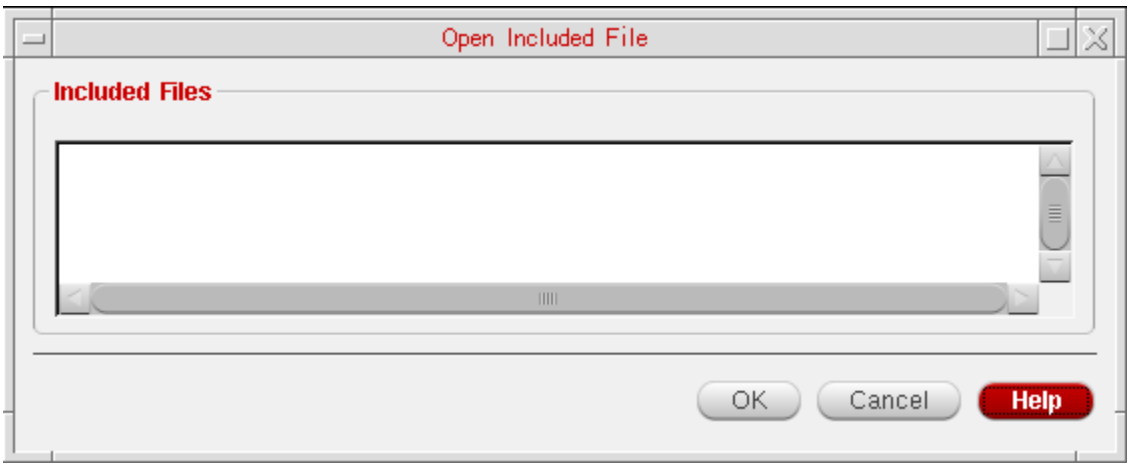
Using Included Files

You can open included `cds.lib` files for use in your design. Included files are other library definitions files that have been included in your library definitions files.

To open an included library definitions file in the library path editor,

1. Choose *File – Open Include*.

   The Open Included File form lists the files included in your `cds.lib` file.



2. Select an included file.
3. Click *OK*.

   The file you select appears in the Library Path Editor window as the current library definition file.

Switching Between Library Definitions Files

Every library definition file that you have called up in a design session is available for editing during that design session.

To edit a `cds.lib` file that you have viewed previously in a design session,

1. Choose *File – File History*.

   The `cds.lib` file you have called up in the current design session appear in a submenu.

2. Select the file you want to use.

   The Library Path Editor window changes to display the file you select.

Viewing Full Paths

To display the full path of libraries in the Library Path Editor window,

➤ Choose *View – Full Paths*.

The paths from the `cds.lib` files are expanded to absolute paths and displayed.

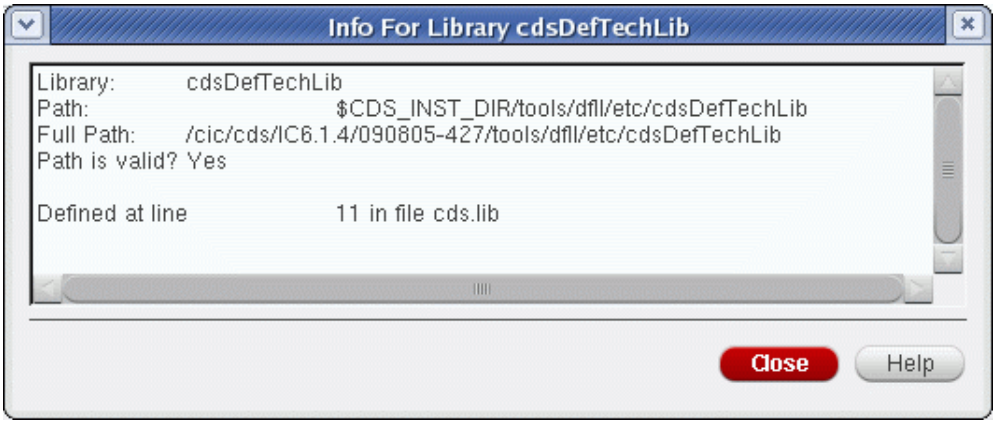
This is a toggle command; choose *View – Full Paths* again to deselect the command.

Finding Library Path information

To locate information about a library name or library path,

1. Click the library name or path you want information on.
2. Choose *View – Library Info*.

   The Info For Library form appears, and shows the library name and path, whether the path is correct, and the line in the library definitions file where the library is defined.



3. Click *Close* to exit the form.

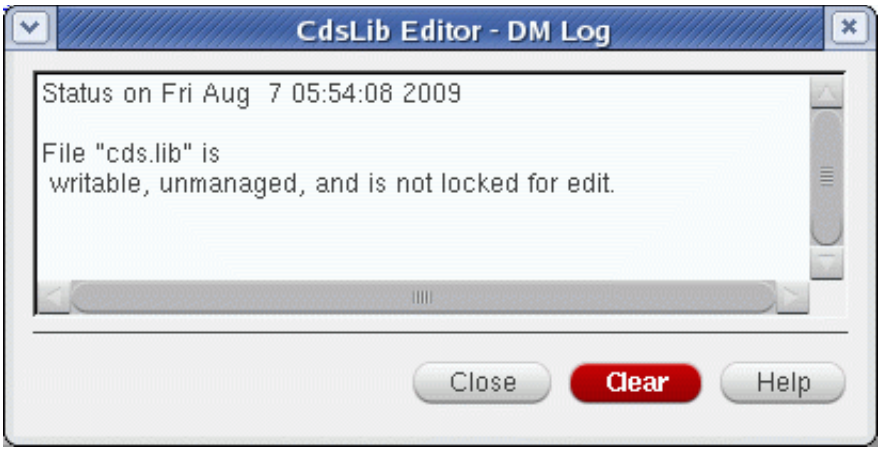
Viewing File Status

You can view the managed status of the `cds.lib` files that are currently open in the Library Path Editor.

To view the file status,

1. Choose *Design Manager – Show File Status*.

   The following form appears:



   The form displays information about each file: whether it is managed by a design management system, whether it is locked, and whether you have write permission for the file.

2. (Optional) Click *Clear* if you want to clear the status information of the files and close the form. By default, the status information is retained and every time you display the form, the new status information is appended to the existing information.

3. Click *Close* to close the form. The status information of the files is retained. The next time you display the form, the new status information is appended to the existing information.

Pop-up Menus

You can access some of the Library Path Editor commands quickly by using the pop-up menus.

To access a pop-up menu,

1. Right-click a path or library field.

2. Click a menu item in the pop-up menu.

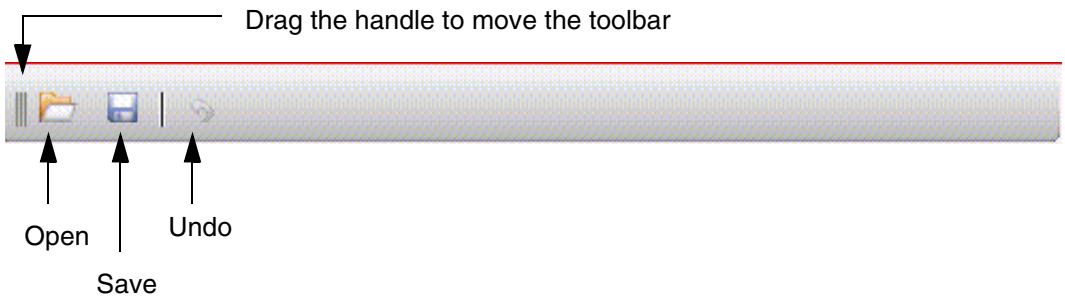
Toolbar

You can access some of the Library Path Editor commands quickly from the toolbar.

To display the toolbar,

➤ Choose *View – Toolbar*.

The toolbar is displayed at the top of the library path editor window.



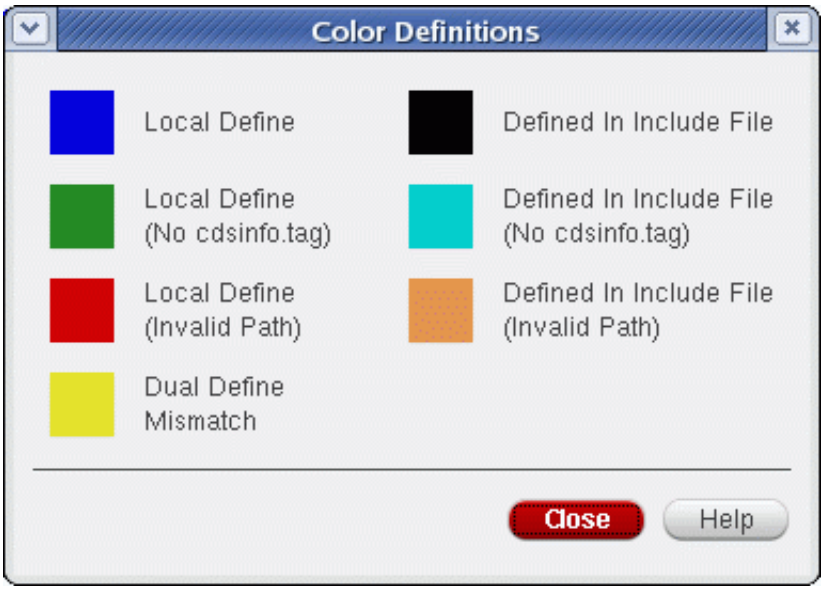
You can move the toolbar to any side of the window.

Color Definitions

The color legend is the key to what the colors represent in the library path editor.

➤ Choose *View – Color Legend*.

The Color Definitions form appears.



Color	Definition
	A valid path to a locally-defined library (a library defined in the current `cds.lib` file)
	A valid path to a locally-defined library with no `cdsinfo.tag` file
	An invalid path to a locally-defined library
	A path found in only one of the two library definitions files indicating that the `lib.defs` and `cds.lib` files are not synchronized
	A valid path to an included library

Color	Definition
	A valid path to an included library with no `cdsinfo.tag` file
	An invalid path to an included library

Error messages

- **Bad Include Path**

  The path to the include file is not valid. An invalid include path ends with an existing directory name or the name of a file that is not a standard ASCII file.

- **Bad Library Path**

  The library path does not exist.

  - Click *OK*.

    The library name and path you specified appear on the Add Library form. You cannot change the name but you can change the path.

  - Click *Apply* or *OK*.

    The new library is created in the directory specified. The library name and path appear in the library path editor window. If, however, you try to create a new library in a directory to which you do not have write permission, or if you type a path rather than a single name in the *Name* field, an error message tells you that the directory could not be created.

- **Directory Doesn't Exist**

  The path you entered does not correspond to an existing directory structure.

  - To continue, click *OK*.

    Specify whether you want to add a library and the library path.

    If you type the name of a library that is already listed in the library definitions file as an included file, the error message tells you that it is removing the definition of that library as an included file and redefining it as a regularly defined file.

- **Invalid Library Name Specified**

  The library name is not valid in the current namespace.

- Click *OK*. Type a valid library name.

  Sometimes, however, the library path editor can and will map the name to a valid name and display `Library Name Has Been Mapped` indicating that it has done so.

  Mixed-case names are sometimes mapped to a lowercase definition. VHDL does allow mixed-case names; you must type the appropriate escape character before and after the name.

- **Invalid Library Path Specified**

  The path is invalid. Paths cannot include spaces (unless enclosed in quotation marks), semicolons, tabs, newlines, or deletes.

  For example, the space in the path `./lib dir/path2` is invalid.

  - Click *OK.* Type a valid path.

- **Library Definition Already Exists**

  The library name has already been defined with a path.

  - Do one of the following:
    - To overwrite the existing path with a new path, click *Yes*.
    - To cancel the entry so you can type different information in the library path editor window, click *No*.

- **No Directory Path**

  You did not specify a library path.

  - To continue, click *OK*.

- **No Library Name**

  You did not specify a library name.

  - To continue, click *OK*.

Keywords and Syntax of Library Definitions Files

For information about the keywords and syntax used in `cds.lib` and `lib.defs` files, see Chapter 5, “Cadence Library Definition File” of the *Cadence Application Infrastructure User Guide*.

Log File

When you run the library path editor, a log file called `cdsLibEditor.log` is created in your current working directory. This file records the commands used in your library path editor session.

A

Form Descriptions

Library Path Editor Window

The window banner indicates which file is open, the namespace it is in, and whether the file is locked or not.

Library displays libraries listed in the `lib.defs` or `cds.lib` file, or both, if both exist in your design hierarchy or have been explicitly opened with the *File – Open* command.

Path displays the path to the library in the library column.

Use the color legend to interpret the color coding used for the libraries.

Select New File Format Form

This forms creates either or both library definitions files listed in the *Edit Format* section.

Edit Format

Cadence "cds.lib" creates a standard `cds.lib` file.

OpenAccess "lib.defs" creates a standard `lib.defs` file.

OK creates the new file.

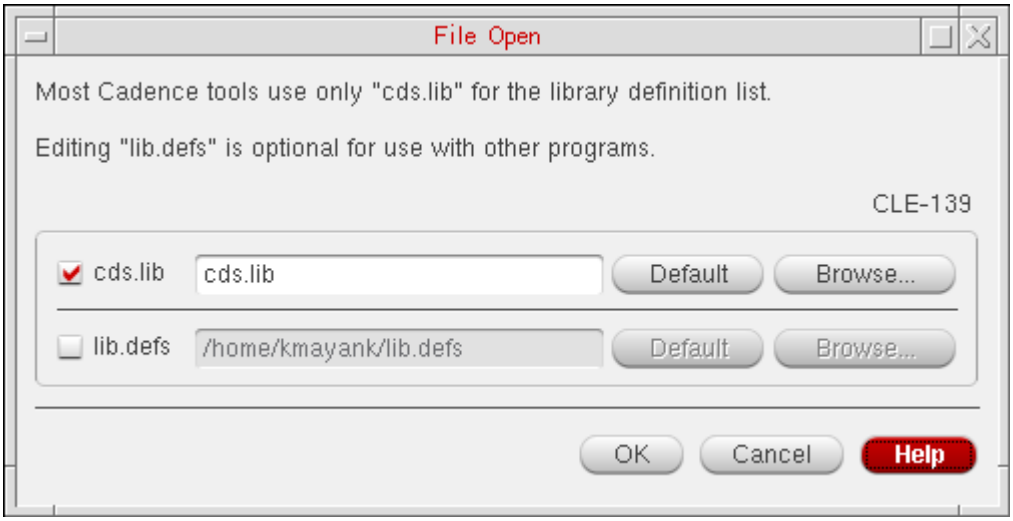
Help open this manual.

Use the *Save As* command to save both files at the same time. Failure to keep these two files synchronized can cause limited interoperability.

See also "Creating Library Definitions Files" on page 16.

File Open Form

The File Open form lets you open both the `cds.lib` and the `lib.defs` files for editing. The contents of both files are displayed in the library path editor window with the paths that are not common to both files highlighted in yellow.



To open one of the two files, deselect the check next to the file name you do not want to open. The file you want to open should have a check next to its name.

Default displays the `cds.lib` or `lib.defs` in the directory from which the library path editor was started. To open a file from another directory, click *Browse*.

Browse opens one of the following forms:

- **Select cds.lib File To Open**
- **Select lib.defs File To Open**

  which let you select `cds.lib` and `lib.defs` files from other locations.

  Look in displays the directory currently selected.

  The list box shows the directories and files available in the directory currently selected (filtered by the value of the *Files of type* field).

  File name lets you type the name (and path) to the file you want to open.

  Files of type filters the files displayed in the list box.

Open puts the path of the selected file in the File Open form.

Cancel cancels your selection and closes the form.

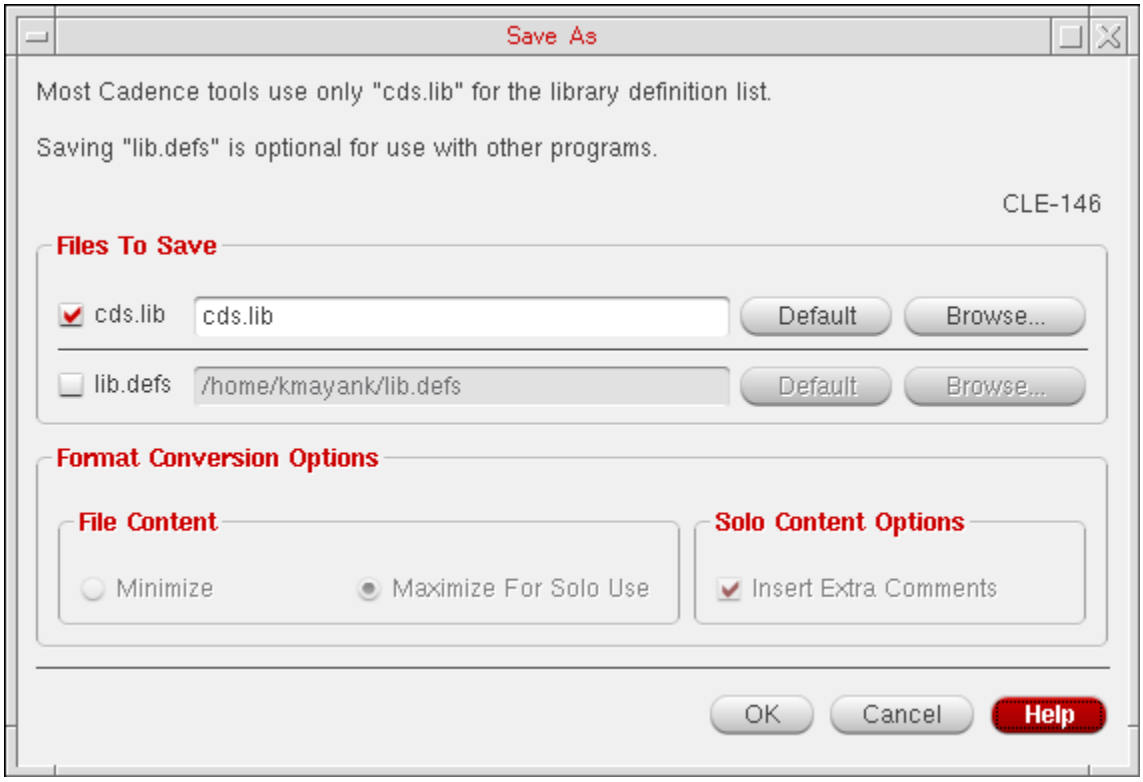
OK opens the selected files.

Cancel cancels your selection and closes the File Open form.

Help opens this manual.

Save As Form

The Save As form lets you save both the `cds.lib` and the `lib.defs` files at the same time. Save As updates both files so they contain the same library path information. Save As removes duplicate entries and adds a date stamp to the ASCII version of both files.



Files to Save

Lets you choose which file to save.To save both files, ensure the checks appear next to both the file names. To save just one of the two files, deselect the check next to the file you do not want to save. The file you want to save should have a check next to its name.

Default displays the `cds.lib` or `lib.defs` in the directory from which the library path editor was started.

Browse opens one of these two forms:

- **Select cds.lib File To Save As**
- **Select lib.defs File To Save As**

Look in displays the directory currently selected.

The list box shows the directories and files available in the directory currently selected (filtered by the value of the *Files of type* field).

File name lets you type the name (and path) to the file you want to save.

Files of type filters the files displayed in the list box.

Save puts the path of the selected file in the Save As form.

Cancel cancels your selection and closes the form.

Format Conversion Options

To save just one of the two files, deselect the check next to the file you do not want to save. The file you want to save should have a check next to its name.

File Content

This option is available when you are saving only one file.

Minimize creates a `cds.lib` or `lib.defs` file from the merging of the two files, taking only the necessary data from both files. Duplicate library paths are removed if the two files contain the same path information, and a date stamp is added for reference. Select this option if you intend to use both files.

Maximize for Solo Use also creates a `cds.lib` or `lib.defs` file from the union of the two files but takes all the information from the files and is therefore potentially larger. The extra data is usually comments. Select this option if you intend to use only the selected file, not both files.

Solo Content Options

Insert Extra Comments adds extra comments to the saved file. This option is used with the *Maximize for Solo Use* option.

OK saves the selected files and closes this form.

Cancel cancels your selections and closes this form.

Help opens this manual.

Add Library Form

Library lets you choose your library name and path.

Name specifies the library name, whether adding an existing library or adding a new library.

Use Mapped Name, when marked, maps a directory name that appears in the *Library* list back to the application's name space. The mapped name is displayed in the *Name* field. If this option is not selected, the directory name is added the way it is.

Note: The directory name must be a valid name for it to be mapped.

For example, a directory name `sample#2elib` is mapped to `sample.lib` when this option is selected.

Directory lets you select the directory that contains the library or the directory into which you want to put the new library (not applicable when using Virtuoso with design management).

Library lists directories containing a valid `cdsinfo.tag` file. (For more information about the `cdsinfo.tag` file, see the *Cadence Application Infrastructure User Guide*.)

Note: The library added is always the library specified by *Name* with the path shown at the bottom of the *Directory* list box.

Design Manager lets you choose your design management setup.

Use <*DesignManagementSystem*> specifies the design management system to use for the library.

Use No DM specifies that the library should not be managed.

Note: If you are not running the software from a design-managed workarea, the *Design Manager* field is grayed-out. If you are running the software from a design-managed workarea, the appropriate design management system is listed as a choice.

OK adds the library and closes this form.

Apply adds the library and leaves this form open.

Cancel cancels your selections and closes this form.

Help opens this manual.