Rational. software

ClearCase-Cadence Design Framework II Integration Reference

1 Introduction

This document describes the commands and features that are supported in ClearCase-Cadence integration. It also describes the user interface for the ClearCase integration with Virtuoso.

2 GDM Integration Commands

2.1 Checkin

You can checkin design artifacts (library, cells, cell views) andnon design artifacts (property file, category file etc.) in ClearCase using the Library Manager GUI or the gdmci command line interface.

2.1.1 Command Line Interface: *gdmci*

gdmci

Usage: gdmci [-cdslib <filename>] [-recurse] [-xtra <str>] [-initial]
[-description <des_str>] [-dfile <file>]
[-help] [-lib <lib.cell:view/file>] [-file <file>]

Description

Checks in the specified files and registers files that were previously unmanaged. Provides checked-out and previously unmanaged files to the repository so that files can be shared. Co-managed files in a view are always checked in as a group. Co-managed set behavior applies only when directories or files are specified as library elements; that is, with the -lib argument.

Options and arguments

-cdslib <filename>
Specifies the library definition file to be used for mapping library names to library directories.

-recurse
If a non library specification is a directory name, by default it refers only to the files immediately below that directory. If the argument -recurse is specified, it selects the entire directory hierarchy.

-xtra <str >

Allows additional arguments, specific to the design management checkin command, to be passed through gdmci to that command.
Note: The -xtra flag takes a string as the parameter; it must be enclosed within "" if there is a space in the string.

The following ClearCase specific flags are supported as part of -xtra. The meaning of these flags is same as defined in the reference page for cleartool checkin.

-nwarn

-keep | -rm

-ptime

-identical

The following -xtra flags are provided for the gdmci -initial operation. The meaning of these flags is same as defined in the reference page for cleartool mkelem.

-eltype <eltype_name>

-master
ClearCase version 7.1.2.11 or 8.0.0.7 onwards, in a multisited environment checkin of unmanaged artifacts will always be performed using -master flag. This behavior can turned off by setting CCASE_CDS_MASTER environment variable to 'false'. Once the behavior is turned off user can explicitly specify -master as an xtra arg to the checkin command and it will be honored. If the feature was turned off previously, user can re-enable it by setting CCASE_CDS_MASTER environment variable to 'true'.

The following ClearCase flags are “not” supported in gdmci -xtra

-comment
The description argument of gdmci becomes the comment

-nc
This is the default.

-cfile
The -cfile argument of gdmci becomes the comment

-atomic
This is the default if the VOB is enabled for atomic checkin operations.

-from
The Cadence tools deal with multiple files at a time when a library, cell, or cellview is checked in. -from is irrelevant in this context.

-initial
Checks in all specified files and registers files that were previously unmanaged. If unspecified, new or unmanaged files that are members of a registered co-managed set are checked in. All arguments are case-insensitive and can be shortened to any abbreviation that is unique across all gdm command arguments. Hence, -initial can be specified as -Ini.

-description <des_str>
Cannot be used with -dfile. The string is the comment for the checkin operation.

-dfile <file >
Cannot be used with -description. Use -dfile to enter a multiline description. The content

of the file becomes the description for the checkin operation.

-help
Displays information about this command and its arguments.

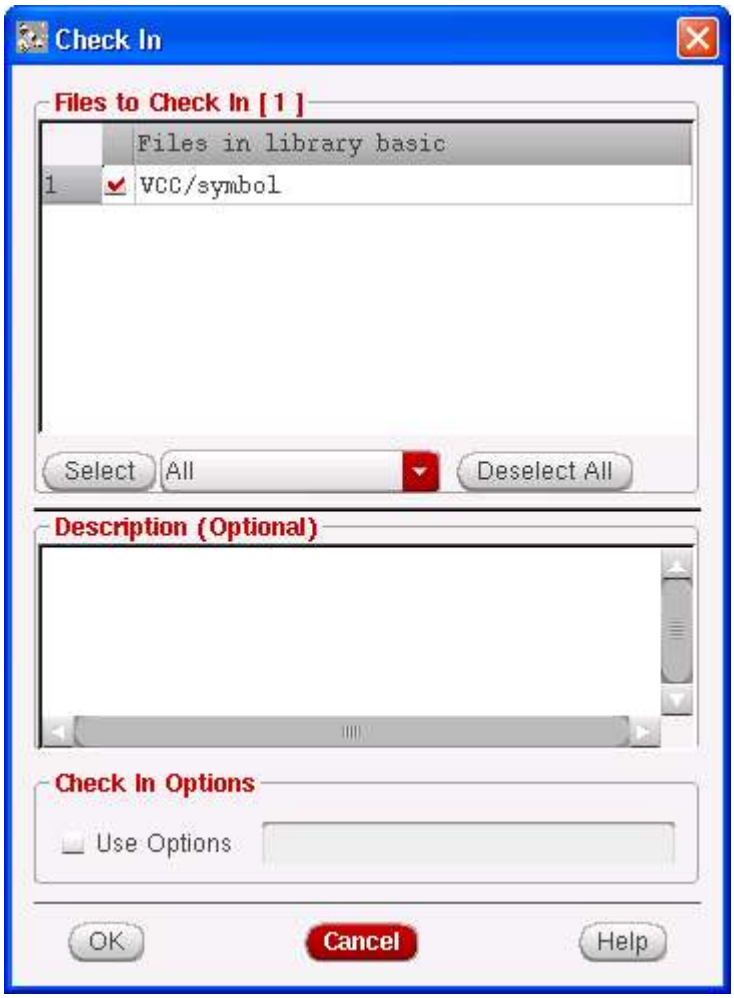
-lib <lib.cell:view/file>
Library elements to check in. Names listed without -lib or -file are treated as -file arguments.

-file <file >
Files and directories (non library elements) to be checked in. Arguments specified without -lib or -file are treated as -file arguments.

2.1.2 Graphical User Interface: Library Manager

Select a file, library, cell or cellview and then
- Click on the Design Manager Menu and then click on Checkin
- Right click on the file/library/cell/view and click on Checkin

A dialog box enables you to select and unselect the files. You can also select "Use Options" to provide ClearCase-specific options: see the description of the -xtra argument.



All checked out files are checked in. If the files are unmanaged, they are added to

ClearCase source control.

2.1.3 Atomic Checkin Support:

The member files of a cellview that constitute a co-managed set are checked in atomically if the VOB has been enabled for atomic checkins (see the reference page for protectvob).

2.1.4 Auto-Checkin

Auto-checkin applies only to the files/designs that were auto-checked out. If a file was checked out explicitly, you are not prompted to check it in when ending the session.

2.1.5 Checkin of unmanaged artifacts in Multisite

ClearCase version 7.1.2.11 or 8.0.0.7 onwards, checkin of unmanaged artifacts will leverage SRFM feature. For replicated vobs which are srfm enabled, checkin operation will now use the 'srfm' option by default to checkout the non mastered parent folders to add unmanaged artifacts underneath them to ClearCase control. This behavior can be turned off by setting CCASE_CDS_SRFM environment variable to 'false'. Set CCASE_CDS_SRFM environment variable to 'true' to re- enable this feature if it was turned off previously.

Non mastered parent folders which were checked-out during the checkin of unmanaged artifacts can be checked-in only after their branch mastership arrives to the current replica. User can opt to poll the checkin of these folders to commit them to ClearCase automatically.

User can set "CCASE_CDS_POLL_CI" environment variable to 'true' to enable polling. Poll interval must be set in seconds using "CCASE_CDS_POLL_CI_INTERVAL" environment variable. Checkin of parents will be attempted after every specified interval until it succeeds or time-outs. Poll time-out must be set in seconds using "CCASE_CDS_POLL_CI_TIMEOUT" environment variable. User can disable polling feature by setting "CCASE_CDS_POLL_CI" environment variable to 'false'.

While the polling feature is ON, if an invalid value was specified for CCASE_CDS_POLL_CI_INTERVAL environment variable it will be defaulted to 30 seconds. If an invalid value was specified for CCASE_CDS_POLL_CI_TIMEOUT it will be defaulted to 90 seconds. If CCASE_CDS_POLL_CI_TIMEOUT was specified less than CCASE_CDS_POLL_CI_INTERVAL, CCASE_CDS_POLL_CI_TIMEOUT will be set to CCASE_CDS_POLL_CI_INTERVAL.

2.1.6 Cancelling checkouts that would result in identical versions if checked in

When set to TRUE, the environment variable CCASE_CDS_CANCEL_IDENTICAL causes a checkin operation to cancel the checkout of a version that would be identical to its predecessor if it were to be checked in. Default value of this environment variable is FALSE.

2.2 Checkout

You can checkout design artifacts (library, cells, cellviews) andnon design artifacts (property files, category files, and so on) using the Library Manager GUI or the gdmco command line.

2.2.1 Command Line Interface: *gdmco*

```
Usage : gdmco [-cdslib <filename>] [-recurse] [-xtra <str>]
                [-version <version>] [-help]
```

[-lib <lib.cell:view/file>] [-file <file>] ...

Description

Checks out the specified files. Co-managed files are checked out in the same grouping in which they were checked in. Co-managed set behavior applies only when directories or files are specified as library elements, that is, with the -lib argument.

Options and arguments

-cdslib <filename>
Specifies the library definition file to be used for mapping library names to library directories.

-recurse
If a non library specification is a directory name, then by default it refers only to the files immediately below that directory. If -recurse is specified, it selects the entire directory hierarchy.

-xtra <str>
Allows additional arguments that are specific to the design management checkout command, to be passed through gdmco to the checkout command.
Note: The -xtra flag takes a string as the parameter; it must be enclosed within "" if there is a space in the string.

The following ClearCase-specific flags are supported by -xtra. The meaning of these flags is same as defined in ClearCase reference pages.

-unreserved [-nmaster]
Unreserved checkouts can be checked in only when there is no reserved checkout of the element on the branch and no new version of this element is created on the branch after the unreserved checkout.

In a replicated environment, to perform an unreserved checkout using -nmaster, the environment variable CCASE_CDS_SRFM must be set to false if the VOB is enabled for synchronous request for mastership (SRFM). Refer section 2.2.3 for more details.

-nwarn

-ptime

-comment
The comment string must be enclosed within \"<comment-string>\"; for example, -xtra "-comment \"Hello World\""

-cfile

-query

-nquery

The following ClearCase options are not supported by gdmco -xtra

-cq

-cqe

-nc
This is the default for gdmco.

-ndata
If specified, this option would result in the "checked out but removed" state, which is not handled by the integration.

-out
This option is irrelevant in the context of multiple files (Cadence tools deal with multiple files at a time).

-version
Redundant with respect to gdmco -version.

-usehijack

-version
Allows the checkout of a version that is not the latest on its branch. Checking out a version on different branch is not supported by the integration.

-lib <lib.cell:view/file>
Library elements to check out. Names listed without -lib or - file are treated as -file arguments.

-file <file>
Files and directories (non library elements) to check out. Names listed without -lib or -file are treated as -file arguments.

2.2.2 Graphical User Interface: Library Manager

Select a file, library, cell or cellview and then

- Click on the Design Manager Menu and then click on Checkout
- Right click on the file/library/cell/view and click on Checkout

A dialog box enables you to select and unselect the files. You can also select "Use Options" to specify ClearCase options (see the description of the -xtra option).

All the checked in files are checked out.

2.2.3 Checkout in Multisite

V1.1 and later versions of the ClearCase-Cadence integration support synchronous request for mastership (SRFM). SRFM is supported in ClearCase V7.1.2.2 and later releases. The VOB must be SRFM-enabled (refer to the reference page for protectvob). You can disable the default "-srfm" flag by setting the environment variable CCASE_CDS_SRFM to false or FALSE.

If the VOB is not SRFM-enabled or the environment variable CASE_CDS_SRFM is set to false, you must acquire mastership of the branch before the checkout operation can proceed.

The following -xtra option is supported:

-reqmaster
Initiates a request for mastership. For example, gdmco –xtra –reqmaster –lib basic.vcc:symbol will do the following

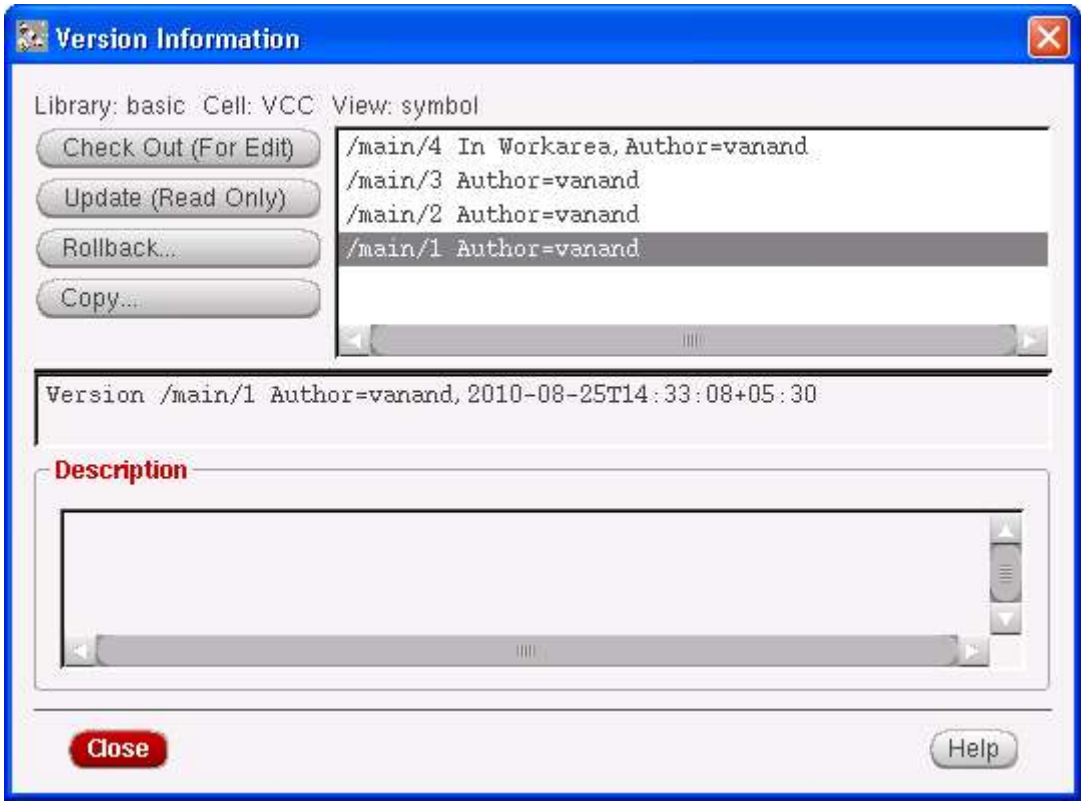
- If the files in the co-managed set are mastered locally, then checkout proceeds.
- If the files are mastered remotely, a request for mastership is issued. The checkout operation will succeed after branch mastership is acquired.

2.2.4 Checkout -version

To checkout a non latest version,

- Open the library manager.
- Select the cellview or files that are to be copied.
- Select Design Manager – Version Info.

The Version Information form appears.



- Select the version of the cellview that you want to check out.
- Click Check Out (For Edit).

Note that the version should be on the same branch as the view selected version. Because the checkout is from a non latest version, ClearCase prohibits a check-in if there is no merge arrow from latest version to the checked out version. A merge arrow is drawn from the latest version to the checked-out version. However, there is no merge; the merge arrow simply allows the version to be checked-in.

2.3 Cancelling Checkouts

You can cancel checkouts of design artifacts (library, cells, cellviews) andnon design artifacts (property file, category file, and so on) using the Library Manager GUI or the gdmco command.

2.3.1 Command Line Interface: *gdmcancel*

Usage: gdmcancel [-cdslib <filename>] [-recurse]
[-xtra <str>] [-help]
[-lib <lib.cell:view/file>] [-file <file>] ...

Description
Cancels the checked-out status of files in the workarea. Co-managed files are always cancelled as a group. Co-managed set behavior applies only to a view that is specified as a library entry, such as -lib lib.cell:view.

Arguments

-cdslib <filename>
Specifies the library definition file to be used for mapping library names to library directories.

-recurse
If a non library specification is a directory name, then by default it refers only to the files immediately below that directory. If -recurse is specified, it selects the entire directory hierarchy.

-xtra <str>
Enables additional arguments that are specific to the design management cancel command to be passed through gdmcancel to that command.

The following ClearCase specific flags are supported as part of -xtra. The meaning of these flags is same as defined in ClearCase manual.

-keep

-nsrfm

The following ClearCase flags supported by cleartool uncheckout command are not supported in gdmcancel -xtra

-rm
Redundant (this is the default).

-lib <lib.cell:view/file>
Library elements for which to cancel checkout. Names listed without -lib or -file are treated as -file arguments.

-file <file>
Files and directories (non library elements) for which to cancel checkout. Names listed without -lib or -file are treated as -file arguments.

2.3.2 Graphical User Interface: Library Manager

Select a file, library, cell or cellview and then
- Click on the Design Manager Menu and then click on Cancel Checkout
- Right click on the file/library/cell/view and click on Cancel Checkout

Use the dialog box to select and unselect the files. Select "Use Options" to specify ClearCase-specific options.

2.4 Auto Checkin/Checkout

By default, when you open properties, files, or cellviews that are not checked out, the integration software checks out the artifacts and prompts you to confirm the check-out by displaying the Auto Checkout form.

By default, properties or files that were checked out are closed, or the user tries to exit a session without closing properties or design files that were automatically checked out, the software begins an automatic check-in process and prompts you to confirm the automatic check-in by displaying the Auto Checkin form.

You can configure Auto Checkin/Checkout behavior using Virtuoso.

2.5 Deleting artifacts

You can delete design artifacts (library, cells, cellviews) andnon design artifacts(property file, category file, and so on) in ClearCase using the Library Manager GUI or the gdmdelete command.

2.5.1 Command Line Interface: *gdmdelete*

Usage: gdmdelete [-cdslib file] [-xtra str]
[-local] [-help][-keepunmanaged]
[-lib lib.cell:view/file] [-file file]

Description

Deletes managed and unmanaged files from the work area. The element is not deleted from the VOB but removed from the current branch. The other branches can still see and operate on the artifacts. The history of the element is preserved.

[-cdslib file]
Specifies the library definition file to be used for mapping library names to library directories.

[-local]
This option is not supported in the integration. A delete operation always removes the elements for all the users who are working on the branch on which the delete operation was performed.

[-help]
Displays information about this command and its arguments.

[-keepunmanaged]
Filters unmanaged files from the deletion list. For example, if cell1 contains cellviews cv1, cv2 which are managed, and cv3, which is unmanaged, then the command **gdmdelete –keepunmanaged –lib lib.cell1** will delete cv1 and cv2 only.

[-lib lib.cell:view/file]
Library elements to delete. Names listed without -lib or –file are treated as -file arguments.

[-file file]
Files and directories (non library elements) to delete. Names listed without -lib or -file are treated as -file arguments.

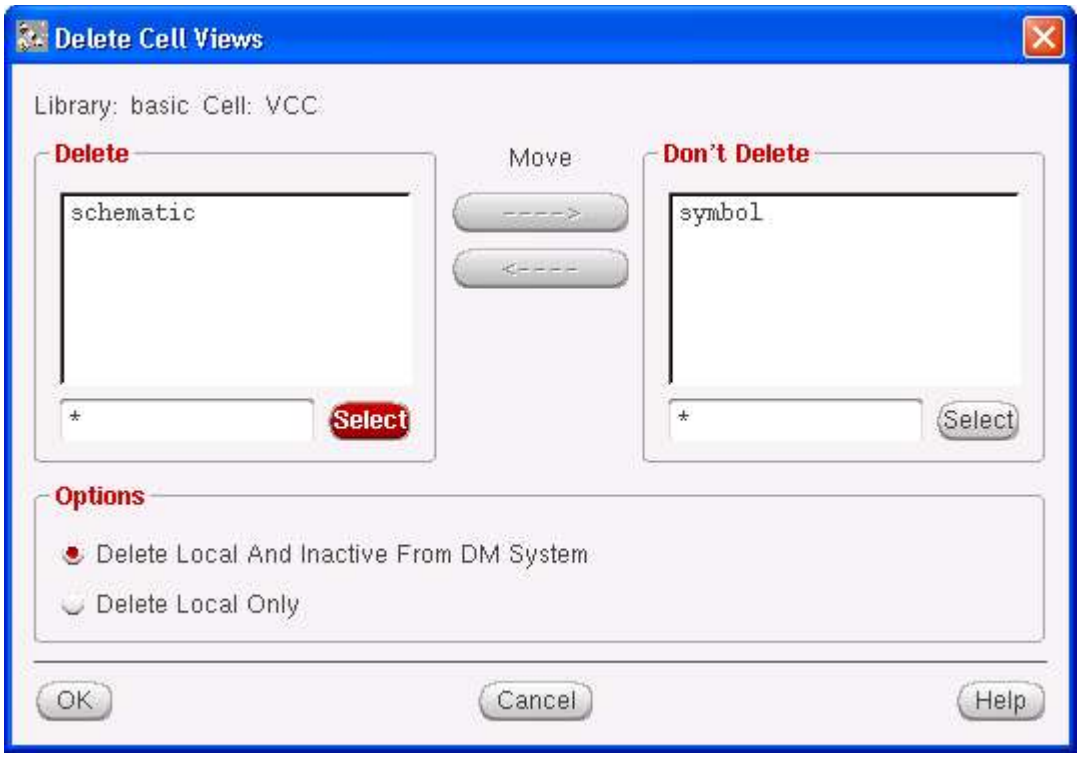
[-xtra str]
Allows additional arguments, specific to the design management (ClearCase) delete command, to be passed through gdmdelete command.

Supported xtra arguments:
By default, a deletion comment is created. You can specify a comment option to override the default comment.

–c/omment *comment* | –cfi/le *comment-file-pname* | –nc/omment

2.5.2 Graphical User Interface: Library Manager

You can delete managed and unmanaged design artifacts using the “Delete...” menus item that is available on the library, cell, cellview, and file nodes. Use the dialog to specify the artifacts that are to be deleted. Both the options “Delete Local and Inactive from DM system” and “Delete Local Only” are treated similarly and perform the same operation.



2.5.3 Known Issues

A delete operation that is issued on a library deletes all of its contents, leaving an empty library directory. However if the delete operation is attempted from libManager on the library, the library node will no longer be visible in the libManager GUI.
Also note that Cadence does not process an empty cell, displaying the message that the cell is empty and cannot be deleted. To delete the orphan cell, create a temporary cellview or files in the cell and then delete the cell.

2.6 Querying Status for Managed/Unmanaged Design Artifacts

2.6.1 Command Line Interface: *gdmstatus*

Usage: gdmstatus [-cdslib <filename>] [-lib <lib.cell:view/file>]
[-file <file>] [-xtra <str>] [-workarea] [-repository]
[-civersion] [-coversion] [-updateversion] [-status]
[-header] [-absolute] [-modified] [-where] [-recurse]

Description

Returns the design management status of files. If no file is present, the command operates on the current directory with -recurse.

-cdslib <filename>
Specifies the library definition file to be used for mapping library names to library directories.

-lib <lib.cell:view/file>
Specifies the library elements. Names listed without -lib or -file are treated as -file arguments.

-file <file>
Specifies files and directories (non library elements) about which status is requested. Names listed without -lib or -file are treated as -file arguments.

-xtra <str>
Not supported.

[-civersion]
Shows the checked in version. If the file is checked out, –civersion prints the version that will be created after checkin.

[-coversion]
Shows the checked out version. If the file is checked out, –coversion prints the version that was checked out. If the file is checked in, –coversion prints the checked-in version.

[-status]
Shows the DM status.

[-header]
Shows the header for every column that is displayed.

[-absolute]
Output shows absolute pathname. (By default, path the is relative to the current directory or is shown as a library specification.

[-modified]
A filename is annotated with an asterisk (*) if it is modified or by a question mark (?) if its modification status is unknown. All checked out files are annotated with asterisks when their status is queried using this option, indicating the files are modified or ready for modification.

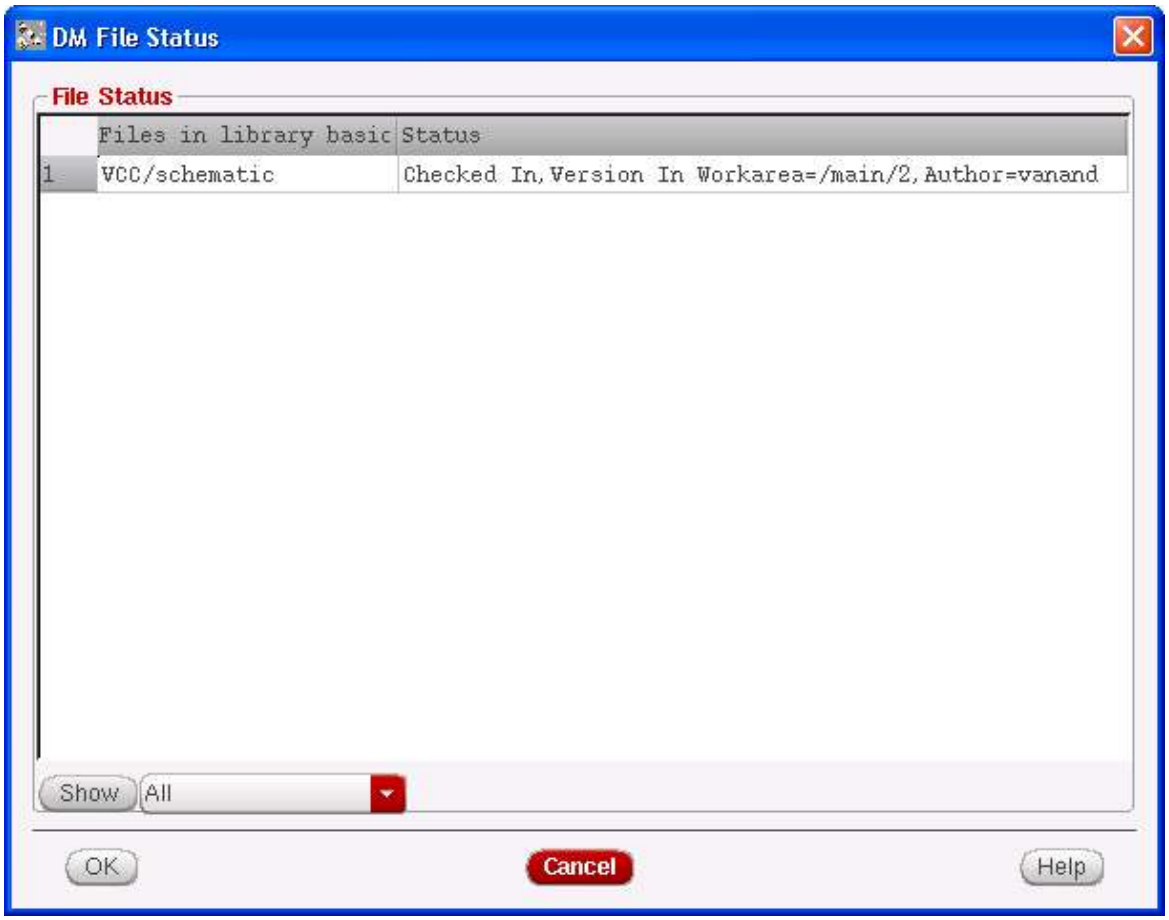
[-where]
Displays the checkout location and the user who performed the checkout. If the checkout location is returned, it is displayed; otherwise, the user is displayed.

[-recurse]
If a non library specification is a directory name, by default it refers only to the files immediately below that directory. If the argument -recurse is specified, it selects the entire directory hierarchy.

[-help]
Displays information about this command and its arguments.

2.6.2 Graphical User Interface: Library Manager

You can obtain the status of a library, cell, orcellview using the “Show file status...” context menu item.



2.7 *Export*

You can export a specified version of a co-managed set of operated cellview to the destination folder. Command is applicable only to single managed cellview.

2.7.1 Command Line Interface: *gdmexport*

Usage: gdmexport [-cdslib file] [-recurse] [-xtra str]
[-lib lib.cell:view/file | -file file] ... -destination path
[-version identifier] [-complete] [-exportpaths] [-help]

Description:
Export the specified version of co-managed set of operated cellview to the destination folder. This command is applicable only to single managed cellview.

Options and Arguments

[-cdslib file]
Specifies the library definition file to be used for mapping library names to library directories.

-destination path
The location to which the co-managed set of cellview is to be exported.

[-lib lib.cell:view]
cellview to be exported.

[-file file]
Not applicable.

[-version identifier]
A string representing the version of the files to be exported

[recurse]
Not applicable.

[-xtra *str*]
Not applicable.

[-complete]
Not applicable.

[-exportpaths]
Not applicable.

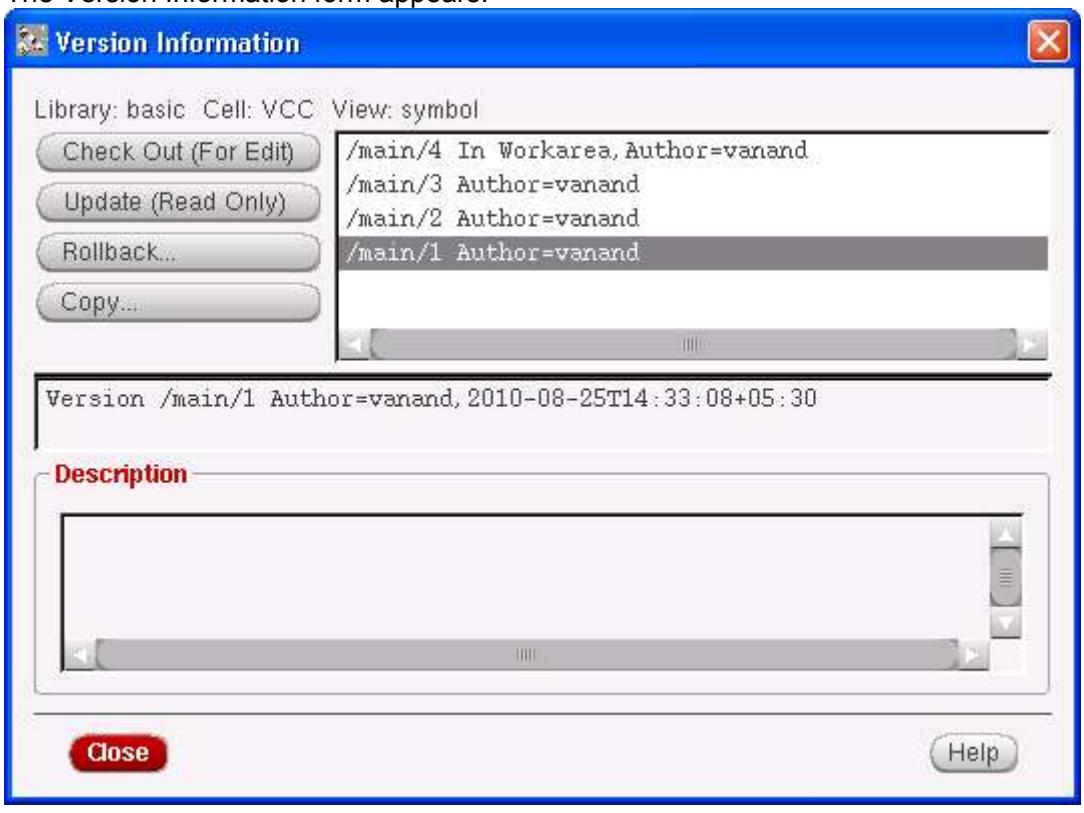
[-help]
Displays information about this command and its arguments.

2.7.2 Graphical User Interface: Library Manager

To copy a version of a file,

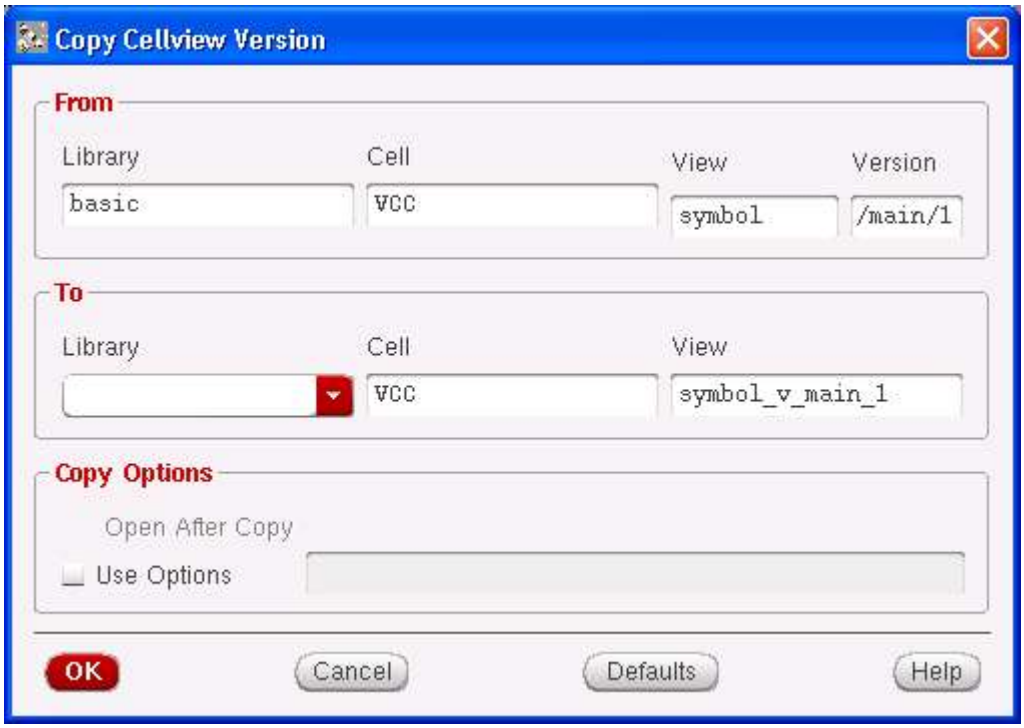
- Open the library manager. The Library Manager form appears.
- Select the cellview/files which is required to be copied.
- Choose Design Manager – Version Info.

The Version Information form appears.



- In the list box, choose the version of the cellview that you want to copy.
- Click Copy.

The Copy Cellview Version form appears.



The From fields default to the name of the source library, cell, and view, and the cellview version number. The To fields default to the same cell name, and the same view name appended with an incremented version number, as the destination (copied) cellview version.

- Type the name of the library you want to copy the cellview version to.
- Select Use Options to pass in options specific to your particular design management system.

2.8 Rollback

You can rollback to any version which is non latest on the current branch. This operation creates a new version in the current branch which is the same branch to which you have rolled back.

2.8.1 Command Line Interface: *gdmsetdefver*

Usage: gdmsetdefver -version version [-cdslib file] [-xtra str] [-name tag] [-help]
[-lib lib.cell:view/file] [-file file]...

[-cdslib file]
Specifies the library definition file to be used for mapping library names to library directories.

[-lib lib.cell:view]
The cellview on which rollback is to be performed.

[-file *file*]
Files and directories (non library elements) to be rolled back. Names listed without -lib or -file are treated as -file arguments.

[-version version]
A string representing the version to which elements will be rolled back.

[-name tag]
Not Applicable.

[-xtra *str*]
Not Applicable.

[-help]
Displays information about this command and its arguments.

2.8.2 Graphical User Interface: Library Manager

To specify an earlier version as the latest version on the branch:

- Open the library manager. The Library Manager form appears.
- Select the cellview/file.
- Choose Design Manager – Version Info.
  The Version Information form appears.
- In the list box, select the earlier version. The software enables the appropriate option buttons.
- Click Rollback. The software prompts you to confirm the rollback operation.
- Click Yes.

Note that the rollback operation is equivalent to issuing the following two commands:
gdmco -version
gdmci -xtra -identical

The operation creates a new version on the branch, the content of which is same as the selected version.

2.9 History

2.9.1 Command Line Interface: *gdmhistory*

Usage: gdmhistory [-cdslib file] [-lib lib.cell:view/file] [-file file] [-xtra str]
[-full] [-author] [-size] [-last nn] [-header] [-status]

Description
Returns information about the version history of a file. The library and file arguments must specify a single file only. This command is applicable to versioned elements only.

Options and arguments

-cdslib <filename>
Specifies the library definition file to be used for mapping library names to library directories.

-lib <lib.cell:view/file>
Specifies library elements. Names listed without -lib or -file are treated as -file

arguments.

-file <filename>
Files and directories (non library elements) for which to get their histories. Names listed without -lib or -file are treated as -file arguments.

-xtra str
Not Applicable.

-full
Full description is printed on separate lines from the other information.

-author
Name of the author of the file.

-size
Size of the file. **Known Issue**: Size is always displayed as 0.

-last nn
Takes a numeric option. Prints each version number and date and the first several characters of the history up to the given number of versions.

-header
Header of the history output.

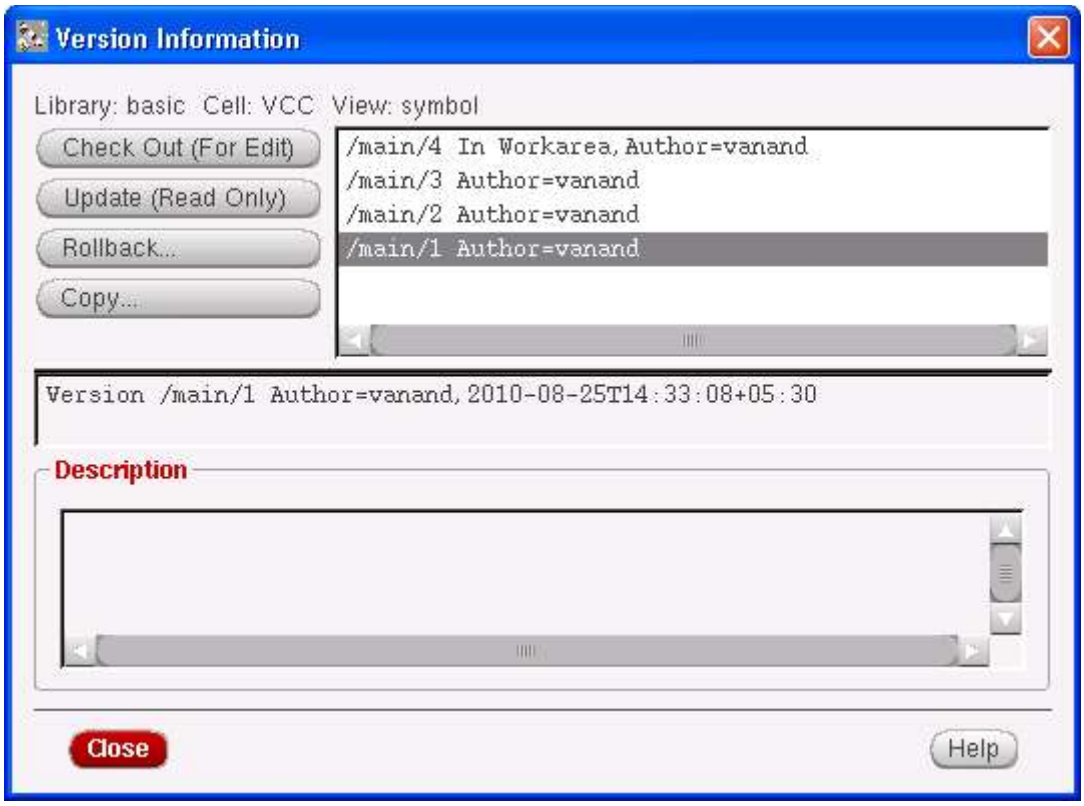
-status
Status of the file.

2.9.2 Graphical User Interface: Library Manager

To see the version history of a file,

- Open the library manager. The Library Manager form appears.
- Select the cellview/file.
- Choose Design Manager – Version Info.

The Version history Information appears.



2.10 Restricting ClearCase operation on design elements to Cadence interfaces

The integration interfaces set an environment variable CLEARCASE_EDA to 'Cadence Virtuoso' upon their invocation which can be leveraged by ClearCase pre-op triggers to identify the process which is going to execute a ClearCase command thereby restricting the execution only to Cadence Virtuoso if required. These triggers can be written by ClearCase site administrator.

2.11 Logging and Tracing

Apart from the Cadence logging in CDS.log and libManager.log, you can set following environment variables to control the output and to enable tracing and logging for the integration.

CCASE_CDS_VERBOSITY

By default all the output is printed on the standard output for the CLI and on the Library Manager console for the GUI. CCASE_CDS_VERBOSITY can be set to the following values :

0 : Error messages only

1 : Error and warning messages

2 : (Default) OK, warning, and error messages

3 : OK, error, warning, and trace messages; with trace messages, the output is extremely verbose.

A log file is created with name ccase_cds.log.<timestamp> in the current directory. One log file is created for one session of Library Manager. For every gdm command, a separate log file is created.

CCASE_CDS_LOG_DIR

If the value of CCASE_CDS_VERBOSITY is 3 then by default the log file is created in the current working directory. Use the environment variable CCASE_CDS_LOG_DIR to specify a different directory to which the log file is to be written.

3 Custom ClearCase GUI Documentation

This section describes the custom user interface for the ClearCase integration into Virtuoso.

3.1 Integration with IC51

3.1.1 How to start the user interface

The user interface is invoked via the Cadence Library manager. To browse the list of all version controlled libraries, choose the Design Manager menu and there the menu item ClearCase WA Manager.

The ClearCase Work Area Manager is displayed, listing the libraries (Fig. 3.1.1). To browse the design hierarchy of a single cellview, select that cellview in the library manager and right click to show the context menu. Select “Browse hierarchy” to open the design hierarchy browser for the selected cellview (Fig. 3.2.2). To browse a single library, select that library in the library manager and right click it to show the context menu. Select “Browse library” to browse that library only.

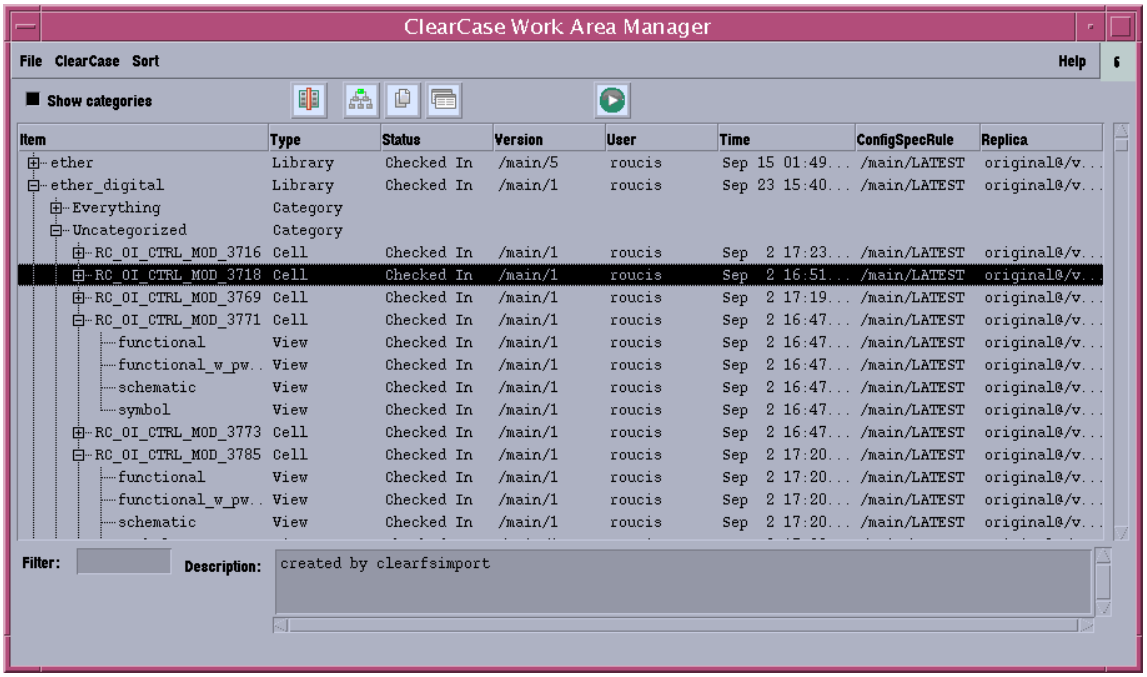


Figure 3.1.1: The ClearCase Work Area Manager browsing libraries

3.1.2 User interface components

The main components of the user interface are:

1. The menu bar
    1. File
        1. Flatten library

            This opens the selected library in a new form, all cellviews in the library in a flat hierarchy
        2. Filter

Opens the Filter form. Only views which match the filter criterion with their name are displayed, together with their corresponding cells and libraries.

3. Edit configspec

   Opens the editor defined by the EDITOR environment variable to edit the active configspec.

4. Close

   Closes the current form

2. ClearCase
   1. Check in

      Check in the selected items

   2. Check out

      Check out the selected items

   3. Cancel

      Cancel the checkout of the selected items

   4. Delete

      Delete the selected items

   5. Rename

      Rename the selected item

   6. Label

      Attach a label to the selected items

   7. Refresh selected

      Refresh the selected items

   8. Refresh all

      Refresh all items displayed

   9. Checkin checked out

      Check in all items currently checked out

   10. Version browser

       Open the ClearCase version browser for the selected item

   11. History

       Open the ClearCase history browser for the selected item

   12. Label type browser

       Open the ClearCase 'Label Type Browser' for locking, locking except some users, or unlocking of label types.

3. Sort
   1. By Name
   2. By Time
   3. By Status
   4. By Version
   5. By User

   Sort the displayed items by the corresponding column. Selecting the same sort criterion again reverses the sort order. To be able to sort cellviews globally, it is recommended to open the flatten library view of the library you are working on and sort there.

2. The tool bar

   The tool bar contains the check button for “Show categories” and several tool buttons described further below. With “Show categories” the display of cell categories is toggled on and off.

3. The central tree widget

   Detail information about the selected item at the bottom

3.1.3 The tree widget

The tree widget is the central place for displaying the information and interacting with the version controlled items. The leftmost column contains the tree. The default display is the hierarchy of library, category, cell, and view. Sub hierarchies can be opened via the + symbol left of the tree items, and closed by the – symbol for expanded items.
The columns to the right of the tree structure contain the properties of the corresponding item. Their visibility can be customized via the edit columns button.
The tree widget supports multiple selections. The menu and toolbar button commands work on all selected items, as far as applicable.

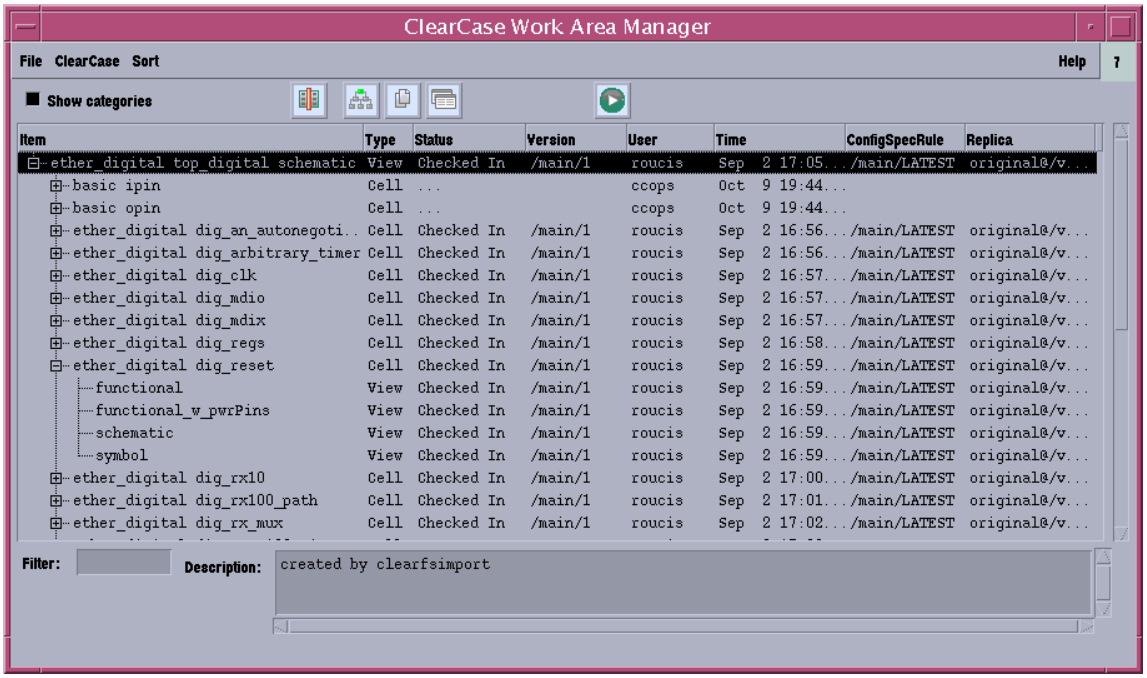


Figure 3.1.2: The ClearCase Work Area Manager browsing the design hierarchy of a cellview.

3.1.4 Description of the Toolbar buttons

 Filter

Invokes a dialog for the filter criterion for the items displayed.

 Expand all

Expands all trees that are contained in the tree widget.

 Collapse all

Collapses all tree nodes.

 Edit columns

Edit the list of columns shown in the user interface. This opens a new form to perform the editing.

 Descend into hierarchy

For every selected cellview that contains instances, the instantiated cells are inserted into the tree as subnodes to the corresponding cellview.

3.1.5 The action forms

All commands from the ClearCase menu that affect more than one item potentially open a form that enables you to select the items on which to act and supply additional options for the command that is to be run. The form is shown in figure 3.1.3. In the center is a list box with the items the command would run on. This list box is seeded with the selection from the tree widget. You select items in the list box and remove them with the “Remove selected” button. The command will be run only on those items remaining in the list box. The description field allows specifying a description for the operation, which will be stored in ClearCase. Special options for the command to be run can be added in the “Use Options:” field.

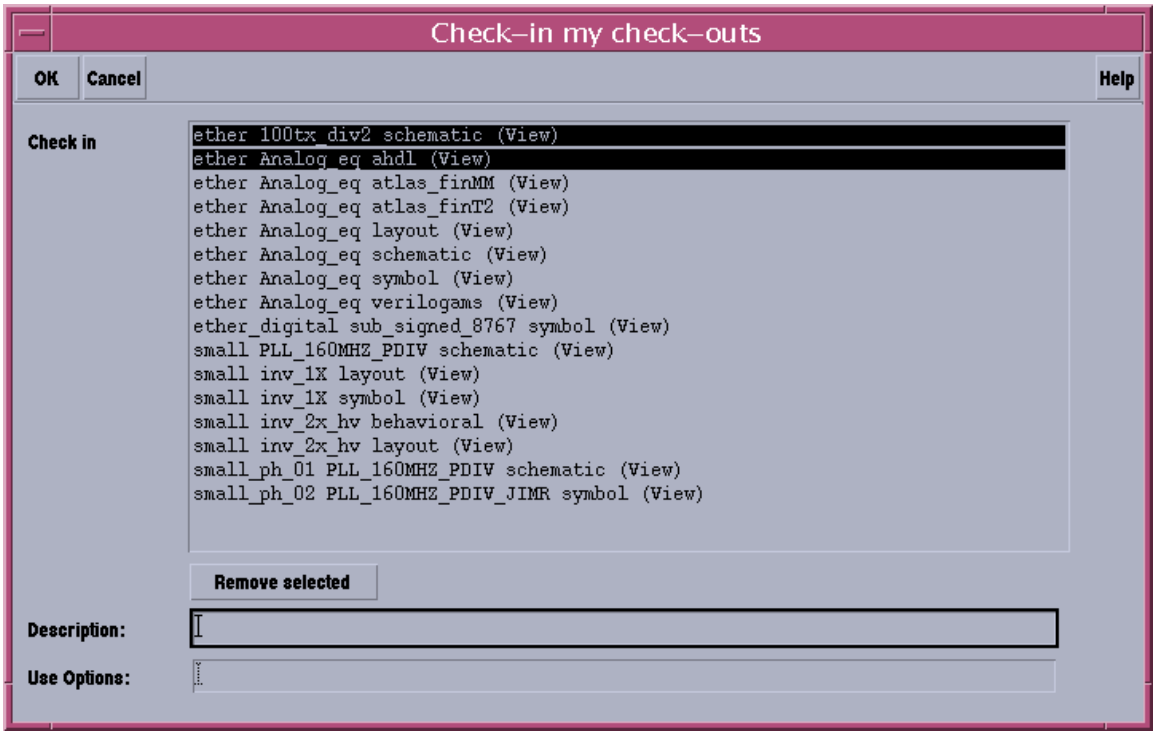


Figure 3.1.3: The form for checking in checked out items.

3.1.6 The filter form

When the filter button or menu item is selected, the filter form is shown. Use it to specify a filter for the tree widget. Only views that contain the filter string in their names are displayed, together with their corresponding cells and libraries.

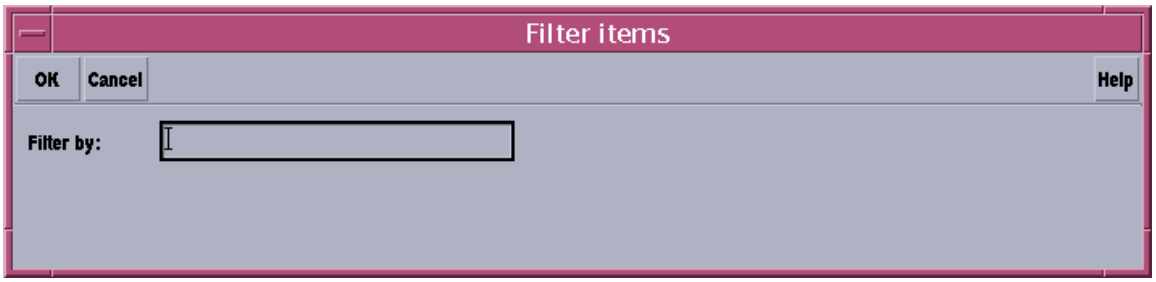


Figure 3.1.4: The filter form.

To cancel filtering, leave the filter criterion empty.

3.1.7 The column select form

The column select button displays the column selection form.

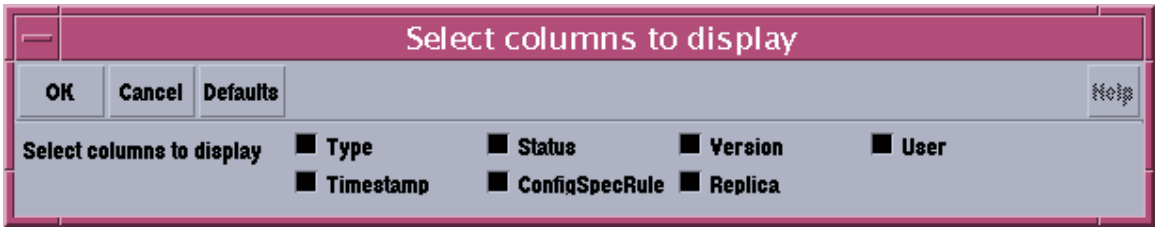


Figure 3.1.5: The column select form.

3.1.8 The rename form

The rename form renames a selected library, cell, or view. If the “Update instances in library” option is selected, all instances in the same library will be updated to instantiate the new item.

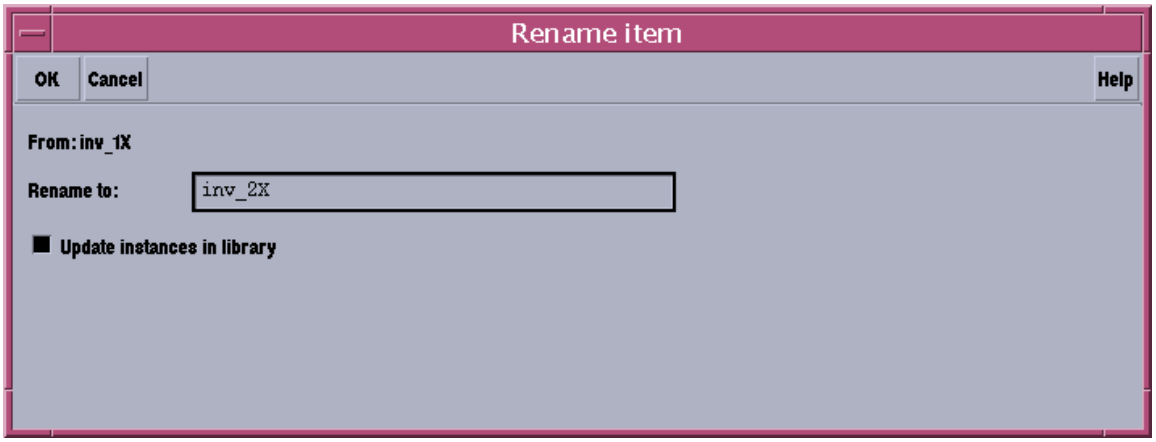


Figure 3.1.6: The rename form.

NOTE: When the “Update instances in library” option is enabled, one or more cellviews may be modified in the library containing the renamed cell. In a ClearCase managed library, this results in checkouts for the affected cellviews. You can automatically check-in these cellviews by setting preferences in Virtuoso. The renamed cell/cellview/file is not added to source control by default. Enable the auto-checkin to make them DM-managed. See the Cadence Library Manager User Guide for more information.

3.2 Integration with IC61

3.2.1 How to start the user interface

The user interface is invoked via the Cadence Library manager. To browse the list of all version controlled libraries, choose from the Design Manager menu the menu item ClearCase WA Manager.

The ClearCase Work Area Manager lists the libraries (Fig. 3.2.1). To browse the design hierarchy of a single cellview, select that cellview in the library manager and right click to display the context menu. Select “Browse hierarchy” to open the design hierarchy browser (Fig. 3.2.2). To browse a single library, select that library in the library manager and right click. Select “Browse library” to browse that library only.

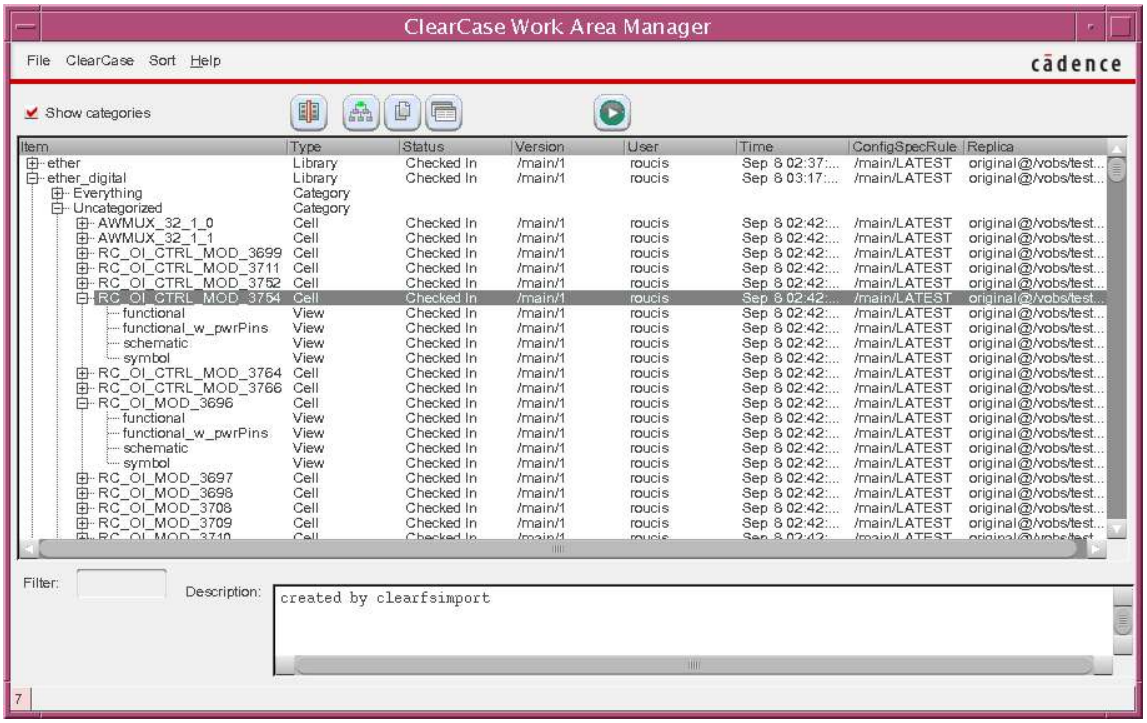


Figure 3.2.1: The ClearCase Work Area Manager browsing libraries

3.2.2 User interface components

The main components of the user interface are:

1. The menu bar
   1. File
      1. Flatten library

         This opens the selected library in a new form, all cellviews in the library in a flat hierarchy
      2. Filter

         Opens the Filter form. Only views which match the filter criterion are displayed, together with their corresponding cells and libraries.
      3. Edit configspec

Opens the editor defined by the EDITOR environment variable to edit the active configspec.

4. Close

   Closes the current form

2. ClearCase
   1. Check in

      Check in the selected items
   2. Check out

      Check out the selected items
   3. Cancel

      Cancel the checkout of the selected items
   4. Delete

      Delete the selected items
   5. Rename

      Rename the selected item
   6. Label

      Attach a label to the selected items
   7. Refresh selected

      Refresh the selected items
   8. Refresh all

      Refresh all items displayed
   9. Checkin checked out

      Check in all items currently checked out
   10. Version browser

       Open the ClearCase version browser for the selected item
   11. History

       Open the ClearCase history browser for the selected item
   12. Label type browser

       Open the ClearCase 'Label Type Browser' for locking, locking except some users, or unlocking of label types.
3. Sort
   1. By Name
   2. By Time
   3. By Status
   4. By Version
   5. By User

   Sort the displayed items by the corresponding column. Selecting the same sort criterion again reverses the sort order. To be able to sort cellviews globally, open the flattened library view of the library you are working on and sort there.

2. The tool bar

   The tool bar contains the check button for “Show categories” and several tool buttons described below. With “Show categories,” the display of cell categories is toggled on and off.

3. The central tree widget

   Detail information about the selected item at the bottom

3.2.3 The tree widget

The tree widget is the central place for displaying the information and interacting with the version

controlled items. The left-most column contains the tree. The default display is the hierarchy of library, category, cell, and view. Sub hierarchies can be opened via the + symbol left of the tree items, and closed by pressing the – symbol for expanded items.
The columns that are to the right of the tree structure contain the properties of the corresponding item. Their visibility can be customized via the edit columns button.
The tree widget supports multiple selections. The menu and toolbar button commands work on all selected items, as far as applicable.

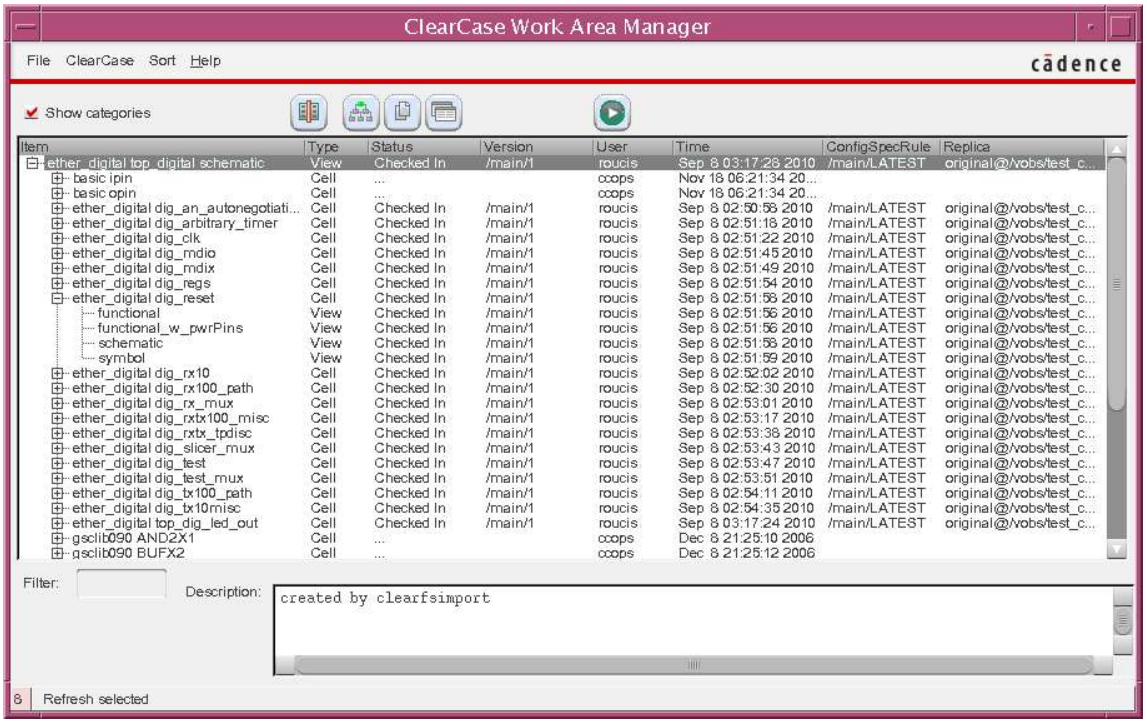


Figure 3.2.2: The ClearCase Work Area Manager browsing the design hierarchy of a cellview.

3.2.4 The context menu

Right-clicking the tree widget displays a context menu with operations to be called on the selected items. The menu items correspond to those in the menu bar.

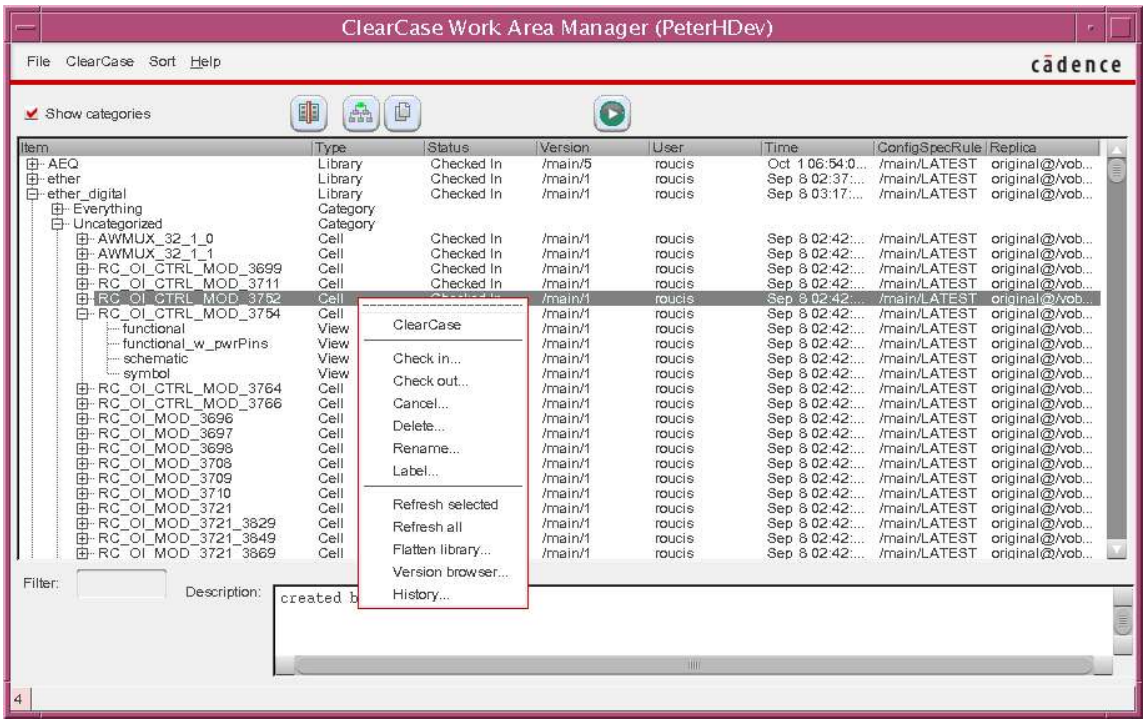


Figure 3.2.3: The context menu

On some platforms, the context menu may be converted to an orphan dialog box. To prevent this from happening, navigate to “Options ->User Preferences” (in Virtuoso) and turn off the “Tear-Off Menus.”

3.2.5 Description of the Toolbar buttons

 Filter

Displays a dialog box to define the filter criteria for the items displayed.

 Expand all

Expands all trees in the tree widget.

 Collapse all

Collapses all tree nodes.

 Edit columns

Edit the list of columns shown in the user interface. This opens a new form to perform the editing.

 Descend into hierarchy

For every selected cellview that contains instances, the instantiated cells are inserted into the tree as sub nodes to the corresponding cellview.

3.2.6 The action forms

All commands from the ClearCase menu that affect more than one item potentially open a form which you use to select the items on which to operate. The form is shown in figure 3. The list box is populated with the selections from the tree widget. Use the “Remove selected” button to remove items on which the operation is not to be performed. Use the description field to add a description for the operation, which will be stored in ClearCase. Special options for the operation that is to be run are specified in the “Use Options:” field.

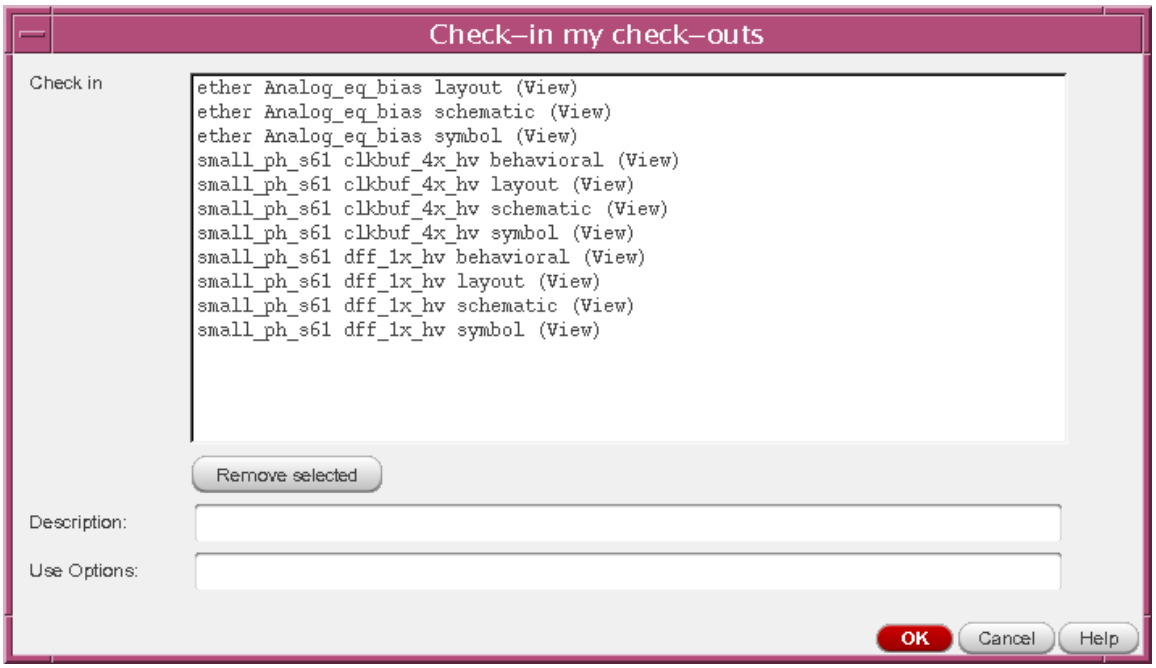


Figure 3.2.4: The form for checking in checked out items.

3.2.7 The filter form

When the filter button or menu item is selected the filter form is shown. Use it to specify multiple filter criterion for the tree widget. 'Prefetch data' option will fetch design management information for every item before applying filter. Filtering with this option may take time.

Figure 3.2.5: The filter form.

To cancel filtering, leave the filter criterion empty.

3.2.8 The column select form

Click the column select button to display the column select form and select the columns that you want displayed.

Figure 3.2.6: The column select form.

3.2.9 The rename form

Use the rename form to rename a selected library, cell, or view. If the “Update instances in library” option is selected, all instances in the same library are updated to instantiate the new item.

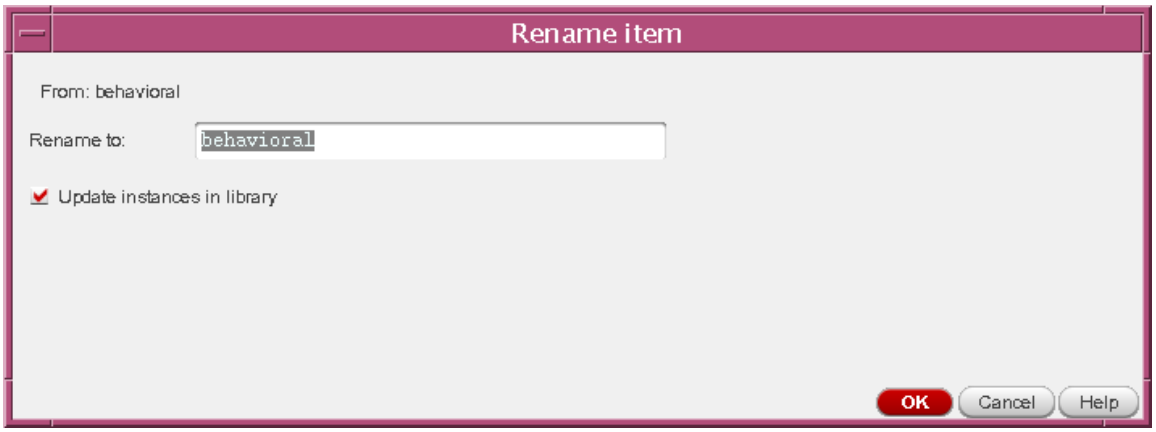


Figure 3.2.7: The rename form.

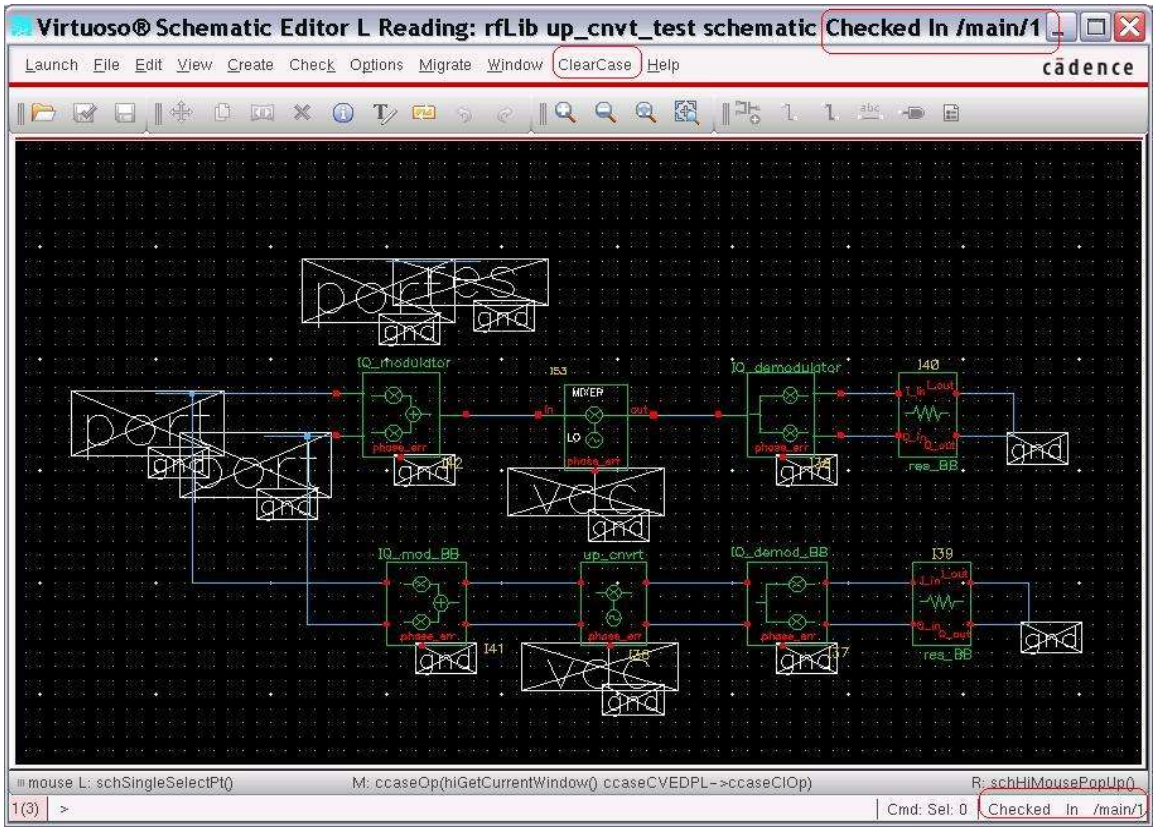
NOTE: When the “Update instances in library” option is enabled, one or more cellviews may be modified in the library containing the renamed cell. In a ClearCase managed library, this results in checkouts for the affected cellviews. You can automatically check-in these cellviews by setting preferences in Virtuoso. Also, the renamed cell/cellview/file is not be added to source control by default. Enable the auto-checkin to make them DM managed. Refer to the Cadence Library Manager User Guide for more information.

4 The ClearCase integration with Cadence Cell View Editors

This section describes the ClearCase integration with Cadence Schematic and Layout Cell View Editors, which are referred to collectively as Cell View Editors (CVEs) .

4.1 Version and status information

Version information, which is displayed on the CVE title bar, includes the current Design Management (DM) state and the version of the design artifacts that are loaded in the work area. In Cadence 6.x and later releases, version information is also displayed on the CVE banner (refer to the README for Cadence versions that support this feature). If the design is checked out to another user, the information includes the user's identity and the work area to which the design is checked out.



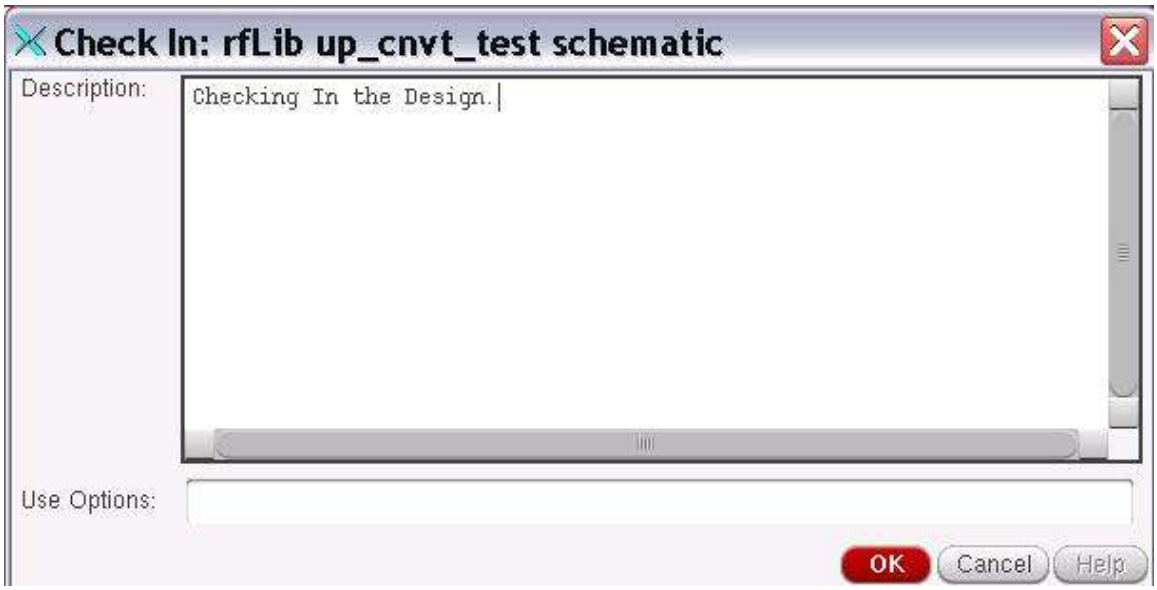
4.2 Using ClearCase from Cadence Cell View Editors

The Cell View Editors display a ClearCase menu in the menu bar. This section explains the ClearCase operations that can be performed on a design that has been opened in an editor. All information, error, warning messages from ClearCase operations are routed to the Cadence Command Interpreter Window (CIW).

4.2.1 Checking in designs

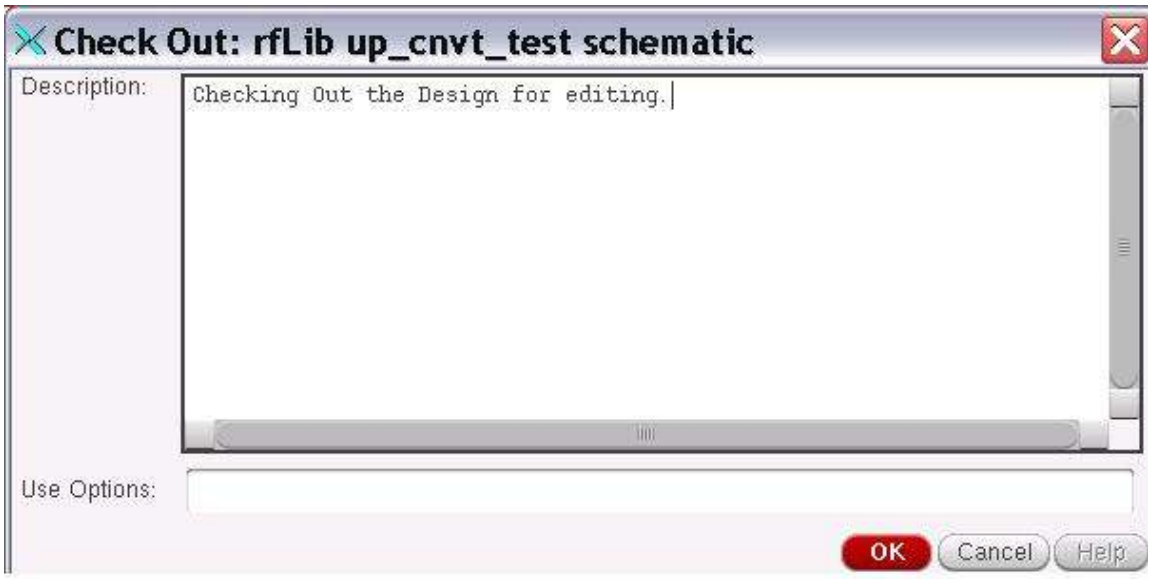
For unmanaged designs, the checkin operation implicitly adds the designs to source control. Managed designs that are checked out are simply checked in. The dialog enables you to

specify checkin options (refer Section 2.1 for information on supported checkin options).



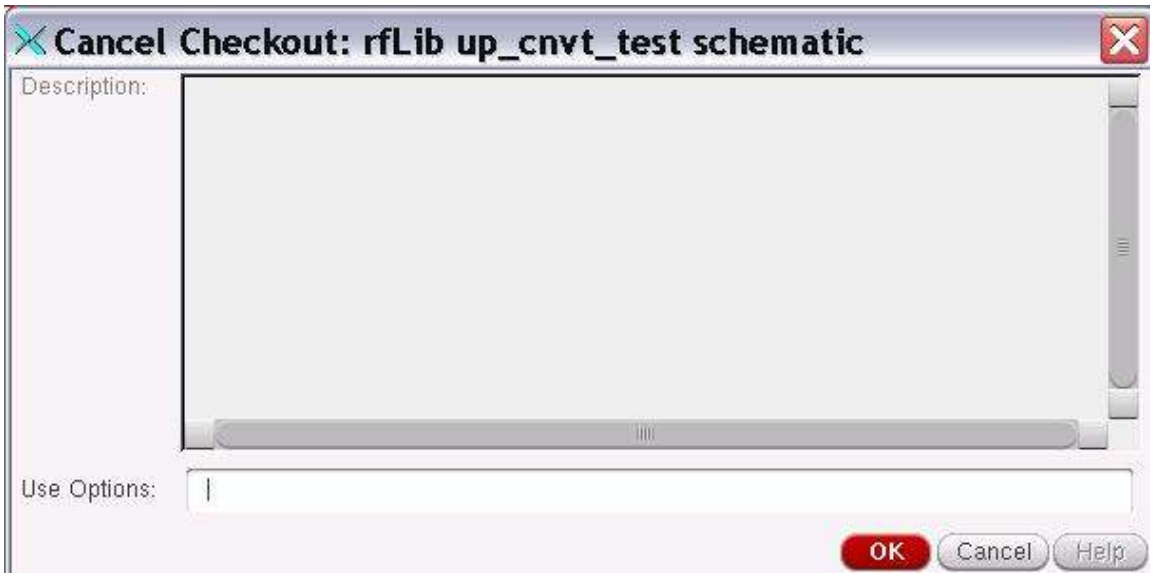
4.2.2 Checking out designs

The “Check out” menu item displays a dialog with which you can specify checkout options (refer to Section 2.2 for information on supported checkout options).



4.2.3 Canceling checkouts

The “Cancel Checkout” operation cancels a checkout and discards the changes made to a design since it was checked out. Refer to Section 2.3 for information on the options that you can specify for this operation.



4.2.4 Managing Hierarchy

Use the “Manage Hierarchy” menu item to launch the ClearCase Hierarchy Manager. Refer section 5 for information about the Hierarchy Manager user interface.

4.2.5 Applying labels

Use the “Label” menu item to apply labels to design artifacts that are under source control. Assuming the label type exists, it is applied recursively to the entire co-managed set, including the cell view directory. If the label type does not exist, you are prompted to create it, and then the label instances are applied.

The dialog enables you to specify the same options that the cleartool mklabel command supports, except -recurse (the default for this operation, as noted above). Refer to the mklabel reference page for descriptions of the options.

4.2.6 Using the version tree browser

Use the ‘Version Browser’ menu item to invoke the ClearCase version tree browser.

4.2.7 Using the history browser

Use the ‘History Browser’ menu item to invoke the ClearCase history browser.

4.2.8 Opening a previous version

Use the ‘Open Version’ menu item to invoke the ClearCase version selector dialog and further select a version from the dialog to be opened in a new editor instance in read only mode.

4.2.9 Using the property sheet

Use the ‘Properties’ menu item to invoke the ClearCase property sheet. The following ClearCase properties are displayed in the property sheet.

4.2.9.1 DM state: The design management state (checked in, checked out or checked out elsewhere) of the cell view.

4.2.9.2 Config spec rule: The rule that is applied to select the version of the cell view that is loaded into the workspace.

4.2.9.3 Version: The version of the cell view that is loaded into the workspace. If the version is checked out, this field also indicates whether the checkout is reserved or unreserved.

4.2.9.4 Permissions: The user, group and other ClearCase permissions on the cell view.

4.2.9.5 View: The view context.

4.2.9.6 View path: The view storage path.

4.2.9.7 Replica: The master replica name.

4.2.9.8 Labels: The labels that have been applied to the version of the cell view loaded into the workspace.

4.2.9.9 Description: The checkin or checkout comments associated with the current version.

4.2.10 Refreshing the DM information

The “Refresh” menu item displays the design's most recent DM state and enables or disables ClearCase menu items as appropriate for the state of the DM. This operation can be used to update the DM state in CVE, which may be changed by auto-checkout, auto-checkin and by DM operations performed from outside the CVE.

5 The Hierarchy Manager

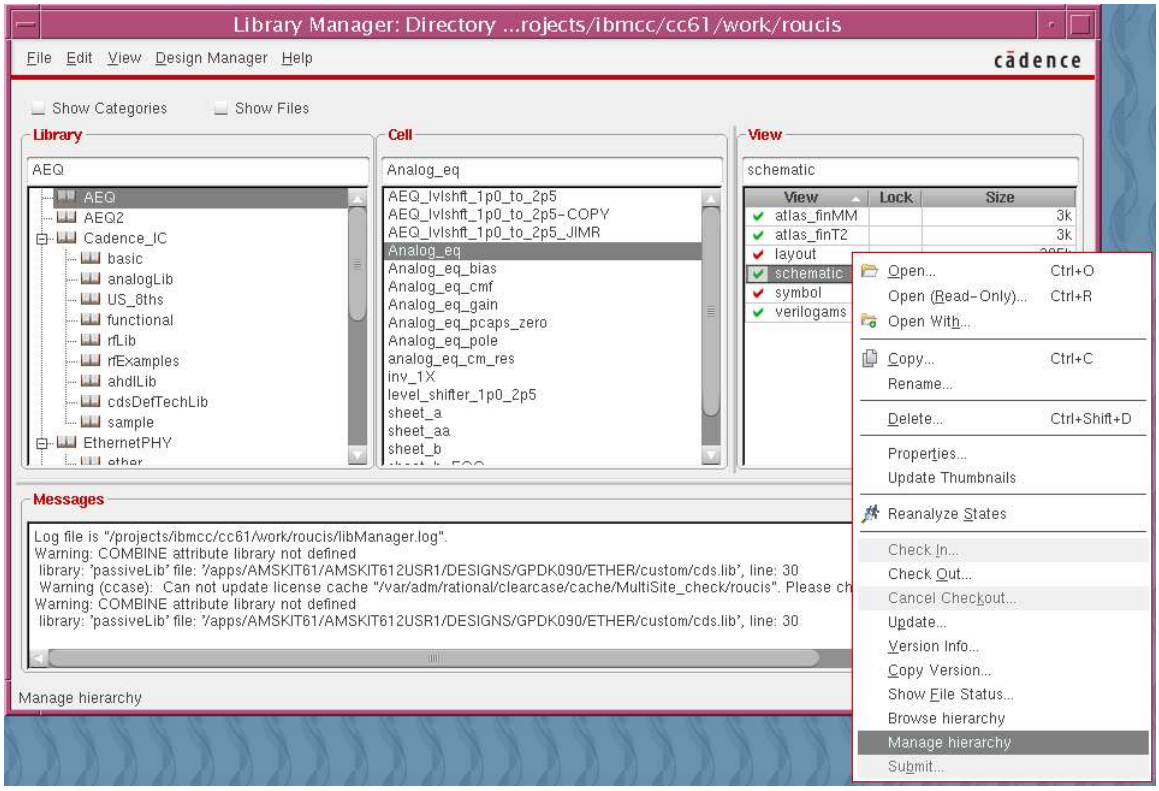
The Hierarchy Manager is similar to the Work Area Manager in certain respects but differs in its format. Along with the usual *File*, *ClearCase*, and *Help* menus, the Hierarchy Manager interface displays two tabs:

1.The *Specification* tab contains fields for specifying a Virtuoso configuration (which contains a list of all cellviews to be considered) or a top-level cellview and other controls for determining the set of cellviews that constitute the target hierarchy. Use filtering criteria to add or remove cellviews.

2.The *Cellviews* tab lists the contents of the configuration or hierarchy. You interact with the cellview list in much the same manner as with the items displayed in the Work Area Manager and its related forms.

Whereas the Work Area Manager presents a set of libraries that can be expanded into cells, which in turn can be expanded into views, the Hierarchy Manager presents a list of cellviews that constitute a design hierarchy. You can perform a DM operation on all cellviews in the hierarchy or on a subset only. Typical operations such as check in, check out, cancel check out, label, and so forth, are available.

You can invoke the Hierarchy Manager from several locations in the Library Manager or from the *ClearCase* menu that is available from most Virtuoso editor windows. The Hierarchy Manager is invoked from the Library Manager in this illustration:

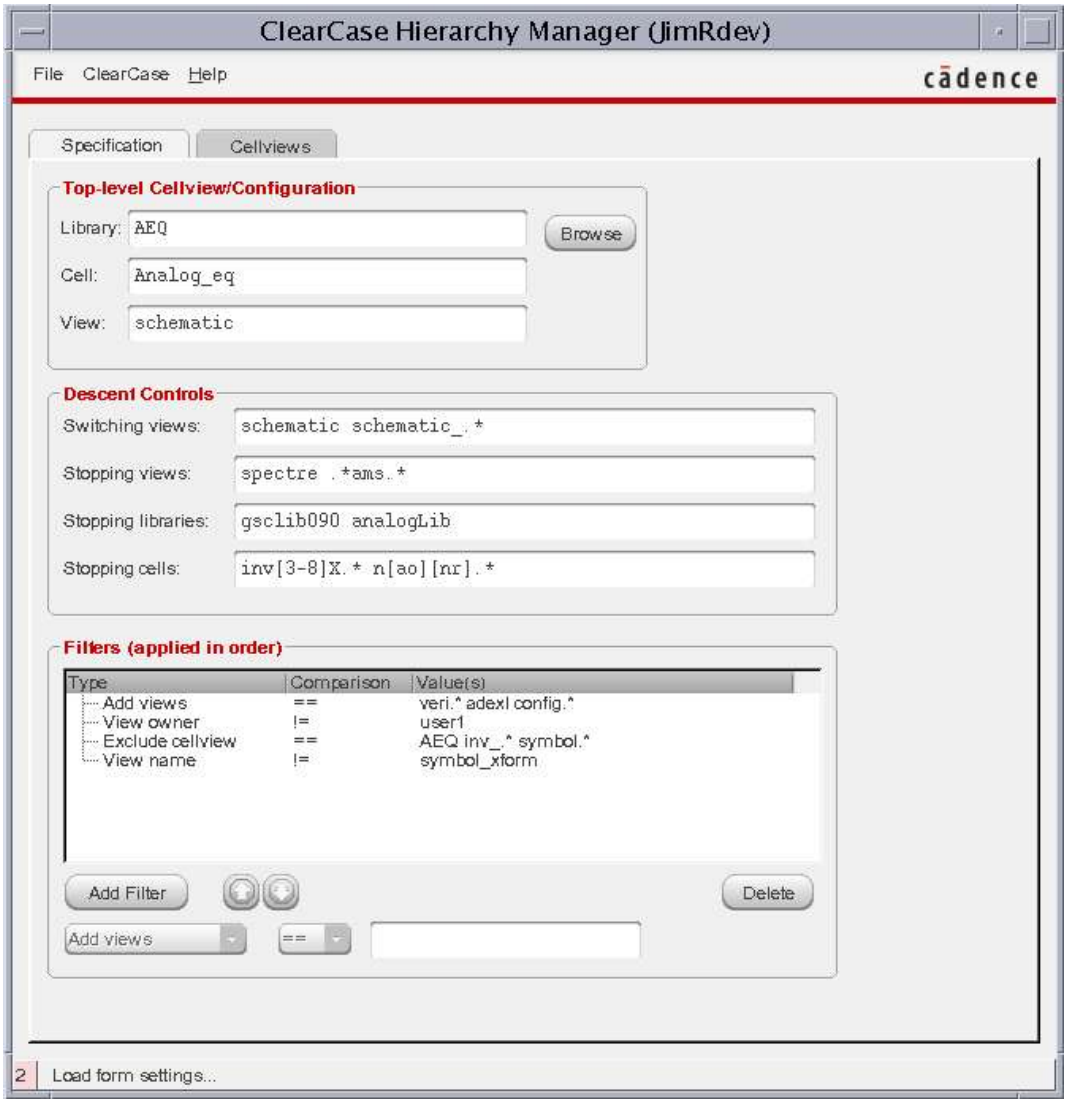


In this example, right-clicking the *schematic* view displays the pop-up menu that includes the

menu item, *Manage hierarchy.*

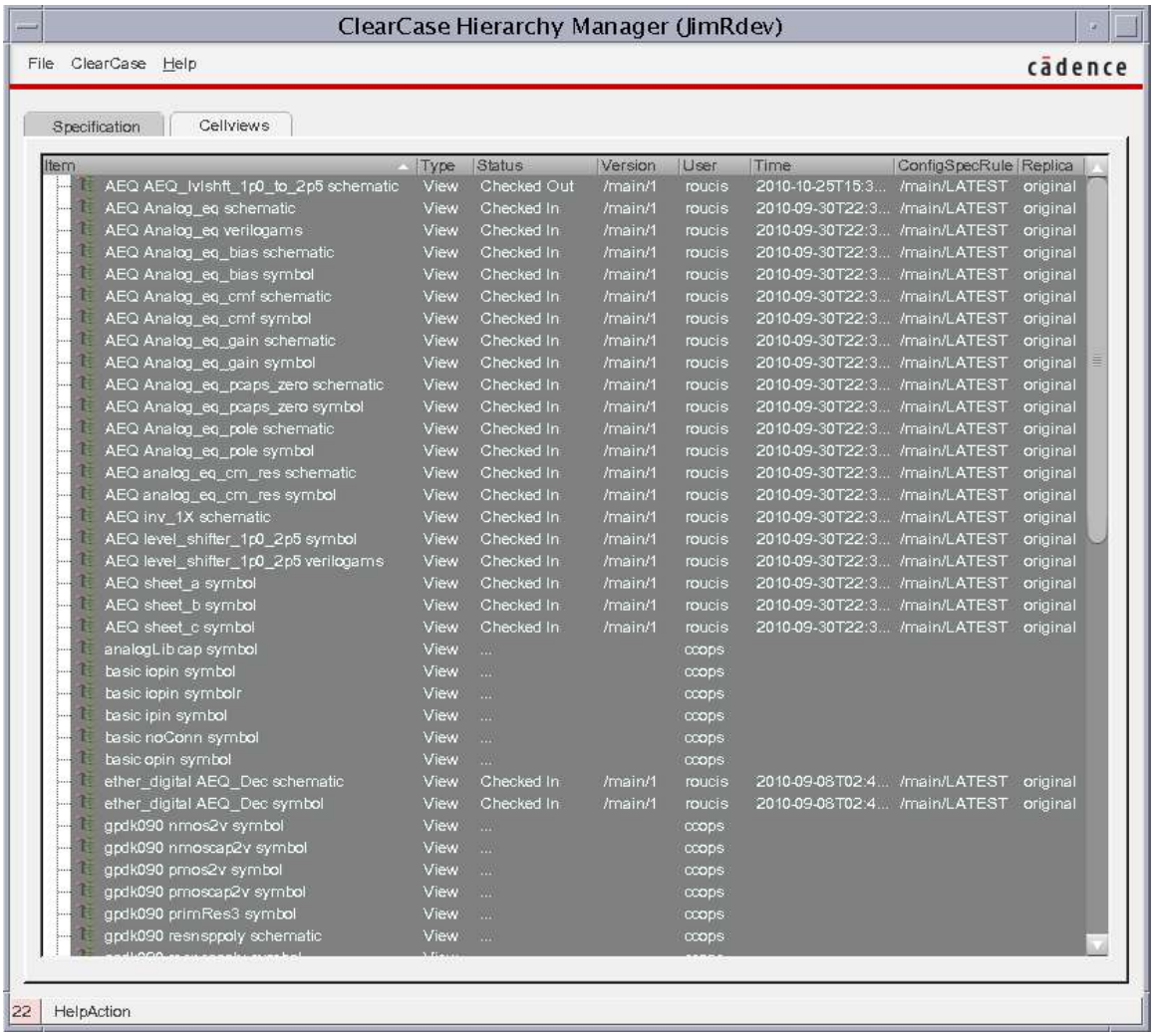
NOTE: You can invoke as many Hierarchy Managers as you need within a given Virtuoso session.

The image below shows the Hierarchy Manager in a typical use scenario:



In this situation, a top-level cellview has been specified along with various controls for the hierarchy elaboration and cellview list filtering. Notice the use of SKILL pattern matching meta-characters in many of the fields.

The image below depicts the Hierarchy Manager in the same scenario displaying the resulting cellview list:

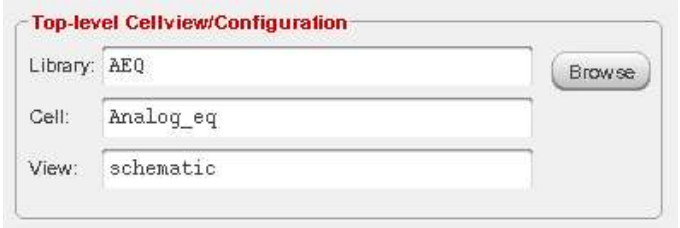


Although the entire cellview list is automatically selected, you can select any subset of the items to operate upon before invoking the desired ClearCase DM operation.

5.1 The Specification *Tab*

The Specification tab contains the controls for producing the set of cellviews to be considered for DM operations. Following is a description of these fields, the expected input, and the associated effect on the cellview list.

5.1.1 The *Top-level Cellview/Configuration* group



This group of fields determines a cellview that specify a Virtuoso configuration or the top-level cellview from which the hierarchy descent is to take place. The group contains these fields:

1.*Library name*

This field accepts a single string with no embedded whitespace characters. No pattern matching is performed on the value – all characters in the name are taken literally. This name must match one of the libraries referenced in your design environment.

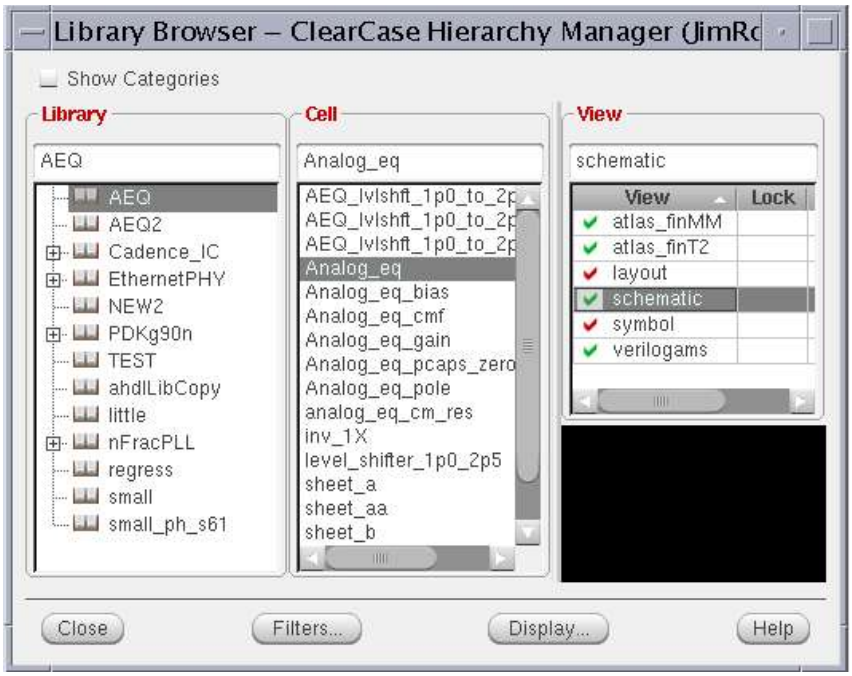
2.*Cell name*

This field accepts a single string with no embedded whitespace characters. No pattern matching is performed on the value – all characters in the name are taken literally. This name must match one of the cells in the library identified in the *Library name* field.

3.*View name*

This field accepts a single string with no embedded whitespace characters. No pattern matching is performed on the value – all characters in the name are taken literally. This name must match one of the views in the cell identified in the *Cell name* field.

This group of fields also possesses a companion *Browse* button. Similar to many other Virtuoso forms, clicking on this button invokes the standard Library Browser interface as shown below:



Selections in this browser directly affect the *Library name*, *Cell name*, and *View name* fields of the Hierarchy Manager. Using this interface can be more convenient and less error-prone than manually typing in the names.

If any of these fields in the *Top-level Cellview/Configuration* group is improperly specified, the *Cellviews* tab will be inaccessible and the items in the *ClearCase* menu will be disabled. Once a feasible set of values is provided, *Cellviews* tab and *ClearCase* menu items will be accessible

again.

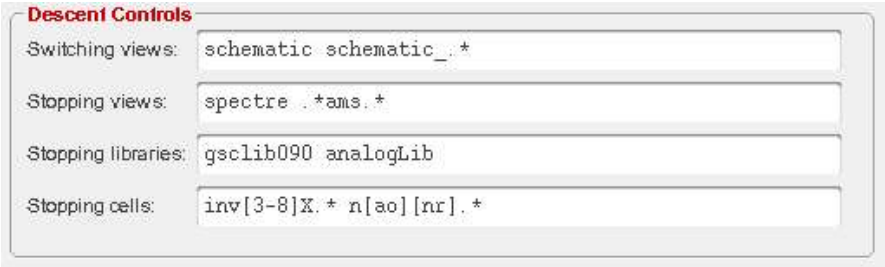
If the values of these fields specify a Virtuoso configuration, its contents provides the list of cellviews that will then be passed to the filtering stage of the Specification tab. In this case, the fields in the *Descent Controls* group will be disabled as they are not applicable in the context of a configuration.

If a design entity cellview is specified by these fields, the *Descent Controls* group will be enabled, and the values within those fields will be used to determine the descent path and depth during hierarchy elaboration.

If a physical implementation design entity (a *layout*) cellview is specified by these fields, the *Switching views* field of the *Descent Controls* group will be disabled. This is done because instances in a layout hierarchy are direct references to the desired design data. No switching views are needed as the descent takes place directly into a given instance's master cellview.

For non layout hierarchies, the *Switching views* field of the *Descent Controls* group will be enabled.

5.1.2 The *Descent Controls* group



This group of fields determines the manner in which the hierarchy descent will take place starting at the top-level cellview. Most controls limit the depth achieved by the hierarchy descent. However, the *Switching views* control, discussed below, determines which view belonging to a given instance's master cell will be used for hierarchical descent.

All fields in this group recognize SKILL pattern-matching meta-characters. Please see the *Pattern Matching of Regular Expressions* section of the Virtuoso document entitled Cadence SKILL Language User Guide for more information regarding SKILL pattern matching.

The *Descent Controls* group contains these three fields:

1.*Switching views*

This field specifies an ordered list of view names. As the hierarchy is traversed, the cell of a given instance's master cellview will be inspected to determine if any of the views it contains match one of these switching view names. The switching view names or patterns will be matched in the order they appear in this field. The first match found is the view of the instance's cell that is used for the hierarchy descent. If no matching view is found, hierarchical descent will stop at that instance and continue with other instances at the current level of the design.

This field accepts zero or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any view of any cell in

the referenced Virtuoso libraries, but typically you would provide traditional names such as *schematic* or *cmos*.

NOTE: The actual names specified are completely dependent upon your design methodology.

NOTE: This control is not applicable in direct-descent hierarchies such as a layout hierarchy – its purpose is best served in schematic oriented design hierarchies.

2.*Stopping views*

This field specifies a list of view names. As the hierarchy is traversed, the cell of a given instance's master cellview will be inspected to determine if any of the views it contains match one of these stopping view names. If a match is found, hierarchy descent will stop at that instance and continue with other instances at the current level of the design.

This field accepts zero or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any view of any cell in the referenced Virtuoso libraries, but typically you would provide traditional names such as *spectre* or *verilog*.
NOTE: The actual names specified are completely dependent upon your design methodology.

3.*Stopping libraries*

This field specifies a list of library names. As the hierarchy is traversed, the library name of a given instance's master cellview will be inspected to determine if it matches one of the "ignore" library names. If a match is found, hierarchy descent will stop at that instance and continue with other instances at the current level of the design.

NOTE: The instance's master cellview will still be considered part of the design hierarchy, but the contents of the hierarchy below that instance will be ignored.

This field accepts zero or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any of the referenced Virtuoso libraries, but typically you would provide names or patterns that *do* match – specifying nonexistent library names is allowed but not useful.

4.*Stopping cells*

This field specifies a list of cell names. As the hierarchy is traversed, the cell name of a given instance's master cellview will be inspected to determine if it matches one of the "ignore" cell names. If a match is found, hierarchy descent will stop at that instance and continue with other instances at the current level of the design.

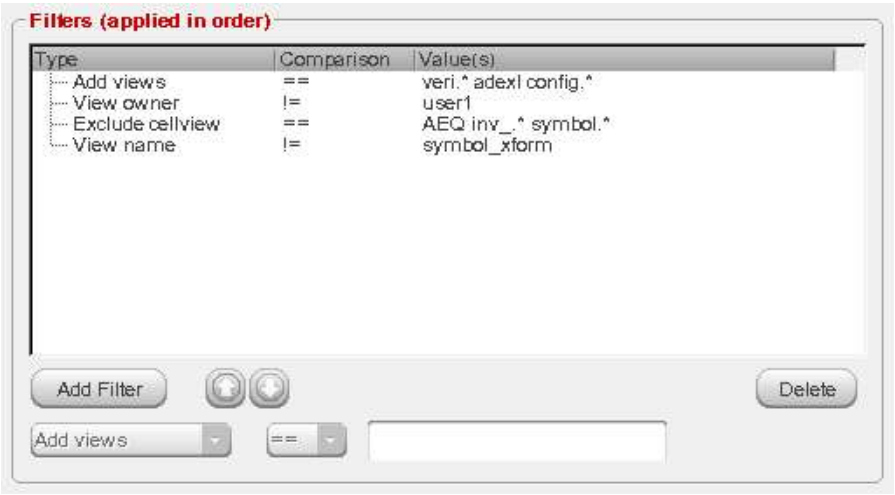
NOTE: The instance's master cellview will still be considered part of the design hierarchy, but the contents of the hierarchy below that instance will be ignored.

This field accepts zero or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any cell of the referenced Virtuoso libraries, but typically you would provide names or patterns that *do* match –

specifying nonexistent cell names is allowed but not useful.

A great deal of flexibility is offered by these controls and the pattern matching mechanisms built into them.

5.1.3 The *Filters* group



Use this optional group of fields to add or remove candidate cellviews starting with the list of cellviews determined by the settings in the *Top-level Cellview/Configuration* and *Descent Controls* fields.

Certain fields in this group recognize SKILL pattern-matching meta-characters. Refer to the section, *Pattern Matching of Regular Expressions* in the Virtuoso document, Cadence SKILL Language User Guide for more information regarding SKILL pattern matching.

The *Filters* group contains a special mechanism for adding, editing, and arranging essentially any number of filtering criteria. It is perfectly acceptable to specify no filters for a given hierarchy because the *Cellviews* tab provides the means to select a specific subset of the resulting cellview list before applying the desired DM operation. Nonetheless, you can use the filter mechanism to eliminate unwanted cellviews from the list, making it easier to review and manage.

NOTE: The filtering operations are applied *in the order they appear in the criteria list*. So, a small change in the order of the filters can dramatically affect, or possibly have no effect at all, on the resulting cellview list.

The *Filters* group comprises two subsets of fields:

1.*Criteria list*

This is the large, multiline pane that lists the criteria you add. An individual entry in this list may be selected and modified or deleted by the various criterion editing controls present in the *criterion editor* discussed below.

Each criterion identifies its type, its comparison operator, and its value or values.

2.*Criterion editor*

This is the collection of buttons at the bottom of the *Filters* group that operate on the selected criterion or criteria. Below is a description of each element in the *criterion editor*:

•The *Add Filter* button: 

Click this button to add a new criterion to the criteria list and select as the target for subsequent actions by the criterion editor.

There may be theoretical limits to the number of criteria you can add to the criteria list, but those limits are far beyond practical usability limits.

•The *move up* and *move down* buttons: 

Use these buttons to move criteria up or down the criteria list.

•The *Delete* button: 

Deletes all selected criteria from the criteria list. If no criteria are selected, clicking this button has no effect.

NOTE: No mechanism exists to “undo” criteria deletion.

•The *criterion type* field: 

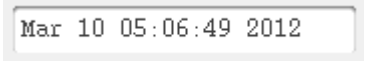
This cyclical field provides a list of supported criterion types from which you can choose.

NOTE: It is likely that this list will grow in future releases.

•The *comparison selector* field: 

Use this cyclical field to choose comparison operators supported by the criterion type.

NOTE: Some criterion types provide only a single comparison type.

•The *value* field: 

This field accepts string data. An empty value field is permitted, but causes the associated criterion to be ignored. The type of data required by this field is a function of the criterion type.

•The *value browser* button: 

If supported by the selected criterion type, clicking this button will perform at least some level of assistance for entering data in the value field.
NOTE: Not all criterion types possess a value browser.

The following types of filters are currently supported:

•*Add views*

This criterion specifies a list of view names. The companion views of each cellview in the hierarchy will be inspected to determine if any match one of these view names. All matching cellviews are added to the cellview list.

This criterion's value field accepts one or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any view of any cell in the referenced Virtuoso libraries, but typically you would provide names meaningful to your design methodology.

NOTE: If no value is provided, the criterion is ignored.

NOTE: No browser is available for this criterion.

•*Cell name*

This criterion specifies a list of cell names. The cell name of each cellview in the cellview list is compared according to the criterion's comparator. All matching cells are either excluded from or retained within the cellview list depending upon the matching cell name and comparator setting.

This criterion's value field accepts one or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any actual cells in the Virtuoso libraries, but typically you would provide meaningful names.

The comparator for this criterion allows == (equal) and != (not equal).

NOTE: If no value is provided, the criterion is ignored.

NOTE: No browser is available for this criterion.

•*Cell owner*

This criterion specifies a list of user names in the host operating system. The ownership of each *cell* involved in the cellview list is compared according to the criterion's comparator. All matching cells are either excluded from or retained within the cellview list depending upon the matching user name and comparator setting.

This criterion's value field accepts one or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any actual user

names in the host operating system, but typically you would provide meaningful names.

The comparator for this criterion allows == (equal) and != (not equal).

The browser button for this criterion enters your user ID into the value field.
NOTE: If no value is provided, the criterion is ignored.

•*Exclude cellview*

This criterion specifies the library name, cell name, and view name of a cellview to be removed from the cellview list. Ideally, three whitespace separated strings are provided in this criterion's value field. If more are provided, the additional strings are ignored. If only one or two are provided, the missing strings are assumed to be the SKILL pattern to match any combination of characters.

This criterion's value field accepts one or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any actual Virtuoso library names, cell names, or view names, but typically you would provide meaningful names.

NOTE: If no value is provided, the criterion is ignored.

NOTE: The only comparator allowed for this criterion is == (equal).

NOTE: No browser is available for this criterion.

•*Library name*

This criterion specifies a list of library names. The library name of each cellview in the cellview list is compared according to the criterion's comparator. All matching cellviews are either excluded from or retained within the cellview list, depending upon the matching library name and comparator setting.

This criterion's value field accepts one or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any actual Virtuoso libraries, but typically you would provide meaningful names.

The comparator for this criterion allows == (equal) and != (not equal).

NOTE: If no value is provided, the criterion is ignored.

NOTE: No browser is available for this criterion.

•*Modify date*

This criterion retains or excludes a given cellview based on the cellview's modification date and the criterion's comparator setting. The following comparisons are supported by this criterion:

○== (equal)
The cellview is *removed* from the cellview list if its modification date **exactly matches** the value

specified by the criterion.

○!= (not equal)

The cellview is *removed* from the cellview list if its modification date **does not** match the value specified by the criterion.

○<= (less than or equal, meaning the same or *older*)

The cellview is *removed* from the cellview list if its modification date is the same or *older* than the value specified by the criterion.

○>= (greater than or equal, meaning the same or *newer*)

The cellview is *removed* from the cellview list if its modification date is the same or *newer* than the value specified by the criterion.

○< (less than, meaning *older*)

The cellview is *removed* from the cellview list if its modification date is *older* than the value specified by the criterion.

○> (greater than, meaning *newer*)

The cellview is *removed* from the cellview list if its modification date is *newer* than the value specified by the criterion.

The browser button for this criterion displays this form to assist in choosing an appropriate date and time:

•*View name*

This criterion specifies a list of view names. The view name of each cellview in the cellview list is compared according to the criterion's comparator. All matching cells are either excluded from or retained within the cellview list depending upon the matching cell name and comparator setting.

This criterion's value field accepts one or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any actual views in the Virtuoso libraries, but typically you would provide meaningful names.

The comparator for this criterion allows == (equal) and != (not equal).

NOTE: If no value is provided, the criterion is ignored.

NOTE: No browser is available for this criterion.

- *View owner*

This criterion specifies a list of user names in the host operating system. The ownership of each *cellview* involved in the cellview list is compared according to the criterion's comparator. All matching cellviews are either excluded from or retained within the cellview list depending upon the matching user name and comparator setting.

This criterion's value field accepts one or more strings separated by whitespace characters. Each string in the field can be a literal name or SKILL regular expression containing pattern-matching meta-characters. The names and patterns provided in this field need not match any actual user names in the host operating system, but typically you would provide meaningful names.

The following comparisons are supported by this criterion:

- **==** (equal)

The cellview is *removed* from the cellview list if its owner **matches** one of the specified user names.

- **!=** (not equal)

The cellview is *removed* from the cellview list if its owner **does not match** one of the specified user names.

The browser button for this criterion enters your user ID into the value field.

NOTE: If no value is provided, the criterion is ignored.

5.2 The Cellviews Tab

The Cellviews tab contains the set of cellviews resulting from the settings on the *Specification* tab. When you click on the *Cellviews* tab, the Hierarchy Manager attempts to open the configuration or top-level cellview specified in the *Top-level Cellview/Configuration* fields. If successful, the Hierarchy Manager determines the type of data in the cellview. If it is a Virtuoso configuration, the contents of the configuration specifies the initial cellview list. If it is a Virtuoso design data cellview (such as a schematic or a layout), the Hierarchy Manager applies the descent controls as it elaborates the hierarchy to produce the initial cellview list. If any are present, the Hierarchy Manager applies filter criteria in the order they appear in the *Specification* tab. The resulting cellview list is then populated in the *Cellviews* tab.

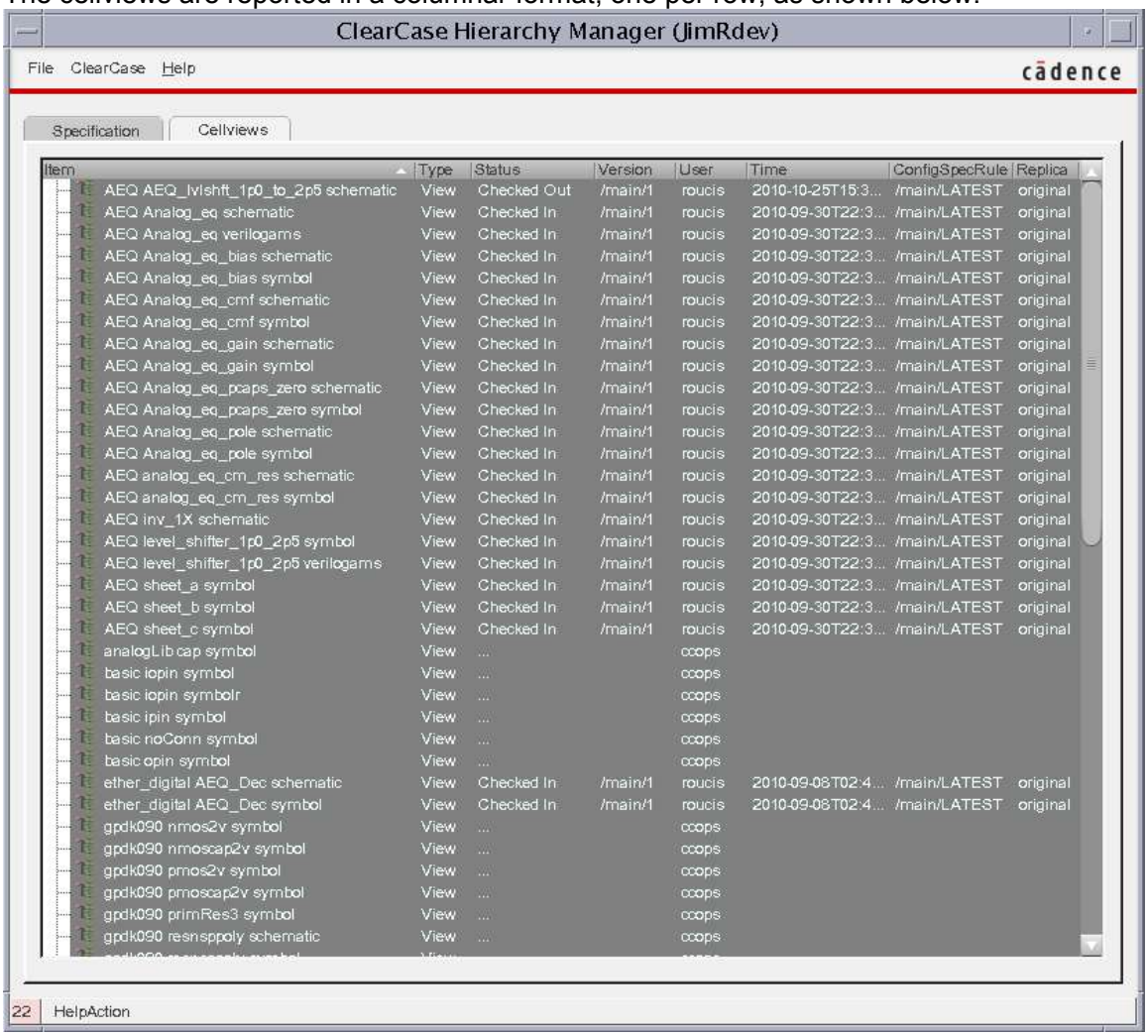
NOTE: Large design hierarchies may result in a noticeable delay when initially switching to the *Cellviews* tab. This delay may be experienced again if changes are made to the contents of the *Specification* tab.

NOTE: If the information on the *Specification* tab is incomplete, refers to a non-existent cellview, refers to a cellview that is other than a Virtuoso configuration or Virtuoso design data, or refers to a cellview that contains no hierarchy, the *Cellviews* tab shows an empty list.

NOTE: Certain combinations of filter criteria can eliminate all cellviews found in the hierarchy. In this case, the *Cellviews* tab will show an empty list.

Cellviews List Contents

The cellviews are reported in a columnar format, one per row, as shown below:



The columns contain the following information:

- *Item*

The library name, cell name, and view name of each cellview appear in this column, separated by spaces.

- *Type*

The type of data appears in this column. Currently, only *View* type information is listed here.

- *Status*

The data management status of each cellview appears in this column. The information in this column is initially … but is asynchronously updated for managed data. For unmanaged data, this column retains the … notation.

- *Version*

The data management version of each cellview appears in this column. The information in this column is initially empty but is asynchronously updated for managed data. For unmanaged data, this column remains empty.

- *User*

The user ID of each cellview owner appears in this column.

- *Time*

The modification date and time of each cellview owner appears in this column. The information in this column is initially empty but is asynchronously updated for managed data. For unmanaged data, this column remains empty.

- *ConfigSpecRule*

The config spec rule used to derive each cellview appears in this column. The information in this column is initially empty but is asynchronously updated for managed data. For unmanaged data, this column remains empty.

- *Replica*

The replica information for each cellview appears in this column. The information in this column is initially empty but is asynchronously updated for managed data. For unmanaged data, this column remains empty.

5.2.1 Selecting Items in the Cellviews List

Initially, all items in the cellview list are selected. Use the following methods to select and unselect items in the list:

To clear the current selection and select a single item, click on that item.

To select all items:

1. Press and hold the Control key.
2. Press the A key.

To select a continuous range of items:

1. Press the left mouse button.
2. Drag the cursor over the items you wish to select.
3. Release the mouse button.

Each time the left mouse button is pressed, a new selection process begins, and the previously selected items are unselected.

To modify a selection:

1. Press and hold the Shift key.

2.Move to the new endpoint of selected items.

3.Press the left mouse button.

The items between the initial start point and the new endpoint are selected; any previous selections are not unselected.

To add or delete items from the selected set:

1.Press and hold the Control key.

2.Move to the item you wish to add or delete.

3.Press the left mouse button (inverts the selection state of this item).

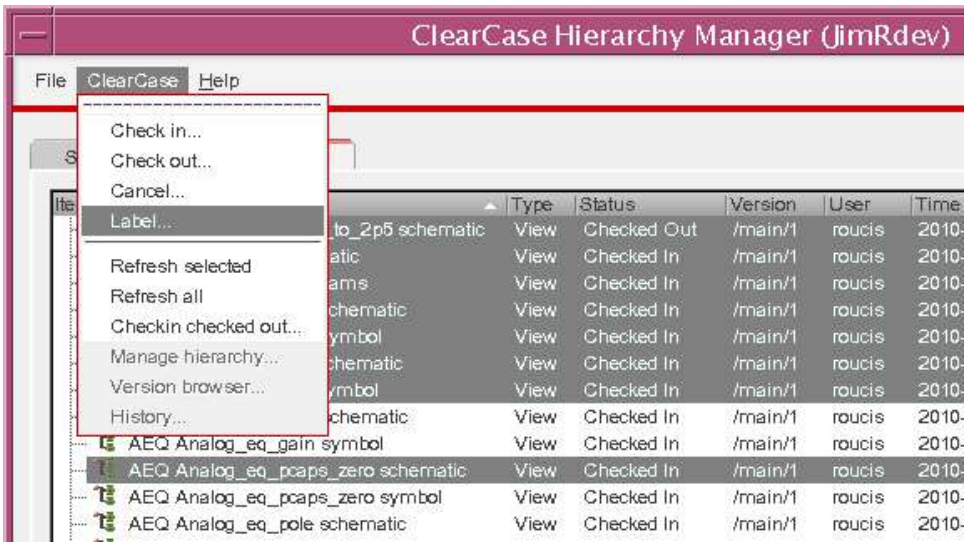
4.Drag the mouse button if you wish to add or delete a range of items.

The item(s) selected or unselected are added to or deleted from the selected set. Any previous selections remain unaffected.

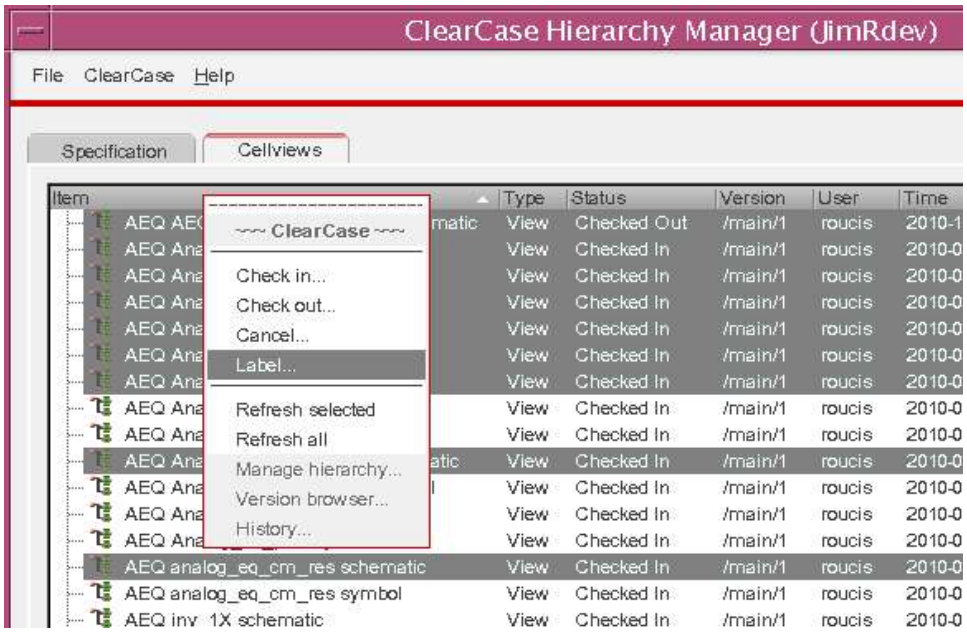
5.2.2 Performing Data Management Operations

The Hierarchy Manager provides two menus from which you can choose ClearCase data management operations:

•The *ClearCase* pull-down menu in the Hierarchy Manager banner menus.



•A pop-up menu accessed by right-mouse clicking any selected item in the cellview list.

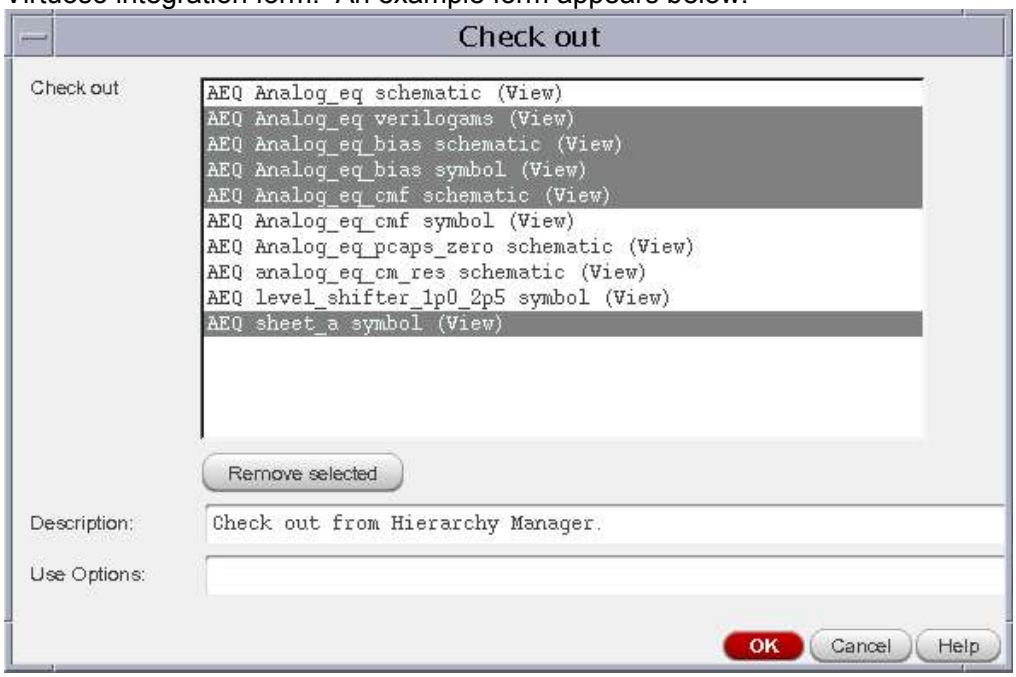


NOTE: If you right-click an unselected item, the existing selection is cleared, the item is selected, and the pop-up menu appears.
NOTE: These menu items are enabled only when a *single cellview* is selected in the list:

•*Manage hierarchy…*

•*Version browser…*

•*History…*
When you choose a data management action, you may be presented with a standard ClearCase Virtuoso integration form. An example form appears below:



You can interact with this form and others like it in the usual manner.

Check the Command Interpreter Window (CIW) for important messages arising from the data management operation.

5.3 The Hierarchy Manager File *Menu*

The Hierarchy Manager includes a *File* pull-down menu in its banner menus. This menu provides various utility functions related to hierarchical data management operations:

- *Save to CSV…*

This menu item displays a file dialog allowing you to choose a destination file. The file is written with the contents of the Hierarchy Manager form in comma-separated-values (CSV) format. This includes the settings in the *Specification* tab as well as the paths to the cellviews in the *Cellviews* tab.

- *Save managed item paths…*

This menu item displays a file dialog allowing you to choose a destination file. The file is written with the paths to managed items contained in the cellview list.

- *Load form settings…*

This menu item displays a file dialog allowing you to choose a file containing Hierarchy Manager settings. The various Hierarchy Manager fields will be set to the values saved in the file.

- *Save form settings…*

This menu item displays a file dialog allowing you to choose a file. This file will contain the current Hierarchy Manager settings. These settings can be loaded at a later time.

- *Close*

This menu item closes the Hierarchy Manager.

6 Library Manager Customization

This section describes customizations made to libManager to support additional ClearCase operations.

6.1 Comparison

Two compare operations are provided on Design Manager's property menu as well as on the context menu of the cellviews.

6.1.1 Compare with Predecessor

A convenient way to display current and predecessor version in their individual editors side-by-side.

6.1.2 Compare with Another Version

This option presents a version selector dialog from which a previous version can be selected and compared with the current version by displaying both versions in their individual editors side-by-side.