
Allegro® EDM

Version Management Utilities Guide

Product Version 23.1
September 2023

© 2023 Cadence Design Systems, Inc. All rights reserved.

Portions © Apache Software Foundation, Sun Microsystems, Free Software Foundation, Inc., Regents of the University of California, Massachusetts Institute of Technology, University of Florida. Used by permission. Printed in the United States of America.

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

Allegro EDM contains technology licensed from, and copyrighted by: Apache Software Foundation, 1901 Munsey Drive Forest Hill, MD 21050, USA © 2000-2005, Apache Software Foundation. Sun Microsystems, 4150 Network Circle, Santa Clara, CA 95054 USA © 1994-2007, Sun Microsystems, Inc. Free Software Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA © 1989, 1991, Free Software Foundation, Inc. Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, © 2001, Regents of the University of California. Daniel Stenberg, © 1996 - 2006, Daniel Stenberg. UMFPACK © 2005, Timothy A. Davis, University of Florida, (davis@cise.ulf.edu). Ken Martin, Will Schroeder, Bill Lorensen © 1993-2002, Ken Martin, Will Schroeder, Bill Lorensen. Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts, USA © 2003, the Board of Trustees of Massachusetts Institute of Technology. vtkQt, © 2000-2005, Matthias Koenig. All rights reserved.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission. All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information. Cadence is committed to using respectful language in our code and communications. We are also active in the removal and/or replacement of inappropriate language from existing content. This product documentation may however contain material that is no longer considered appropriate but still reflects long-standing industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

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

Contents

<u>Preface</u>	7
<u>About This Guide</u>	7
<u>Related Documentation</u>	7
<u>Cadence Web Resources</u>	8
<u>Related Tools and Flows</u>	8
<u>Typographic and Syntax Conventions</u>	9
<u>1</u>	
<u>Managing Component Versions</u>	11
<u>Component Version Mismatch</u>	11
<u>Common Version Management Tasks</u>	12
<u>How Library Revision Manager works</u>	12
<u>LRM Directives</u>	12
<u>Starting Library Revision Manager</u>	16
<u>Automatic Launching of LRM</u>	16
<u>Launching Library Revision Manager from Flow Manager</u>	17
<u>Starting Library Revision Manager from the Allegro EDM System Console</u>	17
<u>Comparing Component Versions</u>	18
<u>Interpreting Library Revision Manager Information</u>	19
<u>Understanding the Status of a Component</u>	21
<u>Correcting the Part Status of the Component</u>	23
<u>2</u>	
<u>Synchronizing Design Libraries</u>	29
<u>Updating Cells and Blocks</u>	29
<u>Updating Cells</u>	30
<u>Updating Blocks</u>	34
<u>Highlighting Component Instances on the Schematic</u>	36

3

Reverting to Older Versions of a Library..... 37

<u> Launching Rollback</u>	37
<u>Discarding Changes Made to a Block</u>	38
<u>Discarding Changes Made to a Cell</u>	38

4

Removing Redundant Information from the Cache..... 41

<u>Project Directives for Purge</u>	41
<u>Starting Purge</u>	43
<u>Specifying What to Remove</u>	44

5

Cache-Enabled Projects in Design Entry HDL..... 47

<u>Adding a Block</u>	47
<u>Importing a Block</u>	48
<u>Importing a Sheet</u>	48
<u>Managing Identical Block Names in Both Designs</u>	49
<u>Resolving Sheet Content Mismatch</u>	50

6

Symbol Revision Manager..... 53

<u>Overview</u>	53
<u>Symbol Versions</u>	54
<u>Footprint Files</u>	55
<u>Updating Symbols to Newer Versions</u>	55
<u>Updating to the New Footprint</u>	56
<u>Recovering Old Footprint Versions</u>	57
<u>Parameter File</u>	57
<u>Limitations of Symbol Revision Manager</u>	59

Allegro EDM Version Management Utilities Guide

7

<u>Library Verification</u>	61
-----------------------------------	----

8

<u>Design Verification</u>	1
----------------------------------	---

A

<u>User Interface</u>	5
-----------------------------	---

<u>Library Revision Manager Dialog</u>	6
--	---

<u>Cell Status</u>	8
--------------------------	---

<u>Part Status</u>	10
--------------------------	----

<u>Block Mismatch Details Dialog</u>	12
--	----

<u>Block Re-Import Dialog</u>	13
-------------------------------------	----

<u>Sheet Content Mismatches Dialog</u>	14
--	----

<u>Rollback</u>	16
-----------------------	----

<u>Purge</u>	19
--------------------	----

Allegro EDM Version Management Utilities Guide

Preface

About This Guide

Allegro® EDM Version Management Utilities User Guide introduces Library Revision Manager (LRM) and explains the tasks you can perform on cache-enabled projects.

Related Documentation

You can also refer the following documentation to know more about related tools and methodologies:

- For information about the Allegro Data Manager solution, see *Allegro® EDM Solution Overview*.
- For information about new features, see *Allegro® EDM: What's New in Release 17.4*.
- For information about Flow Manager, see *Allegro® EDM Flow Manager User Guide*.
- For information about Part Information Manager, see *Part Information Manager User Guide*.
- For information about Database Administrator, see *Allegro® EDM Database Administrator User Guide*.
- For information about Database Editor, see *Allegro® EDM Database Editor User Guide*.
- For information about Library Import, see *Allegro® EDM Library Import User Guide*.
- For information about Library Distribution, see *Allegro® EDM Library Distribution User Guide*.
- For information on how to configure Allegro EDM, see *Allegro® EDM Configuration Guide*.
- For information on how to migrate non-Allegro EDM designs and libraries into Allegro EDM, see *Allegro® EDM Migration Guide*.

Cadence Web Resources

For the latest Allegro EDM-related Known Problems and Solutions, visit [Cadence Online Support](#).

Related Tools and Flows

- For information on various PCB design working environments, such as a team of designers working on a Design Entry HDL project, implementing FPGAs in designs, working with high-speed constraints, importing IFF files for radio-frequency designs, and reusing existing modules, see *Allegro PCB Design Flows*.
- For information on how to create new Design Entry HDL projects and define the project settings, see *Allegro Project Manager User Guide*.
- For information on capturing a logical design, see *Design Entry HDL User Guide*.

Typographic and Syntax Conventions

This list describes the syntax conventions used for this user guide:

<code>literal</code>	Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names.
<i>argument</i>	Words in italics indicate user-defined arguments for which you must substitute a name or a value.
	Vertical bars (OR-bars) separate possible choices for a single argument. They take precedence over any other character.
[]	Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices. You can choose one argument from the list.
{ }	Braces are used with OR-bars and enclose a list of choices. You must choose one argument from the list.

Allegro EDM Version Management Utilities Guide

Preface

Managing Component Versions



Library Revision Manager (LRM) is available **only** in a board flow where the `cache_enabled` variable is set to *True*.

Component Version Mismatch

When creating designs in Allegro EDM, designers connect to an Allegro EDM Server with Part Information Manager, search for components, and then add them to a schematic. These components are copied to cached libraries. Designers can then disconnect from the Allegro EDM Server and use the cached versions of the components in the design. Additionally, designers can also add components from local libraries into their designs. In other words, the components in a design can be from cached libraries, which are subsets of the reference library, or from local libraries.

When a designer is working on a schematic, the librarian might modify a component that is being used in the schematic and then run library distribution. This leads to a version mismatch between the component in the reference library versus the component in the cache. In addition to version changes, there can be a number of differences between the component version in the reference library as compared to the component used in the design.

The objective of LRM is to ensure the following:

- That the design uses the latest components from the reference library.
- That cached parts are in sync with the reference libraries. All the cached parts are overwritten with the latest components from the reference library.

That no local parts are used in the design. With LRM, designers can replace local parts used in the design with the ones from the reference library. As cached libraries are not always connected, or in sync, the components in the cache libraries might be different from those in the reference libraries.

Common Version Management Tasks

When working with designs that use components from reference libraries, cached libraries, and local parts, LRM performs version management tasks such as:

- Identifying the source of the components used in the design
- Comparing the versions in the design (cached or local) with the reference libraries
- Locating changed usage information
- Traversing through the blocks in the design
- Updating the locally cached libraries
- Saving older versions of cached components

How Library Revision Manager works

When a project is opened, Library Revision Manager (LRM) reads the cache and the project. The LRM Directives in the `<project.cpm>` control the behavior of LRM. If the same components are in the local libraries and the reference library, LRM prompts you to update the design project's cache with the components in the reference library.

The information displayed in LRM includes the following:

- Components that are in the cache, and if they are used in the design
- If the component version matches the reference version, PTF differences, if any
- Based on the directive values set in the CPM, the PTF rows or cells that can be fixed automatically
- Components from local libraries and if they are found in the reference libraries

LRM Directives

The following table lists the directives in the `<project>.cpm` file that control LRM.

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Directive	Description
<code>auto_fix_ptf '<TRUE/FALSE> '</code>	<p>When this directive is set to <code>TRUE</code>, LRM fixes the autofixable PTF row (corresponding to the cell selected in the <i>Cell/Block Details</i> pane) if the <i>Update</i> button is clicked.</p> <p>This directive can be used with the following directives:</p> <p><code>sync_properties</code></p> <p><code>auto_update_minor_ptf</code> when set to <code>TRUE</code></p>
<code>auto_update_minor_cell '<TRUE/FALSE> '</code>	<p>When this directive is set to <code>TRUE</code>, LRM automatically selects cells that have minor differences as soon the <i>Update</i> button is clicked.</p>

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Directive	Description
<code>auto_update_minor_ptf '<TRUE/ FALSE/AUTO>'</code>	<p>When <code>auto_update_minor_ptf</code> and the <code>auto_fix_ptf</code> directives are both set to <code>TRUE</code>, LRM marks the part row as Autofixable. The <code>auto_update_minor_ptf</code> directive also provides flexibility in scenarios where you may not want to auto-update part table rows but want to use the auto-update feature with the <code>sync_properties</code> directive.</p> <p>When <code>auto_update_minor_ptf</code> is set to <code>AUTO</code>, LRM automatically updates injected property value and injected header mismatches.</p> <p>If both <code>auto_update_minor_ptf</code> and <code>check_injected_order</code> are set to <code>TRUE</code>, LRM considers both directives when updating the mismatches.</p> <p>If you do not want LRM to auto-update certain injected property mismatches, you can specify those properties using the <code>exclude_autoupdate_props</code> directive.</p> <p>If there is a header change along with an injected property value change which is defined in the <code>exclude_autoupdate_props</code> directive list, LRM will not update any of the parts</p> <p>If the same property is a key and an injected property, LRM will not auto-update the mismatch.</p>
<code>check_local_modified '<TRUE/ FALSE>'</code>	<p>Verifies (when set to <code>TRUE</code>) whether any cell has been modified directly on the disk without using LRM.</p>

Allegro EDM Version Management Utilities Guide

Managing Component Versions


Directive

Description

`check_injected_order`

Verifies (when set to *TRUE*) whether the order of any of the injected property headers has changed and indicates the change in LRM:

Status
Injected Property Order Mismatch
Injected Property Order Mismatch
Not in Reference
Injected Property Order Mismatch



`dump_FileName '<name>'`

Specify the name of the dump file that contains up-to-date revision information about cells and blocks in Library Revision Manager.

If you do not provide a file name, the default name, `lrmDumpFile.lrmDump`, is used.

`exclude_autoupdate_props`
`'<prop1>' '<prop2>'`
`'<prop3>'`

Use this directive to specify injected property value mismatches that you do not want LRM to auto-update even when the `auto_update_minor_ptf` has been set to *AUTO*.

`lrm_logfile '<name>'`

Specify the name of the log file that is generated during the update operation. This file is created at the location defined by the `adwconfigdir` directive in the ADW section.

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Directive

`sync_properties '<KEY1>
<KEY2> <INJ1> <INJ2>'`

Description

Using this directive, you can specify a preference for a key or injected property to decide which mismatched part should be auto-fixed.

For example, in the following figure, there is a mismatch in Key2:

	Key1	Key2	Inj1	Inj2
Local	10k	5%	def	cell
Reference	10k	7%	def	cell

Assume that the value of the `sync_properties` directive is `key1`, and in the reference PTF row, the value for `key2` has changed from 5% to 7%. In this case, Library Revision Manager displays the row as *Autofixable* in the *Cell/Block Details* pane, and displays a grid if you select the *Show Differences* pop-up menu option.

Starting Library Revision Manager

The following topics show you how to start Library Revision Manager:

- [Automatic Launching of LRM](#)
- [Launching Library Revision Manager from Flow Manager](#)
- [Starting Library Revision Manager from the Allegro EDM System Console](#)

Automatic Launching of LRM

If there are differences between the cached libraries and design, and the reference libraries, LRM is automatically launched when you open an Allegro EDM project (board flow only). The cache folder is in the project directory and information about the reference libraries is stored in the `REF_PPT` directive in `<project>.cpm`.

```
START_ADW
ADWCONFIGDIR './atdmdir'
REF_PPT '$PCBDW_LIB/reflib/model_sym/part_table.ptf'
END_ADW
```


Allegro EDM Version Management Utilities Guide

Managing Component Versions

Part Information Manager User Guide describes how you can use Part Information Manager to work with libraries in the cached or reference (database) mode.

To be able to use LRM, ensure that all the cells have versioning metadata. See *Part Developer User Guide* for more information on versioning metadata.

Launching Library Revision Manager from Flow Manager

1. Depending on the implementation of Allegro EDM, LRM can be launched in the following ways:
 - a. Click *Board Design Flow* in the Flow Tree pane (left panel).
 - b. Choose *Project Admin – Update Project Library* in the *Flow Specific Tools* pane.

If you have administrator rights for Allegro EDM, you can use *File – Update – Projects*.

Starting Library Revision Manager from the Allegro EDM System Console

Type the following command at the Allegro EDM System Console:

```
lrm -proj <project cpm path>
```

For example:

```
lrm -proj allegroEDMprojects\lrm\cell_mismatches\cell_mismatches.cpm
```

Comparing Component Versions

When components in the project and the cached libraries are compared with those in the reference libraries, the results show components that are:

- The same in both the libraries but the PTFs have differences
- Updated in the reference libraries
- Have changed properties in either, or both, libraries, that is, cache or reference
- In the local library but not in the cached or reference library

In the local library but not in the cache but another version in the reference library The *Library Revision Manager* dialog displays information for all cells that have conflicting or changed library information.

- The component changes could be for any, or all, of the following:
 - ☐ Cells
 - ☐ Blocks
 - ☐ Physical part (PTFs) for a cell

Cell/Block data for design 'design1'

Cell Name	Used	Status	Part Status
74f07	Yes	Reusable ...	Autofixable
74f08	No	Reusable ...	In Sync
c_7461	Yes	Local Cell	Need Manual Update

Cell: 74f07
 Local Version = 2.0.0
 Reference Version = 2.0.0
 Library in Local = crystal
 Library in Reference = 74f

-----User Comments-----

Cell will be deleted from the local library and copied from reference to the cache library.

Detailed Comments...

Physical Part Rows for Cell "74f07"

Source	Physical Pa	Status	Used	PART_NUM	PACK_TYPE	MATE
Design		Sync Propert	Yes	100607-251	SOIC	EMPTY

Schematic Instance(s) = [1]

Instance	Refdes
design1:pag	

Update Close Help

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Interpreting Library Revision Manager Information

Library Revision Manager compares the cached library with the design and shows you the following information:

1. Whether the component is used in the design

Components where the *Used* column shows *No* are not included in the `<project>.cpm` but exist in the cache. Consequently, changes for these components are made only to the libraries and not to the `<project>.cpm` file.

2. Tooltip explaining the cell status

- ☐ The results of clicking Update in each highlighted cell
- ☐ If the cell cannot be updated

3. The component versions in the cache and reference libraries

4. The component status

- ① Shows Yes if the design uses this component
- ④ Shows the status of the component. See [Understanding the Status of a Component](#).

The screenshot shows the 'Cell/Block data for design 'design1'' dialog box. It contains several sections:

- Table 1: Cell/Block data**

Cell Name	Used	Status	Part Status
74107	Yes	Cell missing in Reference	Need Manual Update
c_7461	Yes	Local Cell	Need Manual Update

- Warning Message:** Cell exists locally but is not in the cache or ref lib. Cannot be updated.
- Table 2: Physical Part Rows for Cell "c_7461"**

Source	Parts	Status	Used
Design		Not in Cache	Yes

- Warning Message:** Part not found in the cache. Update will add this part from the reference area to the cache.
- Table 3: Component Versions and Libraries**

Cell: c_7461	
Local Version	= 1.0.0
Reference Version	= -
Library in Local	= c_jc
Library in Reference	= -
-----User Comments-----	

- Table 4: Schematic Instance(s) = [1]**

Instance	Refdes
design1.pag	

5. The physical part status

Allegro EDM Version Management Utilities Guide

Managing Component Versions

If required, you can update the physical part row used.

Important

When you update a PTF row, all the schematic instances are updated. Library Revision Manager does not allow you to update the row only for the block or for an instance. If you want to do so, use Part Manager.

- After you have reviewed the component status, part status, and set the part table row, specify the components that need to be updated.
- Check the details of the PTF.

6 Select components to update

5 Shows physical data changes for the component. (See [Correcting the Part Status of the Component](#).)

Cell Name	Used	Status	Part Status
74f07	Yes	Reusable ...	Autofixable
74f08	No	Reusable ...	In Sync
c_7461	Yes	Local Cell	Need Manual Update

Cell: c_7461
Local Version = 1.0.0
Reference Version = -
Library in Local = c_ic
Library in Reference = -
-----User Comments-----

Cell exists locally but is not in the cache or ref lib. Cannot be updated.

Detailed Comments...

Physical Part Rows for Cell "c_7461"

Source	Parts	Status	Used
Design		Not in Cache	Yes

Part not found in the cache. Update will add this part from the reference area to the cache.

Update Close Help

7 Shows the available physical parts / details for the selected component's PTF

8 Creates backup and synchronizes the libraries

- Click *Update*.

Understanding the Status of a Component

The rows in the *Cell/Block data for design <design name>* table report what is out of sync in the design. When you select a row, you see a description of what is wrong with the cell/block and if it can be updated without your inputs. The [Analyzing the Cell/Block Status](#) table summarizes the different values for the Cell Status column in the *Library Revision Manager* dialog along with a description of the action that you can take.

Table 1-1 Analyzing the Cell/Block Status (Sheet 1 of 3)

Status	Indicates that	Corrective action
Cell in different library	The cell is in different cache and reference libraries.	The location of the cell in your project libraries needs to be corrected. LRM moves the cell to the correct library.
Cell missing in Reference	The cell is in the cache but not the reference library.	Check with the librarian. If this is a deleted cell, use an alternative cell in the design. If needed, ask the librarian to update the reference library with the new cell.
Error	The cell has an unknown error that Library Revision Manager cannot identify. For example, cell metadata is missing from the cache as well as from the reference library.	Cache cell: Open the cell using Part Developer to determine the cause of the error. Reference cell: Ask the librarian to check the cause of the error. Updating the cell will replace the modified cell in the cache with the reference library cell.
In Sync	The cell in the cache and reference libraries are identical.	No action required.
Incorrect Metadata	The cell has been modified manually (possibly without using Part Developer) resulting in inconsistencies in the cell metadata.	Instead of making changes manually, ask the librarian to make the modifications and distribute the updated cells to all the sites. Updating the cell will replace the modified cell in the cache with the reference library cell.

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Table 1-1 Analyzing the Cell/Block Status (Sheet 2 of 3)

Status	Indicates that	Corrective action
Local cell/block	Cell/block is not in the reference libraries but is used in the design. Not there in the cached libraries. This could be from an entry in the <i>cds.lib</i> file.	Nothing, if the designer has done this intentionally. If the cell is required in the reference library, ask the librarian to add it.
Major Update	The cell has undergone considerable changes. For example, a pin could have been added, deleted, or moved within a symbol. The changes in the cell can impact packaging.	Analyze the impact of the change and make sure that updating the design with such cells does not affect your design. If you decide to update the cell, ensure you check the schematic and update it, if required.
Minor Update	The cell has undergone a small change. For example, a change in the text of a reference designator. This usually does not impact the packaging data in the design.	Analyze the impact of the minor change and ensure that updating the design with such cells does not affect your design.
Modified in Cache	The cache cell (under the project library) has been updated (for example, using Part Developer).	<p>If the changes made locally need to be retained, ask the librarian to make the modifications in the reference library cells. Updating the cell will replace the modified cell in cache with the reference library cell.</p> <p>If the local cell is used by other users, update after checking with them. The update might affect their designs.</p>

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Table 1-1 Analyzing the Cell/Block Status (Sheet 3 of 3)

Status	Indicates that	Corrective action
Modified in Reference	The reference cell has been modified (for example, using Part Developer).	Update the design with the modified reference cell. Updating the cell will replace the modified cell in the cache with the reference library cell.
New	A new cell has been added to the cache. Such cases arise when you import, or add a block to your design.	No action required. The new cell will be automatically copied from the reference library.
Nonreadable cache metadata	Library Revision Manager is unable to read cell-level metadata from the cache. As a result, the correct status is unavailable.	Verify the cached cell using Part Developer and take a decision about whether you should update the cache cell. Updating the cache cell will replace it with the cell in the reference library.
Nonreadable reference metadata	Library Revision Manager is unable to read cell-level metadata from the reference library. As a result, the correct status is unavailable.	Contact the librarian and request a correction to the reference metadata. When you have the cell with the correct metadata in the reference library, update the design with that cell.
Reuseable cell/block	The cell/block is in the reference libraries and is used in the design, but it is not there in the cached libraries.	Clicking Update gets the cell/block from the reference library to the cached library. The cell from the local library will be deleted. The cell from the reference library is copied into the library into the cache.

Correcting the Part Status of the Component

For each row in the *Cell/Block data for design <design name>* table, the *Part Status* can have the following values:

- Autofixable

Allegro EDM Version Management Utilities Guide

Managing Component Versions

- In sync
- Needs manual update

If the values are *Autofixable* or *Needs manual update*, when you select a component, you will see the details of PTF mismatches. If is *In Sync*, the *Physical Part Rows* table remains empty for the selected row.

The Analyzing the Physical Part Rows Status table summarizes the different values for the *Status* column in the *Physical Part Rows for Cell “<selected cell>”* along with a description of the action that you can take. When you select a part row, you can see a description of the issues LRM found and if it can correct the problem without your inputs.

Table 1-2 Analyzing the Physical Part Rows Status (Sheet 1 of 5)

Part Status	Indicates that	Action required
Added Property Mismatch	<p>There is an added property mismatch between the cache and reference PTFs. Some cases in which this arises are:</p> <ul style="list-style-type: none">■ The added property is in the cache PTF but is missing in the reference PTF.■ The added property is in the reference PTF but is missing in the cache PTF.■ The value of the added property in the cache and reference PTFs is different.	<p>View the differences between the cache and reference PTFs using the <i>Show Differences</i> pop-up menu option.</p>

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Table 1-2 Analyzing the Physical Part Rows Status (Sheet 2 of 5)

Part Status	Indicates that	Action required
Sync properties match	<p>If there is no header or added property mismatch, and the part is not found in the reference libraries, then only sync properties are matched.</p> <p>The PTF row property can be automatically updated using the <i>sync_properties</i> directive defined in the project file.</p> <p>The mismatched cache PTF row is compared with the other reference PTF rows. If a reference PTF row matches as defined by <i>sync_properties</i>, it is replaced with the cache PTF row. There is no need for manual intervention.</p>	<p>View the differences between the cache and reference PTFs using the <i>Show Differences</i> pop-up menu option.</p> <p>Check the cell whose PTF you want to update and click <i>Update</i>. This will automatically update all the PTF rows.</p>

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Table 1-2 Analyzing the Physical Part Rows Status (Sheet 3 of 5)

Part Status	Indicates that	Action required
Autofixable	<p>The part rows in the cache can be autofixed with respect to the reference library PTF. This fix can be based on either a minor mismatch (injected property mismatch) or the <i>sync_properties</i> directive.</p> <p>You can see the corresponding PTF row (with which it is going to be replaced) using the Show Differences pop-up menu option.</p> <p>This status will not appear in the <i>Physical Part Rows for Cell <cell name></i> table.</p>	Select the check boxes against the cell for which you want the PTF to be autofixed.
Block PTF	Indicates the PTF added to the cache as a result of an imported or added block. It shows that the new PTF is being used in the block being imported or added to the design.	No action required. As the block is added or imported, these PTF rows will be cached automatically.
In Sync	<p>The PTF rows in the cached and reference libraries are identical and there are no differences.</p> <p>This status will appear only in the <i>Cell/Block data for design <design name></i> Part Status.</p>	No action required.

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Table 1-2 Analyzing the Physical Part Rows Status (Sheet 4 of 5)

Part Status	Indicates that	Action required
Injected Header Mismatch	The injected property headers between the reference and cache PTF rows are different.	Replace with the correct PTF row using the <i>Replace with</i> pop-up menu option. If you do not update all the part rows of the cache PTF with the reference library PTF, none of the parts are updated for the cache PTF.
Injected Value Mismatch	There is an injected property value mismatch between the cache and reference PTFs.	You can replace the cached PTF row with a correct PTF row using the <i>Replace with</i> pop-up menu option.
Key Header Mismatch	The key property headers between the reference and cache PTF rows are different.	Replace with the correct PTF row using the <i>Replace with</i> pop-up menu option. If you do not update all the part rows of the cache PTF with the reference library PTF, none of the parts are updated for the cache PTF.
Key Value Mismatch	There is a key property value mismatch between the cache and reference PTFs.	You can replace the cached PTF row with a correct PTF row using the <i>Replace with</i> pop-up menu option.
Manually Replaced	You have changed the erroneous PTF with a correct PTF (using the <i>Replace with</i> pop-up menu option)	Once done, you can update the design by clicking <i>Update</i> .
Missing in Ref	The reference library PTF is missing.	Report this issue to your librarian who will take corrective action.

Allegro EDM Version Management Utilities Guide

Managing Component Versions

Table 1-2 Analyzing the Physical Part Rows Status (Sheet 5 of 5)

Part Status	Indicates that	Action required
Need Manual Update	<p>The PTF row is not autofixable. You need to manually update it.</p> <p>This status will appear only in the <i>Cell/Block data for design <design name> Part Status</i>.</p>	You can replace the PTF row with a correct PTF row using the <i>Replace with</i> pop-up menu option.
New	<p>A new PTF will be added to the cache. Such cases arise when you import or add a block to your design.</p> <p>Parts that are used in the blocks but are missing from the cache have this status.</p>	No action required. These parts will be cached automatically as the block is added or imported to the design.
Not in cache	The part is used in design and is in the reference libraries, but is not in the cache.	<p>Update will add this part from the reference area to the cache.</p> <p>The local part will be will lost. Other designers using the local part will be affected.</p>

Synchronizing Design Libraries

The result of the comparison of the design, cached libraries, and reference libraries is displayed in the *Library Revision Manager* dialog. As the designer, you need to specify what needs to be done by LRM to synchronize the cache with the reference libraries.

Updating Cells and Blocks

A row in the Cell/Block pane has the same columns regardless of its being a cell or a block. However, some additional information is processed for blocks.

- The Updating Cells topic has the steps for setting up cell options.

When setting options for a block, you have a choice. You can:

- ☐ Set one option for the entire block

or

- ☐ Navigate to cells and blocks within a block and set options for each cell, and continue this for all the blocks in the design hierarchy

- The Updating Blocks topic explains how Library Revision Manager handles cells within blocks and you can drill-down to the lowest level of cells in the design and specify how it gets updated.

Updating Cells

To update cache cells with the latest reference library versions, do the following in the *Library Revision Manager* dialog.

1. Click a row in the *Cell/Block Details* pane.

The details of the component are displayed in the *Comments* area, *PPT Details* pane, and the *Schematic Instances* area.

2. Check the status of the component.

See “[Understanding the Status of a Component](#)” on page 21 for the possible values of the Cell Status column and what, if any, action you need to perform. If the *Cell Status* value is *NEW* or *In Sync*, you do not need to perform any action and can move to the next component.

3. Check the part status of the component.

The possible values of the Part Status column and what, if any, action you need to perform are as follows:

Part Status	Action Required	Move to
NEW	None	The next component
In Sync		
BLOCK PTF		
Manual Update	Replace the cached PTF	Step 4
Key Value Mismatch	row	
Injected Value Mismatch		
Key Header Mismatch		
Injected Header Mismatch		

Allegro EDM Version Management Utilities Guide

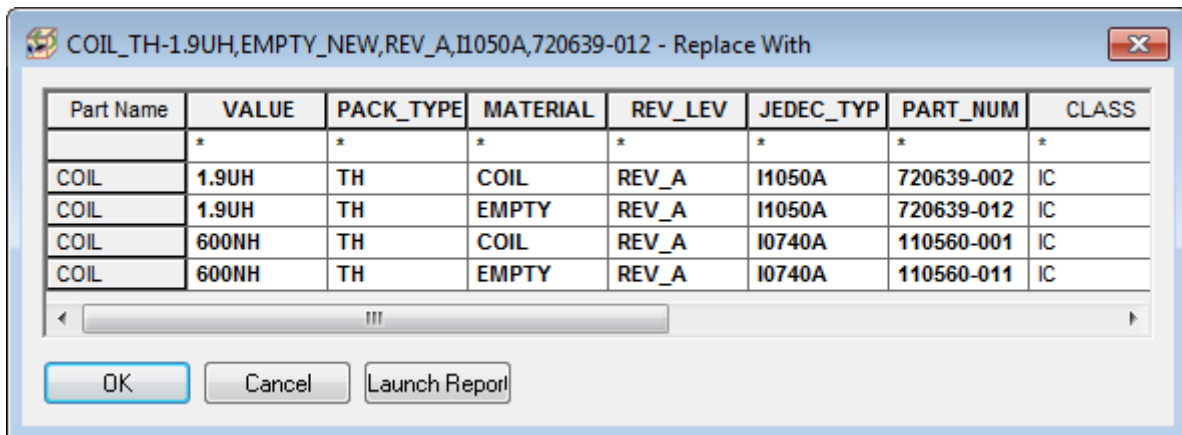
Synchronizing Design Libraries

Part Status	Action Required	Move to
Add prop mismatch	View the differences	Step <u>6</u>
Auto_sync_prop	between the cache and	
Autofixable	reference PTFs	

See “[Correcting the Part Status of the Component](#)” on page 23 for details of all possible values.

4. Replace the cached PTF row with a correct PTF row by doing the following:
 - a. Right-click a PTF row.
 - b. Choose *Replace with* from the pop-up menu.

The *Replace with* dialog appears with a list of the PTF rows for the selected cell.



- c. Select a PTF row in the *Replace with* dialog.
- d. Click *OK*.
- e. Repeat this for all the PTF rows.

When any row is replaced, the status of the PTF changes to *Manually Replaced*.

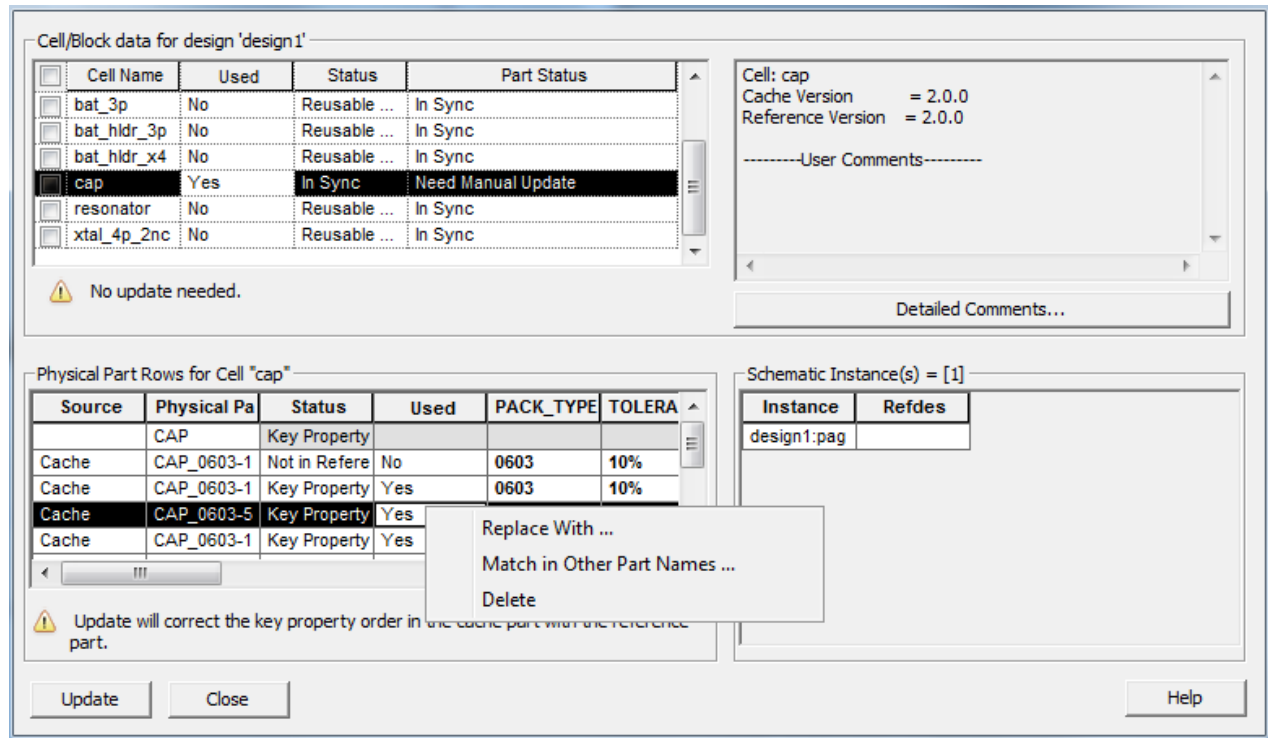
The *Replace with* dialog closes.

5. Change the part name for the PTF row.

Allegro EDM Version Management Utilities Guide

Synchronizing Design Libraries

- a. Right-click a PTF row.
- b. Choose *Match in other Part Name* from the pop-up menu.



Note: This requires Allegro EDM Server access. Note that updating the difference in part name requires DE-HDL to be already launched.

- a. Select a part.
- b. Click *OK*.
- c. Repeat this for all the PTF rows.

Based on the newly selected part, the library and cell information for all the instances also change. Additionally, the instances on the schematic also change.

6. View the differences between the cache and reference PTFs:
 - a. Right-click a PTF row.
 - b. Choose *Show differences* from the pop-up menu.

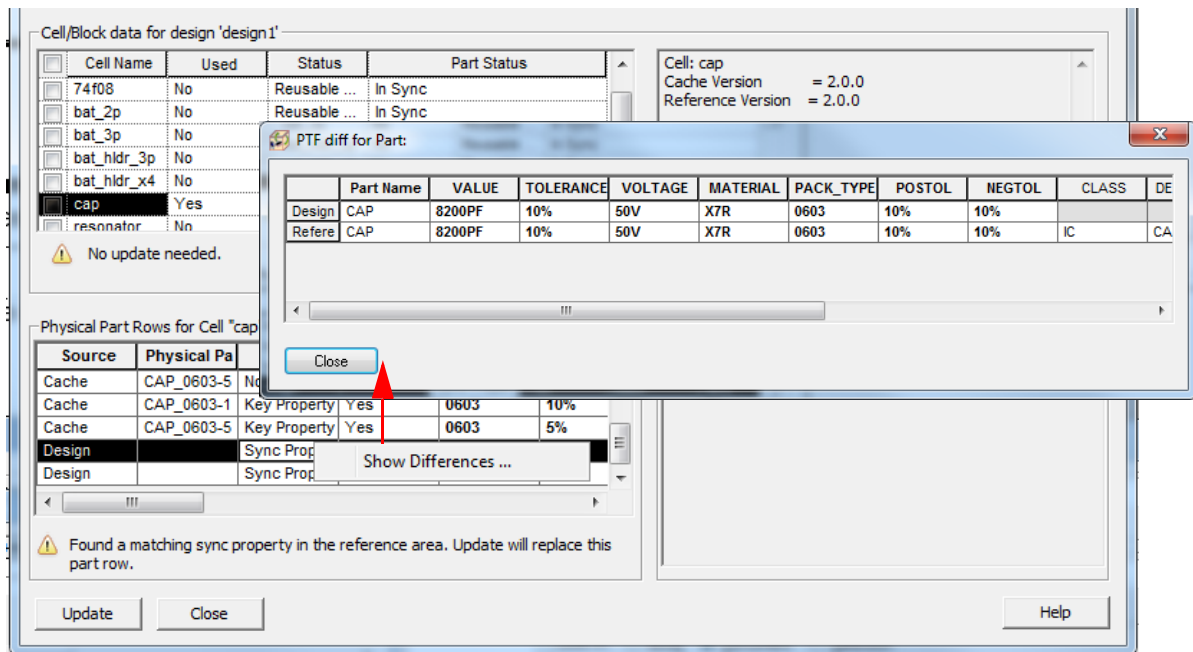
Show Differences comes up only if the `sync_properties` directive is set to TRUE for the project and the Part Status column has one of the following values:

Allegro EDM Version Management Utilities Guide

Synchronizing Design Libraries

- *Autofixable*
- *Added_Property_Mismatch*
- *Auto_Sync_Prop*

The *PTF Diff for Part* dialog appears listing the cache and reference PTF rows for the selected part name.



- c. Click *Close* to close the dialog and return to the *Library Revision Manager* window.

Depending on the extent of differences between cache and reference PTF rows, Library Revision Manager can correct some differences automatically. For such cells, the *Part Status* column is *Autofixable*. You can look at the differences before they are fixed. To see the differences for the cells that are *Autofixable*, use the same steps as listed (step a through step c)

The PTF rows are updated only when you click the *Update* button. The changes might reflect in the *PPT Details* pane as soon as you replace them in the Replace with dialog, but they are saved only when you click *Update*.

7. Repeats step 1 through step 3 for all the components listed in the *Library Revision Manager* dialog.

Remember, step 3 might need you to perform step 4 or step 6 depending on the *Part Status* value of the component.

Allegro EDM Version Management Utilities Guide

Synchronizing Design Libraries

8. Select the check box(es) corresponding to components that you plan to update.
9. Click *Update*.

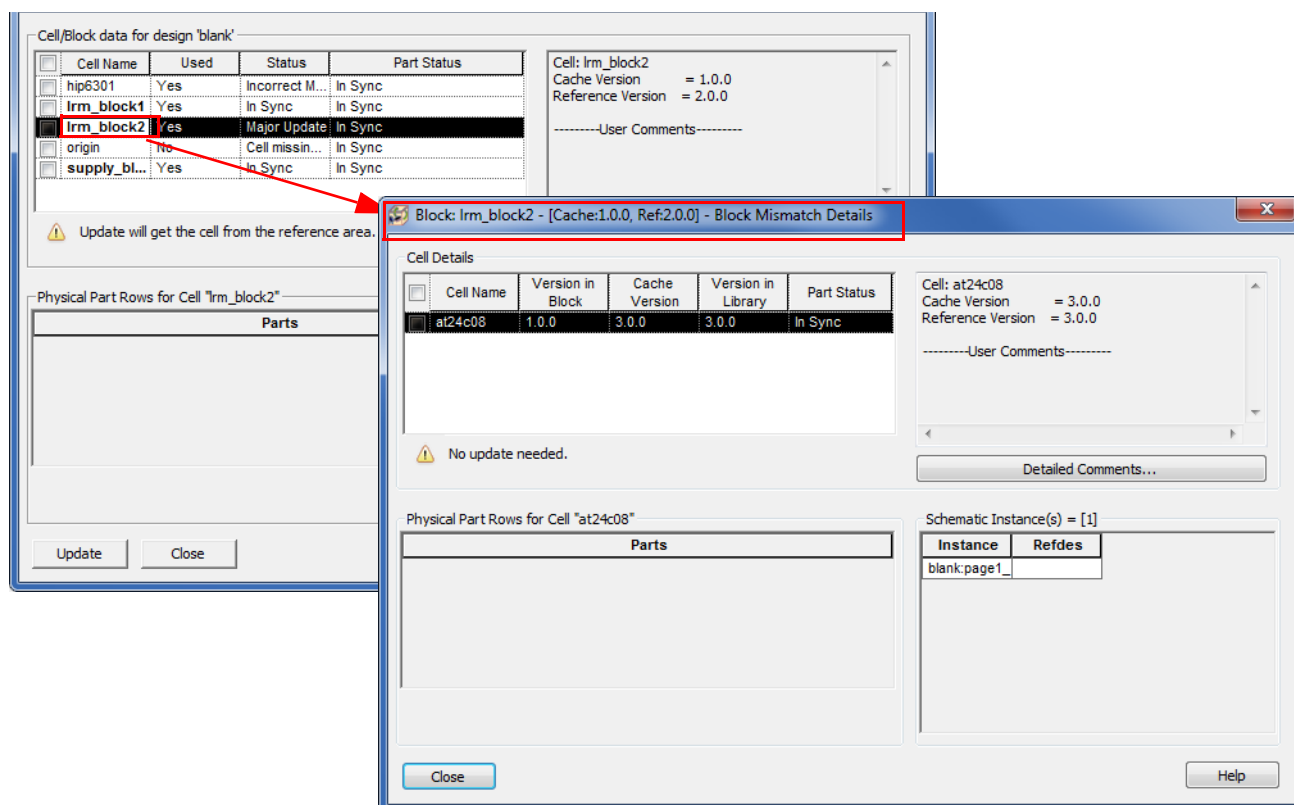
The changes are saved only when you click the *Update* button in the Library Revision Manager dialog.

Updating Blocks

This section describes how to update blocks. The entries in bold contain blocks. To view the block conflict information and update blocks, perform the following tasks:

1. Right-click a component that is in bold.
2. Choose *Block Mismatch Details* from the pop-up menu.

The *Block Mismatch Details* dialog appears.



The layout of this dialog is the same as the already open *Library Revision Manager* dialog.

You can perform the same tasks as step 1 through 8 of the Updating Cells section.

Allegro EDM Version Management Utilities Guide

Synchronizing Design Libraries

Viewing the details of a block is a recursive task that continues until you reach the last nested block.

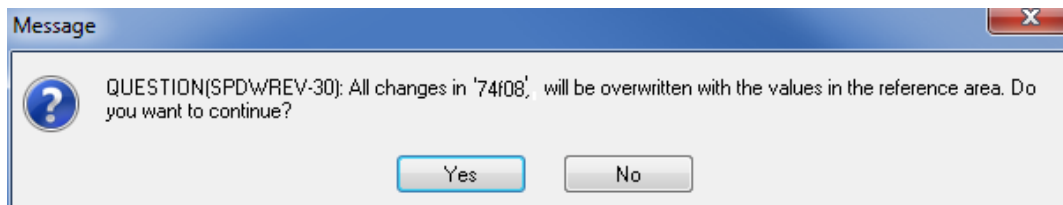
When you click Update, LRM gets the latest version of the cell, for example `LRM_BLOCK2`, from the reference library and updates the design. Also updates the cells used in that block. However if there is any cell which is still using older version in the block that cannot be updated by LRM. The librarian needs to initiate the ECO flow to update that.

Some of the check boxes for cells are checked and you cannot change that selection. This indicates that these cells are part of the block, and will be cached for the first time, therefore these cells cannot be excluded from the block update operation.

After you have specified the options for all blocks and cells, you are ready to start the synchronization of the libraries.

3. Click *Update*.

- ☐ If you try to update a block that has been modified in the cache, a warning appears asking if you want your changes to be overridden by the reference block.



This saves a copy of the cached block in the `temp` directory of your design project.

If you try to update cells that are part of a block, Library Revision Manager alerts you about the blocks that will be affected due to the update.

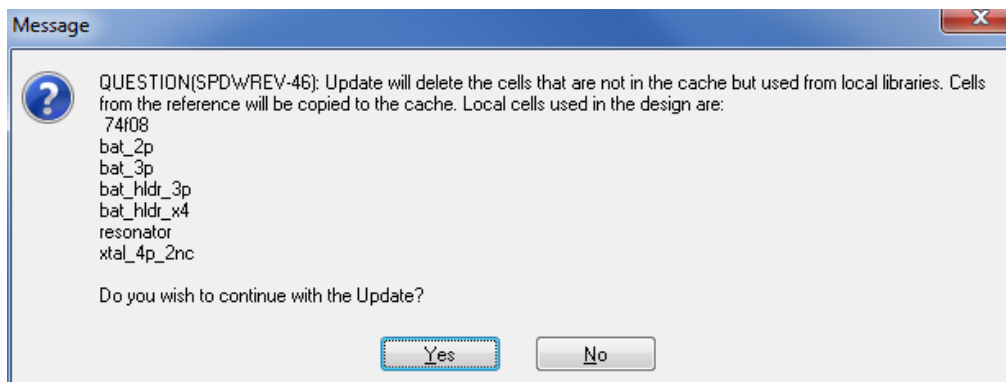
When working with imported blocks in your design, pay special attention to blocks listed.

Depending on your requirements, click *Continue* to go ahead with the update or click *Cancel* to return to the *Library Revision Manager* dialog.

Allegro EDM Version Management Utilities Guide

Synchronizing Design Libraries

- If there are local cells in the design, you will the following message:



4. Click *Yes*.

This completes the libraries synchronization procedure.

Highlighting Component Instances on the Schematic

You can highlight instances on the schematic by performing these tasks in the *Library Revision Manager* dialog:

1. Select a cell in the *Cell/Block Details* pane.
2. In the *Schematic Instances* area, select an instance and right-click it.
3. Choose *Highlight Schematic* from the pop-up menu.

The selected instance of the cell is highlighted on the schematic if Allegro Design Entry HDL is already installed and running.

Reverting to Older Versions of a Library

When the cache and reference libraries are synchronized, a backup of the current state of the library and the history of changes to the cache is maintained by Library Revision Manager.

The history files are in the `atmdir/history_lib` folder of the project directory. If you want to revert to blocks or cells (in the project cache) to any of its previous available versions, use the Rollback utility.

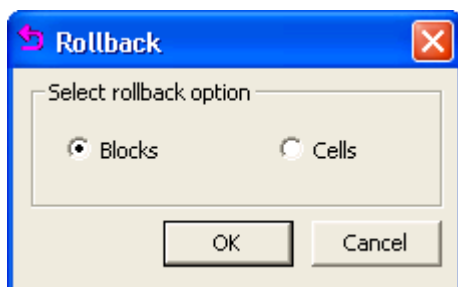
Launching Rollback

Start Rollback in any of these ways:

- Choose *File – Old Models Version Recovery* in Flow Manager.
- Type the following on the Allegro EDM System Console:

```
rollback -proj <path of the project cpm> [-block] [-cell]
```

If you do not specify the arguments (block or cell) in the command, the Rollback dialog prompts you to select the rollback option.



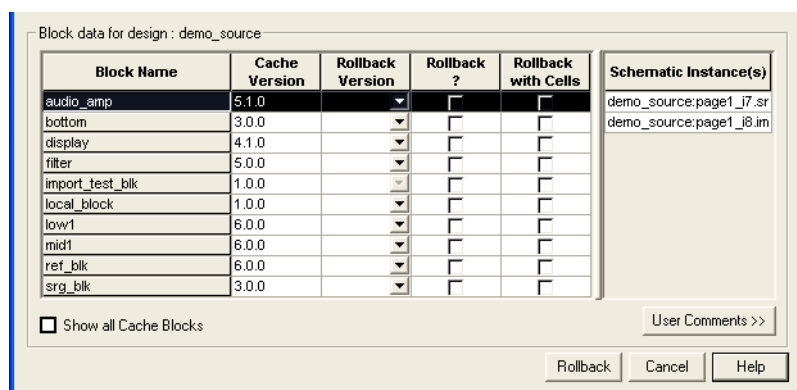
As you can see from this dialog, you can roll back:

- Blocks
- Cells

Allegro EDM Version Management Utilities Guide

Reverting to Older Versions of a Library

For example, if you select *Blocks* and click *OK*, the following screen appears:



Discarding Changes Made to a Block

To revert to an older version of a block:

1. In Flow Manager, choose *File – Old Models Version Recovery – Rollback Block*.

The *Rollback* dialog appears.

2. Select block(s) in the *Cell/Block Details* pane.
3. Select the version to roll back from the drop-down list under the *Rollback Version* column.

To view all the cached blocks in the Rollback dialog, select *Show all Cache Blocks* check box.

4. Select the check box under the *Rollback?* column.
5. If you want to roll back the cells within the block also, select the check box in the *Rollback With Cells* column.
6. Click *Rollback*.

Discarding Changes Made to a Cell

To undo the changes made to a cell:

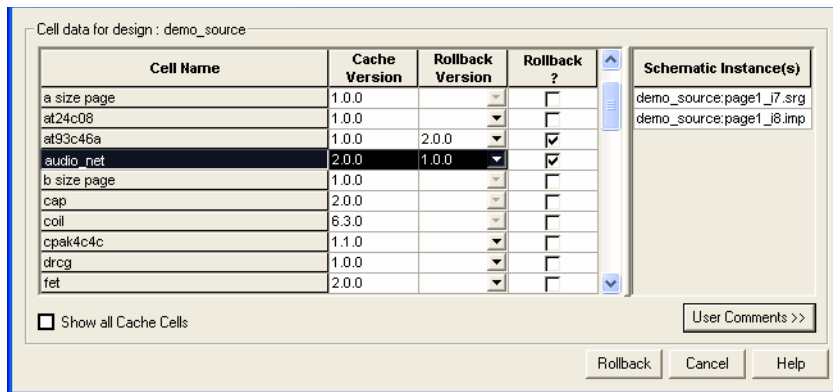
1. In Flow Manager, choose *File – Old Models Version Recovery – Rollback Cell*.

The *Rollback* dialog appears.

Allegro EDM Version Management Utilities Guide

Reverting to Older Versions of a Library

2. Select a cell in the *Cell/Block Details* pane in the Rollback dialog.
3. Select the version to roll back from the drop-down list in the *Rollback Version* column.



4. Select the check box under the *Rollback?* column.
5. Click *Rollback*.

To view all the cached cells in the Rollback dialog, select *Show all Cache Cells* check box.

Allegro EDM Version Management Utilities Guide

Reverting to Older Versions of a Library

Removing Redundant Information from the Cache

In Allegro EDM, all the components that you add to a design are saved (cached) in the cache folder of the design project.

As you update the cache with the latest version of cells (in the form of jar files) from reference libraries, the older versions of the jar files are transferred and stored in the history folder of the design project.

Typically, a project contains both used and unused parts, and as the design becomes complex, the size of the project folder keeps increasing. After you complete the design, you might want to remove all the unused items from the design. In this way, you can synchronize your shopping cart with the latest parts used in the design.

With the availability of the purge feature, you can verify the design to check the blocks, cells and physical parts that are in the cache but are not used by the design. Using this feature, you can:

- Select the design components (cells, blocks, and physical parts) that you want to remove from the cache without affecting the viability of the design
- Reduce the contents of the history folder to a judicious level
- Remove unused parts from the Shopping Cart

Project Directives for Purge

The following table lists the directives for controlling Purge options using the project cpm file.

Allegro EDM Version Management Utilities Guide

Removing Redundant Information from the Cache

Directive	Lets you
<code>purge_block '<TRUE/FALSE>'</code>	Specify whether you want to purge unused blocks in the design.
<code>purge_cell '<TRUE/FALSE>'</code>	Specify whether you want to purge unused cells in the design.
<code>purge_part '<TRUE/FALSE>'</code>	Specify whether you want to purge unused physical parts in the design.
<code>purge_shoppingcart '<TRUE/FALSE>'</code>	Specify whether you want to update the Shopping Cart of your design.
<code>purge_history '<TRUE/FALSE>'</code>	Specify whether you want to purge the history of cells in the design.
<code>purge_depth 'n'</code>	Specify the level (n = 1, 2, 3...) to which you want to remove the history.

Important

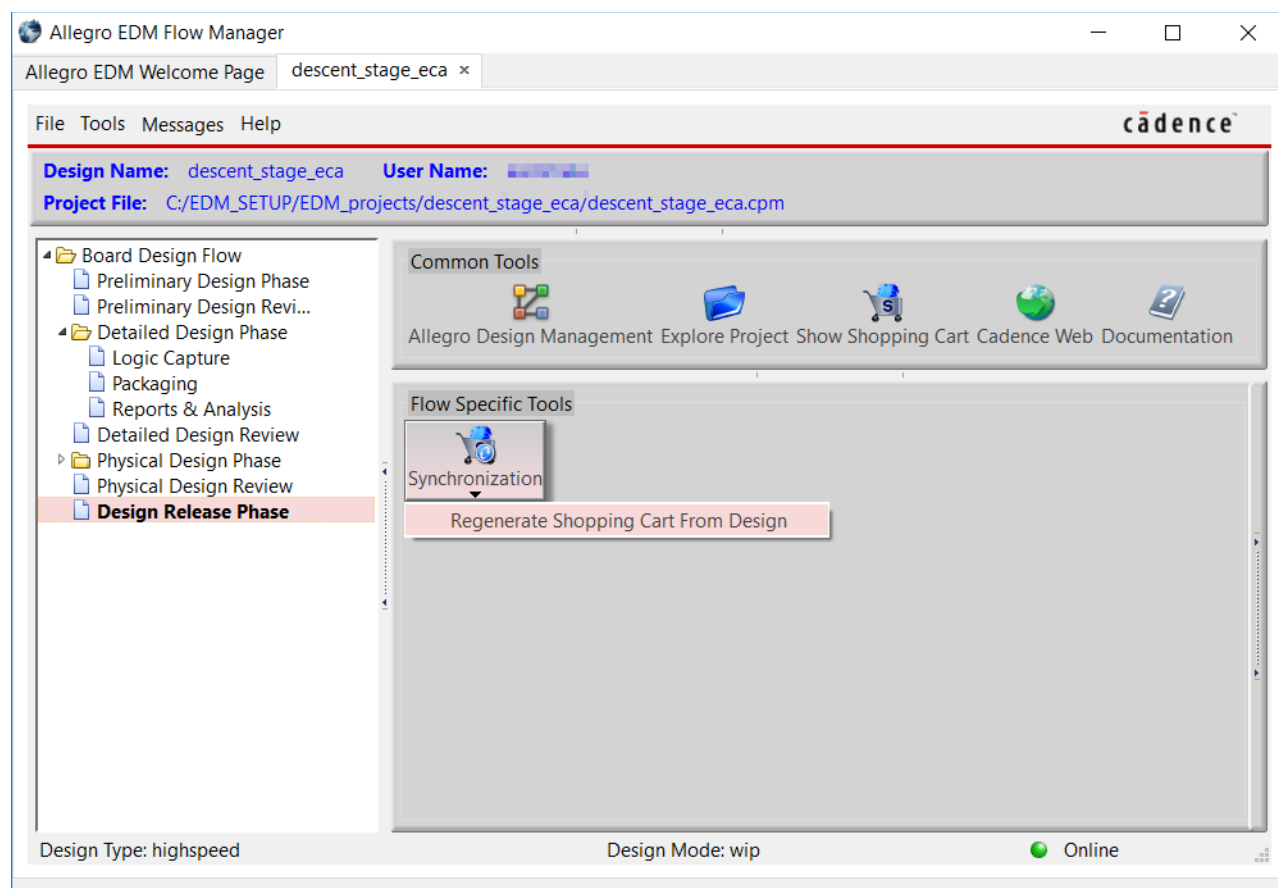
Besides `<TRUE/FALSE>`, you can use standard CPM values, that is, `<YES/NO>` or `<ON/OFF>`.

Starting Purge

You can launch Purge from:

- Flow Manager

In the *Design Release Phase*, choose *Synchronization – Regenerate Shopping Cart* from the *Flow Specific Tools* panel.



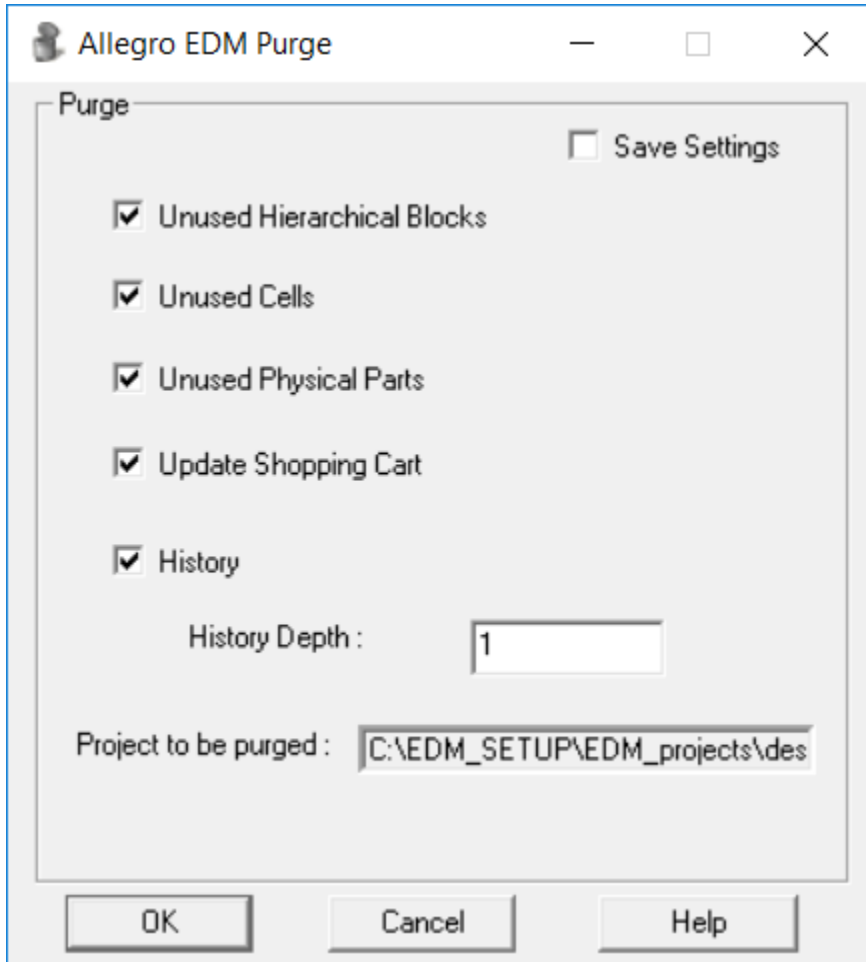
- Allegro EDM System Console

Use the following command at the Allegro EDM System Console:

Allegro EDM Version Management Utilities Guide

Removing Redundant Information from the Cache

```
purge -proj <project cpm> [-product <license string>]
```



■ Run Purge from the Command Line

You can also run Purge as a batch command without the user interface.

```
purge -proj <project cpm> -all -depth 1 -nogui
```

If you want to run Purge to remove unused cells for example, you can run:

```
purge -proj <project cpm> -cells -nogui
```

Specifying What to Remove

The *Purge* dialog contains a simple user interface that allows you to select the desired option or a combination of multiple options and purge unused blocks, cells, physical parts, and

Allegro EDM Version Management Utilities Guide

Removing Redundant Information from the Cache

history with a single-click. In addition, you can also update your shopping cart. The following table explains the user interface elements of the Purge dialog.

Interface Element	Lets you
Save Settings	Set your default purge options in the Purge dialog.
Unused Hierarchical Blocks	Delete all the hierarchical blocks present in the cache but not instantiated on the schematic. It also deletes the associated history files.
Unused Cells	Remove all unused cells, their associated part tables, and associated jar files from the history of the design project.
Unused Physical Parts	Remove all unused physical part table rows from the cache as well as from the cell level part table files (of the cached cells).
Update Shopping Cart	Update the Shopping Cart (the shoppingcart.xml file) whenever you remove a part table row from the cache. As a result, the corresponding details of the part table row are removed from the Shopping Cart.
History	<p>Delete the history (jar) files for the design project.</p> <p>Use the <i>History Depth</i> field to specify the number of versions (major or minor) to delete. For example:</p> <p>Enter 0 to delete all the versions of the history files, including the most current version. Using this clears your history folder completely.</p> <p>Enter 1 to retain the current version and delete all the other versions from the history.</p> <p>Enter 2 to retain the last two versions and delete all the other versions from the history.</p>
Project to be purged	Display the location of the design project (cpm file).
OK	Perform the purge operation using the selected option(s).
Cancel	Cancel the purge operation.
Help	View online help for the Purge dialog.

Allegro EDM Version Management Utilities Guide

Removing Redundant Information from the Cache

Cache-Enabled Projects in Design Entry HDL

You can use Allegro Design Entry HDL to capture the logic of a design as a schematic. The design and its related information is stored in a `<project>.cpm`. The *Design Entry HDL User Guide* explains the Design Entry HDL product, its interface, and the tasks you can perform.

When you launch Design Entry HDL in the Allegro EDM mode, or for an cache-enabled project, you will notice differences in the following:

- [Adding a Block](#)
- [Importing a Block](#)
- [Importing a Sheet](#)

Adding a Block

When you add a top-level block to your design, Library Revision Manager first checks whether all the cells or PTFs for the block are in sync. If the cells and PTFs are in sync, Library Revision Manager caches the block, the block PTFs, cells of the block, and their PTFs and adds the block to the schematic.

If there is a cell version or PTF mismatch, the *Block Mismatch Details* dialog displays.

If you click *OK*, the *Library Revision Manager dialog* prompts you to select the block or cell or the appropriate PTF row to update cache cells or PTF rows with their respective reference library versions, and update the schematic with the changes.

If you choose the cell or block, Library Revision Manager caches the block and the block PTF. Otherwise, the block addition process is canceled.

Importing a Block

When you import a top-level block, Library Revision Manager performs a check on the block and all the lower-level blocks under it. The changed information is displayed, but you cannot make any changes. Based on the displayed information, you can choose to continue or cancel the import.

Importing a Sheet

When you import a design, Library Revision Manager checks for any conflicting blocks, cells, or PTF rows. As a result, after you start the design import, Library Revision Manager displays alerts in the following situations and prompts you to cancel or resume the import:

- Blocks with the same name found in both the open design and the sheet being imported

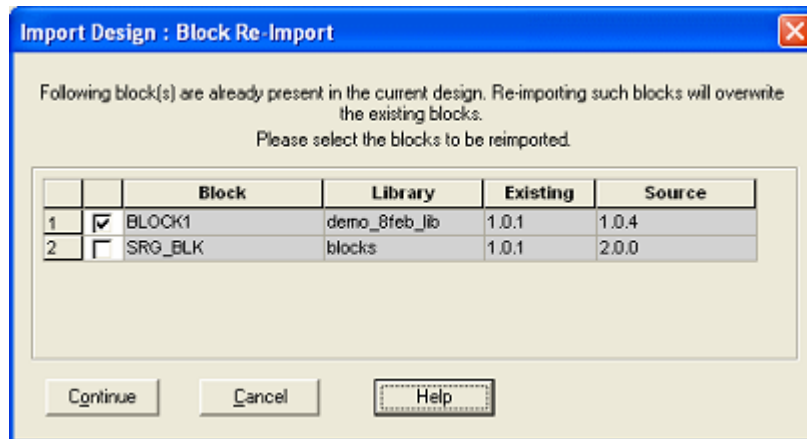
[Managing Identical Block Names in Both Designs](#) has details of how Library Revision Manager displays the conflicts in blocks and prompts you for inputs.

- Sheet contents of the design being imported conflict with components of the destination design

[Resolving Sheet Content Mismatch](#) has details of how Library Revision Manager displays the sheet content conflicts and prompts you for inputs.

Managing Identical Block Names in Both Designs

When blocks in an opened design and the design that is being imported have the same names, Library Revision Manager shows you the names of the blocks and also the version information for each block in the *Block Re-Import* dialog.



When the *Block Re-Import* dialog appears, do the following:

1. Review the version information the blocks in the design.

You can overwrite the blocks in the open design or retain them and ignore the blocks in the design that is being imported.

To overwrite blocks move to step 2 and to retain the blocks move to step 3.

2. Select the check box for the block that you wish to overwrite.

Repeat this for all the blocks that you wish to overwrite.

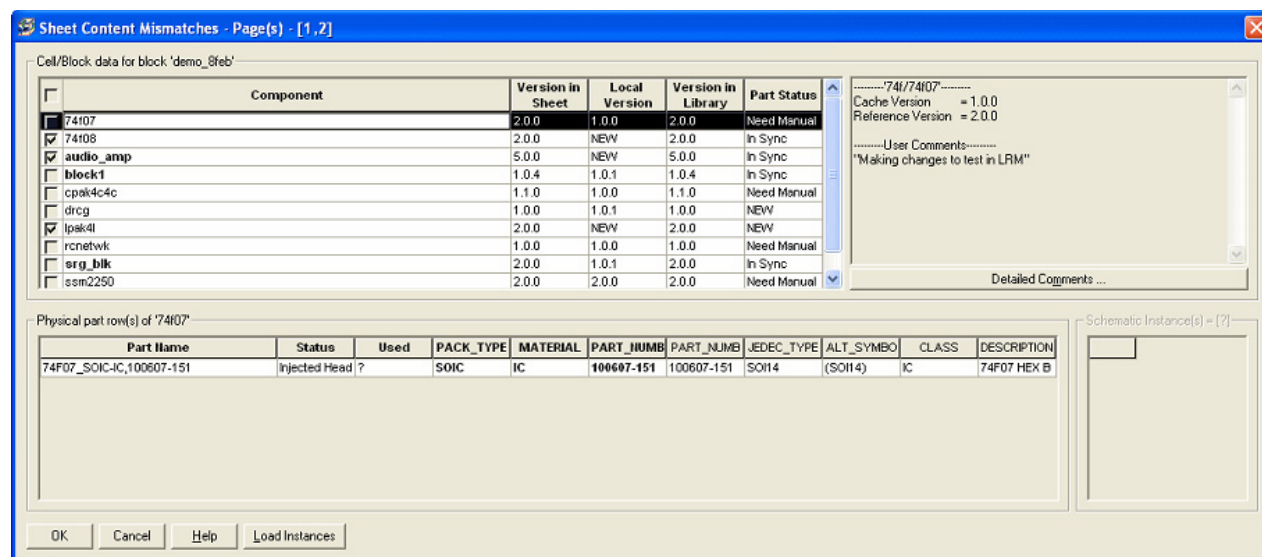
3. Click *Continue* to resume the import or click *Cancel* to stop the import process.

Allegro EDM Version Management Utilities Guide

Cache-Enabled Projects in Design Entry HDL

Resolving Sheet Content Mismatch

If there is any mismatch in the cell/block versions, or the PTF inside the sheet that is being imported and the destination design, the *Sheet Mismatch Details* dialog is displayed.



The *Sheet Mismatch Details* dialog shows the following differences for the cells or blocks:

- Version in sheet
- Local Version
- Version in Library
- Part Status

Note: This is a read-only dialog and you cannot make any changes here.

Here is how the sheet information is processed by Library Revision Manager:

- Sheet Primitives (cells) are copied from the reference library into cache. If a primitive does not exist in reference library, it will not be brought into destination design.
- Similarly, only PTF rows will also be copied from the reference library.

Note: If a sheet primitive or PTF row already in cache folder, that will not be re-cached.

- Sheet block(s) not in the destination project will be cached from the reference library. If the block is not in the reference library, it is copied from the source project into the library that you specify.

Allegro EDM Version Management Utilities Guide

Cache-Enabled Projects in Design Entry HDL

- Re-importing of a block is done only if it is outside cache folder.

Allegro EDM Version Management Utilities Guide

Cache-Enabled Projects in Design Entry HDL

Symbol Revision Manager

Overview

Allegro Symbol Revision Manager (SRM) provides the following functionality for Allegro PCB Editor board designers:

- Synchronizes a PCB Editor board design with a PCB Editor symbol library
- Alerts you to changes made to the symbol library that may affect your design
- Alerts you to footprint symbol changes
- Compares the PCB Editor symbols in your existing design with the symbols in the library
- Displays the librarian's comments in the change log to determine whether you want to upgrade to the latest library versions
- Updates the symbol in your PCB Editor design with a newer symbol in the library. If you need to, you can recover an older version of a symbol.

In the PCB Editor database, each footprint (package symbol in PCB Editor terminology) is tagged with a property called `ALLEGRO_MODEL_REVISION`.

When you launch PCB Editor from Allegro EDM, Allegro Symbol Revision Manager automatically runs the following tasks:

1. Scans the open drawing for symbols that do not contain the `ALLEGRO_MODEL_REVISION` property and reports them as "unknown".
2. Reads the current library paths to locate a `<symbol_name>.status` symbol file, which identifies the symbols' current revision.
3. Opens the Symbol Update Manager for you to manage any symbols that are not up-to-date with the current library. For more information, see [Updating Symbols to Newer Versions](#).

Allegro Symbol Revision Manager operates at the PCB Editor database level, not at the project level. Thus, you can launch it only when you open the PCB Editor database; you

Allegro EDM Version Management Utilities Guide

Symbol Revision Manager

cannot launch it when an Allegro EDM project is opened. For more information on limitations, see [Updating Symbols to Newer Versions](#).

When you change or add symbols to a board design, you can manually run Allegro Symbol Revision Manager by choosing *Place — Symbol Rev Mgmt*.

Note: Cadence supports Allegro Symbol Revision Manager on all supported platforms of release 14.2 and later. Allegro Symbol Revision Manager is limited to revision management of certain types of PCB Editor symbols such as package symbols, mechanical symbols, and format symbols. Each of these may be in a library managed by a librarian, and, as a result, are subject to similar revision controls as package symbols.

Allegro Symbol Revision Manager requires the user-defined property called `ALLEGRO_MODEL_REVISION`, a `<symbol_name>.status` file and a `<symbol_name>.jedec.log` file in the library directories to function properly. On newly imported boards, the `ALLEGRO_MODEL_REVISION` property is not present on footprint symbols. In this case, SRM adds the `ALLEGRO_MODEL_REVISION` property with the value "unknown".

Revisions of padstacks, pad flashes and Flash symbols are not managed by Symbol Revision Manager.

Symbol Versions

Symbol version names are represented by three decimal-separated numbers, such as 1.0.14.

- The first number indicates a major revision — a change to the footprint part that impacts the etch of a board, such as changing pin locations or changing padstacks.
- The second number indicates a minor revision — a change to non-electrical data, such as changing the refdes text size or changing a component outline. Librarians make major and minor revisions.
- The third number indicates the major version of the PCB Editor from which the footprint was built. For example, version 1.0.17.4 indicates that the footprint was built on version 17.4 of PCB Editor.

Version numbers are compared between the version tag on the footprint in the Allegro database and the version tag on the footprint in the Allegro library. Version differences are listed in a report.

Footprint Files

The footprint library of Allegro EDM contains footprint files (.psm).

Each .psm file has associated files that have the same file name but with different file extensions. These associated files contain version information.

For example, the Dip14 footprint symbol has the following files:

- Dip14.dra - PCB Editor source file that contains footprint symbol data.
- Dip14.psm - Contains the compiled version of the Dip14.dra file.
- Dip14.status - The .status file contains the version of the footprint symbol.
- Dip14.log.jedec - This log file contains the unformatted history of footprint changes.

Updating Symbols to Newer Versions

When you launch PCB Editor from Allegro EDM, the Symbol Update Manager appears if the board symbols are not up-to-date with the current library.

When you change or add symbols to a board design, you can manually run the Symbol Update Manager by choosing *Place —Symbol Rev Mgmt — Check Lib for New Version*. The following figure shows a sample Symbol Update Manager. To update each selected footprint in the *To Be Updated* list box, click the *Update Symbols* button.

Allegro EDM Version Management Utilities Guide

Symbol Revision Manager

Lists the name of symbols found in the .brd file.

Shows the revision number of board symbols

Auto move makes selection go to other list with single click.

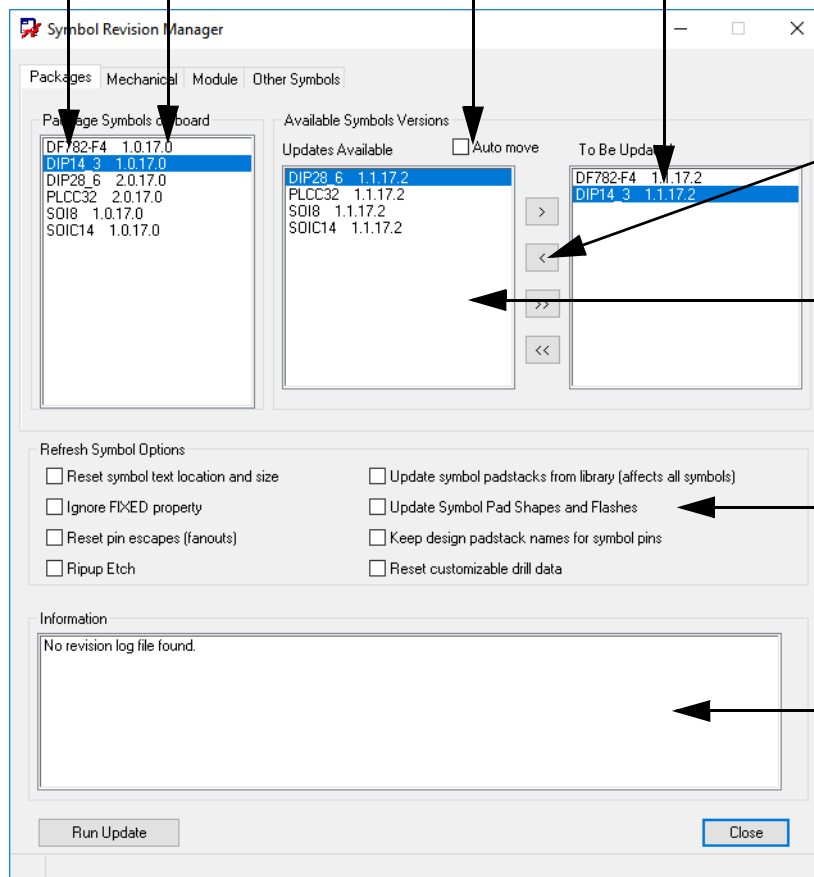
Lists the symbols that are scheduled for update.

You can select one (< and >) or all (<< and >>) symbols.

Lists the symbols that are out-of-date.

Standard refresh options.

The Information window shows progress messages as refresh runs and the contents of the symbol's status file when you select an individual symbol from the Updates Available box.



Updating to the New Footprint

Allegro Symbol Revision Manager preserves symbols before replacing them with newer symbols by saving them to a .jar file. This is so you can recall symbols that have been replaced, if you want to. The .jar file contains the footprint symbol (.dra and .psm), all padstacks used by the footprint, and all the Flash symbols and pad shapes used by the padstacks. The .jar file is named as follows:

```
<symbolName>-<type>-<revision info>.jar
```


where *type* is:

pkg, mech, fmt

Example: dip14-pkg.1.0.17.4.jar

The .jar file is written to: *<same directory that has the .brd file>/old_symbols:*

After the old footprint is safely preserved, Allegro Symbol Revision Manager refreshes the footprint with the newer version from the library. After each symbol is refreshed, the new symbol is tagged with the ALLEGRO_MODEL_REVISION property and the updated revision number.

After you refresh your existing footprint with the new one in the library, Symbol Revision Manager creates the ALLEGRO_MODEL_REVISION property, if not already defined in the current drawing, and attaches the ALLEGRO_MODEL_REVISION property and current symbol revision as the property value to the symbols.



Changes made in symbols placed on the board are not retained when the symbols are updated. This includes silkscreen changes, pin moves, and so on.

Recovering Old Footprint Versions

You can recover an old footprint version by choosing *Place — Symbol Rev Mgmt — Old Model Recovery*. The Symbol Update Manager appears with a list of footprints that are on the board that have corresponding archived files within the project. When you click the *Roll Back Symbols* button, Allegro Symbol Revision Manager does the following for each footprint you select:

1. Unjars the archived file
2. Refreshes the symbol so that the board's symbol is swapped out and the locally resurrected symbol is inserted into its place.

Parameter File

The symbol Revision Manager Parameter (symbolrevchk.par) file allows you to define the visibility and default values for padstack updates, and pad shape/pad Flash updates. The parameter file is located at the *<Installation_Directory>/tools/pcbaw/pcb_tools/pcbstart/archindep/par* directory.

Allegro EDM Version Management Utilities Guide

Symbol Revision Manager

You can copy this file to the `<Allegro EDM Conf Root>/<company>/<site>/pcbstart` directory. The file contains five entries: two for padstack updates, two for pad shape/Flash symbol updates, and one to make it obligatory for designers to update their footprint symbols if the symbols have undergone major revisions. The values set in this file apply to both, the update and rollback functions, in Symbol Revision Manager.

- `_ADW_PAD_UPDATE_ENABLE` - this parameter enables the visibility of the update padstack checkbox in the Update and Rollback dialogs. 't' makes the checkbox visible and 'nil' makes the checkbox invisible.
- `_ADW_PAD_UPDATE_VALUE` - this parameter sets the default value for the padstack update checkbox. 't' selects the padstack update checkbox, and 'nil' unchecks the padstack update box.

Note: If the padstack checkbox is selected but disabled, padstack updates will run.

- `_ADW_PADSHAPE_UPDATE_ENABLE` - this parameter enables the visibility of the update pad shapes and Flash symbols checkbox in the Update and Rollback dialogs. 't' makes the checkbox visible, and 'nil' makes the checkbox invisible.
- `_ADW_PADSHAPE_UPDATE_VALUE` - this parameter sets the default value for the pad shapes and Flash symbols update checkbox. 't' selects the pad shape/Flash symbol update checkbox, and 'nil' unchecks the padstack update box.

Note: If the padstack checkbox is selected but disabled, pad shape and Flash symbol updates will run.

- `_ADW_FORCE_FOOTPRINT_UPDATE_ON_MAJOR_REV` - If this parameter is set to true, that is, if its value is 't', designers are prompted to update their footprint symbols if the symbols have undergone major updates. It will be mandatory for designers to update their footprint symbols.

Details of the symbols refreshed are saved to `refresh_forced.log`, which is available in the same directory as the `.brd` file.

Note: Designers do not need to update their symbols if they have undergone only minor revisions.

- `_ADW_RESET_DRILL_DATA_VALUE` - If this parameter is set to true, that is, if its value is 't', designers are prompted to update or refresh drill customizable data fields in the Drill Customization spreadsheet during a update or refresh of padstacks.

If this parameter is not set, updates or refreshing padstacks deletes any changes previously made to these customizable fields in the Drill Customization spreadsheet.

- `_ADW_RESET_PIN_ESCAPES_ENABLE` - this parameter enables the visibility of the reset pin escapes checkbox in the Update and Rollback dialogs. 't' makes the checkbox visible, and 'nil' makes the checkbox invisible.

Allegro EDM Version Management Utilities Guide

Symbol Revision Manager

- `_ADW_RESET_PIN_ESCAPES_VALUE` - If this parameter is set to true, that is, if its value is 't', predefined pin escapes from the symbol are reset.
- `_ADW_RIPUP_ETCH_ENABLE` - this parameter enables the visibility of the ripup etch checkbox in the Update and Rollback dialogs. 't' makes the checkbox visible, and 'nil' makes the checkbox invisible.
- `_ADW_RIPUP_ETCH_VALUE` - If this parameter is set to true, that is, if its value is 't', etch associated with symbol pins is removed when symbols are refreshed.
- `_ADW_PRESERVE_PADSTACK_NAMES_ENABLE` - this parameter enables the visibility of the preserve padstack name checkbox in the Update and Rollback dialogs. 't' makes the checkbox visible, and 'nil' makes the checkbox invisible.
- `_ADW_PRESERVE_PADSTACK_NAMES_VALUE` - If set to 't', symbol pins will maintain their padstack names when they are refreshed.
- `_ADW_RESET_SYM_TEXT_ENABLE` - this parameter enables the visibility of the reset symbol text checkbox in the Update and Rollback dialogs. 't' makes the checkbox visible, and 'nil' makes the checkbox invisible.
- `_ADW_RESET_SYM_TEXT_VALUE` - If this parameter is set to true, that is, if its value is 't', the symbol text and size are reset based on how they are defined in the symbol definition instead of how they are defined in your design, if different.
- `_ADW_IGNORE_FIXED_ENABLE` - this parameter enables the visibility of the ignore fixed checkbox in the Update and Rollback dialogs. 't' makes the checkbox visible, and 'nil' makes the checkbox invisible.
- `_ADW_IGNORE_FIXED_VALUE` - If this parameter is set to true, that is, if its value is 't', SRM replaces a symbol to which the FIXED property has been assigned.

Limitations of Symbol Revision Manager

- Update and recovery of symbols does not retain symbol edits. Writing out the symbol definitions means that local edits are lost; this includes silkscreen changes, pin moves, and so on.
- Refreshes do not retain symbol edits. Local edits to pin locations, silkscreen, and so on are lost when the symbol is replaced by the newer version from the library.
- Padstacks, Flash symbols and pad shapes are not under revision control

Allegro EDM Version Management Utilities Guide

Symbol Revision Manager

Library Verification

The Library Design Verification program is used to run a set of predefined verification rules on an active schematic model.

Launching Library Design Verification

You can launch the library design verification program in one of the following ways:

- a. Open a library project.
- b. Create a new schematic model for the project or check out an existing schematic model.
- c. Create or edit the schematic model.
- d. Click the *Verify Symbol* button in the Flow Specific Tools pane.

The Library Design Verification window appears.

Alternatively:

- a. Open the required library project.
- b. Check out the required schematic model.
- c. Open the Allegro EDM System Console.
- d. Enter the `library_verif` command.

The Library Design Verification window appears.

Library Design Verification User Interface

The user interface is divided into the following sections:

- ❑ *Cell*: This field displays the schematic model to be verified.

Allegro EDM Version Management Utilities Guide

Library Verification

- ❑ *Setup*: Clicking this button allows you to review or configure the verification rules to be run.
- ❑ *DRC*: This section displays the number of errors, warnings, and oversights found while running the Design Rule Checks (DRC). (Rules Checker)
- ❑ *Report*: Clicking this button opens the DRC report.
- ❑ *Status*: This section displays a message indicating whether or not the information displayed is up-to-date or was generated during a previous session.

Note: By default, you can only view the already configured rules. Only the library administrator can change the configuration settings.

Working with Library Design Verification

1. Click the *Run* button.

The Verification of: <cell_name> window appears.

2. Click *OK* when the verification process is complete.
3. Click the *Report* button.

The log file opens.

When you run this verification program, the verification data and report files are available at the following location:

<Allegro
EDM_project_directory>\<library_project>\flatlib\model_sym\<library_name>\<cell
_name>\atdm_verif

The Library Design Verification program generates a marker file (.mkr) that can be used in Design Entry HDL to graphically indicate each error in the design. To use this file, do the following:

1. Open Design Entry HDL.
2. Choose *Tools – Markers – Load*.

The Markers window appears.

3. Choose *File – Load*.

The Load Marker File dialog box appears.

4. Navigate to the atdm_verif folder to open the .mkr file.

Allegro EDM Version Management Utilities Guide

Library Verification

5. Click the *Next Marker* button to view each error.

Allegro EDM Version Management Utilities Guide

Library Verification

Design Verification

The Schematic Design Verification program is used to apply a set of predefined verification rules on the schematic design or on a schematic block.

Launching Schematic Design Verification

You can launch the schematic design verification program in one of the following ways:

- a. Open the required design project.
- b. Choose Detailed Design Phase – Logic Capture.
- c. Choose Verification – Automatic Mode from the Flow Specific Tools pane.

The Allegro Design Workbench: Schematic design verification window appears.

Alternatively:

- d. Open the required design project.
- e. Open the Allegro EDM System Console.
- f. Enter the `verif` command.

The Allegro Design Workbench: Schematic design verification window appears.

Schematic Design Verification User Interface

The user interface is divided into the following sections:

- ❑ *Design block*: This drop-down combo box allows you to select the design block you want to check.
- ❑ *params*: Clicking this button allows you to review or configure the verification rules to be run.

Types of checks:

Allegro EDM Version Management Utilities Guide

Design Verification

- DRC: Design Rules Checks. The Rules Checker utility runs these verification rules.
- PTF Synchronization: This will run LRM on the selected design.
- PXL: Packager-XL, which is used for packaging verification
- ERC: Electrical Rules Checks

The number of errors, warnings, and oversights corresponding to each type of check is displayed after the verification process is complete.

Report: Clicking this button opens the respective verification report.

Status: This section displays a message indicating whether or not the information displayed is up-to-date or was generated during a previous session.

Note: By default, you can only view the already configured rules. Only the library administrator can change the configuration settings.

Working with Schematic Design Verification

1. Choose the design block for which you want to run the verification process.
2. Click the Run button.

The Verification of: <design_block_name> window appears.

3. Click *OK* when the verification process is complete.
4. Click the *Report* button for each type of check.

The corresponding log file opens.

The name and location of the log files for each check are as follows:

- DRC: <Allegro EDM_project_directory>\<design_project>\worklib\<design_block_selected>\atdm_verif\cp.log
- PTF: <Allegro EDM_project_directory>\<design_project>\atdmdir\logfiles\LRM_Update.log
- PXL: <Allegro EDM_project_directory>\<design_project>\worklib\<design_block_selected>\atdm_verif\pxl.log

Allegro EDM Version Management Utilities Guide

Design Verification

- ERC: `<Allegro EDM_project_directory>\<design_project>\worklib\<design_block_selected>\atdm_verif\erc.rpt`

When the Schematic Design Verification program runs DRC (Rules Checker), it generates a marker file (`.mkr`) that can be used in Design Entry HDL to graphically indicate each error in the design. To use this file, do the following:

1. Open Design Entry HDL.
2. Choose *Tools – Markers – Load*.
The Markers window appears.
3. Choose *File – Load*.
The Load Marker File dialog box appears.
4. Navigate to the `atdm_verif` folder to open the `.mkr` file.
5. Click the Next Marker button to view each error.

Allegro EDM Version Management Utilities Guide

Design Verification

User Interface

This section explains the following Library Revision Manager interface controls.

- [User Interface](#)
- [Block Mismatch Details Dialog](#)
- [Block Re-Import Dialog](#)
- [Sheet Content Mismatches Dialog](#)

Library Revision Manager Dialog

The Library Revision Manager dialog has the following parts:

- Cell/Block Details pane
- Libraries and version information pane
- Comments area
- PTF Details pane
- Schematic Instances area

Field	Description
Cell/Block data for design <design name>	<p>Displays the cells and blocks in the cached libraries and the design that do not match their versions in the reference libraries. The cache part table row (<i>part_table.ptf</i>) and the reference library for the rows gets used to annotate values to the cell instances in the block.</p> <p>The grid contains the following column headers:</p> <p>Cell name: Lists the name of the cell or block. The column shows cells (in cache) present in the block selected.</p> <p>If a cell is used only in a block and not at the top-level design, it will also appear in the grid if it has a version or ptf mismatch.</p> <p>Entries in bold denote that the cell contains blocks. The cell in the block has a version mismatch with the reference or cache. You can right-click and select <i>Block Mismatch Details</i> to view those details. This opens another dialog that has the same columns as the Library Revision Manager window. See Block Mismatch Details Dialog for more details.</p> <p>Used: Shows if the cell is used in the design or not.</p> <p>Status: Shows the Cell Status value.</p>

Allegro EDM Version Management Utilities Guide

User Interface

Field	Description
	<p>Part Status: Indicates whether the physical data for the cell/block is in sync with the reference libraries. The possible values are:</p> <ul style="list-style-type: none">■ In Sync The reference and cache PTFs match. No change is needed. Selecting a cell with In Sync status will not display anything in the <i>Physical Part Rows for Cell <selected cell></i> pane.■ Autofixable The cell/block PTFs in the cache do not match the reference. LRM can correct the mismatch based on the sync properties.■ Needs manual update The cell is in sync but the:<ul style="list-style-type: none"><input type="checkbox"/> PTF is found in the local library but not found in the cache or reference library.<input type="checkbox"/> PTF is found in the local library, nor in the cache, but exists in the reference library. <p>LRM cannot fix these mismatches and you need to specify what action to take. See Part Status.</p> <p>A description of the selected cell/block row is displayed. The description includes what will be done if Update is selected. Or, if a manual update is needed.</p>
Comments area	<p>Displays the cache, reference, and local version information about the cell selected. It also lists the differences between these two versions.</p>

Allegro EDM Version Management Utilities Guide

User Interface

Field	Description
Physical Part Rows for Cell <selected cell>	<p>Shows the following:</p> <ul style="list-style-type: none">■ Part table rows used in the selected cell that do not match the reference library for the cell selected in the <i>Cell/block data for design</i> pane.■ PTFs used by design that are from a local library and are missing from the cache and/or reference libraries.■ Part table rows with manually annotated properties. <p>In addition to the information under the property headers of the part table, the Status column displays the status of the physical row with respect to the reference library data.</p> <p>The columns for each part table row are based on the PTF. Source and Status are always there.</p> <p>A description of the selected PTF row is displayed. The description includes what will be done if you click Update.</p>
Schematic Instances area	Lists all of the instances of the cell (in the design) selected in the Cell/Block Details pane.
Update	Based on the selections made, LRM makes the changes in the design libraries.
Close	Closes the LRM window.
Help	Launches the Cadence Help and displays information related to the window.

Cell Status

The [Implication of the Cell Status](#) table summarizes the different values for the Cell Status column in the *Library Revision Manager* dialog along with their descriptions.

Table A-1 Implication of the Cell Status

Status	Indicates that
<i>Cell in different library</i>	The cell is present in different libraries in the cache and reference.

Allegro EDM Version Management Utilities Guide

User Interface

Status	Indicates that
<i>Cell missing in Reference Error</i>	<p>The cell is in the cache but not in the reference library.</p> <p>The cell has an unknown error that Library Revision Manager cannot identify.</p> <p>For example, one possible case, cell metadata is missing from the cache as well as from the reference library.</p>
<i>In Sync</i>	<p>The cell in the cache and reference libraries are identical.</p>
<i>Incorrect Metadata</i>	<p>The cell has been modified manually (possibly without using Part Developer) resulting in inconsistencies in the cell metadata.</p>
<i>Local cell/block</i>	<p>Cell/block is not there in the reference libraries but is used in the design. This could be from an entry in the CDS.LIB other than the cache.</p>
<i>Major Update</i>	<p>The cell has undergone a considerable change. For example, a pin could have been added, deleted, or moved within a symbol. The changes in the cell can impact packaging.</p>
<i>Minor Update</i>	<p>The cell has undergone a small change. For example, a change in the text of a reference designator. This usually does not impact the packaging data in the design.</p>
<i>Modified in Cache</i>	<p>The cache cell (under the project library) has been updated (for example, using Part Developer) by the librarian.</p>
<i>Modified in Reference</i>	<p>The reference cell has been modified (for example, using Part Developer).</p>
<i>New</i>	<p>A new cell has been added to the cache. Such cases arise when you import a block or add a block to your design.</p>
<i>Nonreadable cache metadata</i>	<p>Library Revision Manager is unable to read cell-level metadata from the cache. As a result, the correct status is unavailable.</p>

Allegro EDM Version Management Utilities Guide

User Interface

Status	Indicates that
<i>Nonreadable reference metadata</i>	Library Revision Manager is unable to read cell-level metadata from the reference library. As a result, the correct status is unavailable.
<i>Reusable cell/block</i>	Cell/block is from an entry in the cds.lib other than the cache. And is also found in the reference library.

Part Status

The [Implication of Part Status](#) table summarizes the different values for the *Part Status* column in the *Library Revision Manager* dialog along with their descriptions.

Table A-2 Implication of Part Status (Sheet 1 of 2)

Part Status	Indicates that
<i>In Sync</i>	The part in the cached and reference libraries are identical and no difference exists.
<i>Need Manual Update</i>	The part is not autofixable. You need to manually update it.
<i>Missing in Ref.</i>	The part is not there in the reference PTF.
<i>NEW</i>	A new part will be added to the cache. Such cases arise when you import a block or add a block to your design. Parts that are used in the blocks but are missing from the cache have this status.
<i>Autofixable</i>	The part rows in the cache can be autofixed. This fix can be based on either minor mismatch (injected property mismatch) or the <i>sync_properties</i> directive. You can see the corresponding part row (with which it is going to be replaced) using the <i>Show Differences</i> context menu option.
<i>Block PTF</i>	The part has been added to the cache, as a result of block addition or import. It shows that the new part is being used in the block being imported or added to the design.
<i>Key Value Mismatch</i>	There is a key property value mismatch between the cache and reference parts.

Allegro EDM Version Management Utilities Guide

User Interface

Table A-2 Implication of Part Status (Sheet 2 of 2)

Part Status	Indicates that
<i>Injected Value Mismatch</i>	There is an injected property value mismatch between the cache and reference parts.
<i>Added Property Mismatch</i>	<p>There is an added property mismatch between the cache and reference PTFs. Some cases in which this arises are:</p> <p>Added property is available in cache PTF but missing in reference PTF.</p> <p>Added property is available in reference PTF but missing in cache PTF.</p> <p>Value of the added property in the cache and reference PTFs is different.</p>
<i>Key Header Mismatch</i>	The key property headers between the reference and cache PTF rows are different.
<i>Injected Header Mismatch</i>	The injected property headers between the reference and cache PTF rows are different.
<i>AUTO_Sync_Prop</i>	<p>The PTF row property can be automatically updated using the <i>sync_properties</i> directive defined in the project file.</p> <p>The properties defined in this directive are used as a sensitivity list. The mismatched cache PTF row is compared with the other reference PTF rows. If a reference PTF row matches, it is replaced with the cache PTF row. There is no need for manual intervention.</p>
<i>Manually Replaced</i>	You have changed the old part with a new part (using the <i>Replace with</i> context menu option)

Block Mismatch Details Dialog

The dialog is similar to the User Interface and contains the following areas:

- Cell/Block Details pane
- Comments area
- PPT Details pane
- Schematic Instances area

Block Re-Import Dialog

The Block Re-Import dialog comprises the following user interface elements:

Field	Description
Block	Displays the names of the blocks that mismatch in the design that is open and the one that is being imported.
Library	Displays the name of the library where the block is.
Existing	Shows the version of the cell/block in the open (destination) design.
Source	Shows the version of the cell/block in the design from where the block is being imported (source).
Continue	Resumes the import process.
Cancel	Stops the design import.
Help	Launches the Cadence Help and displays information related to the window.

Sheet Content Mismatches Dialog

The *Sheet Content Mismatches* dialog shows the same content as the *Library Revision Manager* dialog with the exception of the command buttons.

Field	Description
Block/Cell Details pane	<p>Contains the blocks or cells that are instantiated at least once, on the schematic. The grid contains the following column headers:</p> <ul style="list-style-type: none">■ Cell/Block Name: Displays the name of the cell or block that have a mismatch■ Version in Sheet: Shows the version of the cell in the design which is being imported■ Local Version: Shows the version of the cell in the currently open design.■ Version in Library: Shows the version of the cell in the library.■ Part Status: See Part Status.
Comments area	Displays the cache and reference version information about the cell selected in the Cell Details pane. It also lists the differences between these two versions.
Detailed Comments button	Displays user comments between the cached version and the rollback version of the selected block or cell.
Physical Part Rows pane	<p>Lists the part table rows used in the design that do not match the reference library for the cell selected in the Cell Details pane.</p> <p>In addition to the information under the property headers of the part table, the Status header displays the status of the physical row with respect to the reference library data.</p>
Schematic Instance(s) area	<p>Lists all the block or cell instances of the selected block or cell in the Block/Cell Details pane.</p> <p>For multiple instances of a block, the schematic linkages for the cells it contains appear only once.</p>
OK	Resumes the sheet import process.
Cancel	Cancels the import and returns to the Import Design dialog.

Allegro EDM Version Management Utilities Guide

User Interface

Field	Description
Help	Launches the Cadence Help and displays information related to the window.

Rollback

The Rollback dialog comprises the following user interface elements:

Allegro EDM Version Management Utilities Guide

Rollback

Field	Description
Block/Cell Details pane (Depends whether you have blocks or cells to rollback)	<p>Contains the blocks or cells that are instantiated at least once, on the schematic. The grid contains the following column headers:</p> <ul style="list-style-type: none">■ Cell/Block Name: Lists the name of the block or cell. The column will show cells or blocks in the project cache.■ Cache Version: Lists the version of the block or cell in the project cache.■ Rollback Version: Choose the version you want to revert. The drop-down box lists all the previous versions available for the selected block or cell.■ A disabled drop-down box indicates the absence of the cached version.■ Rollback: Check this next to the block or cell to roll back. the options under this column indicate either of the following:<ul style="list-style-type: none">■ Cached and rollback versions are same■ No rollback version is available■ Rollback With Cells: Check this to roll back cells of the blocks.■ This column is available only when you launch rollback in the block mode.■ Show All Cache Blocks or Cells: Select this to list all the cached blocks or cells in the Block/Cell Details pane.■ Blocks or cells that do not belong to the cache appear italicized in the grid.■ Detailed Comments: Click this to view system-generated message history for the selected block or cell.
User Comments button	<p>Displays user comments between the cached version and the rollback version of the selected block or cell.</p> <p>To view user comments, the rollback version should be lower than the current cached version. If the cached version is greater than the version listed under the <i>Rollback?</i> column, the differences do not appear.</p>

Allegro EDM Version Management Utilities Guide

Rollback

Field	Description
Schematic Instance(s) area	<p>Lists all the block or cell instances of the selected block or cell in the Block/Cell Details pane.</p> <p>For multiple instances of a block, the schematic linkages for the cells it contains appear only once.</p>
Rollback	Initiates the rollback on the selected block or cell.
Cancel	Closes the <i>Rollback</i> dialog.
Help	Displays the online help for the rollback feature.

Purge

The Purge utility helps you synchronize the shopping cart with your design.

Field	Description
Save Settings	Sets the options that you select as the default.
Unused Hierarchical Blocks	Deletes all the hierarchical blocks that are present in cache but not instantiated on the schematic. It also deletes the associated history files.
Unused Cells	Removes all unused cells, their associated part tables, and associated jar files from the history of the design project.
Unused Physical Parts	Removes all unused physical part table rows from the cache as well as from the cell level part table files (of the cached cells).
Update Shopping Cart	Updates the shopping cart (<code>shoppingcart.xml</code> file) whenever you remove a part table row from cache. As a result, the corresponding details of the part table row are removed from the shopping cart.
History	<p>Deletes the history (jar) files for the design project.</p> <p>Use the <i>History Depth</i> field to specify the number of versions (major or minor) to be deleted. For example:</p> <p>Enter 0 to delete all the versions of the history files, including the most current version. Using this will empty your history folder completely.</p> <p>Enter 1 to retain the current version and delete all the other versions from the history.</p> <p>Enter 2 to retain the last two versions and delete all the other versions from the history.</p>
Project to be purged	Displays the location of the design project (cpm file)
OK	Starts the purge operation using the selected option(s).
Cancel	Closes the <i>Purge</i> dialog.

Allegro EDM Version Management Utilities Guide

Purge

Field	Description
Help	Displays the online help for the purge feature.
