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 seg. or its successor.

## **Contents**

Duefees	
<u>Preface</u>	
About This Guide	
Related Documentation	
Related Tools and Flows	
Typographic and Syntax Conventions	9
<u>1</u>	
	11
Overview of Database Administration	11
How Database Entities are Related	12
<u>Model Types</u>	
<u>Libraries</u>	
Preferred Parts List (PPL)	
<u>Lifecycle Status</u>	
<u>User Accounts</u>	14
Launching Database Administrator	
Flow Manager	
Command Line	
<u>2</u>	
= <u>Managing Classifications</u> <sub>1</sub>	19
Understanding Allegro EDM Classification	
Classification Versioning	
Hierarchical Classifications and Properties Behavior	
·	
Creating Child Classifications       2         Revising Classifications       2	
Revising Hierarchical Classifications	
Updating Classifications	
Working with Classification Properties	
Adding Properties	25

Editing Properties
Deleting Properties
Working with Predefined Values28
Working with Schematic Model Properties
Adding Schematic Model Properties29
Editing Schematic Model Properties30
Deleting Schematic Model Properties
Modifying Classification Property Order for Display in Search Results
Setting Property Display Order
Copying Property Display Order 34
Exporting Property Display Order
Importing Property Display Order
Resetting Property Display Order
Highlighting Classifications with Modified Display Order
Working with Obsolete Classifications
Marking a Classification Obsolete
Viewing Obsolete Classifications
Revising Obsolete Classifications40
Copying Classifications
Renaming Classifications
Deleting Classifications
<u>3</u>
Working with Allegro EDM Entities 43
Managing Component-Specific Entities
Creating Entities43
Editing Entities
<u>Deleting Entities</u>
Attributes and Relations
Working with Attributes47
Working with Relations49
<u>4</u>
Model Management Support 53
Adding a New Model to Allegro EDM

	_
Creating an Adapter for New Models56Return Values for Adapter Calls58Supported Commands/Modes58Location for New Adapters69Using a New Model Type69	5 5 0
<u>5</u>	
Managing Lifecycle Status of Parts	3
<u>6</u>	
Managing User Accounts 6	7
Creating User Accounts	
Editing User Attributes	8
Deleting Users	8
Changing User Password	8
Searching Users and Roles	9
7	
Verifying Objects with Rule Manager	1
List of Part and Model Rules	
List and Description of Part Rules	
List and Description of Model Rules80	
List and Description of Global Rules8	5
Default Definitions of Rule Sets80	
Creating a Rule Set	
Merging Rule Sets 89	9

<u>8</u>
Importing PPT Option Sets
···· <del>·································</del>
Α
Database Administrator User Interface
<u>Menu Help</u> 9
<u>File Menu</u> 99
View Menu
Search Menu90
Tools Menu9
Window Menu9
<u>Tabs</u> 98
Search Tab98
<u>Details Tab</u> 98
Search Results Tab99
Log Report Tab99
Error Tab
Classification Tabs
Part Classification Tab99
Block Part Classification Tab
Mechanical Part Classification Tab
Model Classification Tab
Dialog Box Help
New Dialog Boxes
Open Dialog Boxes
Add Dialog Box
Delete Property Dialog Box109
Set User Password
<u>B</u>
About Attributes and Relations11
Default Attributes
Default Relations
<u>Doladit Holadolio</u>

6

<u>C</u>	
Searching Database Entities	. 117
Searching the Allegro EDM Component Database	. 117
Search-Related Procedures	. 118
Working with Search Results	. 123
Viewing Recently Visited Details Tabs	. 124

8

## **Preface**

#### **About This Guide**

The Allegro® EDM Database Administrator User Guide explains how an Allegro Engineering Data Management (EDM) administrator or librarian can create and manage component-specific entities (such as Model Type, Library, and PPL), and user entities.

#### **Related Documentation**

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

- Allegro EDM Library Flow User Guide
   Describes the library and library components flow offered by Allegro EDM.
- Allegro EDM Database Editor User Guide
   Describes the other Allegro EDM library components management tasks.

#### **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 *PCB Design Flows*.
- For learning how to create and configure Design Entry HDL projects, see the *Project Manager User Guide*.

## Typographic and Syntax Conventions

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

literal	Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names.
argument	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.

1

# Getting Started with Database Administrator

## **Overview of Database Administration**

Allegro EDM is meant to streamline the creation and availability of parts for EDA designers and librarians. In a nutshell, ECAD designers access component parts from the Allegro EDM Component Database. Each part gets to the database after testing and verification. Librarians are responsible for ensuring that the parts available to the designers are easily searchable.

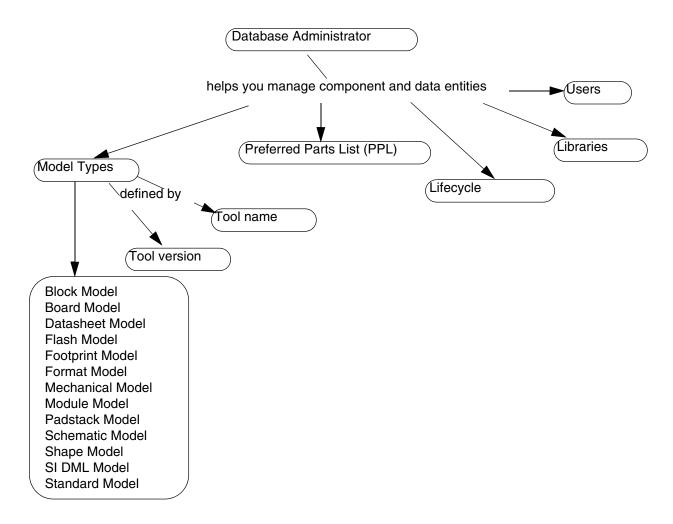
The Allegro EDM Component Database stores two types of data:

- Administrator Data: This describes the type of data entities that can exist in the component database, such as model types, libraries, users, and preferred parts list. In other words, these entities act as metadata for your library data. These entities are based on your enterprise data schema.
  - This type of information is accessed and managed by using Database Administrator (dbadmin).
- Library Data: This describes the end-user component database elements, such as parts and models that designers use in their designs.

#### **How Database Entities are Related**

The following diagram illustrates the types of database entities (supported by the component database) and their relationships with one another.

Figure 1-1 Understanding the Allegro EDM Component Database



#### Applications Used by the Allegro EDM Librarian

- Allegro EDM librarians use many Allegro tools such as:
  - Part Developer to create and verify the parts
  - PCB Editor to create footprints
  - Padstack Editor to create padstacks

Getting Started with Database Administrator

- Database Editor (dbeditor) is used by the librarians to create the meta information for library elements. This information is used to classify and search parts.
- Database Administrator (dbadmin) helps you manage the following elements in the Allegro EDM component database:
  - Model Types
  - □ <u>Libraries</u>
  - Preferred Parts List (PPL)
  - □ <u>Lifecycle Status</u>
  - User Accounts

This guide explains the tasks that are done by a librarian before the actual parts are created, using dbadmin.

#### **Model Types**

Database Administrator references a design or library tool, say Allegro (Allegro PCB Editor) or Design Entry HDL, by the name to be used in the flow. Model Types are specific to the models they are used. For example, Design Entry HDL is associated with schematic models. Database Administrator names directories in the database as model\_<tool name>. For example, model\_concept. These directories contain the archive files in the integration area and vault area.

#### Libraries

Design libraries are collections of a type of component. For example, a library res could contain all the resistors. Database Administrator allows you to define logical libraries. A library name, in conjunction with a tool name and model name, helps you define a unique database entity. For example, if you have ALLEGRO defined as a tool, and JEDEC and PADSTACK as two models, then you can define FOOTPRINTS as a library and can create two database entities as follows:

- Entity A: Tool: ALLEGRO, Model: PADSTACK, Library: FOOTPRINTS
- Entity B: Tool: ALLEGRO, Model: JEDEC, Library: FOOTPRINTS

Getting Started with Database Administrator

#### Preferred Parts List (PPL)

Preferred Parts Lists (PPLs) represent the part lists to be used at a site or in an enterprise. These lists help librarians ensure that the design teams use the approved parts in their designs. For example, you can create a PPL containing all the parts of a preferred vendor and can help design groups use them at various sites. One part can be in multiple PPLs, if required. You can also associate a color to it so as to visually differentiate various PPLs.

#### Lifecycle Status

Library parts can also have an additional associated business lifecycle state, which is configurable by the library administrator. For example, *Approaching EOL*, *EOL*, and *Approved*.

Such states are specific to companies. The site administrator can also specify actions to be taken by Part Information Manager when designers add parts with certain business lifecycle states to their designs.

#### **User Accounts**

In many electronics design organizations, the designer librarian community is grouped into the following categories:

- Designers
  - Logical Designers and Administrators
  - Physical Designers and Administrators
  - High Speed Analysis Engineers
- Librarians
  - Librarians
  - Library Administrators

Database Administrator allows you to manage users and roles depending on the access to the components in the database. User and role management provides different degrees of access and control to different groups of database users. To accomplish this, you can create and manage user database objects. The available default roles in Database Administrator are as follows:

■ *ECAD Library Administrator*: The ECAD Library Administrator can access both, Database Administrator, and Database Editor, and also manage classifications in

Getting Started with Database Administrator

Database Editor. A user with this role has access to all administration operations, such as creation of custom models, library creation, password management, lifecycle creation.

- Senior ECAD Librarian: A Senior ECAD Librarian can manage classifications using Database Editor.
- *ECAD Librarian*: The ECAD Librarian cannot manage classifications but can manage parts and models.
- *ECAD Designer*: The ECAD Designer role has read-only privileges for the Allegro EDM Component Database.

15

Getting Started with Database Administrator

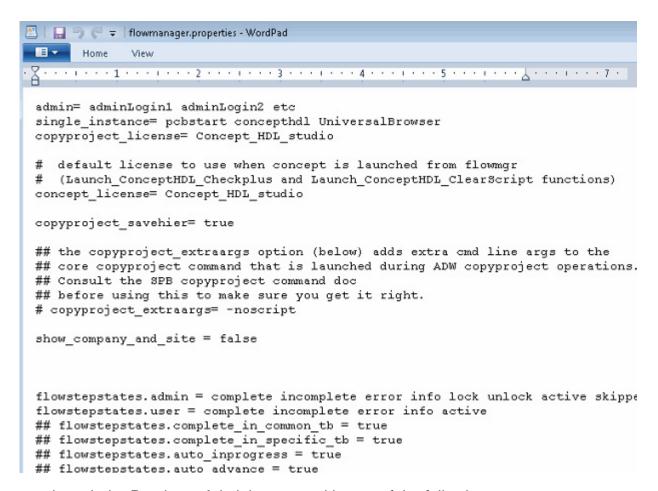
## **Launching Database Administrator**

## Important

You must have database administration privileges before you work with Database Administrator. To assign Allegro EDM administrator privileges, open the

flowmanager.properties file at

 $< adw\_conf\_root > \ < company > \ < site > \ \\ codssetup \ projmgr \ flows and add the required login id to the admin = line.$ 



You can launch the Database Administrator tool in one of the following ways:

- Flow Manager
- Command Line

Getting Started with Database Administrator

## Flow Manager

There are two ways in which you can launch Database Administrator using Flow Manager interface.

- Choose Tools Database Administration.
- In the flow-specific tools section, choose *Tools Admin Database Administration*.

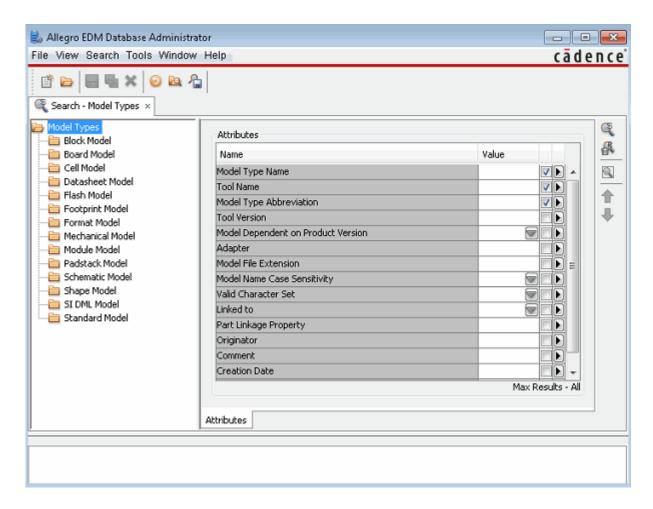
#### **Command Line**

- 1. In the Allegro EDM system console, type dbadmin, and press the Return key.
- 2. Provide your login details in the Login dialog box.



Getting Started with Database Administrator

The Database Administrator interface appears.



The Database Administrator User Interface section describes the interface.

2

## **Managing Classifications**

## **Understanding Allegro EDM Classification**

Classification is a hierarchical categorization that helps you organize your Allegro EDM database components (parts and models). Classifications enable faster searching. They also play a critical role in Part Table File (PTF) generation. Schematic Model Classifications define all the PTF Properties: Global, Key, Injected, Additional, and so on. Classifications are defined in the Database Administrator (dbadmin) tool.

Database Administrator supports the following types of classifications:

- Part Classification (such as transistors, resistors, capacitors, ICs, and so on)
- Block Part Classification
- Mechanical Part Classification (such as sockets, screws, washers)
- Model Classification (such as block model, board model, cell model, datasheet model, flash model, footprint model, format model, mechanical model, padstack model, schematic model, shape model, standard model, and SI DML model)

Classification helps you logically arrange parts and models for your design site. Moreover, common attributes can be defined for a parent classification and these attributes can be linked to parts, models, or child classifications that are under the parent classification. For example, you can specify data type, default value, mandatory and searchable attributes for the Transistor parent classification that are inherited by the parts and child classifications under the Transistor classification.

## Classification Versioning

Revision management is one of the key features available that helps you subclassify versions of a classification (part and model), which can be used by your librarians. The default version for all classifications is 1.0. For example, if you have an existing part classification, say  $LOGIC\_GATE$  [1.0], and want to update it with changes, you can create an incremental version of this classification, say  $LOGIC\_GATE$  [2.0], and can save the changes to it. This

Managing Classifications

way you can maintain the integrity of the designs using LOGIC\_GATE [1.0] parts. The decision and the timing for design update, therefore, remains with the designers.

#### **Hierarchical Classifications and Properties Behavior**

In a hierarchy of classifications, if you try to add, edit, or delete a property inherited by child classifications, the changes made in the parent classifications are not propagated to the child classifications. All such properties are overridden by the property changes done in the child classifications. All overridden properties appear with the overridden property icon in the right pane.

The tasks you can perform while working with classifications are as follows:

- Creating Child Classifications
- Revising Classifications
- Working with Classification Properties
- Working with Schematic Model Properties
- Modifying Classification Property Order for Display in Search Results
- Working with Obsolete Classifications
- Copying Classifications
- Renaming Classifications
- Deleting Classifications

Managing Classifications

## **Creating Child Classifications**

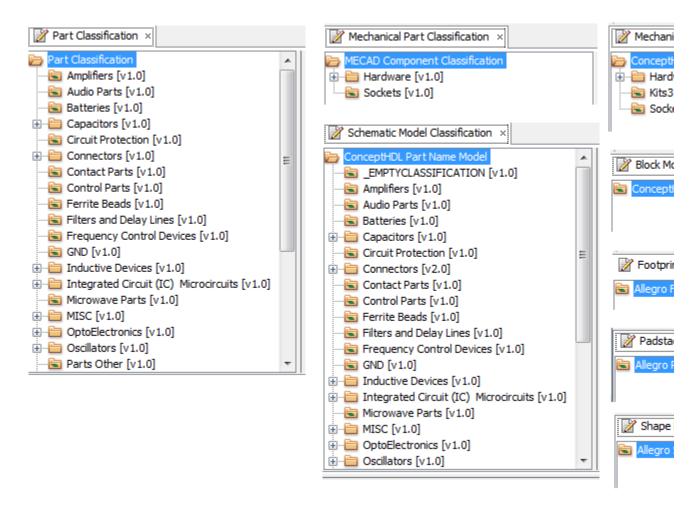
The Library Import utility allows you to specify the default classification for your site libraries. When these are available, you can make modifications to the existing classifications:

To create a child classification, do the following:

- 1. Select one of the following:
  - □ File Manage Classification Part Classification
  - □ File Manage Classification Mechanical Part Classification
  - □ File Manage Classification Block Part Classification
  - File Manage Classification Model Classification <Model\_Type> Model Classification

Managing Classifications

The corresponding classification tab appears.



- **2.** Select a classification node in the explorer pane.
- 3. Right-click and choose Add Child.

Alternatively, use the shortcut key Ctrl+N.

The New dialog box appears.

**4.** Enter the name for the new classification and click *OK*.

The new classification appears under the selected node with [v1.0] as the suffix and is added to the database. For new child classifications with inherited properties, an icon appears against the property row in the right pane.

Managing Classifications

## **Revising Classifications**

Revising a classification creates another copy of the classification with a higher version number.

To revise a classification, do the following:

- 1. Open a classification tab.
  - □ File Manage Classification Part Classification
  - □ File Manage Classification Mechanical Part Classification
  - □ File Manage Classification Block Part Classification
  - File Manage Classification Model Classification <Model\_Type> Model Classification

The appropriate classification tab appears.

**2.** Select a classification node in the explorer pane you want to revise and choose *Revise* from the pop-up menu.

A message prompts you to confirm the revision.

3. Click Yes.

A new version of the classification is added with the incremental version number as the suffix. For example, if the previous version is 1.0, the new version will be 2.0.



## **Important**

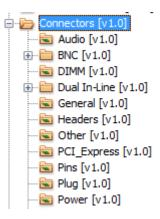
You can not revise a classification if its latest version already exists. For example, if a 2.0 version of a classification exists, you cannot revise the 1.0 version even if it is marked obsolete. To revise the version, first delete the latest version.

Managing Classifications

#### **Revising Hierarchical Classifications**

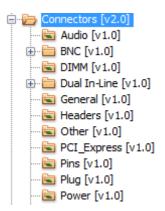
When a classification contains subclassifications and you revise the immediate parent classification, the version of child classifications are not revised. For example, you have a part classification, say Connectors with version 1.0 with child classifications as shown below.

Figure 2-1 Hierarchical Part Classifications



After you revise this, Connectors[v2.0] is created. Note the [v1.0] of its child classifications.

Figure 2-2 Hierarchical Part Classification after Revision



Similarly, a change made to a subclassification (of a specific version) does not apply to other subclassification that has the same name but a different parent classification. For example, if you make changes to Audio[v1.0] under Connectors[v2.0], these changes does not affect Audio[v1.0] under Connectors[v1.0].

Managing Classifications

#### **Updating Classifications**

The pop-up menu items for classifications contain the *Refresh* command. This command helps you refresh the view for the selected nodes in the explorer pane.

**Note:** In hierarchical classifications, this command becomes important for making property changes. To help you understand this, assume you have two hierarchical classification nodes where part2[v1.0] is a subclassification of part1[v1.0], and part2[v1.0] has parts linked to it, and you want to edit properties for this version. To do so, you revise part2[v1.0] to part2[v2.0], and make property changes to it. Before you save the hierarchy, you must select part2[v1.0], choose Refresh pop-up command. When done, select part1[v1.0] and choose the  $Save\ Hierarchy$  pop-up menu command.

## **Working with Classification Properties**

For classifications, you can specify properties, which in turn, are transferred to the parts or models specified under them. The tasks you can perform with classification properties are:

- Adding Properties
- Editing Properties
- Deleting Properties
- Working with Predefined Values

However, if you are working with a Schematic Model, see <u>Working with Schematic Model</u> <u>Properties</u>.

## **Adding Properties**

To add a property, do the following:

**1.** Select a classification node in the explorer pane.

The default properties, if available, appear in the right pane. If there are no values, then the right pane is grayed out.

- **2.** Select a property row in the right pane.
- **3.** Right-click and choose *Add* from the pop-up menu. Alternatively, click the *Add* button in the right pane.

The Add dialog box appears.

Managing Classifications

**4.** Enter the name of the property to add, and click *Create*.

The property is added and appears in a new row in the right pane, and a new interface property icon appears.

**5.** Enter appropriate values in the row.

For each property you can enable *Predefined Values*. To learn how to add or edit predefined values, see <u>Working with Predefined Values</u>.

For detailed information on the kind of information that you provide with a property, see <u>Appendix B. "About Attributes and Relations"</u> of this guide.

- **6.** Click the *Property Display Order* tab if you need to modify the display order of the new property, which appears as the last property in this tab.
- **7.** Choose *Save* from the pop-up menu to save changes in the database or click the *Save* button.

If you have made more than one change to a hierarchy, then select the parent classification and choose *Save Hierarchy* from the pop-up menu to save all the changes.

#### **Editing Properties**

To edit a property:

1. Select a classification node in the explorer pane.

The default properties, if available, appear in the right pane.

2. Select a property row in the right pane and edit the values.

For each property you can enable *Predefined Values*. To learn how to add or edit predefined values, see <u>Working with Predefined Values</u>.

- **3.** Click the *Property Display Order* tab if you need to modify the display order of any property.
- **4.** Choose *Save* from the pop-up menu to save changes in the database or click the *Save* button.

If you have made more than one change to a hierarchy, then select the parent classification and choose *Save Hierarchy* from the pop-up menu to save all the changes.

An asterisk (\*) next to a node in the explorer pane signifies unsaved classification.

Managing Classifications

#### **Deleting Properties**

To delete a property:

**1.** Select a classification node in the explorer pane.

The default properties, if available, appear in the right pane. If there are no values, the right pane is grayed out.

- 2. There are two ways to delete a property:
  - Select a property by clicking the corresponding row in the right pane, and choose Delete from the pop-up menu. Alternatively, click the Delete button in the right pane.
  - □ Without selecting any property, click *Delete*. The Property Name dialog box appears. Enter the name of the property to delete, and click *Delete*.

As soon as you delete a property row, it is grayed out and a delete mark icon appears against the property row.

**3.** Choose *Save* from the pop-up menu to save changes into the database or click the *Save* button.

If you have made more than one change to a hierarchy, then select the parent classification and choose *Save Hierarchy* from the pop-up menu to save all the changes.



You can add, edit, and delete properties for classifications not linked to a part or model. For linked classifications, first create a revision, and then make the property changes to it.

Managing Classifications

#### **Working with Predefined Values**

For each new property or an existing property, you can add Predefined Values. This allows you to select values from a given list. To add these values:

- **1.** Choose a property row in the right pane.
- 2. Click the button next to Predefined Values.

The Add/Remove dialog box appears.

**Note:** If there are any existing values for this property, the database retrieves and populates them in the Add/Remove dialog box. If you want to use these values, click *OK*.

3. Add values and press the Enter key.

Alternatively, right-click the *Predefined Values* area and choose *Add Value* from the pop-up menu.

- **4.** Choose any value and right-click it to perform any of the following operations, if required:
  - □ Delete Value(s)
  - □ Move Up
  - □ Move Down
  - Move to Top
  - Move to Bottom

**Note:** You cannot delete a value if it is being used in a part or model.

- **5.** Select the *Restrict Predefined Values* check box if you do not want the librarian to add value other than the ones in the predefined list.
- 6. Click OK.
- 7. Choose *Save* from the classification node pop-up menu to save changes in the database or click the *Save* button.

After you associate this classification with a part or block part, you will be able to select from the list of predefined values.

Managing Classifications

## **Working with Schematic Model Properties**

A schematic Model is an ECAD model, and plays a special role in the generation of PTF files for your library data, which is why Schematic Model properties are defined in a slightly different manner in Database Administrator.

The right pane in the *Schematic Model Classification* tab contains two panes. The upper pane contains the properties for the model, while the lower pane (*Informational Attributes*) contains the ECAD properties. For detailed information on the kind of information you provide with a property, see <u>Appendix B</u>, "About Attributes and Relations" of this guide.

The tasks you can perform with Schematic Model properties are as follows:

- Adding Schematic Model Properties
- Editing Schematic Model Properties
- Deleting Schematic Model Properties

#### **Adding Schematic Model Properties**

**1.** Choose File – Manage Classification – Model Classification – Schematic Model Classification.

The Schematic Model Classification tab appears.

**2.** Select a classification node in the explorer pane.

The default properties and the information attributes appear in the respective areas of the right pane.

**3.** Select a property row and choose *Add* from the pop-up menu. Alternatively, click the *Add* button in the right pane.

The Property Name dialog box appears.

**4.** Enter the name of the property to add and click *Create*.

The property is added and appears in a new row in the *Properties* area.

**5.** Enter appropriate values in the row.

**Note:** If *ECAD Type* property is set to *True*, the added property appears in the *Informational Attributes* pane.

**6.** Specify the ECAD properties for the recently added property in the *Informational Attributes* area in the right pane.

Managing Classifications

## /Important

Depending on the *PTF Mapping* column value you specify, the *Key Property Order*, *Optional Property*, and *Global Property Value* columns may or may not be available.

**Note:** For schematic models, the ECAD property name is used as it is in the PTF (generated by the PTF Generator tool). Therefore, ensure that the property name specified is valid according to PTF property naming convention. For more information on part table files, see *Part Table Editor User Guide*, a Cadence document.

- **7.** Click the *Property Display Order* tab if you need to modify the display order of the new property, which appears as the last property in this tab.
- **8.** Choose *Save* from the pop-up menu to write the changes to the database or click the *Save* button.

If you have made more than one change to a hierarchy, select the parent classification and choose *Save Hierarchy* from the pop-up menu to save all the changes.

#### **Editing Schematic Model Properties**

The editing process is similar to adding properties where you can add new properties and also modify existing properties. For more information on how to do this, see <u>Adding Schematic Model Properties</u>.

## /Important

While editing schematic models, ensure that no duplicate key property orders exist between a classification and any of its subclassifications. In such cases, using the *Save Hierarchy* and the *Save* commands displays an error prompting you to specify unique key property order values.

## **Deleting Schematic Model Properties**

To delete an existing property, do the following:

- 1. With the *Schematic Model Classification* tab open, select a classification node in the explorer pane.
  - The default properties and the information attributes appear in the respective areas of the right pane.
- **2.** There are two ways to delete a property:

Managing Classifications

- Select a property in the *Properties* or the *Informational Attributes* area, and click the *Delete* button.
- ☐ Without selecting any property, click *Delete*. The Property Name dialog box appears. Enter the name of the property to delete, and click *Delete*.

As soon as you delete a property row, it is grayed out and a deleted mark icon appears against the property row.

**3.** Choose *Save* from the pop-up menu to update the database or click the *Save* button.

If you have made more than one change to a hierarchy, select the parent classification and choose *Save Hierarchy* from the pop-up menu to save all the changes.

Managing Classifications

# Modifying Classification Property Order for Display in Search Results

As an administrator, you can predefine which classification properties should be displayed and in which order, when search is performed on the library data. You can:

- Set the order of properties for display in the search results
- Control if a certain property should be displayed in the search results

The order defined by the administrator becomes the default property display order that appears in the search results in the Allegro Library Manager. This will ensure that all librarians and designers see a consistent view of the search results.

As an administrator, you can perform the following tasks:

- Setting Property Display Order
- Copying Property Display Order
- Exporting Property Display Order
- Importing Property Display Order
- Resetting Property Display Order
- Highlighting Classifications with Modified Display Order

## **Setting Property Display Order**

At a selected classification node, you see the union of:

- Properties on the selected classification
- Properties inherited from the parent classification
- Properties from its child classifications

This enables you to set the display and also order the properties at the highest classification node in the explorer pane. This order will serve as the default order of properties for display in the Search Results tab.

Thus, to be able to predefine the classification property order:

- **1.** Choose one of the following:
  - □ File Manage Classification Part Classification

Managing Classifications

File – Manage Classification – Mechanical Part Classification
File – Manage Classification – Block Part Classification
File - Manage Classification - Model Classification - <model_type> Model Classification</model_type>

- 2. Click any classification node in the explorer pane.
- **3.** Click the *Property Display Order* tab in the right pane. In this tab, you will see that:
  - This child classification shows both the properties inherited from the parent classification and properties at this subclassification.
  - □ When you add a new property to this child classification, the same is displayed in the parent classification.
  - ☐ The *Display in Search* check box is selected for all the properties, and as a result, all the properties will be visible in the search results.

**Note:** The Display in Search check box is not selected for all the schematic model classification properties. The Display in Search setting is on for ECAD properties that have the PTF Mapping attribute as Global, and non-ECAD properties that are searchable. You can change the display settings for other properties, if required.

- **4.** Clear the *Display in Search* check box for the properties that should not be visible in the search results.
- **5.** Select the property name and click up or down arrow to define the order in which it should appear in the search results.

Alternatively, drag and drop the property name to the desired position.

**6.** Choose *Save* from the classification node pop-up menu to save changes in the database or click the *Save* button.

## **Important**

The display and order that you specify at a classification node is inherited by all its child classifications. However, if you modify the display order at a child classification, and then modify the display order of its parent classification, any change in the order at the parent classification is not inherited by any of its child classifications.

7. Search for objects related to the updated classification.

The specified property display order is visible in the search results. In addition, all nonsearchable properties will be shown as disabled. You cannot search using any of the nonsearchable properties but can choose to see them in the search results.

Managing Classifications

#### **Copying Property Display Order**

You can copy the display order of properties specified for one classification on to another classification by doing the following:

- **1.** Choose a classification node in the explorer pane.
- **2.** Choose *Property Display Order Copy* from the pop-up menu.

**Note:** The *Copy* option is disabled if no display order is specified for the selected classification.

- **3.** Choose the classification node to which you want to copy the display order.
- **4.** Choose *Property Display Order Paste* from the pop-up menu.

## Important

The display order of the properties common to both the classifications is applied to this classification. These properties now appear before those properties that are not common.

#### **Exporting Property Display Order**

You can export and import the display order of properties for all classifications.

To export the display order of properties, do the following:

1. Choose the root node.

## /Important

The Export and Import options are enabled only at the root node.

**2.** Choose *Property Display Order – Export* from the pop-up menu.

The Export dialog box opens.

- **3.** Specify the name of the *property\_display\_order>.csv file.*</code>
- **4.** Click Export.

The CSV file will be saved in the project directory and will contain the property display order for all the classifications in the explorer tree.

A message appears to indicate successful completion of export.

Managing Classifications

5. Click OK.

#### **Importing Property Display Order**

While importing, the existing display order for properties is deleted from the database and the display order being imported is applied.

To import the display order of properties:

1. Choose the root node.

## /Important

The Export and Import options are enabled only at the root node.

**2.** Choose *Property Display Order – Import* from the pop-up menu.

The Import dialog box appears.

- **3.** Choose the cproperty\_display\_order>.csv file.
- 4. Click Import.

A message appears to confirm the import because it will delete the existing display order for properties and then apply the display order being imported.

- **5.** If you click *Yes*, a message appears to indicate successful completion of import and the classification node icon changes.
- 6. Click OK.
- **7.** Click *Save All* to save the display order changes that have been imported in the database.

#### **Format of Property Display Order File**

The property display order file is a comma separated file with file extension as .csv. The header contains three columns:

- Classification Name: Name of the classification on which the property display order exists.
- Property Name: Property name which is ordered.
- Display In Search: Indicates if the property is selected for display in search results.

Managing Classifications

Each entry in this file contains the information for a property on the classification based on these three headers.

The order of the appearance of properties for a classification in this file is same as the property order on the classification. You can change the ordering in and import this file to change the property order on a classification.

Example of the property display order file:

```
"Classification Name", "Property Name", "Display in Search"

"CAD Component Classification.Electrical Parts [v1.0].Contact Parts [v1.0]", "COST LAST UPDATED", "false"

"CAD Component Classification.Electrical Parts [v1.0].Contact Parts [v1.0]", "EST ANNUAL USAGE", "false"

"CAD Component Classification.Electrical Parts [v1.0].Contact Parts [v1.0]", "DESCRIPTION", "true"

"CAD Component Classification.Electrical Parts [v1.0].Contact Parts [v1.0]", "COST", "true"

"CAD Component Classification.Electrical Parts [v1.0].Contact Parts [v1.0]", "ROHS", "true"
```

When you export the property display order for the complete classification tree, the classification entries in the exported .  $\mathtt{CSV}$  file have the same property order as seen in the Property Display Order tab.

## **Resetting Property Display Order**

You can reset the display order of properties on the selected classification and its children.

To reset property display order, do the following:

- **1.** Choose a classification node in the explorer pane.
- **2.** Choose *Property Display Order Reset* from the pop-up menu.

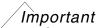
A message appears to confirm the reset operation because it will permanently remove the existing property display order on this classification and its children.

3. Click Yes.

The display order preferences are reset and the classification node icon changes.

Managing Classifications

4. Click Save All.



After this operation, if parent classification of the selected classification node has an order defined, that order will be inherited on the selected classification and its children.

### **Highlighting Classifications with Modified Display Order**

You can highlight the child classifications on which you have specified a display order.

To identify the child classifications which have a property display order set:

- **1.** Choose a classification node in the explorer pane.
- **2.** Choose *Property Display Order Highlight Modified Child Classifications* from the pop-up menu.

A message appears to inform you that the selected classification node has been expanded and all the child classifications under this node have been highlighted to indicate that the property display order has been modified for them.

**Note:** The selected classification node or its parent node will not be highlighted even if it has a modified display order.

3. Click OK.

Managing Classifications

## **Working with Obsolete Classifications**

An obsolete classification is the one no longer used. This happens when you have incremental versions of classifications and you are confident that they are no longer required by the design groups. In such cases, you can mark the classification <code>obsolete</code> classification, is not visible to the design groups. However, it is available in the database. This ensures that as a database administrator, you have access to it and should you can make it available (using the copy-paste operation) if the need arises.

A classification name is unique within a parent classification node. As a result, you can create more than one classification with the same name but under different parent classifications. Moreover, a parent and child classification can share the same name. For example, the following hierarchy is valid.

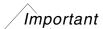
```
ClassP1 [v1.0]
......ClassC1 [v1.0]
ClassP2 [v1.0]
......ClassC1 [v1.0]
```

In this example, you cannot add or copy another classification with name ClassC1 (under ClassP1 [v1.0]). But, you can do this when you mark the ClassC1 obsolete.

Note: Database validations are not performed on an obsolete classification.

The procedures related to obsolete classifications are:

- Marking a Classification Obsolete
- Viewing Obsolete Classifications
- Revising Obsolete Classifications



You can copy, delete, and revise an obsolete classification. All other classification-related operations are not available.

## Marking a Classification Obsolete

You cannot set a classification to obsolete if it has any objects linked to it. After you have ensured that a classification is not in use, you can mark it as obsolete. To do so:

Managing Classifications

- **1.** Open the appropriate classification tab.
- 2. Select a node in the explorer pane and choose *Mark Obsolete* from the pop-up menu.

**Note:** A classification is always active before it is marked obsolete.

**3.** An alert appears asking if you want to mark the selected classifications (and its children, if any) as obsolete.

If you mark a classification that contains subclassifications obsolete, all the subclassifications are also marked obsolete.

## **Important**

Once a classification along with its child classifications, if any, is marked obsolete, it becomes a dormant classification. You can view it through the *Show Obsolete Classification* pop-up menu command, but cannot use for classifying part and model data. However, you can copy-paste this classification.

4. Click Yes.

**Note:** If the classification you want to mark obsolete is linked to any of the active database objects, the *Error Report* tab appears listing the error. To view the details of error, choose *Show Extended Error* from the pop-up menu.

### **Viewing Obsolete Classifications**

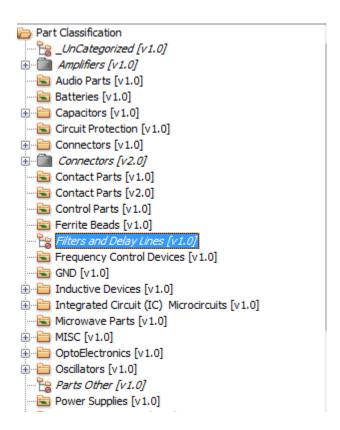
To view obsolete classifications, do the following:

- **1.** Open the appropriate classification tab.
- 2. Select the highest node Part Classification node for part classification or Model Type node for model classification in the explorer pane and choose *Show Obsolete Classifications* from the pop-up menu.

All the successfully marked obsolete classifications nodes appear in black italics in the explorer pane.

Managing Classifications

Figure 2-3 Obsolete Classifications Appear in Black Italics



### **Revising Obsolete Classifications**

To revise an obsolete classification, do the following:

- **1.** Open the appropriate classification tab.
- 2. Select the highest node Part Classification node for part classification or Model Type node for model classification in the explorer pane and choose *Show Obsolete Classifications* from the pop-up menu.
  - All the successfully marked obsolete classifications nodes appear in black italics in the explorer pane.
- **3.** Right-click the required obsolete classification and choose Revise. A message is displayed.
- **4.** Click *Yes* to revise the classification.

A new version of the selected classification is created, which is visible and available for use.

Managing Classifications

## **Copying Classifications**

To facilitate the existence of same part, model classifications or their properties under different classifications, Database Administrator allows you to copy a node to another classification node. These operations are available as:

Paste As	Description
Classification Node	Allows you to copy-paste only the selected classification node to another classification node.
Subtree Classification	Allows you to copy-paste the selected classification node along with the child nodes under it to another classification node.

To copy a node, irrespective of the copy-paste modes, you select:

- 1. Open the appropriate classification tab.
- **2.** Select a source node in the explorer tree, for example pre\_release [v1.0], and choose *Copy* from the pop-up menu. Alternatively, use the shortcut key Ctrl+C.
- **3.** Select a destination node in the explorer pane, for example, highspeedflow [v1.0], and choose *Paste* from the pop-up menu. Alternatively, use the shortcut key Ctrl+V.

In this case, only the selected classification node (without its child classifications) is copied to the target classification. However, if you choose  $Paste\ Classification\ Subtree$  from pop-up menu, the pre\_release v[1.0] classification along with its child classifications are copied to the target classification, highspeed [v1.0]. Regardless of the paste command you use, the classifications (whether alone or with child classifications) are replicated along with their properties.

The paste operation, by default, makes the copied classification version as 1.0.

You can copy-paste obsolete classifications. They are copied as active classifications. To do so, select the root classification, and choose *Show Obsolete Classifications* from the pop-up menu, and then follow the steps mentioned above.

Managing Classifications

## **Renaming Classifications**

To rename a classification node:

- **1.** Open the appropriate classification tab.
- 2. Select the desired node in the explorer pane, and choose *Rename* from the pop-up menu.

The Rename dialog box appears.

3. Enter the new name and click OK.

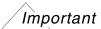
The new name of the node appears in the explorer pane.

**Note:** When you rename a parent classification, the child classifications are not affected by this.

## **Deleting Classifications**

To delete a classification:

- 1. Open the appropriate classification tab.
- 2. Select the target node in the explorer pane, and choose *Delete* from the pop-up menu. An alert appears asking if you want to delete the classification and all its children.
- 3. Click Yes.



You cannot recover a deleted classification.

## **Working with Allegro EDM Entities**

This chapter provides a basic understanding of the tasks you can perform using Database Administrator. They are:

- Managing Component-Specific Entities
- Working with Attributes
- Working with Relations

## /Important

Many of the tasks performed on database elements follow similar steps and approaches. Therefore, a generic methodology has been provided in the documentation.

**Note:** The use model of these tasks depend on your enterprise-or site-specific flow that has been implemented.

## **Managing Component-Specific Entities**

Managing component-specific entities involves the following procedures:

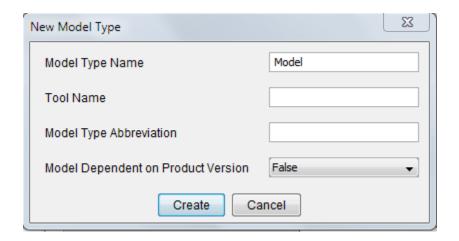
- Creating Entities
- Editing Entities
- Deleting Entities

## **Creating Entities**

To create an entity, do the following:

Working with Allegro EDM Entities

**1.** In the Database Administrator window, choose *File - New - <Entity\_Name>*. For example, if you are creating a model type, you will choose *File - New - Model Type*.



The New  $\langle Entity \rangle$  dialog box appears prompting you for values of key attributes. These key attributes combine to specify a unique database object.

2. Specify the values in all of the attribute fields (mandatory).

The fields that appear depend on the key attributes defined in your database.

3. Click Create.

The <*Entity\_Detail> – Key Attribute1 Key Attribute 2... Key Attribute n* tab appears. For example, if you create a PPL entity, the tab name will be: *Preferred Part List – <1ist\_name>* 

**Note:** The number of key attributes required depend on the entity you are creating.

- **4.** In the tabbed view, you can:
  - **a.** Use the explorer pane to specify relation instance for the entity. For example, if you have a *Preferred Part List* tab, then you can create relation instances for the relations, *Parent PPL*, *Linked Parts*, and *Linked Block Parts*.
  - **b.** Use the *Informational Attributes* area to view the key and schema-defined attributes of the entity.
  - **c.** Use the *Attributes* area to edit mandatory and optional attributes of the entity.
    - O For information on specifying attribute values, see Working with Attributes.
    - For information on creating relations, see <u>Working with Relations</u>.

Working with Allegro EDM Entities

- O For a detailed list of default attributes, relations, and their explanations, see Appendix B, "About Attributes and Relations".
- **5.** Choose *File Save* to save the entity created into the component database. Alternatively, click the *Save* button on the toolbar.

**Important** 

The procedure for creating other entities is similar, except that the attributes and relations appear depending on the entity you create.

#### **Editing Entities**

You can edit component-specific entities (model type, library, or PPL) by:

- Specifying new and editing existing attributes
- Associating new and editing existing relations

To edit component-specific entities, do the following:

**1.** Choose File – Open – <Entity>.

The Open *<Entity>* dialog box appears.

- **2.** Specify the requisite information in the appropriate fields.
- 3. Click Open.

The *<Entity\_Detail> – Key Attribute1 Key Attribute 2...Key Attribute n* tab appears. For example, if you open a PPL entity, the name of the tab that opens will be: *Preferred Part List – <1ist\_name>* 

**Note:** The number of key attributes required depend on the entity you are editing.

- **4.** In the tabbed view, you can:
  - **a.** Use the explorer pane to edit relation instances for the entity.
  - **b.** Use the *Informational Attributes* area to view the key and schema-defined attributes for the tool.
  - **c.** Use the *Attributes* area to edit mandatory and optional attributes for the tool.
    - For information on how to edit attribute values, see <u>Working with Attributes</u>.
    - O For information on editing relations, see Working with Relations.

Working with Allegro EDM Entities

- O For information on the default attributes, relations, and their explanations, see Appendix B, "About Attributes and Relations"
- **5.** Choose *File Save* to save the information in the component database.

Alternatively, click the Save button on the toolbar.

**Important** 

The procedure for editing other entities is similar, except that the attributes and relations that appear depend on the entity you edit.

#### **Deleting Entities**

To delete an entity, do the following:

- **1.** Search for the entity to be deleted.
- **2.** Choose the entity in the *Search Results* tab.
- **3.** Choose the *Show Details* pop-up menu command.

The entity details appear in the right pane.

- **4.** Do one of the following:
  - □ Choose *File Delete*.
  - □ Click the *Delete* button on the toolbar.

An alert appears asking if you want to delete the entity.

5. Click Yes.



Ensure that you delete the desired entity. You cannot recover a deleted entity. The entity history is removed when you delete it.

Working with Allegro EDM Entities

## **Attributes and Relations**

Attributes and relations help you specify properties and linkages on component-specific and user entities. When you open an *<Entity\_Detail>* tab, then the explorer pane (left pane) contains the relations for the selected entity. These appear in a tree-like structure, where all the available relations appear as second-level nodes. The last-level nodes denote specific relation instances.

The *Detail* tab on the right pane contains two areas: *Informational Attributes* and *Attributes*. The former displays all the schema defined and non-editable attributes, while the latter contains the editable attributes, if any.

**Note:** Informational attributes are attached to the entities when you create them. You cannot edit them in Database Administrator.

#### **Working with Attributes**

Attributes are the name-value pair properties defined on an entity. For example, a model type can have properties, such as <code>Tool Name</code>, <code>Creation Date</code>, <code>Originator</code>, <code>Modification Date</code>, <code>Comment and so on</code>. Some of these properties are key properties (<code>Tool Name</code>, <code>Model Type Name</code>, and <code>Tool Version</code>) defined at the time of creating the model type, while other properties are predefined in the component database schema, which can be mandatory or optional.

**Note:** For information on the default attributes, relations and their explanations, see <u>Appendix B, "About Attributes and Relations."</u>

The tasks you can perform on attributes using Database Administrator are:

- Adding Attribute Values
- Editing Attribute Values
- Deleting Attribute Values

#### **Adding Attribute Values**

To add attribute values, do the following:

**1.** Open the *<Entity\_Detail>* tab.

Working with Allegro EDM Entities

## **Important**

Before you add attributes, you need to create the entity. For information on how to do so, see <u>Creating Entities</u>.

2. Select the root node in the explorer pane.

The Informational Attributes and Attributes areas appear in the right pane.

- **3.** In the *Attributes* area, select the row that lists the attributes under the *Name* column.
- **4.** Add a value in the field next to the attribute (under the *Value* column).
- **5.** Choose *File Save* to save changes.

#### **Editing Attribute Values**

To edit attribute values:

**1.** Open the *<Entity\_Detail>* tab.

## Important

Before you edit attributes, you need to edit the entity. For information on how to do so, see Editing Entities.

**2.** Select the root node in the explorer pane.

The *Informational Attributes* and *Attributes* areas appear in the right pane.

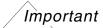
- **3.** In the *Attributes* area, select the row (that lists the attributes) under the *Name* column.
- **4.** Edit the value in the field next to the attribute (under the *Value* column).
- **5.** Choose *File Save* to save changes.

**Note:** An asterisk (\*) within the < *Entity\_Detail>* tab signifies that there are unsaved changes in the tab.

#### **Deleting Attribute Values**

To delete an attribute value, you have to remove the value against the attribute name. In such cases, the value field remains blank.

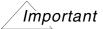
Working with Allegro EDM Entities



You cannot delete key or mandatory attributes.

#### **Working with Relations**

Relations are associations created between component-specific entities. In the default component database, only the Preferred Part Lists entity has relations, namely Parent PPL, Linked Parts, and Linked Block Parts, which helps you map the PPL with its parent PPL, linked parts, and block parts.



Relations, by default, are defined in the database schema. You can only specify instances of the default relations.

**Note:** For information on the default attributes and relations available, see <u>Appendix B</u>, "About Attributes and Relations."

You can perform the following tasks using Database Administrator.

- Adding and Editing Relation Instances
- Deleting Relation Instances

#### **Adding and Editing Relation Instances**

To add or edit a relation instance, do the following:

**1.** Open the *<Entity\_Detail>* tab.



Before you open an existing entity, you may need to search for them. For information on how to do so, see <u>Searching Database Entities</u>.

2. In the explorer pane, click a node (except the root node).

All the second-level folders in this pane specify the schema-defined relations. A chain icon represents an existing instance of the relation.

When you click a relation or an instance of the relation, the right pane displays the corresponding information in a table.

3. To specify a relation, you can do either of the following:

Working with Allegro EDM Entities

**a.** If you already know the exact instance to specify, choose *Associate* <*Relation*> from the pop-up menu.

A blank row is added to the table in the right pane.

**b.** Specify the values in the row under each column.

Alternatively, you can:

**a.** Choose Associate <Relation> From Tree.

The Associate <Relation> dialog box appears.

**b.** Select a value from the dialog box and click *Associate*.

A row with the values specified is added to the table in the right pane.

- **4.** You can specify as many relation instances as limited by the cardinality defined for the relation.
- **5.** Choose File Save.



To create relation instances quickly:

- **a.** Search for an entity.
- **b.** Select a search result row in the *Search Results* tab.
- **c.** Choose the *Copy* pop-up menu command.
- **d.** Select the relation node in the *<Entity\_Detail>* tab where you want to create a relation instance.
- **e.** Choose the *Paste* pop-up menu command.

#### **Deleting Relation Instances**

To delete an instance of a relation, do the following:

**1.** Open the *<Entity\_Detail>* tab.

All the second-level nodes in the explorer pane specify the schema-defined relations.

- 2. Do one of the following:
  - □ Right-click a relation node ( ).

Working with Allegro EDM Entities

Choose Delete from the pop-up men
-----------------------------------

□ Click the *Delete* button on the toolbar.

An alert appears prompting you to delete the relation instance.

- **3.** Click *OK*.
- 4. Choose File Save.

# Allegro EDM Database Administrator User Guide Working with Allegro EDM Entities

4

## **Model Management Support**

In Allegro EDM, librarians can add new model types and tool types.

## Adding a New Model to Allegro EDM

A new model in Allegro EDM consists of ECAD data and metadata information. When the librarian creates a new model, the tool type and model name are specified in Database Administrator. This creates an object which defines all the metadata related to the new model.

The librarian associates the classifications and defines the libraries associated with the tool and model name. The libraries and classifications can be defined interchangeably.

Before a new model is created in Database Administrator, an adapter needs to be created. This task is done outside of Allegro EDM.

Model Management Support

## **Creating an Adapter for New Models**

This section lists the requirements for any new adapter that you create. A new adapter is a pre-requisite to creating a new model in Allegro EDM.

Allegro EDM needs to know how to get the data for import, what files constitute a model, will a model be in a compound file, such as capture and if so - how to extract a single model from the big file, and so on.

To solve this problem, Allegro EDM has provided hooks for data collection of ECAD files. Model information can be extracted from industry-standard format.

The .dml files include models, such as packages, boards, and cables. These .ibis file (industry-standard device model files) are converted into the .dml files with the ibis2SI utility.

An adapter needs to be written for each custom model being managed in Allegro EDM.

An adapter needs to link to an xmlWrite DLL so that the xmlWrite APIs can extract model information in the XML format. This format can then be processed by Library Import (libimport) and Database Editor (dbeditor).

An adapter named DWDmlModelAdapter.exe performs the commands on the .dml files and each command needs to return any of the following status depending on the outcome of the operation.

Model Management Support

#### **Return Values for Adapter Calls**

All adapter calls should return one of the following values:

Rule Value	Meaning
0	STATUS_SUCCESS
1	STATUS_FAILURE
2	STATUS_MODE_SUPPORTED
3	STATUS_MODE_NOT_SUPPORTED
4	STATUS_NO_MODELS_PRESENT

#### **Supported Commands/Modes**

An adapter needs to support the following commands:

#### **Adapter Read Command**

In the read mode, the adapter extracts the model(s) information from the input file to the xmloutput file in the XML format. The debug messages are written to the log file.

#### Syntax

```
adapter -mode read -input inputFileName -xmloutput xmlOutputFileName -log
    logFileName
```

### Example

```
DWDmlModelAdapter -mode read -input D:\comhier\v16-4-
0\wint\share\pcb\signal\cds_samples.dml -xmloutput c:\cds_samples.xml -log
c:\dml.log
```

#### Used by

- Library Import to read model information from/to the database.
- Database Editor to read model information while syncing models before check in or release.

Model Management Support

#### **Adapter Diff Command**

In the diff mode, the adapter compares the files input1 and input2. Information about the models updated, inserted, or deleted are written to the xmloutput file and the debug messages are written to the log file.

#### **Syntax**

```
adapter -mode diff -input1 inputFile1Name -input2 inputFile2Name -xmloutput
xmlOutputFileName -log logFileName
```

#### Example

```
DWDmlModelAdapter -mode diff -input1 D:\comhier\v16-4- 0\wint\share\pcb\signal\cds_samples.dml -input2 c:\cds_samples_split.dml -xmloutput c:\cds_samples_diff.xml -log c:\dml.log
```

#### Used by

Database Editor at the time of check in and release of model to compare the split file to the compound file. Also to report new models added or models deleted from the file.

#### **Adapter Iscapable Command**

In the iscapable mode, the adapter reports if it can handle the *<operation>*. The debug messages are written to the log file.

#### Syntax

```
adapter -mode iscapable -operation
    [merge|read|diff|split|delete|rename|validate|edit] -log logFileName
```

#### Example

```
DWDmlModelAdapter -mode iscapable -operation merge -log C:\dml.log
```

#### Used by

Database Editor to decide how to handle the check out, check in, or release of a single model or all the models

Model Management Support

#### **Adapter Split Command**

In the split mode, the adapter extracts model definition from the input file and writes to the output file. The debug messages are written to the log file.

#### **Syntax**

#### Example

DWDmlModelAdapter -mode split -model p14u1 -input D:\comhier\v16-4- 0\wint\share\pcb\signal\cds\_samples.dml -output c:\cds\_samples\_split.dml -log c:\dml.log

#### Used By

Database Editor, only when merge is available, to extract the model definition when checking out a model.

#### **Adapter Merge Command**

In the merge mode, the adapter merges the model definitions from input files to the output file with model definitions of models present in '-model modelName1, modelName2,...' extracted from inputFile1 and not from inputFile2.

All models are extracted from inputFile1 onto outputFile if '-model all' is given. The debug messages are written to the log file.

#### Syntax

```
adapter -mode merge -model [all | modelName1, modelName2, ...] -input1
   inputFile1Name -input2 inputFile2Name -output outputFileName -log logFileName
```

#### Example

DWDmlModelAdapter -mode merge -model p14u1 -input1 c:\cds\_samples\_split.dml -input2 D:\comhier\v16-4-0\wint\share\pcb\signal\cds\_samples.dml -output c:\cds\_samples\_merge.dml -log c:\dml.log

57

Model Management Support

#### Used by

Database Editor to merge the model definition from compound files when a model is released.

#### **Adapter Delete Command**

In the delete mode, the adapter extracts all model definitions from the input file to the output file with the exception of the models present in '-model modelName'. The debug messages are written to the log file.

#### Syntax

#### Example

DWDmlModelAdapter -mode delete -model p14u1 -input D:\comhier\v16-4- 0\wint\share\pcb\signal\cds\_samples.dml -output c:\cds\_samples\_delete.dml - log c:\dml.log

#### Used by

Database Editor to delete the model from the compound file and mark the model for deletion on library distribution.

**Note:** Database Editor marks the compound file for deletion only when the compound file has no model definitions present in it.

#### **Adapter Rename Command**

In the rename mode, the adapter extracts all the model definitions from the input file to the output file after renaming the models present in '-model modelName' to new name present in '-newmodel modelName'. The debug messages are written to the log file.

#### Syntax

Model Management Support

#### Example

DWDmlModelAdapter -mode rename -model p14u1 -newmodel p14u1NEW -input D:\comhier\v16-4-0\wint\share\pcb\signal\cds\_samples.dml -output c:\cds\_samples\_rename.dml -log c:\dml.log

#### Used by

Database Editor to rename a model.

#### **Adapter Validate Command**

In the validate mode, the adapter validates the steps in the input file. The debug messages are written to the log file.

#### Syntax

adapter -mode validate -input inputFileName -log logFileName

#### Example

DWDmlModelAdapter -mode validate -input D:\comhier\v16-40\wint\share\pcb\signal\cds\_samples.dml -log c:\dml.log

#### Used by

Database Editor to validate a model when checking in or releasing the model.

#### **Adapter Edit Command**

In the edit mode, the adapter opens the input file for editing. The debug messages are written to the log file.

#### Syntax

```
adapter -mode edit -input inputFileName -log logFileName
```

#### Example

DWDmlModelAdapter -mode edit -input D:\comhier\v16-4-0\wint\share\pcb\signal\cds\_samples.dml -log c:\dml.log

Model Management Support

#### Used by

Database Editor to call the adapter edit mode in the 'launch viewer' panel of a model detail.

#### **Location for New Adapters**

Any new, customized model adapter is at the following location:

```
<ADW_CONF_ROOT> \ < company> \ < site > \cdssetup\pcbdw\bin
```

Allegro EDM provides a default adapter name, <code>DWFileNameAdapter</code>. This value is populated in the <code>Adapter</code> field when you create a new custom model type in Database Administrator. Objects of such a customized model type that are created either through Library Import or Database Editor have the:

- same model name as the file being read
- extension name as the value in the Model File Extension field (specified during custom model type creation in Database Administrator)

The tarballs of such a custom model will have the file included in it.

## **Using a New Model Type**

Tasks to be performed for using a new model:

- 1. Creating a New Model Type
- 2. Creating a New Library
- 3. Classifying the New Model Type

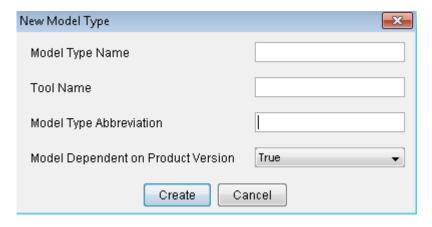
#### **Creating a New Model Type**

To create a new model type, do the following:

- 1. Open Database Administrator.
- **2.** Choose File New Model Type.
- **3.** Specify the following attributes:
  - □ *Model Type Name*: Name of the new model type
  - □ *Tool Name*: Tool the new model type belongs to

Model Management Support

- □ *Model Type Abbreviation*: Short name for the new model type
- Model Dependent on Product Version: Indicates whether or not the files of this model type needs change across releases

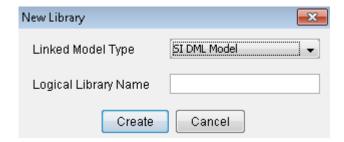


- 4. Save this new model type.
- **5.** Stop the server and restart the server so that the new model type is visible in Database Administrator and Database Editor.

#### **Creating a New Library**

To create a new library corresponding to the new model type, do the following:

- 1. Open Database Administrator.
- **2.** Choose *File New Library*.



**3.** Specify the library name and click *Create*.

#### **Classifying the New Model Type**

To classify a new model type, do the following:

Model Management Support

- 1. Open Database Editor.
- 2. Verify that new model type is visible in Database Editor.
- **3.** Verify that a root classification is created with the same name as the new model type.
- **4.** Select File Manage Model Classification <New\_Model\_Type> Classification.
- **5.** Create new classifications within this root classification, if needed.
- **6.** Define properties on this classification.
- 7. Create a new model of this new model type.

This custom model type can be used and managed as any other Allegro EDM model types. Rule Manager rule sets are applicable to them too, and each part/model is verified at each development state change.

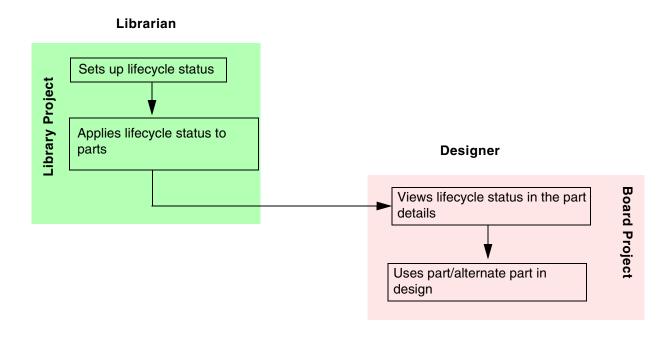
## **Managing Lifecycle Status of Parts**

Most electronic parts pass through several product life cycle stages, such as introduction, growth, maturity, decline, phase out, and obsolescence. The product life cycle stage that a part is in can be used as a basis for forecasting the obsolescence date for a part.

For example, the term *prerelease* in the library development flow implied that the part has completed the library development process, whereas, *prerelease* in the part lifecycle means that the part has completed all states of the library development process but is not yet available to designers for use because of enterprise reasons.

To track the stages of development for a part or a model in the librarian's world, as well as the lifecycle stage of the part from a designer's perspective, Allegro EDM uses two terms:

- Lifecycle status refers to the designer's view of the part lifecycle.
- **Development status** denotes the library development view of the part lifecycle.



Managing Lifecycle Status of Parts

The Lifecycle Status attribute is independent of the library flow (Development Status) and can be created for each installation of Allegro EDM. This attribute is **not** available for models.

#### Creating a Lifecycle Status

A new lifecycle status can only be created in Database Administrator.

To create a lifecycle status with a corresponding action, do the following:

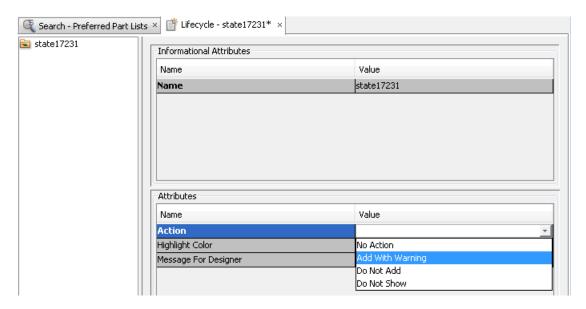
- 1. Open Database Administrator.
- **2.** Choose *File New Lifecycle*.
- **3.** Specify the name of the new lifecycle status.
- 4. Click Create.

A tab appears showing the new lifecycle status.

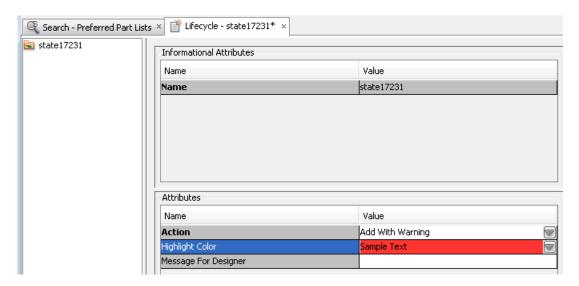
- **5.** Select from any of the following actions for the new status:
  - ☐ Add to Design
  - ☐ Add with Warning
  - ☐ Do not add
  - ☐ Do not show

Managing Lifecycle Status of Parts

For example, for the lifecycle status, *state17231*, select the action, *Add With Warning*.



**6.** You can also associate highlight colors for each lifecycle status. In Part Information Manager, a part in a particular lifecycle status is displayed in the color you define here. This makes it easy for designers to identify parts when they search for them in Part Information Manager.



7. Click Save.

#### Applying the Lifecycle Status to Parts

1. Open Database Editor.

Managing Lifecycle Status of Parts

- 2. Open a part and associate the lifecycle status.
- 3. Save the part.

**Note:** The lifecycle status of a part can be changed regardless of its development state, including the released state. There is no need to check out a part.

#### Viewing the Effect of the Lifecycle Status

- 1. Open a design project.
- 2. Open Part Information Manager.
- **3.** Search for the part with these statuses applied.
  - These parts will show up with background colors in the *Lifecycle Status* column.
  - ☐ The tool tip for part rows in the *Lifecycle Status* column shows the associated lifecycle along with its action.

*	*	*	*	*	*	*	*		
*	0	PPL	TOLERANCE	COST LAST UPDATED	DESCRIPTION	DIELECTIRC	HEIGH		
			10%		Chip Coil 180nH 50mA 10% 25MHz 0.60R 0603LQM18	DSMAX	0.9mm		
			10%		Chip Coil 0.1uH 250mA 10% 25MHz 0.26R 0805 LQM2	DSMAX	0.9mm		
	Approved, No Action								

## **Managing User Accounts**

The user management related tasks are:

- Creating User Accounts
- Editing User Attributess
- Deleting Users
- Changing User Password
- Searching Users and Roles

## **Creating User Accounts**

To create a user, do the following:

- 1. Open Database Administrator.
- **2.** Choose File New User.

The New User dialog box appears asking the new user ID.

**3.** Enter a unique user ID and click *Create*.

The *User* < *Detail* > tab appears.

- **4.** In the tabbed view, you can:
  - ☐ Use the *Informational Attributes* area to view key and schema-defined attributes.
  - Use the *Attributes* area to specify other mandatory and optional attributes.
- **5.** Choose *File Save* to save the new user in the database.

Alternatively, click the *Save* button on the toolbar.

**6.** To assign a password to the new user, see <u>Changing User Password</u>.

**Note:** The default password for a new user is blank.

## **Editing User Attributes**

To edit user attributes, do the following:

- 1. Open Database Administrator.
- **2.** Choose File Open User.

The Open User dialog box appears asking for the user ID.

**3.** Enter a unique user ID and click *Open*.

The *User Detail* tab appears.

- **4.** In the tabbed view, you can:
  - Use the *Informational Attributes* area to view the key and mandatory attributes.
  - Use the *Attributes* area to edit user-specific attributes, such as role.
- **5.** Choose *File Save* or click the *Save* button on the toolbar.

## **Deleting Users**

To delete a user, do the following:

**1.** Select a user in the explorer pane.

Alternatively, you can also search for a user and select it in the Search Results tab.

2. Choose File - Delete.

Alternatively, click the *Delete* button on the toolbar.

## **Changing User Password**

As an administrator, you can manage the password of users of the Allegro EDM Component Database. To do so:

1. Choose Tools – Set User Password.

The Set User Password dialog box appears.

- **2.** Enter an existing user name in the *User ID* field.
- **3.** Enter a new password for the user in the *New Password* field.

Managing User Accounts

- **4.** Confirm the password by entering it again in the *Confirm Password* field.
- 5. Click Change.

## **Searching Users and Roles**

To list the users and their roles in the component database, do the following:

- In the Database Administrator window, choose Search User.
   The Search Users tab opens.
- 2. Select *Role* from the *Attributes* area in the right pane.
- 3. Click the Search button ( ).

You can see the list of users and their role in the Search Results tab.

Managing User Accounts

7

## Verifying Objects with Rule Manager

Rule Manager acts as a data verification utility. It updates the status of parts and models based on default rules that run during different development stages of a part of model, and updates the status depending on custom rules that you may choose to define.

In Allegro EDM, a **rule** is a check to perform a specific task. For example, Allegro EDM provides a predefined rule that parts in the EDM database must have all their mandatory models linked to them. Whether a model is mandatory for a particular part is defined in the Allegro EDM schema. For example, EDM schema defines that an electrical part must have a Footprint Model. Another built-in rule is that when a part is being checked in, its linked model should *not* be in a checked-out state.

Each rule validates parts and models stored in the Allegro EDM Component Database. Often, Allegro EDM performs multiple checks (in a specific sequence), before you can proceed to the next development stage.

Allegro EDM has a set of built-in validation rules, which Database Editor runs when library objects go through various development stages, such as check-in, prerelease, and release.

For example, a part or model can move from:

- Preliminary or Checkout to Checked-In & Verified
- Checked-In & Verified to Pre Released or Released

When you select an object and a development stage in Rule Manager, the utility displays the rules that Allegro EDM runs by default for that stage and the selected object. You can choose to run any rule at any stage. For example, if you want to flow-verify a part, you might run the following checks (in sequence):

- Linked models should not be in checked-out state. For example, models can be linked to other models, or a model can be linked to a part.
- 2. Parts must have mandatory models linked to them.
- **3.** Part table file should be correctly generated.

Verifying Objects with Rule Manager

A group of related rules is called a rule set. A rule set is a collection of rules to be run for a particular task (for example, during the check-in of a part) and in a specific order. A task is considered successful if and only if all of its rules pass in a particular order.

Rule Manager has built-in rule sets that cover the basic checks that should be performed to ensure the validity of the parts and models as they go along their development stages. These built-in rule checks run on the following:

- Object types include: Cell Model, Footprint Model, Mechanical Kit, Mechanical Model, Mechanical Part, Model, Part, and Schematic Model
- Development stages include: checked-in, deleted, prereleased, released
- Dependency set checks include: local flow verification, prerelease dependency set, and release dependency set

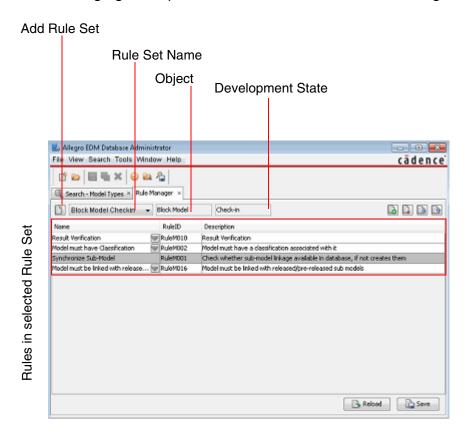
To launch Rule Manager to define a custom rule set or run an existing rule, do the following:

- 1. Open Database Administrator.
- **2.** Choose *Tools Rule Manager*.

The Rule Manager tab appears.

Verifying Objects with Rule Manager

The following figure explains the elements of the Rule Manager tab.



Verifying Objects with Rule Manager

## **List of Part and Model Rules**

For the list of rules and their descriptions, click the following links:

- List and Description of Part Rules
- List and Description of Model Rules
- List and Description of Global Rules global rules apply to both parts and models

Verifying Objects with Rule Manager

## **List and Description of Part Rules**

The following table lists the rules related to parts.

Rule	Name	Description
RuleP000	Current State Validity Check	This rule checks the compatibility of the part with the schematic by running front-to-back checks.
		Front-to-back (F2B) verification checks verify the mapping between the physical pin list of each primitive and that of its associated footprint. A primitive is a symbol with a unique pin list and a set of physical properties.
		The F2B checks also verifies the physical part rows associated with all the primitives in the part.
		The result of these checks are saved in the pdv_validation.txt log file, which is generated and stored in the metadata view.
		By default, this rule runs when any model is released. Allegro EDM checks whether the new version of the model is compatible with the part.
		If it finds that the parts are incompatible with the new version of the schematic, it prevents the release of the cell. A cell is a group of primitives.
		You will either need to fix the schematic, or make the part compatible by checking out the part and modifying it.

Rule	Name	Description
RuleP001	Mandatory Models	This rule checks whether the part has all its mandatory, associated, models.
		By default, this rule is run when a part moves from the Preliminary or Checked state to the Checked In state. A part can only move to the Checked In state when its associated model is linked to it.
		For example:
		If it is a schematic part, Allegro EDM checks whether a Cell Model and one linked Part Name model (schematic), which should itself have a linked cell, is linked to it.
		If it is a block part, a Block Model should be linked to it
		If it is a Capture part, one Capture Model, it the Capture flow is enabled.
		If it is a mechanical part, it should have at least one Part Name model linked to it.
		Essentially, a part should be linked to at least one front-end model.
		If any model is missing, this part fails the check
RuleP002	No Checked-out Models	This rule checks whether the currently selected part has any linked models (of any type, but restricted to model types) that are in the checked out state. If any linked model is in the checked out state, the part fails the rule.

Rule	Name	Des	scription
RuleP003	PTF Generation check		rule tries to generate the Part Table File for part for which the rule is being run.
			legro EDM encounters any of the following es, it does not generate a PTF:
			If key value combinations are duplicated, that is, if there are duplicate PTF rows
			If there are duplicate PTF subtypes across Part Name models (schematics)
		•	If there are multi-byte characters as a value in ECAD properties
			If the schematic is not linked to a cell or library
		•	If the PTF is not generated, the part fails this rule.
RuleP004	Linked Models Tarball check	This	models linked to parts should have tarballs. Is rule extracts all the tarballs for the linked dels in a temporary area.
		exis	tarball of a model linked to parts does not in the vault, or if extraction of any of these alls fails, the part fails this rule.
RuleP005	PTF Syntax and Semantic check	and	part table file (PTF) is checked for semantic syntactical errors. For example,SYMBOL should be within parentheses.
		If the	ere are errors, this part fails the check.

Rule	Name	Description
RuleP006	Front-To-Back Check	This rule runs front-to-back (F2B) verification checks for the selected part. Front-to-back compatibility checks the mapping between the physical pin list of each primitive and that of its associated footprint.
		A primitive is a part with a unique pin list and a set of physical properties.
		It also checks between all the physical part rows associated with all the primitives in the part for front-to-back verification.
		The result of these checks are saved in the pdv_validation.txt log file, which is generated and stored in the metadata view.
		F2B must be successfully completed for the part to pass this rule.
RuleP007	No Higher Released Versions of Parts	This rule ensures that no later version of the part, for which the rule is being run, is released. The rule also prevents a previous version of the part from being in the released state, if a later version is already released.
		If a higher version of this part is released, this part fails this rule.
RuleP009	Part must have classification	This rule checks that the part has a classification associated with it, that is, whether the part has been classified under at least one classification.
		If the part has no classification, the part fails this rule.
RuleP010	Mandatory Footprint Model	This rule checks whether the part has linked Footprint Models. If there are no Footprint Models linked to the part, Rule Manager checks for the JEDEC_TYPE property in the classification linked to the Schematic Model.
		If the Footprint Model is still not found in the JEDEC_TYPE property, the part fails this rule.

Rule	Name	Description
		·
RuleP012	Mandatory Mechanical Schematic Model Relation	This rule checks whether mechanical parts and kits have the mandatory mechanical Schematic Models.
		If the mechanical Schematic Model is not found, the part fails this rule.
		If a part fails rules RuleP006 and RuleP021, then the part fails this rule too.
RuleP013	Check ALT_SYMBOLS relation	The rule checks whether a Schematic Model has ALT_SYMBOLS specified as a property in its linked classification.
		If it does, then this part must also have this relation. If it does not, the part will fail the check.
RuleP014	Front-To-Back Check without Footprint	This rule runs front-to-back checks for a part while ignoring the footprint, and is part of the prerelease process.
RuleP015	Check linked model version	This rule checks whether the part is linked to the latest version of the models.
RuleP016	PACK_TYPE check against schematic model	This rule checks whether PACK_TYPE has a value matching one of the values in the chips.prt file of the cell associated with the schematic that is linked to the part.
RuleP017	Mandatory Module Model	This rule checks whether the part has a Module Model linked to it.
RuleP018	Performing Additional Data Integrity Checks	This rule checks whether the Schematic, Block, or Mechanical Models related to the part are modified.

Verifying Objects with Rule Manager

Rule	Name	Description
RuleP021	Front-To-Back Check with Capture Model	This rule verifies that there are no differences between the physical pin numbers in the footprint, and alternate symbols of the physical pins in the Capture part.
		The verification check reads footprints and ALT SYMBOLS linked to the part as well as the PCB footprint or ALT_SYMBOLs defined for the Capture model as properties.
		Important
		This rule does not cover errors in the Capture Model, such as duplicate logical or physical pins, property assignment across logical pins, and so on.

## **List and Description of Model Rules**

Rule	Name	Description
RuleM000	Current State Validity Check	This rule checks whether the linkages between the selected object and its linked parts and models are still valid.
		If this rule finds that the parts are incompatible with the new version of the schematic, it prevents the release of the model.

Rule	Name	Description
RuleM001	Synchronize Sub- Model	Checks whether submodel linkages are available in the Allegro EDM database for the following models:
		■ Schematic
		■ Footprint
		■ Block
		■ Padstack
		■ SI DML
		■ Capture
		Creates linkages between these models and submodels by creating, deleting, or adding a linkage, where required, after running the rule.
RuleM002	Model must have classification	Checks that the model has a classification associated with it, that is, that the model is in at least one classification. If the model has not been classified, it fails the check.
RuleM003	PTF Generation check	On running this rule, the PTF for the selected model is generated after traversing all the parts linked to the Model. This rule is only available for Schematic Models.
		If the PTF is not generated, check ${\tt dbeditor.log}$ for details.
RuleM004	PTF Syntax and Semantic check	The part table file (PTF) is checked for semantic and syntactical errors. For example, ALT_SYMBOL should be within parentheses.
		If there are errors, this model fails the check.
RuleM005	No Checked-out parts	This rule checks if the selected model has any checked-out part linked to it.
		If any parts linked to the model are checked out, the part fails the check.
RuleM006	Mandatory Cell Model	Checks that the model has a Cell Model linked to it.
		If the model does not have a Cell Model linked to it, it will fail the check.

Rule	Name	Description
RuleM007	No Higher Released Versions of Models	Checks that there are no higher released versions of the model for which the run is being run.
RuleM008	No Checked-out Sub Models	Checks that submodels linked to the model, for which the rule is being run, are not checked out.
		If any linked submodels are checked out, the model fails the check.
RuleM009	Baseline Models	When run, this rule compares the model being released to a baselined version of itself that Allegro EDM generated internally on previous check-ins of the model.
		If the model matches the baselined version, it passes this check.
RuleM010	Result Verification	This rule is run when the model, for which the rule is being run, has been modified in a utility other than Allegro EDM Database Editor, but is now being checked in using Database Editor.
		The rule verifies whether the model passed all the Checkplus rules, and checks the symchk files for Footprint Models. If the files are not present or have errors, this rule fails.
RuleM011	Check if a model can be marked for delete	This rule checks whether a model can be marked for deletion.
RuleM012	Current State Validity Check - Extended	This rule validates the state of the models related to the model for which the rule is being run by looking for the parts of the model and running <i>RuleP000</i> , that is, <i>Current State Validity Check</i> .
RuleM013	Adapter verification of custom models	Custom adapters that you create for custom model types, or the EDM default adapter,  DWFileNameAdapter, run their own checks to verify a custom or SI DML model.
		This rule runs the adapter-specific verification checks.

Rule	Name	Description
RuleM014	Synchronize submodel of custom models having merge capability	This rule checks whether linkages between submodels and their corresponding SI DML file names (checks only for the same file name) are available in the Allegro EDM database.
		If the linkage is not available, it is created. This rule is specific to custom models.
RuleM015	Rule to merge all custom models in the working set, which have merge capability to the compound file, before their release	In Allegro EDM, a model with multiple models within it is a compound model. This compound model is kept on the disk by EDM as a compound file type. The models within the main model are called submodels.
		When you do certain tasks, such as checking out one of the submodels of a compound model, the submodel is extracted from the main model. When the submodel is being released, Allegro EDM merges the submodel (that is, the extracted portion) back with the main model.
		Allegro EDM provides SI DML models as an example of how a custom model needs to be structured. As a result, although provided out of the box in EDM, SI DML models are also considered custom models in Allegro EDM.
		When this rule is run, it merges all SI DML and custom models in the working set back with their compound files, before their release - provided these custom models can be extracted from and merged into a main model.
RuleM016	Model must be linked with released or prereleased submodels	This rule checks whether Block Models are linked with released or prereleased submodels and subparts, and if Module Models are linked with released or prereleased submodels.
		If they are not, the model for which this rule is being run fails the check.

Rule	Name	Description
RuleM017	Mark model updated	Whenever a submodel is revised, its parent model is marked as requiring an update - indicated in Allegro EDM Database Editor as <i>Update Required</i> . You then have the option of accepting the update.
		However, this rule is run when a model is being released. In this case, regardless of whether you have accepted the update, the model is marked as updated. Since the model is being released, EDM assumes that the model is updated.
RuleM018	Linked models not present in the current working set must be released or prereleased at least once.	This rule checks whether submodels, which are linked to the parent model for which the rule is run, and which are not in the current working set, are released or prereleased at least once.
		If the submodels linked to the parent model are not released or prereleased at least once, this model will fail the check.
RuleM019	Linked classification(s) must not be obsolete	This rule checks whether the classifications linked with the model are obsolete. If any one of the linked classifications is obsolete, this model fails the check.
RuleM020	Check if the model is a Standard Model	This rule checks whether the model is a Standard Model on the basis of its file structure. If the model is not a Standard Model, this model fails the check.
RuleM021	Mandatory Capture Model File Type	This rule checks whether the Capture model for which this rule is being run has a Capture Model File Type model linked to it.
		If the relation is not available in the database, this model will fail the check.

Verifying Objects with Rule Manager

## **List and Description of Global Rules**

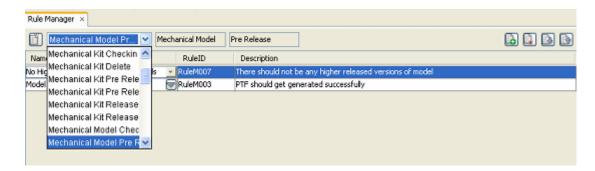
Global rules are run for parts and models.

Della	Nama	Description
Rule	Name	Description
RuleG001	Release Dependency Set	This rule checks the current working set and identifies all the objects that are linked to the model that is being released.
		When this rule is run, it tries to release all such linked objects if the objects are checked in and verified. If any of these objects cannot be released, the object fails the rule.
RuleG002	Pre Release Dependency Set	This rule checks the current working set and finds all the objects which are linked to the model being prereleased. This rule tries to prerelease all such objects. If any of the objects cannot be released, the model fails this rule.
RuleG003	Parent model requires an update	Whenever a submodel is revised, its parent model is marked as requiring an update - indicated in Allegro EDM Database Editor as <i>Update Required</i> . You then have the option of accepting the update.
		If you accept the update, running this rule marks the parent model as updated.
RuleG005	Validate Classification Property Values	This rule checks for invalid characters in property values.
		If you have enabled multibyte support (this is done using the adwschema utility) in the Allegro EDM database, then:
		<ul> <li>UTF-8 characters are allowed in the values of searchable properties</li> </ul>
		<ul> <li>only ASCII characters are supported in PTF property values.</li> </ul>
		if multibyte support is not enabled, then in all property values, only ASCII characters are supported.

Verifying Objects with Rule Manager

#### **Default Definitions of Rule Sets**

Each built-in rule set has rules. The rule sets might contain different rules for each company/ enterprise. To know the rules in a rule set, select the rule set from the drop-down list.



## **Creating a Rule Set**

To create a rule set, do the following:

1. In the Rule Manager tab, click the Add Rule Set button.



The New Rule Set dialog box appears.



2. Choose a type of rule from the *Type* drop-down list.

The list contains all the component database entity types, such as part, model, Footprint Model, Symbol Model, Schematic Model, Shape Model, Flash Model, Mechanical Model, Format Model, Padstack Model, Capture Model (if the Capture schema is enabled using the adwschema utility), and Datasheet Model.

**3.** Choose a development action from the list of predefined actions.

The *Action* drop-down list shows the following options:

- ☐ Check-in
- Major Modification
- ☐ Minor Modification
- ☐ Verify Metadata
- ☐ Flow Verification
- ☐ Release
- ☐ Obsolete
- **4.** Specify a name of your rule set in the *Rule Set* field.

Verifying Objects with Rule Manager

#### 5. Click Create.

The new rule set appears in the drop-down list. You can now use the buttons, highlighted in the following image, to add rules to this newly created rule set.



Verifying Objects with Rule Manager

## **Merging Rule Sets**

Allegro EDM has a set of built-in validation rules, which Database Editor runs when library objects go through various development stages, such as check-in, prerelease, and release. Cadence recommends using the out-of-the-box rules provided by Allegro EDM to ensure the correctness of all released objects. However, you can use a customized set of rules depending on your organization's requirements.

If you have a customized rule set and an updated rule set is available in an incremental or major release of Allegro EDM, when you initiate any library flow task, Allegro EDM prompts you to:

- Use the updated set of rules provided by Allegro EDM (recommended)
- Continue using the existing customized set of validation rules

#### Merging existing customized rules with updated rule sets

If you want to use the updated set of rules but also want to merge your existing customized rules with the updated rule sets, do the following:

- **1.** Rename the rules file, rules.xml at <adw\_conf\_root>\<company>\<site> to rules.xml.backup.
- 2. Launch Database Administrator and open Rules Manager.

The updated rule set is loaded in Rules Manager.

- **3.** Reapply the previous customizations to these rule sets.
- 4. Click Save to save this customized rule set.
- **5.** Ensure that you restart Database Editor so that the customized rule set is loaded.

#### Removing existing customized rule set

If you want to use only the updated set of rules and want to stop using the existing customized rule set, rename the rules file containing your customized rules, that is,

<adw\_conf\_root>\<company>\<site>\rules.xml to rules.xml.backup.

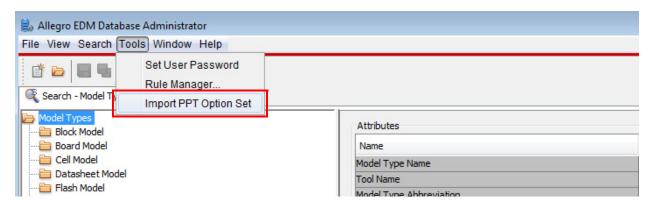
8

## **Importing PPT Option Sets**

You can import existing PPT options sets using Library Import or Database Administrator. For details about importing using Library Import, see the Configuring Library Import Options section in *Allegro EDM Library Import User Guide*.

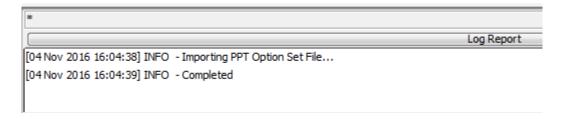
In Database Administrator, to import an existing PPT option set, do the following:

1. Choose Tools - Import PPT Option Set.



2. In the Select PPT Option Set File (.dat File) dialog, select the .dat file you want to import and click *Open*.

The Log Report pane indicates the progress and completion of the import.



When this option set is uploaded to the database, Allegro EDM applies the settings in the option set file to all parts in the project. The default will be applied to those parts for which settings have not been defined.

Importing PPT Option Sets

When this option set is uploaded to the database, by default, Allegro EDM sets the annotate to design and visibility attributes for the newly-imported part properties:

- □ Injected property: *Annotate to Design* is set to No, and *Visibility* is set to Invisible. Both attributes can be edited using Allegro EDM Database Editor.
- □ Key Property: *Annotate to Design* is set to Yes, and *Visibility* can be set by the librarian using Database Editor.



## **Database Administrator User Interface**

The following section describes the user interface of Database Administrator.

## Menu Help

- File Menu
- View Menu
- Search Menu
- Tools Menu
- Window Menu

#### File Menu

Command		Description
New	Model Type	Creates component database entities for Cadence tools, such as Allegro and Allegro Design Entry HDL.
	Library	Creates component database entities for libraries.
	Preferred Part List	Creates component database entities for preferred part lists (PPL).
	User	Creates users for different roles in the component database.
	Lifecycle	Creates a new lifecycle state that can be applied to parts in the database.

Database Administrator User Interface

Command		Description
Open	Model Type	Displays the component database entities for Cadence tools.
	Library	Displays library component database entities.
	Preferred Part List	Displays PPL component database entities.
	User	Displays user details.
	Lifecycle	Displays details of a lifecycle.
Manage Part Classification		Opens the <i>Part Classification</i> tab listing the part classifications in the database.
Manage Block Part Classification		Opens the <i>Block Part Classification</i> tab listing the block part classifications in the database.
Manage Mechanical Part Classification		Opens the MECAD Component Classification tab listing the mechanical classifications in the database.

Database Administrator User Interface

Command			Description
Manage Model Classification	<model< td=""><td>Type&gt;</td><td>Choose any of the submenus to view a model-specific tab.</td></model<>	Type>	Choose any of the submenus to view a model-specific tab.
			■ Schematic Model
			■ Mechanical Model
			■ Block Model
			■ Footprint Model
			■ Padstack Model
			■ Flash Model
			■ Shape Model
			■ Board Model
			■ Format Model
			■ Module Model
			■ Datasheet Model
			■ SI DML Model
			Opens the < Model Type> Model tab listing the model classifications defined in the database.
Save			Save the changes made into the component database.
Save All			Save all the changes.
			Important
			Save and Save All commands are disabled in the Search and Classification tabs.
Delete			Remove the selected data entity permanently.
			Important
			The Delete command is not available in Search and Classification tabs.

Database Administrator User Interface

Command	Description
Exit	Closes the Database Administrator.

## **View Menu**

Command	Description
Search Results	Displays the Search Results tab.
Show Errors	Shows the Error Report tab.
Toolbar	Toggles the display of the toolbar.
Refresh	Refreshes the Search Results tab.

### **Search Menu**

Command	Description
Model Type	Open the <i>Search – Tools</i> tab for performing search on tool entities.
Library	Open the Search – Libraries tab for performing search on library entities.
Preferred Part List	Open the Search – PPLs tab for performing search on PPL entities.
User	Open the Search - Users tab for performing search on users.
Lifecycle	Open the Search – Lifecycle tab for performing search on types of lifecycle status.
Search Option	Lets you select a search mode for performing search. Your options are: Match Any and Match All.
Save Search Criteria	Save a new search criteria.
Load Search Criteria	Load an existing search criteria.

Database Administrator User Interface

## **Tools Menu**

Command	Lets you
Set User Password	Reset the password for a user
Rule Manager	Open the Rule Manager tab.

## **Window Menu**

Command	Lets you
Single Detail Window	Control tab-based views in the Database Administrator window.
Recently Viewed	View the history of the tabs you have recently opened in the main window.

Database Administrator User Interface

### **Tabs**

The Data Administrator window allows you to open as many tabs as you want to. A tab can be of the following types:

- Search Tab
- Details Tab
- Search Results Tab
- Log Report Tab
- Error Tab

The Search Results and the Error tab appear at the bottom of the window. Error tab only appears when you encounter some errors.

#### Search Tab

This tab appears when you choose to search for data entities such as tool, library, PPL, and user. When this tab is opened, two tabs appear at the bottom of the right pane:

- Attributes: When selected, the default search attributes (for the selected data entity) appear.
- Relations: When selected, the default search relations (for the selected data entity) appear.

#### **Details Tab**

The detail of a component database entity can be viewed with the help of a Detail tab. Whenever you open an administrator data entity (such as tool, library, user, or PPL), its detail appear in form of a *<Entity> Detail* tab.

For example, details of a tool=ALLEGRO, model type= FLASH, and version= 15.7 (found after searching for tools), appear under the *Tool Detail – ALLEGRO->FLASH->15.7* Tab.

**Note:** Tool, Model Type, and Version define a unique tool data entity. This is the information required whenever you create a tool data entity.

Database Administrator User Interface

#### Search Results Tab

All the search results appear at the bottom of the Data Administrator window. This pane also shows the number of results that match your search criteria at the bottom-left of the Search Results tab. The Search Results tab always displays the last search performed, irrespective of the tabs, which is active.

#### Log Report Tab

This tab appears before the *Search Results* tab, at the bottom and shows all the log details for each operation performed in Database Administrator.

#### **Error Tab**

This tab appears beside the Search Results tab, at the bottom and shows all the errors encountered, if there are any, during the process.

## **Classification Tabs**

Following are the types of classification tabs available.

- Part Classification Tab
- Block Part Classification Tab
- Mechanical Part Classification Tab
- Model Classification Tab

#### **Part Classification Tab**

This tab appears when you choose *File – Manage Part Classification*. In this tab, the explorer pane contains the classifications, their versions, and the subclassifications, in a tree-like structure.

The right pane contains the properties corresponding to the classification (selected in the explorer pane). Besides this, the right pane allows you to add, edit, delete, and order

Database Administrator User Interface

properties of the selected classification. The following table lists the various interface elements in the right pane of the classification tab.

Control	Description
Name	Name of the property.
Mandatory drop-down list	Defines whether the property is mandatory. The values you can enter are: True or False.
Searchable drop-down list	Defines whether the property is searchable. The values you can enter are: True or False.
Data Type drop-down list	Defines the data type of the property. The values you can enter are: String or Numeric.
Default Value	Helps you specify the default value for the property.
Predefined Values	Helps you specify any predefined value(s) for a property.
Display in Search	This is available in the <i>Property Display Order</i> tab of the right pane. Selecting this check box lets you specify if the property should be appear in the search results.
Property Status Icon	Specifies whether the property is new, marked for deletion, inherited, over-ridden, or owned.
Add	Click this button to add a new property.
Delete	Click this button to delete a property.

#### Part Classification Tab: Context-Menu Commands

The pop-up menu options available in the explorer pane of the *Part Classification* tab are:

Command	Lets You
Add Child	Add a child node.
Revise	Create an incremental version of the classification.
Mark Obsolete	Make a classification unavailable to library data users.
Rename	Rename a classification.
Save	Save the changes in classifications.
Save Hierarchy	Save all the unsaved data (for all child classification nodes) present under the node selected.

Database Administrator User Interface

Command	Lets You	
Refresh	Refresh the data for the classification node selected.	
Delete	Delete a classification.	
Сору	Copy a classification.	
Paste	Paste a classification node into a desired node.	
Paste Classification Subtree	Paste a classification node along with its child nodes into a desired node.	
Show Obsolete Classifications	View all the obsolete classifications.	
Hide Obsolete Classifications	Hide all the obsolete classifications.	
Property Display Order	■ Copy, paste, export, import, and reset the display order specified for the properties of the selected classification.	
	Highlight the child classifications which have a display order set below the selected classification node.	

#### **Block Part Classification Tab**

This tab appears when you choose *File – Manage Block Part Classification*. In this tab, the explorer pane contains the block part classifications, their versions, and the subclassifications, in a hierarchical tree-like structure.

Similar to the Part Classification tab, the right pane contains the properties corresponding to the classification (selected in the explorer pane). Besides this, the right pane allows you to add, edit, delete, and order the properties of the selected classification.

#### **Mechanical Part Classification Tab**

This tab appears when you choose *File – Manage Mechanical Part Classification*. In this tab, the explorer pane contains the mechanical part classifications, their versions, and the subclassifications, in a hierarchical tree-like structure.

Similar to the Part Classification tab, the right pane contains the properties corresponding to the classification (selected in the explorer pane). Besides this, the right pane allows you to add, edit, delete, and order the properties of the selected classification.

Database Administrator User Interface

#### **Mechanical Part Classification Tab: Context-Menu Commands**

The pop-up menu options available in the explorer pane of the *Mechanical Part Classification* tab are:

Command	Lets You	
Add Child	Add a child node.	
Revise	Create an incremental version of the classification.	
Mark Obsolete	Make a classification unavailable to library data users.	
Rename	Rename a classification.	
Save	Save the changes in classifications.	
Save Hierarchy	Save all the unsaved data (for all child classification nodes) present under the node selected.	
Refresh	Refresh the data for the classification node selected.	
Delete	Delete a classification.	
Сору	Copy a classification.	
Paste	Paste a classification node into a desired node.	
Paste Classification Subtree	Paste a classification node along with its child nodes into a desired node.	
Show Obsolete Classifications	View all the obsolete classifications.	
Hide Obsolete Classifications	Hide all the obsolete classifications.	
Property Display Order	Copy, paste, export, import, and reset the display order specified for the properties of the selected classification.	
	Highlight the child classifications which have a display order set below the selected classification node.	

#### **Model Classification Tab**

This tab appears when you choose *File – Manage Model Classification – < Model Type>*. In this tab, the explorer pane contains the model classifications, their versions, and the subclassifications, in a hierarchical tree-like structure.

Database Administrator User Interface

Similar to the Part Classification tab, the right pane contains the properties corresponding to the classification (selected in the explorer pane). Besides this, the right pane allows you to add, delete, and order properties of the selected classification. The following table lists the various interface elements in the right pane of the classification tab.

Control	Description
Attribute Name	Name of the property.
ECAD Type	Specifies whether the property is an ECAD type. Your options are: True or False
Mandatory drop-down list	Specifies whether the property is mandatory. The values you can choose are: ${\tt True}\ or\ {\tt False}$
Searchable drop-down list	Specifies whether the property is searchable. The values you can choose are: ${\tt True}$ or ${\tt False}$
Data Type drop-down list	Specifies the data type of the property. The values you can choose are: String or Numeric
Default Value	Helps you specify the default value for the property.
Predefined Values	Helps you specify any predefined value(s) for a property.
Display in Search	This is available in the <i>Property Display Order</i> tab of the right pane. Selecting this check box lets you specify if the property should appear in the search results.
Property Status Icon	Specifies whether the property is new, marked for deletion, inherited, over-ridden, or owned.
Add	Click this button to add a new property.
Delete	Click this button to delete a property.

The following table lists the interface elements in the right pane of the classification tab, which are exclusive to Schematic Models, Mechanical Models, and Block Models.

Property Header	Description
Attribute Name	Name of the property.
CAD Property Name	By default, this is same as Attribute Name. But, you can edit it.

Database Administrator User Interface

Property Header	Description
PTF Mapping	Specifies the PTF mapping type. Your options are: Key, Injected, Optional Key, Optional Key & Injected, Key and Injected, Global Added, and Subtype.
Key Property Order	Specifies the order of the key property in PTF generation.
Optional Property Value	This header is available if you select the Optional Key or Optional Key & Injected value under the PTF Mapping header.
Global Property Value	This header is available if you select the Global value under the PTF Mapping header.
Link To	Helps you create links for Part and Models and their corresponding attributes (Part. <part attribute=""> or Model Type.<model attribute="">). Some examples are:</model></part>
	Footprint Model.ModelName, Part.PartNumber, PartName Model.JEDEC Type

#### **Model Classification Tab: Context Menu Commands**

The pop-up menu options available in the explorer pane of the Model Classification tab are:

Command	Lets You
Add Child	Add a child node.
Revise	Create an incremental version of the classification.
Mark Obsolete	Make a classification unavailable to library data users.
Rename	Rename a classification.
Save	Save the changes in classifications.
Save Hierarchy	Save all the unsaved data (for all child classification nodes) present under the node selected.
Refresh	Refresh the data for the selected classification node.
Delete	Delete a classification.
Сору	Copy a classification.
Paste	Paste a classification node into a desired node.

Database Administrator User Interface

Command	Lets You
Paste Classification Subtree	Paste a classification node along with its child nodes into a desired node.
Show Obsolete Classifications	View all the obsolete classifications.
Hide Obsolete Classifications	Hide all the obsolete classifications.
Property Display Order	<ul> <li>Copy, paste, export, import, and reset the display order specified for the properties of the selected classification.</li> </ul>
	Highlight the child classifications which have a display order set below the selected classification node.

Database Administrator User Interface

## **Dialog Box Help**

This section describes the dialog boxes that prompt you for information. Most of the dialog boxes are intuitive and have been grouped on the basis of the type of tasks they are used for.

### **New Dialog Boxes**

These dialog boxes appear when you are create administrator data for entities such as Model Type, Library, PPL, User, and Lifecycle. The dialog boxes are:

- New Model Type Dialog Box
- New Library Dialog Box
- New Preferred Part List Dialog Box
- New User Dialog Box
- New Lifecycle State Dialog Box

#### **New Model Type Dialog Box**

Field	Description
Model Type Name	Specify a name for the model type.
Tool Name	Specify the software type.
Model Type Abbreviation	Specify another name for the model type.
Model Dependent on Product Version	Select True or False if the model type is version-specific.
Create	Click to confirm the creation of the new model type.
Cancel	Click to cancel the new entity creation process.

#### **New Library Dialog Box**

Field	Description
Linked Model Type	Select the model type to be linked to the library being created.
Logical Library Name	Specify a name for the new library.

Database Administrator User Interface

Field	Description
Create	Click to confirm the creation of a new library entity.
Cancel	Click to cancel the new library creation.

### **New Preferred Part List Dialog Box**

Field	Description
Preferred Part List Name	Specify a new PPL name.
Create	Click to confirm the creation of a new PPL entity.
Cancel	Click to cancel the new PPL creation.

#### **New User Dialog Box**

Field	Description
User ID	Enter a new user ID.
Create	Click to confirm the creation of a new user.
Cancel	Click to cancel the new user account creation.

#### **New Lifecycle State Dialog Box**

Field	Description
Name	Enter a new lifecycle name.
Create	Click to confirm the creation of a new lifecycle.
Cancel	Click to cancel the new lifecycle creation.

## **Open Dialog Boxes**

These dialog boxes appear when you are open database objects (for schema-defined elements) to view their details. The dialog boxes are:

Database Administrator User Interface

- Open Model Type Dialog Box
- Open Library Dialog Box
- Open Preferred Part List Dialog Box
- Open User Dialog Box
- Open Lifecycle Dialog Box

#### **Open Model Type Dialog Box**

Field	Description
Model Type Name	Lets you enter a tool name.
Open	Lets you view the required data entity.
Cancel	Lest you cancel this process.

### **Open Library Dialog Box**

Field	Description
Linked Model Type	Lets you select a tool model.
Logical Library Name	Enter a library name.
Open	Lets you view the required data entity.
Cancel	Lest you cancel this process.

#### **Open Preferred Part List Dialog Box**

Field	Description
Preferred Part List Name	Enter a PPL name.
Open	Lets you view the required data entity.
Cancel	Lest you cancel this process.

Database Administrator User Interface

#### **Open User Dialog Box**

Field	Description
User ID	Enter a user ID.
Open	Lets you view the required data entity.
Cancel	Lest you cancel this process.

## **Open Lifecycle Dialog Box**

Field	Description
Name	Enter a lifecycle name.
Open	Lets you view the required data entity.
Cancel	Lest you cancel this process.

## **Add Dialog Box**

Field	Description
Enter Property Name	Specify the name of the property to add.
Create	Click this to add the property.
Cancel	Closes the dialog box.

## **Delete Property Dialog Box**

Field	Description
Enter Property Name	Specify the name of the property to delete.
Create	Click this to delete the property.
Cancel	Closes the dialog box.

Database Administrator User Interface

## **Set User Password**

Field	Description
User ID	Enter the user ID of a Database Administrator tool user.
New Password	Enter the new password.
Confirm Password	Confirm the new password.

В

## **About Attributes and Relations**

All the administrator data attributes display in the following two panes:

- **1. Informational Attributes**: These are attributes that are derived from the component database schema or can be the key attributes. These attributes are non-editable.
- 2. Attributes: These attributes are editable, and can be modified by the administrator.

Furthermore, depending on their nature and the way they appear in Database Administrator and Database Editor interfaces, these attributes have been classified into the following categories.

Interface Appearance	Signifies	Notation (in Documentation)
Bold	key attribute or mandatory attribute	K
Roman Normal	that the attribute has been derived from the database schema, and it is non- editable	S
	or	
	optional attribute	

**Note:** This notation has been used in the documentation to explain the nature of the attributes.

**About Attributes and Relations** 

## **Default Attributes**

The following table lists the default attributes that are available in the out-of-the-box implementation of component database.

Entity Name	Attribute Name	Description	Туре
Model Type			
	Model Type Name	Specifies a unique name for the model type. It must end with the keyword, Model.	K
	Tool Name	Defines the name of the Cadence PCB tool with which the model type is associated.	K
	Model Type Abbreviation	Specifies the abbreviated name for the model type. By default, the following are available: BLOCK, MECH, SYM, DATASHEET, FLASH, JEDEC, FMT, MODULE, PADSTACK, SCH, PADSHAPE, PCBSI	K
	Tool Version	Specifies the current release of Allegro EDM.	K
	Model Dependent on Product Version	Select True or False to indicate whether the files of the model type need to be changed for every future release of Allegro EDM.	K
	Originator	Specifies the user name who created the model type.	S
	Adapter	Name of the executable file that is used to access information of a model of a particular model type.	K
	Model File Extension	Specifies the file extension that identifies the models of a particular model type.	K
	Model Name Case Sensitivity	Specifies the case sensitivity to be applied to a model of a particular model type when created in Database Editor.	K
	Valid Character Set	Valid set of characters that can be used to create a model of a particular model type.	K
	Comment	Specifies any comments or instructions for a model type.	S

About Attributes and Relations

Entity Name	Attribute Name	Description	Туре
	Creation Date	Specifies the creation date of the model type.	S
	Modification Date	Specifies the last modification date of the model type.	S
	Linked to	Specifies the part type with which the model type is associated.	S
	Part Linkage Property	Specifies the name of the part property that links a part to a model.	K
Library			
	Linked Model Type	Specifies the model type to be linked to the library being created.	K
	Logical Library Name	Defines the name of the logical library.	K
	Originator	Specifies the user name who created the library.	S
	Creation Date	Specifies the date (mm/dd/yyyy : hh:mm: AM/PM) when the library was created.	S
	Modification Date	Specifies the date (mm/dd/yyyy : hh:mm: AM/PM) when the library was modified last.	S
	Comment	Contains remarks for the library.	S
PPL			
	Name	Defines the name of the PPL.	K
	Originator	Specifies the user name who created the PPL.	S
	Creation Date	Specifies the date (mm/dd/yyyy : hh:mm: AM/PM) when the PPL was created.	S
	Modification Date	Specifies the date (mm/dd/yyyy : hh:mm: AM/PM) when the PPL was modified last.	S
	Comment	Contains remarks for the PPL.	S
	Highlight Color	Associates a color to a PPL so as to visually differentiate various PPLs.	S
User			
	User ID	Specifies the user ID.	K

**About Attributes and Relations** 

Entity Name	Attribute Name	Description	Туре
	Originator	Specifies the user name who created the user.	S
	Creation Date	Specifies the date (mm/dd/yyyy : hh:mm: AM/PM) when the user was created.	S
	Modification Date	Specifies the date (mm/dd/yyyy : hh:mm: AM/PM) when the user was modified last.	S
	First Name	Specifies the first name of the user.	K
	Last Name	Specifies the last name of the user.	K
	Role	Lists the available roles for a user in form of a drop-down list.	K
	Email Address	Specifies the e-mail of the user.	K
	Address	Specifies the address of the user.	K
	City	Specifies the city of the user.	K
	State/Region	Specifies the State/Region of the user.	K
	Country	Specifies the Country of the user.	K
	Postal Code	Specifies the Zip code of the user.	K
	Work Phone Number	Specifies the user work phone number	S
	Home Phone Number	Specifies the use home phone number	S
	Comment	Contains remarks for the user.	S

114

**About Attributes and Relations** 

## **Default Relations**

Following are the default relations that are available in the out-of-the-box component database entities.

#### **Preferred Part Lists**

- Parent PPL: This relation contains the Name of the parent PPL.
- Linked Parts: This relation contains the Part Number and their Revision Numbers.
- Linked Block Parts: This relation contains the Part Number and their Revision Numbers.

About Attributes and Relations

C

## **Searching Database Entities**

Once you have a populated component database and you want to locate data specific to an entity, you can use the search features in Database Administrator. The search can be performed using:

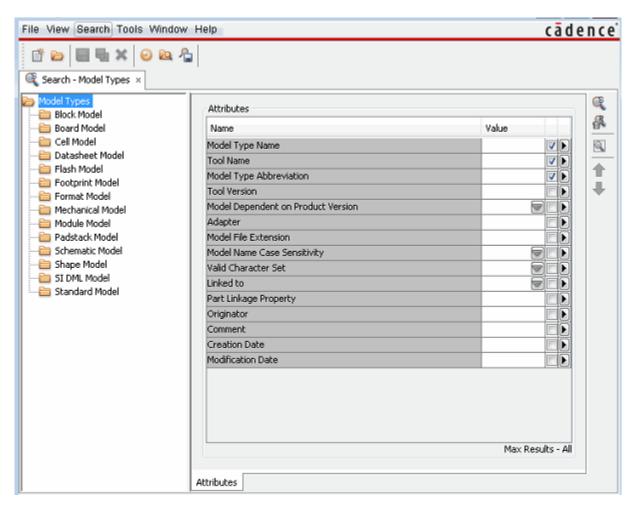
- Attributes: These are search parameters based on entity-specific properties.
- **Relations**: These are search parameters based on relations and their values.

#### **Searching the Allegro EDM Component Database**

In the Database Administrator window, choose Search – <Entity\_Name>.
 For example, choose Search – Model Type to search for tools.

Searching Database Entities

The appropriate *Search - <Entity>* tab appears.



**2.** In the right pane, choose the *Attributes* tab to specify attribute-based parameters for search.

For information on how to specify the search criteria, see <u>Performing Searches</u>.

- 3. After specifying the search criteria, click 👩.
  - Search results appear in the *Search Results* tab at the bottom. Database Administrator supports string search.
- 4. Double-click a search result row to view details.

#### **Search-Related Procedures**

Performing Searches

Searching Database Entities

- Resetting the Search Criteria
- Saving the Search Criteria
- Running a Saved Search Criterion
- Refreshing Search Results
- Working with Search Results

#### **Performing Searches**

Database Administrator supports performing search on attributes and relations in the component database. Depending on the complexity of the search expression, you can perform two types of searches.

- Simple Search
- Complex Search

#### Simple Search

This involves specifying direct and straightforward search parameters. For example, a search on libraries with name equal to <code>capacitor</code>, or a search on tools where version is 16.6.

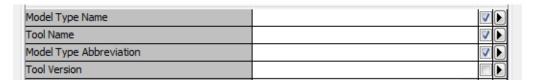
To perform a simple search:

**1.** Choose *Search – <Entity>* command.

The Search – <Entity> tab.

**2.** Choose the *Attributes* tab.

In the *Attributes* tab, the right pane contains a four-column table containing attributes.



In the *Attributes* tab, the right pane contains a four-column table containing attributes. The details of the columns are:

Control	Description
Name	Name of the attribute.

Searching Database Entities

-	
Control	Description
Value	Lets you specify the search value for the attribute.  Depending on the attribute type this can be a drop-down list, text field, or a calendar field.
Check box	Lets you specify whether to display the attribute (as a header) in the search results.
Triangle button ( <b>)</b>	Click this to form a complex search expression involving relational or boolean operators.

**3.** Enter the value next to the attribute or relation in the *Value* column.

## Important

For search parameter values that involve special characters ensure that you use quotes ('') around the specified value.

- **4.** If you have multiple attributes or relations to search:
  - □ Choose Search Search Options Match All to specify the AND operator between all the search parameters.
  - □ Choose Search Search Options Match Any to specify the OR operator between all the search parameters.
- **5.** To include or exclude the visibility of attributes or relations in the search results, select the check box beside the attribute.
- 6. Click 🧣 .

The search results appear in the Search Results tab.

#### Complex Search

Database Administrator allows you to perform complex search queries using a variety of relational operators. The following table lists the operators and their usage.

Operator	Usage
<	The value to search must be less than the specified value.
(Less than)	

# Allegro EDM Database Administrator User Guide Searching Database Entities

Operator	Usage
> (Greater than)	The value to search must be greater than the specified value.
!= (Not Equal to)	The value to search must not match the specified value.
== (Equal to)	The value to search must match to the specified value.
<= (Less than or equal to)	The value to search must be less than or equal to the specified value.
>= (Greater than or equal to)	The value to search must be greater than or equal to the specified value.
(Case insensitive string match)	The pattern of the first value must match the pattern of the second value. The value can be included anywhere in the string. With this operator, character case is ignored so that <code>library</code> is considered a match for <code>LIB*</code> .
!~~ (Case insensitive string not match)	The pattern of the first value must not match the pattern of the second value. The value can be included anywhere in the string.
	With this operator, character case is ignored. For example, a first value of Part Name and a second value of pA* $nA*$ would result in a FALSE comparison since the two are considered a match regardless of the difference in uppercase and lowercase characters.
~= (Case sensitive string match)	The pattern of the first value must match the pattern of the second value. The value can be included anywhere in the string. This includes testing for uppercase and lowercase characters. For example, Part Number is not a sensitive match for the pattern value pa* nu because the uppercase values (P and N) will not match the lowercase specification pattern.
!~=	The pattern of the first value must not match the pattern of the
(Case sensitive string not match)	second value. The value can be included anywhere in the string. For example, if the first value is $Part*$ a second value of $part*$ would produce a true result because the lowercase $p$ is not an exact match to the first value uppercase $P$ .
AND	The first value and the second value must be present.
(Logical AND)	

Searching Database Entities

Operator	Usage
OR	Specifies that either of the values be present.
(Logical OR)	
*	Specifies any number of characters match.
(Any String of Characters)	
?	Specifies exactly one character match.
(Any Single Character)	

- By default, in case you do not provide any operator, the ~~ (case-insensitive string match) operator is used.
- If you specify underscore (\_) as a part of the search string, then it is treated as the? character (single character).
- You can use the **b** button to view the operators supported.

#### Resetting the Search Criteria

To reset a search criteria:

- 1. Select a criterion row in the *Attributes* tab containing the existing search criterion.
- 2. Click \_\_ in the right pane.

The search criterion disappears.

#### Saving the Search Criteria

To save a frequently used search query and its attribute and relation parameters:

- **1.** In the *Search <Entity>* tab, fill in the search criteria.
- 2. Choose Search Save Search Criteria.

The Save search criterion dialog box appears.

- 3. Enter the name of the search criteria in the New Search Criterion field.
- 4. Click Save.

Searching Database Entities

**Note:** All the search criteria, by default, are saved in the home directory (\$HOME/atmdir/search directory) with a filename extension,.ctr.

#### **Running a Saved Search Criterion**

To run a saved search criterion:

1. Choose Search – Load Search Criteria.

The Load search criterion dialog box appears listing all the previously saved search criteria.

2. Choose a search criteria from the Search criteria in current project list.

Alternatively, click *Browse* to navigate to the folder containing saved search criteria files (.ctr).

3. Click Load.

The search criterion for the desired database entity appears.

4. Click the Search button.

#### Refreshing Search Results

To refresh the *Search Results* tab, choose *View – Refresh*. Alternatively, click the *Refresh* button on the toolbar.

## **Working with Search Results**

The *Search Results* tab displays the database elements meeting the search criteria. With these database elements, you can perform the following tasks:

- Copying Database Entities
- Viewing Entity Details

#### **Copying Database Entities**

The ability to copy database elements helps you create relations between database elements. The explorer pane of the window helps you create and manage relations.

To do so:

Searching Database Entities

- 1. Select a search result row.
- **2.** Choose *Copy* pop-up menu command.
- **3.** Paste the data into the explorer pane at the appropriate entity node.

#### **Viewing Entity Details**

- 1. Select a search result row.
- **2.** Double-click the row or choose the *Show Details* pop-up menu command.

The < Entity\_Detail> tab appears.

#### **Viewing Recently Visited Details Tabs**

Choose the *Window – Recently Viewed – <Entity\_Detail>* command to open the recently viewed *<Entity\_Detail>* tabs.

#### Single Tab Viewing

To view only a single *<Entity\_Detail>* tab all the time, choose *Window – Single Detail Window*. This ensures that apart from search, you view only one detail tab for each type of entity.

#### **Supported Operators**

Database Administrator allows you to specify complex search expressions using a variety of relational, logical, and special operators. The search expression supports the following operators.

Operator	Usage
< (Less than)	The value to search must be less than the specified value.
> (Greater than)	The value to search must be greater than the specified value.
!= (Not Equal to)	The value to search must not match the specified value.

## Allegro EDM Database Administrator User Guide Searching Database Entities

Operator	Usage
== (Equal to)	The value to search must match to the specified value.
<= (Less than or equal to)	The value to search must be less than or equal to the specified value.
>= (Greater than or equal to)	The value to search must be greater than or equal to the specified value.
~~ (Case insensitive string match)	The pattern of the first value must match the pattern of the second value. The value can be included anywhere in the string. With this operator, character case is ignored so that <code>library</code> is considered a match for <code>LIB*</code> .
!~~ (Case insensitive string not match)	The pattern of the first value must not match the pattern of the second value. The value can be included anywhere in the string.
	With this operator, character case is ignored. For example, a first value of Part Name and a second value of pA* $nA*$ would result in a FALSE comparison since the two are considered a match regardless of the difference in uppercase and lowercase characters.
~= (Case sensitive string match)	The pattern of the first value must match the pattern of the second value. The value can be included anywhere in the string. This includes testing for uppercase and lowercase characters. For example, Part Number is not a sensitive match for the pattern value pa* nu" because the uppercase values (P and N) will not match the lowercase specification pattern.
!~= (Case sensitive string not match)	The pattern of the first value must not match the pattern of the second value. The value can be included anywhere in the string. For example, if the first value is $Part*$ a second value of $part$ would produce a true result because the lowercase $p$ is not an exact match to the first value uppercase $P$ .
AND	The first value and the second value must be present.
(Logical AND)	
OR	Specifies that either of the values be present.
(Logical OR)	

Searching Database Entities

Operator	Usage
*	Specifies any number of characters match.
(Any String of Characters)	
?	Specifies exactly one character match.
(Any Single Character)	

- By default, in case you do not provide any operator, the ~~ (Case insensitive string match) operator is used.
- If you specify underscore (\_) as a part of search string, then it is treated as the ? character (single character).