EDAConnect

LibSync User’s Guide

Version 3.5.0

Oracle Part Number: E73441_01

Perception Software

EDAConnect LibSync Version: 3.5.0
March, 2016

Contents

Preface

EDAConnect LibSync User's Guide Overview

This manual provides instructions for using EDAConnect LibSync and is organized into the following chapters:

- **Chapter 1**: EDAConnect LibSync Overview — Provides an overview of EDAConnect LibSync operation and the system architecture of a typical EDAConnect LibSync environment.
- **Chapter 2**: User Interface Overview — Describes the EDAConnect LibSync user interface.
- **Chapter 3**: Setting Up Library Synchronization — Describes how to configure library synchronization.
- **Chapter 4**: Managing PLM to ECAD Mappings — Provides instructions for creating mappings between PLM and ECAD data objects.
- **Chapter 5**: Performing Database Synchronization — Describes how to perform ECAD library synchronization.
- **Chapter 6**: Performing PTF File Synchronization — Describes how to PTF file synchronization.
- **Chapter 7**: Problem Reporting — Describes how to report a LibSync problem and obtain technical support.

Font Conventions

In this guide, the following font conventions are used:

- Links, buttons, menus, and icons that are clicked appear in **Bold** Face Type**.**
- Items to select in drop-down menus or navigation trees are **bold** with "→" between entries.
- Dialog box "field names" are in quotes.
- Window and field names are in Initial Capitals.
- Cross references to sections and other chapters appear as underlined text.
- Information to be provided or entered by the user appear in <*italicized text*> between angle brackets.
- References to other Perception Software documentation appear in *italicized text*

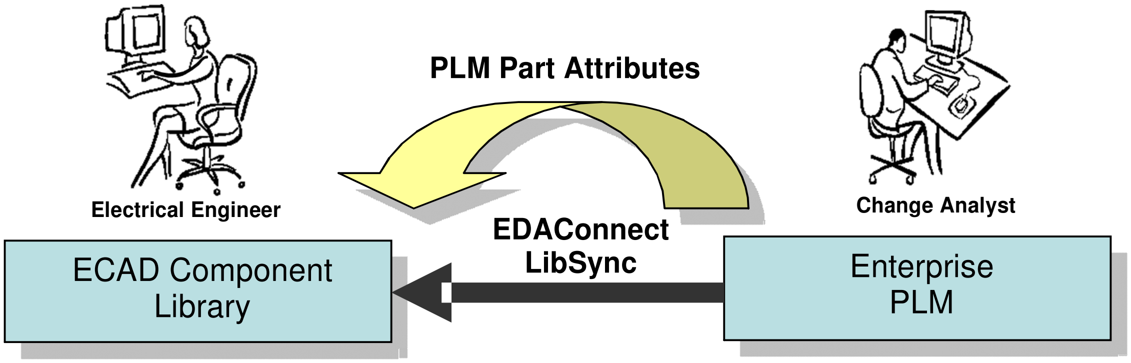
EDAConnect LibSync Overview

Introduction

EDAConnect LibSync is an application that insures that critical enterprise data relevant to the ECAD component library is kept up to date. While the enterprise data is mastered in the PLM system, it is not readily available to ECAD tools and, more importantly, to the engineer at the time of design. LibSync resolves that problem by providing library synchronization between the PLM system and the ECAD component library.

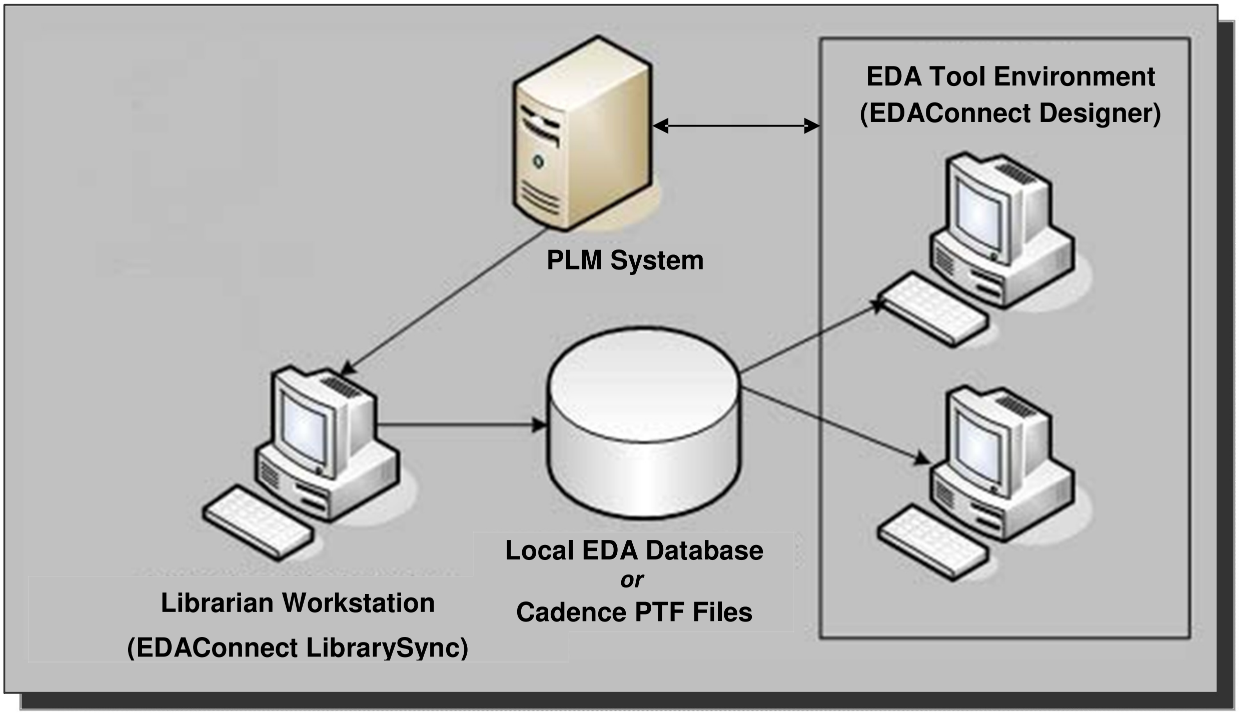
Libsync is designed to be utilized by the ECAD librarian to:

- Specify the parts to synchronize
- Specify the PLM attributes to include
- Map the PLM attributes to ECAD Attributes
- Specify a synchronization schedule.



Installation Architecture

The following diagram shows the system architecture for a typical EDAConnect installation. Each component of this architecture is described below.



PLM System

EDAConnect Designer is a client application to the PLM infrastructure. Designer and LibSync, a companion product to Designer, communicate with the PLM system through the PLM system's API. Designer uses EDA-specific drivers to extract BOM information from ECAD projects and allows the user to compare that information against a previously existing BOM and to publish new BOMs and Engineering Change Orders to the PLM system. LibSync reads part attribute data such as lifecycle phase, lead time, and RoHS compliance and stores that information in the Local EDA Database which can then be read both EDA tools.

Local EDA Database or Cadence PTF Files

Attributes extracted from Agile PLM may be store in one of two place:

- An ODBC Compliant Database
- Cadence PTF Files

If a local EDA Database is used, EDAConnect LibSync manages the table structure in the database to be consistent with what ECAD tools require. The connection to the local database is made via the ODBC standard. The database is not required to be physically local to the EDA environment, but generally must be on a high performance LAN.

If Cadence PTF files are used, then EDAConnect LibSync scans the design library for PTF files and updates the appropriate entries.

Librarian Workstation

The librarian workstation is the machine utilized by the librarian. The librarian workstation is required to have an EDAConnect LibSync application installed for both configuring and executing the automated library synchronization. This application allows the librarian to configure the list of items and attributes that are written to the Local EDA Database. EDAConnect LibSync communicates with the PLM infrastructure through the PLM system's API.

EDA Tool Environment

EDAConnect Designer is installed on the workstations in the EDA tool environment which consists of the EDA tool workstations and associated software. Designer uses EDA-specific drivers to extract BOM information and design files from ECAD projects. The user may then use Designer to check-in design files to the PLM system, compare BOM information against a previously existing BOM in the PLM system, and to publish new BOMs and Engineering Change Orders as required.

LibSync reads part attribute data such as lifecycle phase, lead time, and RoHS compliance and stores that information in the local EDA database. A subset of EDA tools (generally, schematic capture) can then access to the local EDA database to facilitate part selection.

How LibSync Works

EDAConnect LibSync makes critical enterprise data available to the engineering environment in one of two ways:

- By maintaining a Local EDA Database to which the ECAD tools can connect
- By updating the PTF files used by Concept HDL

LibSync operation is governed by a configuration or mapping file which is created by the EDA librarian using the LibSync application. The mapping file stores all the needed information to perform the synchronization:

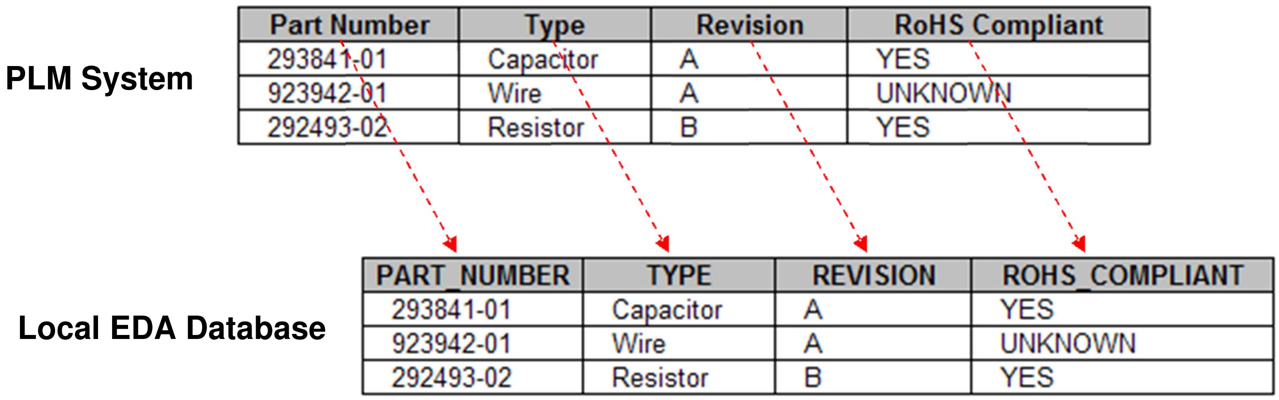
- Source PLM system
- PLM logon credentials
- Target Local EDA Database or PTF Update Template
- Synchronization schedule
- Selected PLM Subclasses
- Mapping of PLM attributes to ECAD attributes for each subclass

The librarian provides the mapping file information for each of these items through LibSync. The file should never be edited directly since LibSync provides an intuitive interface for managing the configuration. The mapping file has the extension ".psm" for PLM System Mapping and is in XML format.

Local EDA Database Mode

LibSync synchronizes the library by updating the Local EDA Database. LibSync executes the following operations for each item in a mapped subclass:

- Create a new table in the Local EDA Database for each Part Type
- Create a new column for each unique attribute name
- Query PLM for the attribute name and value
- Create a new row for each part number
- Place the attribute values into the corresponding column



PLM System

Part Number	Type	Revision	RoHS Compliant
293841-01	Capacitor	A	YES
923942-01	Wire	A	UNKNOWN
292493-02	Resistor	B	YES

Local EDA Database

PART_NUMBER	TYPE	REVISION	ROHS_COMPLIANT
293841-01	Capacitor	A	YES
923942-01	Wire	A	UNKNOWN
292493-02	Resistor	B	YES

Cadence PTF File Mode

For Cadence Concept HDL PTF files synchronization LibSync executes the following operations for each item in a mapped subclass:

- Use a PTF Update Template to locate the PTF files to update
    - Determine whether a property within the PTF is allowed to be updated
- For each Part Type synchronized:
    - Create an in memory table of the Part Type data
        - Create a new column for each unique attribute name
        - Query PLM for the attribute name and value
        - Create a new row for each part number
        - Place the attribute values into the corresponding column
    - Create a backup copy of the PTF file
    - Scan the PTF for updatable properties and perform the update accordingly.
    - Log the PTF transactions in a master log file.
    - Log parts found in PLM but not referenced in the PTF files in an Extra Parts file.

LibSync Synchronization Caveats

EDAConnect LibSync technology is unidirectional from the PLM system to the Local EDA Database or Concept HDL PTF files. LibSync is assumed to be the master of the Local EDA Database at all times and the master of the PTF files during synchronization.

> **WARNING**: Any changes to the Local EDA Database made outside of the LibSync application will not be preserved, or reported to the user.
>
> **WARNING**: The use of international character sets is not fully supported. Use of non-English characters in PLM class names may cause synchronization to fail.

LibSync periodically refreshes the Local EDA Database or local PTF files. The synchronization frequency is determined by the schedule defined in the configuration file. As long as the LibSync application is running on the librarian's workstation, the Local EDA Database will be kept current with the data in the PLM system based on the synchronization schedule.

> **NOTE:** LibSync requires that the librarian's workstation (or other designated machine) and the PLM system be available during the time of synchronization. Additionally, ECAD applications should not be utilizing the library or PTF files during the synchronization time.
>
> **NOTE**: LibSync may be installed as a Windows service so that schedule synchronization resumes automatically following a system reboot. See "*Installing LibSync as a Windows Service*" in the *EDAConnect LibSync Installation Guide.*

Once the Local EDA Database or PTF file set is refreshed with the current PLM data, the ECAD tools may use the database or PTF files for part selection and schematic part verification.

User Interface Overview

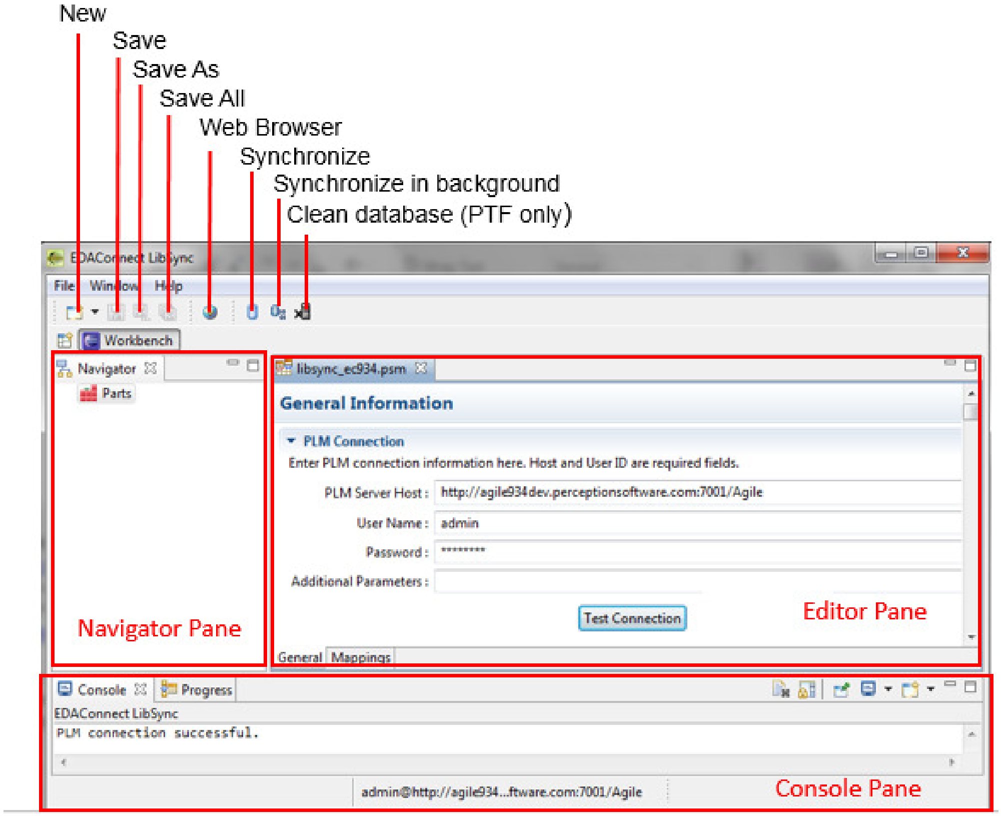
EDAConnect LibSync provides a GUI interface for editing mapping files and performing local EDA Database synchronizations. The EDAConnect LibSync interface has been designed to be intuitive and easy to use. This chapter describes the main features of the interface and provides a foundation for information in the subsequent chapters of this manual.

A command line interface is provided for Cadence PTF synchronizations and is described in the chapter on Performing PTF File Synchronization on page 67.

LibSync Workbench

EDAConnect LibSync uses an Eclipse-based user interface that is divided into three panes or views:

- **Navigator Pane** — Shows PLM Part/Type classes
- **Editor Pane** — Provides access to context specific data editors
- **Console Pane** — Displays Error and Status Messages from all Operations



General Information

PLM Connection

Enter PLM connection information here. Host and User ID are required fields.

PLM Server Host : http://agile934dev.perceptionsoftware.com:7001/Agile

User Name : admin

Password : ********

Additional Parameters :

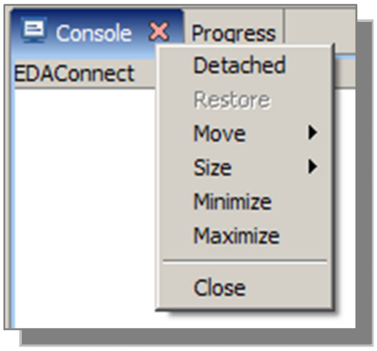
```
EDAConnect LibSync
PLM connection successful.
```

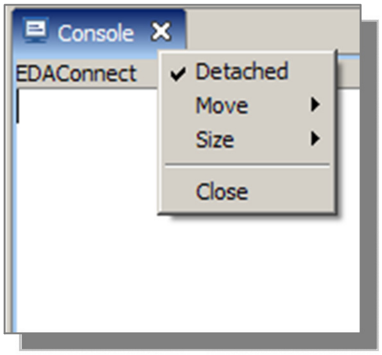
admin@http://agile934...ftware.com:7001/Agile

NOTE: The **Synchronize**, **Synchronize in background**, and **Clean database** buttons are only displayed when a PSM mapping file is open. **Synchronize** and **Synchronize in background** initiate ODBC synchronizations. Cadence PTF file synchronizations are initiated through LibSync's CLI (see PTF Command Line Interface on page 81). The **Clean database** button is used to cleanse the file system of all but the most recent copy of the PTF backup files created when a PTF synchronization is performed.

NOTE: The Eclipse GUI framework provides a number of features not all of which are required by EDAConnect LibSync. Unused Eclipse features are noted in appropriate section of this manual.

Each pane can be dragged-and-dropped to move it to a more convenient location in the EDAConnect Workbench. A pane can also be placed into separate window by right-clicking on a tab and selecting the **Detached** option. To reattach a window to the LibSync Workbench, right-click the tab again and click on the check-marked **Detached** option to toggle the setting.



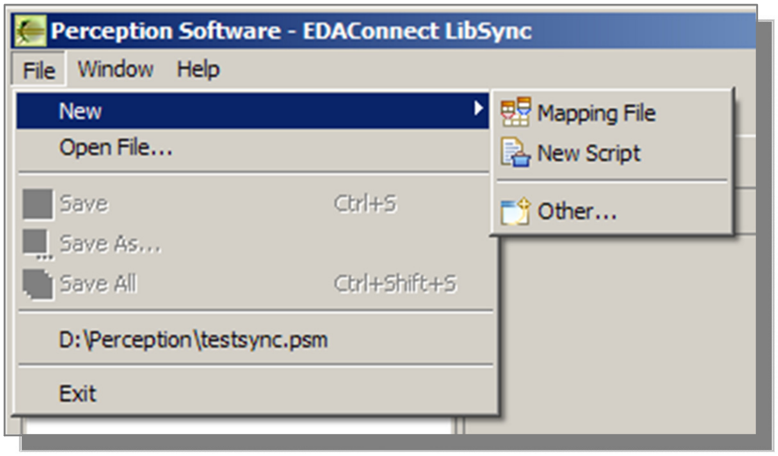


The buttons on the Workbench toolbar provide shortcuts to the File Menu options, the Event Registry, and immediate and background synchronization.

LibSync Menus

File Menu

The **File** menu allows you to create new Mapping Files or Scripts and to open and save existing Mapping Files and Scripts.



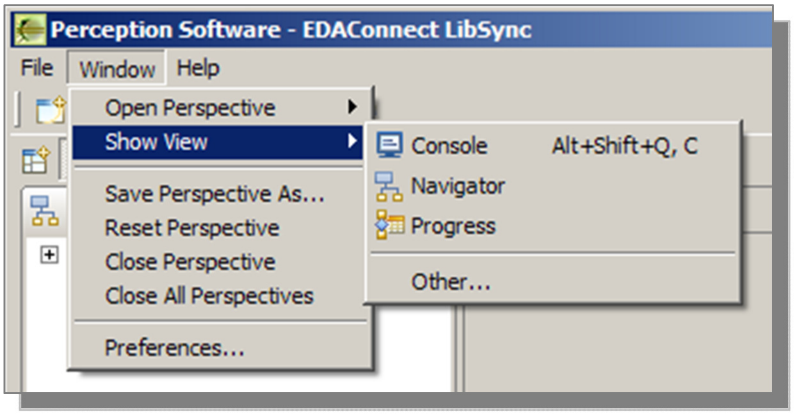
Window Menu

The **Window** menu allows you to select views or set your preferences.

Show View offers the following options:

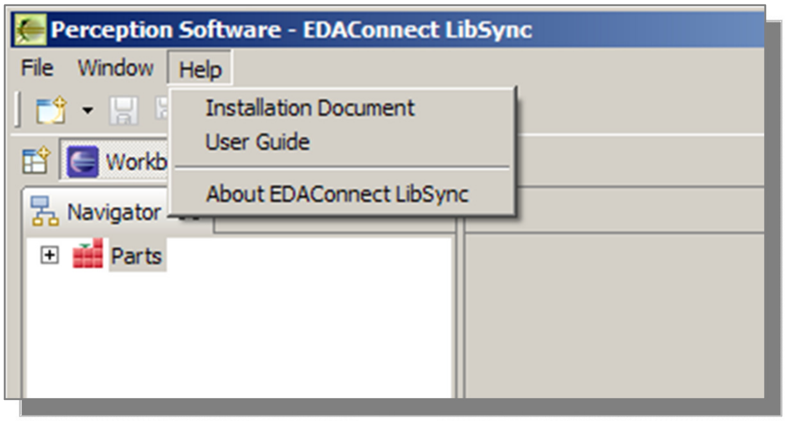
- **Console**: Contains all error and status messages.
- **Navigator**: Shows tree view of projects and templates in Agile.
- **Progress**: Shows progress of any operations running.

Setting preferences is covered in the Setting User Preferences section on page 21.



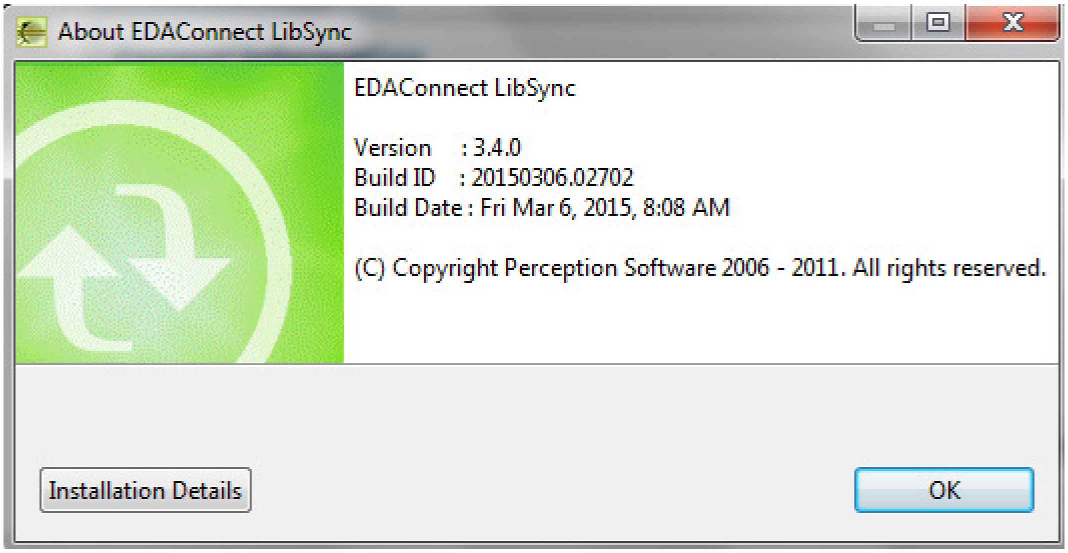
Help Menu

The **Help** menu provides access to user documentation as well as the **Help→About EDAConnect LibSync** screen.



Help→About EDAConnect LibSync

The **Help→About EDAConnect LibSync** screen provides important information regarding product and build number.



EDAConnect LibSync

Version : 3.4.0
Build ID : 20150306.02702
Build Date : Fri Mar 6, 2015, 8:08 AM

(C) Copyright Perception Software 2006 - 2011. All rights reserved.

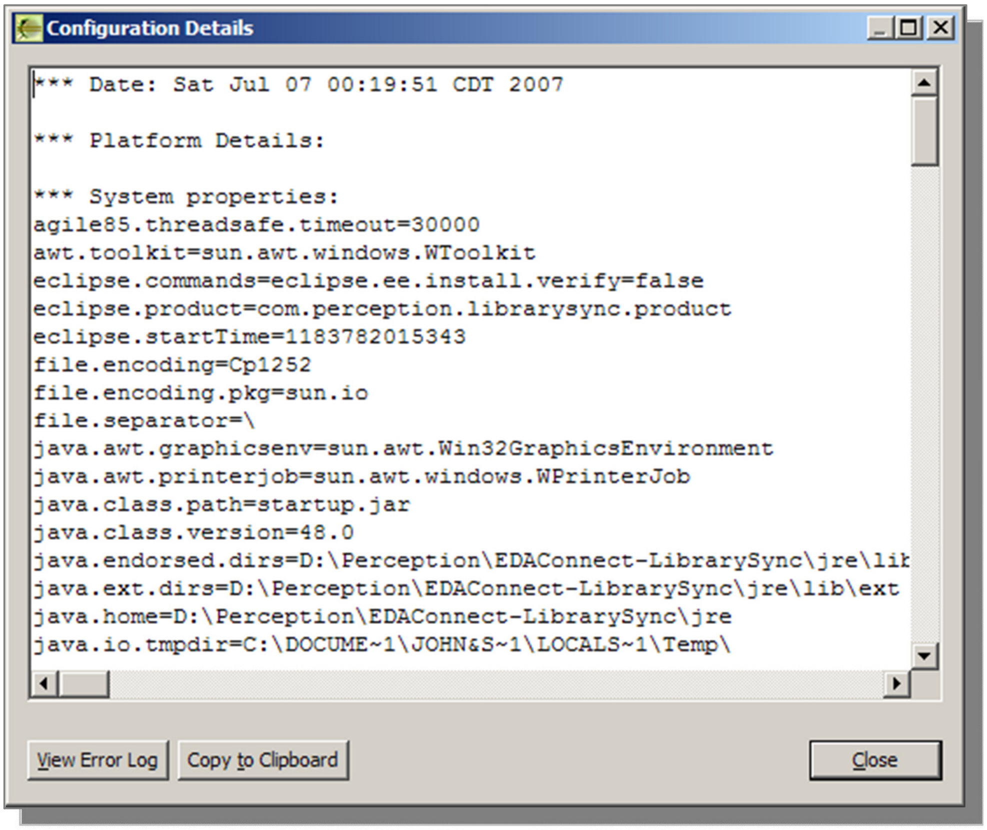
From this dialog box you can access the **Plug-in Details** and **Configuration Details** screens.

Plug-in Details Screen

The **Plug-in Details** screen lists by vendor, the name and version of each plug-in installed including your PLM and ECAD drivers. This information is helpful should you ever need assistance from Technical Support.

Configuration Details Screen

The Configuration Details screen shows detailed information regarding computer settings including environment variables, Java settings, and so on.



```
*** Date: Sat Jul 07 00:19:51 CDT 2007

*** Platform Details:

*** System properties:
agile85.threadsafe.timeout=30000
awt.toolkit=sun.awt.windows.WToolkit
eclipse.commands=eclipse.ee.install.verify=false
eclipse.product=com.perception.librarysync.product
eclipse.startTime=1183782015343
file.encoding=Cp1252
file.encoding.pkg=sun.io
file.separator=\
java.awt.graphicsenv=sun.awt.Win32GraphicsEnvironment
java.awt.printerjob=sun.awt.windows.WPrinterJob
java.class.path=startup.jar
java.class.version=48.0
java.endorsed.dirs=D:\Perception\EDAConnect-LibrarySync\jre\lib
java.ext.dirs=D:\Perception\EDAConnect-LibrarySync\jre\lib\ext
java.home=D:\Perception\EDAConnect-LibrarySync\jre
java.io.tmpdir=C:\DOCUME~1\JOHN&S~1\LOCALS~1\Temp\
```

Eclipse Error Log

The **View Error Log** button on the Configuration Details dialog provides access to the Eclipse Error Log. Typically, this information is not required when reporting a problem and can be ignored.

Perception Log

LibSync maintains an information and error log that records transaction information and exceptions. The log file is located at:

<install_path>\EDAConnect-Designer\logs\perception.log

NOTE: The **perception.log** logging level for Designer scripts can be set via the **scripts.logging.level** property in the **config.ini** file. Please see the *EDAConnect LibSync Installation Guide* for more information.

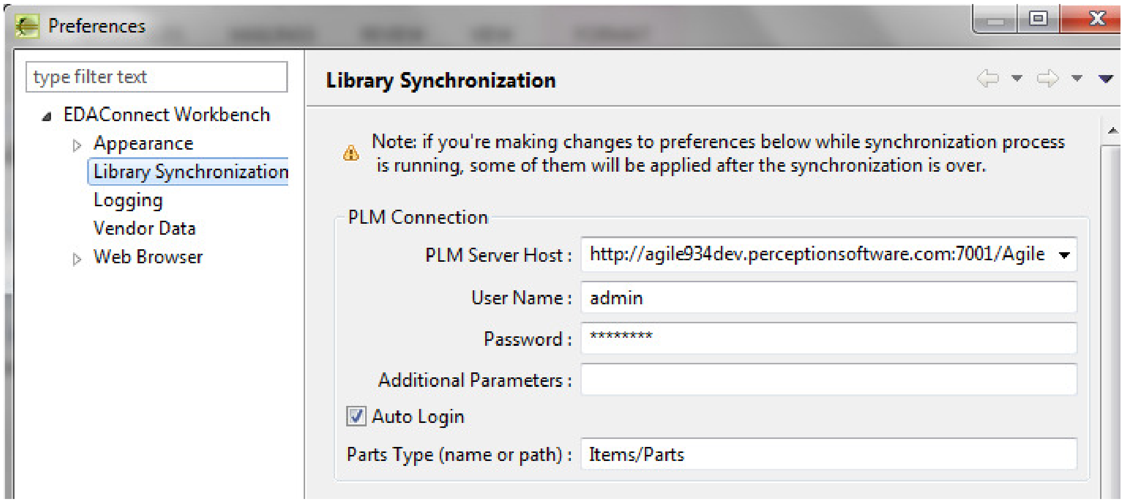
Setting User Preferences

The **Preferences** dialog box provides access to a number of user preference settings to control attributes for Library Synchronization and Workbench general appearance.

LibSync Preferences

► To set the PLM Connection Preferences and Log File

1. Select **Window→Preferences** from the Window menu on LibSync's taskbar.
2. Select **Library Synchronization** from the Preferences navigation tree. The Library Synchronization dialog box is displayed.



Note: if you're making changes to preferences below while synchronization process is running, some of them will be applied after the synchronization is over.

PLM Server Host : http://agile934dev.perceptionsoftware.com:7001/Agile

User Name : admin

Password : ********

Additional Parameters :

☑ Auto Login

Parts Type (name or path) : Items/Parts

3. Enter PLM Connection information:

   A. Choose the desired PLM server URL from the “PLM Server Host” drop-down menu. If the desired server URL is not included in the list, enter the URL manually.

      NOTE: The PLM system administrator will need to supply the exact URL, but it is typically of the form: http://plmserver.domain/VirtualDirectory.

      NOTE: If the URL uses HTTPS protocol, it will be necessary to add the PLM server’s CA certificate key to Designer’s JRE trusted keystore in order to access the PLM server. Please see “*To set PLM connection preferences for LibSync*” in the LibSync Installation Guide.

   B. Enter the appropriate “User Name” and “Password” for the PLM server.

      NOTE: The password is encrypted for security.

   C. If your PLM system requires additional parameters such as port number assignments, enter the information in the “Additional Parameters” field.

   D. Check the **Auto Login** checkbox to enable auto-login.
      Uncheck the **Auto Login** checkbox to disable auto-login.

4. Enter Miscellaneous information:

   A. Enter the Items/Parts type in the “Parts Type” field.

   B. Enter the “Default Log File Name”.

   C. Select either the option to **Append** or **Overwrite** the log file for each LibSync session.

5. Enter Email Notification information:

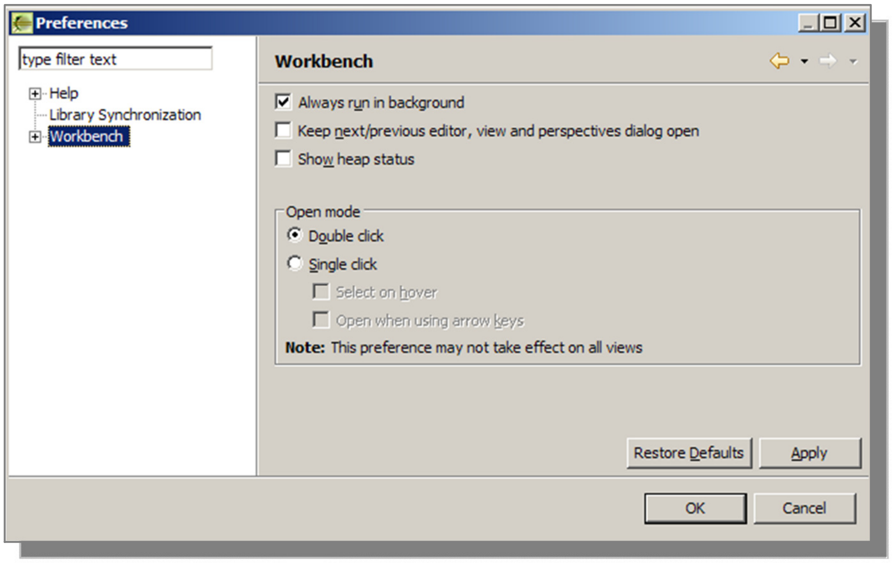
   A. Select **On Error** to send email notification only if there is a synchronization error. Select **Always Send** to send email notification for each synchronization whether errors occur or not.

   B. Enter any email body text desired.

Workbench Preferences

Use the **Preferences: Workbench** screen to configure workbench characteristics.

NOTE: Select “Always run in background” to display progress status in LibSync’s Status Bar rather than in pop-up dialog boxes.



Workbench

☑ Always run in background

□ Keep next/previous editor, view and perspectives dialog open

□ Show heap status

Open mode

● Double click

○ Single click

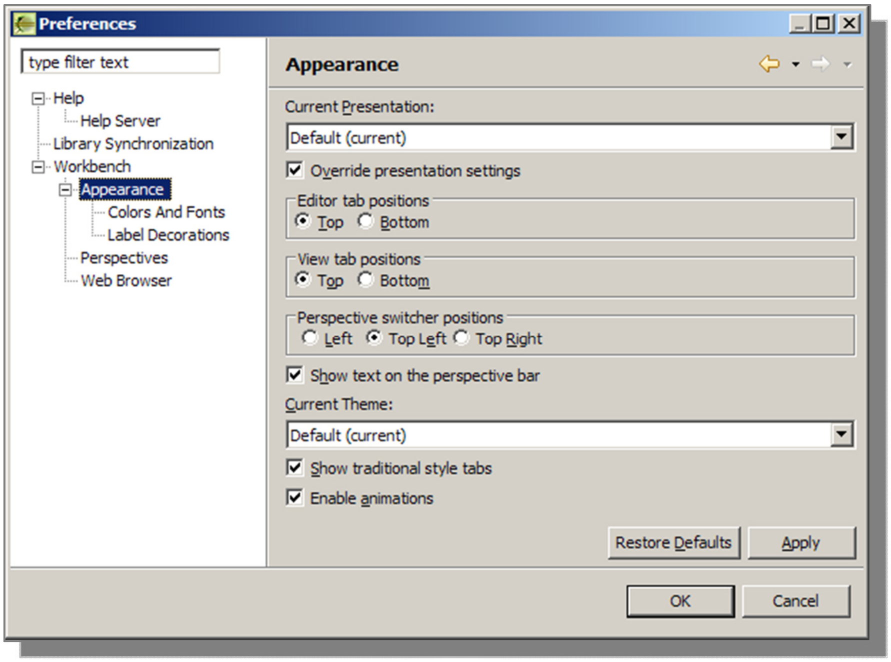
□ Select on hover

□ Open when using arrow keys

Note: This preference may not take effect on all views

Appearance Preferences

From the **Preferences: Workbench→Appearance** screen you can configure various presentation attributes of Workbench such as tab placements, colors and fonts, and label decorations.



Appearance

Current Presentation: Default (current)

☑ Override presentation settings

Editor tab positions: ● Top ○ Bottom

View tab positions: ● Top ○ Bottom

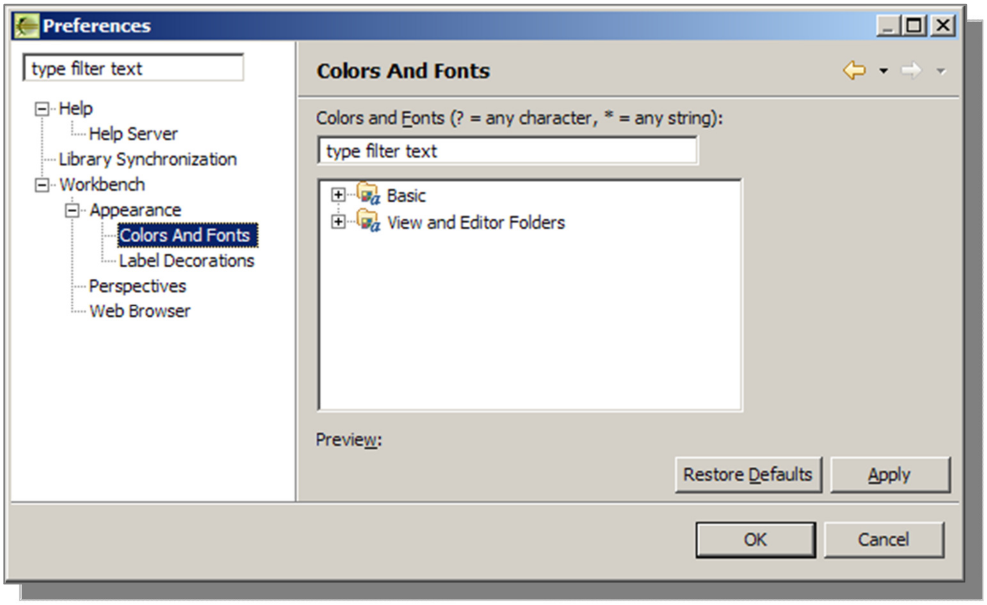
Perspective switcher positions: ○ Left ● Top Left ○ Top Right

☑ Show text on the perspective bar

Current Theme: Default (current)

☑ Show traditional style tabs

☑ Enable animations

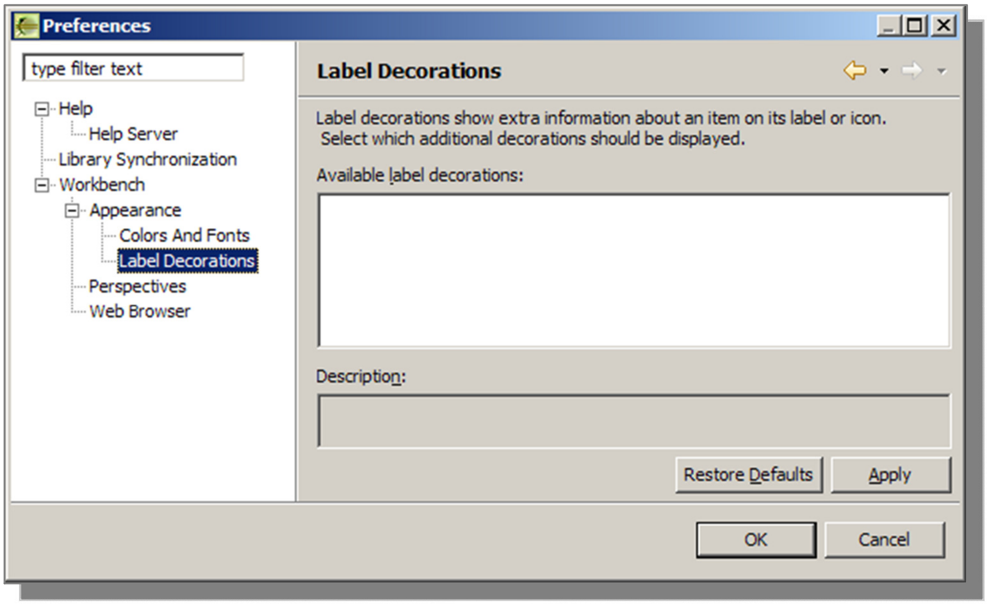


Colors And Fonts

Colors and Fonts (? = any character, * = any string): type filter text

- Basic
- View and Editor Folders

Preview:



Label Decorations

Label decorations show extra information about an item on its label or icon. Select which additional decorations should be displayed.

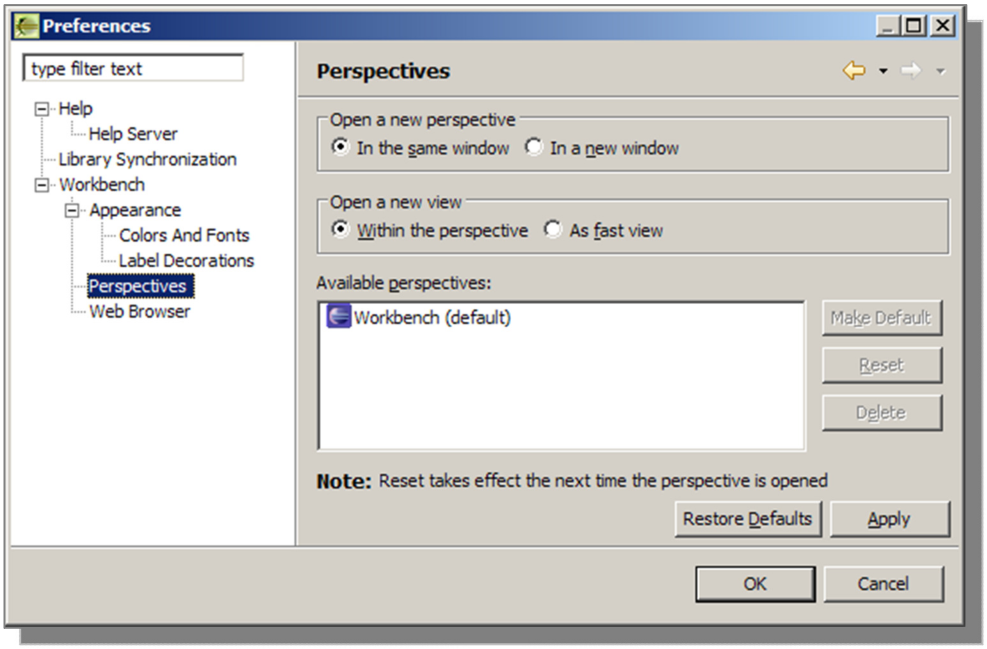
Available label decorations:

Description:

Perspective Preferences

Use the **Preferences: Workbench→Perspectives** screen to configure Perspective characteristics.

> **WARNING**: LibSync uses only the Workbench perspective and is pre-configured with the appropriate settings. These settings should not be modified by the user.



Perspectives

Open a new perspective
● In the same window ○ In a new window

Open a new view
● Within the perspective ○ As fast view

Available perspectives:
Workbench (default)

Note: Reset takes effect the next time the perspective is opened

Web Browser Preferences

There Web Browser Preferences option is an artifact of the Eclipse framework. LibSync does not utilize a web browser to display information, and hence, does not use the Web Brower Preferences settings.

Setting Up Library Synchronization

This chapter describes the basic steps for setting up and maintaining library synchronization.

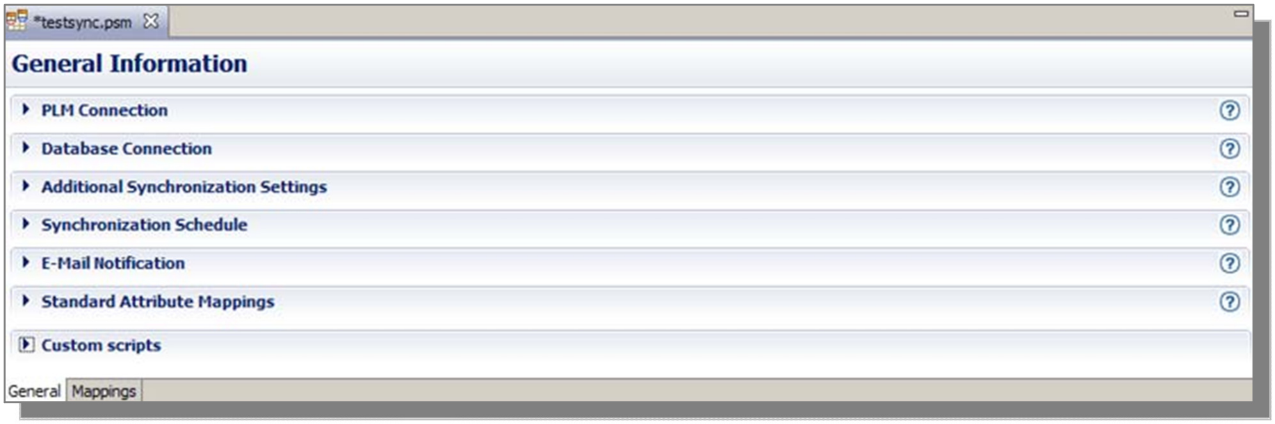
Library synchronization is controlled by a PLM System Mapping (PSM) file which has a file extension of ".psm". The mapping file specifies the following:

- PLM connection parameters
- Database connection parameters
- Additional synchronization settings
- Synchronization schedule
- Email notification
- Standard (default) attribute mappings
- Custom attribute mappings
- Custom scripts

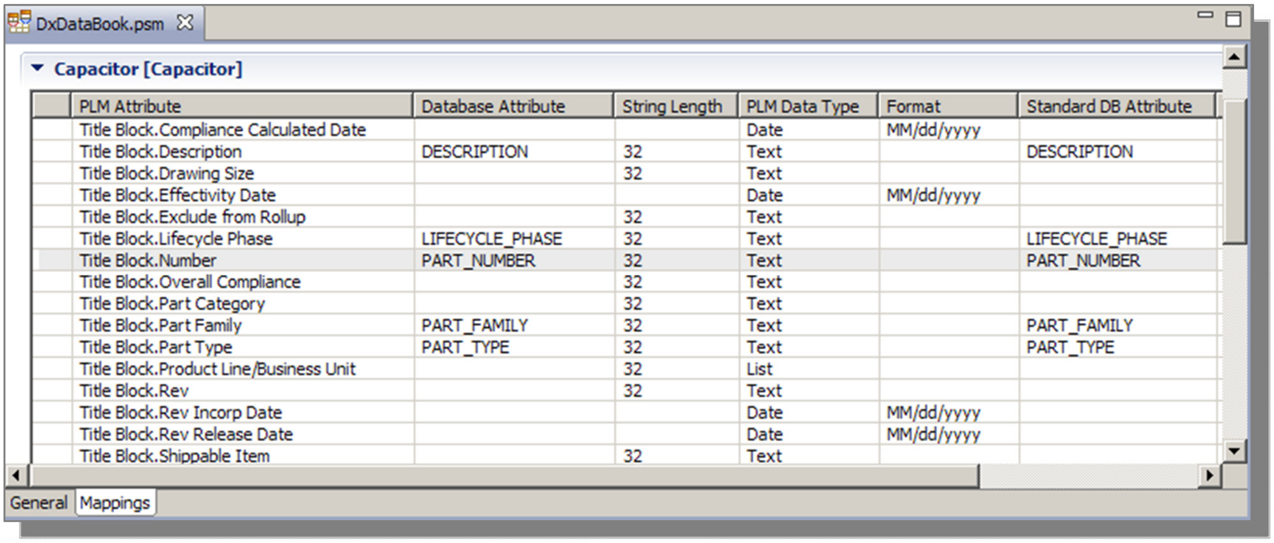
A mapping file is created and maintained through the LibSync mapping file editor. The editor displays the mapping file content in two tabs in the Editor pane:

- General Information
- Mappings

The **General Information** tab allows you to specify all parameters other than Part Type specific attribute mappings:



The **Mappings** tab allows you to specify any Part Type specific attribute mappings. A separate table on the **Mappings** tab is used for each Part Type.



PLM Attribute	Database Attribute	String Length	PLM Data Type	Format	Standard DB Attribute
Title Block.Compliance Calculated Date			Date	MM/dd/yyyy	
Title Block.Description	DESCRIPTION	32	Text		DESCRIPTION
Title Block.Drawing Size		32	Text		
Title Block.Effectivity Date			Date	MM/dd/yyyy	
Title Block.Exclude from Rollup		32	Text		
Title Block.Lifecycle Phase	LIFECYCLE_PHASE	32	Text		LIFECYCLE_PHASE
Title Block.Number	PART_NUMBER	32	Text		PART_NUMBER
Title Block.Overall Compliance		32	Text		
Title Block.Part Category		32	Text		
Title Block.Part Family	PART_FAMILY	32	Text		PART_FAMILY
Title Block.Part Type	PART_TYPE	32	Text		PART_TYPE
Title Block.Product Line/Business Unit		32	List		
Title Block.Rev		32	Text		
Title Block.Rev Incorp Date			Date	MM/dd/yyyy	
Title Block.Rev Release Date			Date	MM/dd/yyyy	
Title Block.Shippable Item		32	Text		

You may create as many mapping files as desired.

First Time Setup

When configuring LibSync for the very first time, the following procedure should be used:

► To set up library synchronization for the very first time

1. Configure the Local EDA Database and obtain the ODBC connector information.
2. Launch EDAConnect LibSync.
3. Configure LibSync preferences for Library Synchronization.
4. Create a mapping file.
   A. Specify the PLM Connection.
   B. Specify the DB Connection.
   C. Specify the Synchronization Schedule.
   D. Specify Email Notification.
   E. Define Part Type independent attribute mappings.
   F. Define Part Type specific attribute mappings.

G. Save the mapping file.

NOTE: Library synchronization cannot occur unless the PSM file has been saved.

5. Validate the mapping file.

   A. Execute an Immediate or Background synchronization.

   B. Verify the results.

   C. Modify the mapping file as required and re-validate if necessary.

Once library synchronization is up and running, you'll want to periodically review LibSync's operation to keep synchronization running smoothly.

► **To maintain a library synchronization configuration**

1. Periodically review log files for errors.
2. Periodically check the PLM system for new or removed part subclasses.
3. Edit the PSM mapping file to be consistent with PLM.

The steps for creating a PSM mapping file are described in greater detail in the subsequent sections.

Configuring the Local EDA Database

LibSync requires access to the Local EDA Database via ODBC. Please see Configuring an ODBC Database in the Appendix section of the *EDAConnect Installation Guide* for information on how to configure the type of database being used. Additionally, the following consideration should be kept in mind when configuring the Local EDA Database.

Table Management

- LibSync requires complete control over the tables it manages.
- LibSync completely drops and recreates the tables for each synchronization.
- If transactions from other processes are open on these tables, then the synchronization may fail with a cryptic error message.
- It is highly recommended that the tables in the Local EDA Database not be modified by any application other than LibSync.

Database Username

- It is highly recommended that a new database username be created and assigned exclusively for use with LibSync.
- The database must be configured such that the username has full permissions to create, drop, query, and add items in the database.

NOTE: Username is not supported by Microsoft Access.

Performance Considerations

Both LibSync and the Local EDA Database instance should reside as close as possible to the PLM server. Best performance is achieved if all three applications are running on the same system.

SQL Reserved Words

SQL Reserved Words cannot be used for table names or column names. See Appendix B – SQL Reserved Words on page 97 for a list of SQL Reserved Words and also check the documentation for your database application since each application has its own documented set of reserved words.

International Character Sets

The use of international character sets is not fully supported. For example, non-English characters in PLM class names results in local database table names containing non-English characters which may cause synchronization to fail.

Launching LibSync

If Start menu entries were installed, launch EDAConnect LibSync from the Start menu:

Start→Programs→Perception Software→EDAConnect-LibSync→LibSync

Otherwise, double-click on the LibSync batch file:

<install_path>\EDAConnect-LibSync\EDAConnect-LibrarySync.bat

where *<install_path>* is typically c:\Perception Software.

> **NOTE**: Running multiple instances of LibSync is not recommended. Care should be taken to insure that only a single instance of LibSync is active.

Configuring Library Synchronization Preferences

If this is a first time set up for LibSync, then Library Synchronization preferences must be set up as described in LibSync Preferences on page 22. It is important to specify PLM default login credentials as well as the PLM Parts Type.

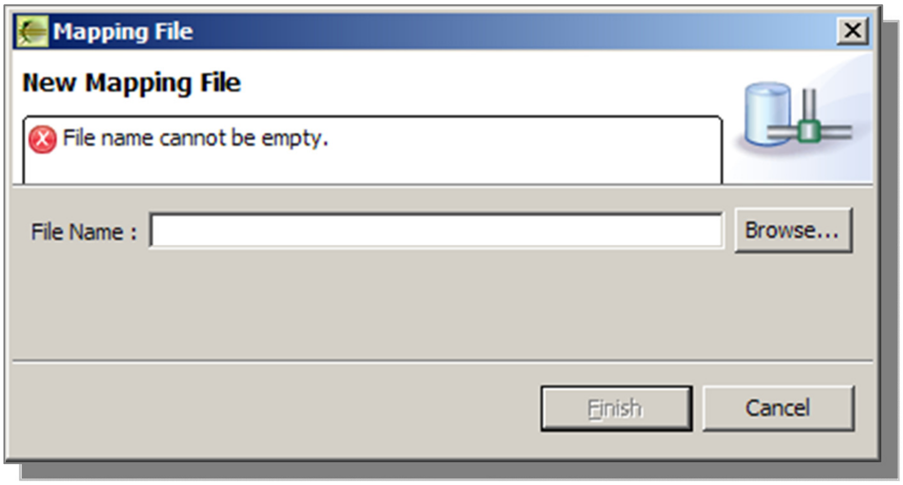
Creating a Mapping File

The PLM System Mapping file holds all of the information to perform a library synchronization. The file is stored locally on the librarian's workstation. Use the following procedure to create a new mapping file.

► **To create a new Mapping File**

1. Select **File→New→Mapping File**.

   The New Mapping File dialog box is displayed.



Mapping File

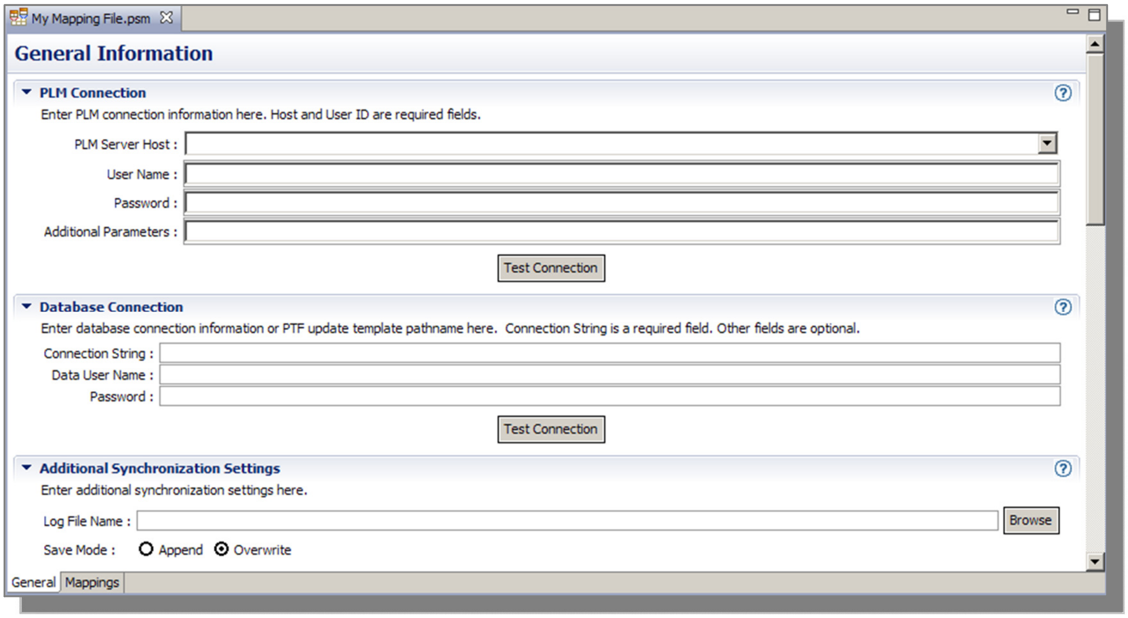
New Mapping File

File name cannot be empty.

2. Specify a full pathname to where you wish to store the Mapping file or use the **Browse…** button to select the location.

   > **NOTE**: LibSync will automatically add the ".psm" extension to the filename if it isn't specified.

   LibSync opens the Mapping File Editor to the General Information tab with all entry groups expanded:



My Mapping File.psm

General Information

PLM Connection

Enter PLM connection information here. Host and User ID are required fields.

PLM Server Host :

User Name :

Password :

Additional Parameters :

Test Connection

Database Connection

Enter database connection information or PTF update template pathname here. Connection String is a required field. Other fields are optional.

Connection String :

Data User Name :

Password :

Test Connection

Additional Synchronization Settings

Enter additional synchronization settings here.

Log File Name :

Browse

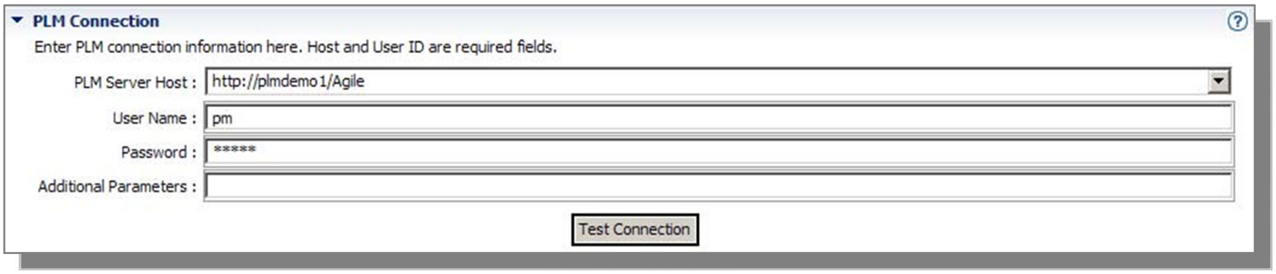
Save Mode : ○ Append ● Overwrite

General | Mappings

You are now ready to configure the Mapping file.

PLM Connection

The PLM Connection group box contains the information that will be used to login to the PLM system when library synchronization occurs.



PLM Connection

Enter PLM connection information here. Host and User ID are required fields.

PLM Server Host : http://plmdemo1/Agile

User Name : pm

Password : *****

Additional Parameters :

Test Connection

► To specify the PLM Connection

1. Expand the PLM Connection group box on the **General** tab, if not already expanded.
2. Choose the desired PLM server URL from the “PLM Server Host” drop-down menu. If the desired server URL is not included in the list, enter the URL manually.

> **NOTE**: The PLM system administrator will need to supply the exact URL, but it is typically of the form: http://plmserver.domain/VirtualDirectory.
>
> **NOTE:** If the URL uses HTTPS protocol, it will be necessary to add the PLM server’s CA certificate key to Designer’s JRE trusted keystore in order to access the PLM server. Please see “*To set PLM connection preferences for LibSync*” in the LibSync

Installation Guide.

3. Enter the appropriate “User Name” and “Password” for the PLM server.

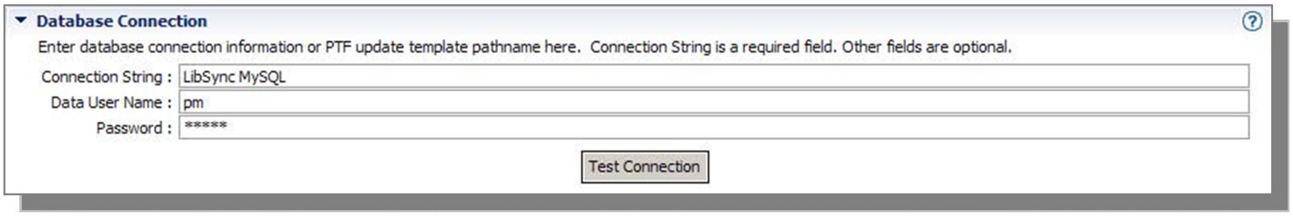
   NOTE: The “User Name” supplied must have sufficient privileges to read the component subclasses and attributes within the PLM system.

   NOTE: The password is encrypted for security.

4. If your PLM system requires additional parameters such as port number assignments, enter the information in the “Additional Parameters” field.
5. Click the **Test Connection** button to verify communication and authentication with the PLM server. If the test fails, modify the PLM Connection entries and try again.

Database Connection

The Database Connection group box contains the information that will be used to login to the Local EDA Database or to access the PTF Update Template when library synchronization occurs.



Database Connection

Enter database connection information or PTF update template pathname here. Connection String is a required field. Other fields are optional.

Connection String : LibSync MySQL

Data User Name : pm

Password : *****

► To specify the Database or PTF Update Template Connection

1. Expand the Database Connection group box on the **General** tab, if not already expanded.
2. Enter Data Source Name for the Local EDA Database in the “Connection String” field if you are synchronizing to a database. Enter the pathname to the PTF Update Template if you are synchronization to Concept HDL PTF files.

   NOTE: Please see Configuring an ODBC Database in the Appendix section of the *EDAConnect Installation Guide* for information on how to set up an ODBC connection.

3. Enter the appropriate “Data User Name” and “Password” for the Local EDA Database. If you are synchronizing PTF files, leave these fields blank.

NOTE: For Oracle, MySQL and SQL Server, this is the database Username/Password. For Microsoft Access, these fields may be left blank.

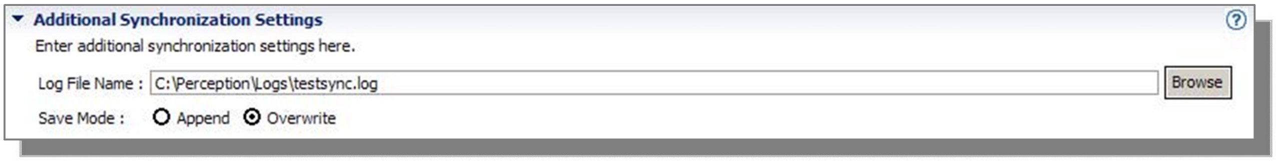
NOTE: The “Data User Name” supplied must have sufficient privileges to create, drop, query and add items to the local database.

NOTE: The password is encrypted for security.

4. Click the **Test Connection** button to verify communication and authentication with the Local EDA Database or to verify access to the PTF Update Template. If the test fails, modify the Database Connection entries or check the pathname for the PTF Update Template and try again.

Additional Synchronization Settings

Use the Additional Synchronization Settings group box to specify a mapping-file-specific log filename and to select whether to **Append** or **Overwrite** the log file for each LibSync session.



Additional Synchronization Settings

Enter additional synchronization settings here.

Log File Name : C:\Perception\Logs\testsync.log

Save Mode : ○ Append ● Overwrite

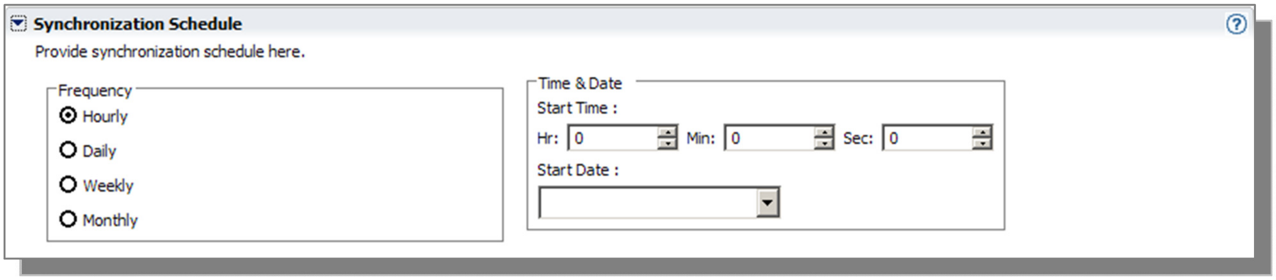
► To specify a mapping-file-specific log file

1. Expand the Additional Synchronization Settings group box on the **General** tab, if not already expanded.
2. Browse to the desired log file or enter the desired file pathname.
3. Select **Append** or **Overwrite** as the desired Save Mode.

Synchronization Schedule

LibSync performs library synchronization according to a specified schedule unless the user invokes an immediate synchronization. Typically, the immediate synchronization is only used for initial setup or for propagating a change to the ECAD library outside of a background synchronization schedule

The Synchronization Schedule group box contains the information for the library synchronization schedule.



Synchronization Schedule

Provide synchronization schedule here.

Frequency

● Hourly

○ Daily

○ Weekly

○ Monthly

Time & Date

Start Time :

Hr: 0 Min: 0 Sec: 0

Start Date :

► To specify a Synchronization Schedule

1. Expand the Synchronization Schedule group box on the **General** tab, if not

already expanded. 2. Select the Frequency of library synchronization.

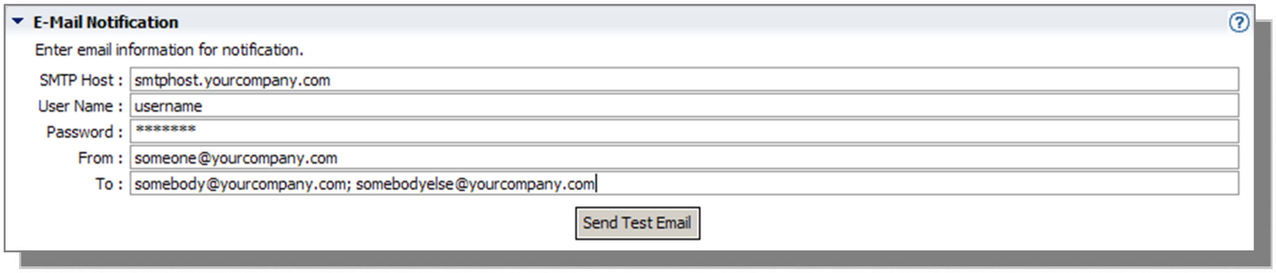
3. Specify the Start Time and Start Date for synchronization to begin.

> **NOTE**: It is best to schedule library synchronizations to occur during non-production hours for the engineers.

E-Mail Notification

LibSync can automatically email to a specified list of users the status of a completed library synchronization. The email contains the synchronization log file as an attachment.

The E-Mail Notification group box allows you to specify a list of email addresses, the SMTP host, and login information.



E-Mail Notification

Enter email information for notification.

SMTP Host : smtphost.yourcompany.com

User Name : username

Password : *******

From : someone@yourcompany.com

To : somebody@yourcompany.com; somebodyelse@yourcompany.com

► To specify Email Notification

1. Expand the E-Mail Notification group box on the **General** tab, if not already expanded.
2. Enter the name of the corporate SMTP server and its associated port in the "SMTP Host" field. You will need to obtain this information from your IT department.
3. Enter the SMTP "User Name" and "Password". Typically, this is the librarian's email address and password used to log into Microsoft Outlook or Outlook WebAccess. In some cases, it is simply your NT or UNIX login.
4. Enter the "From" and "To" email addresses in fully qualified internet

   email format: <name@address>

   Multiple entries should be separated by a semicolon.
5. Click on the **Send Test Email** button to verify the configuration of E-Mail Notifications.

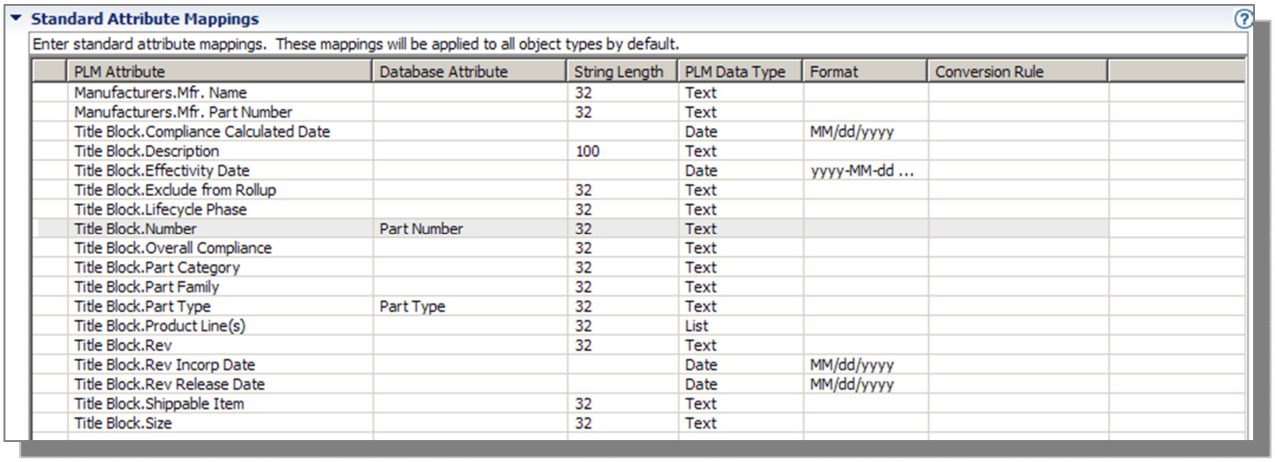
Standard Attribute Mappings

Many attributes are common across various PLM objects. For example, all parts typically have the following attributes:

- Part number
- Revisio
- Part family
- Part type
- Lifecycle phase

Rather than having to enter PLM to ECAD mappings for these attributes for each part type, a set of default mappings can be entered once in the Standard Attribute Mappings table on the **General** tab and then applied automatically by LibSync to Attribute Mapping tables on the **Mappings** tab.

The Standard Attribute Mappings table consists of six columns:



Standard Attribute Mappings

Enter standard attribute mappings. These mappings will be applied to all object types by default.

PLM Attribute	Database Attribute	String Length	PLM Data Type	Format	Conversion Rule
Manufacturers.Mfr. Name		32	Text		
Manufacturers.Mfr. Part Number		32	Text		
Title Block.Compliance Calculated Date			Date	MM/dd/yyyy	
Title Block.Description		100	Text		
Title Block.Effectivity Date			Date	yyyy-MM-dd ...	
Title Block.Exclude from Rollup		32	Text		
Title Block.Lifecycle Phase		32	Text		
Title Block.Number	Part Number	32	Text		
Title Block.Overall Compliance		32	Text		
Title Block.Part Category		32	Text		
Title Block.Part Family		32	Text		
Title Block.Part Type	Part Type	32	Text		
Title Block.Product Line(s)		32	List		
Title Block.Rev		32	Text		
Title Block.Rev Incorp Date			Date	MM/dd/yyyy	
Title Block.Rev Release Date			Date	MM/dd/yyyy	
Title Block.Shippable Item		32	Text		
Title Block.Size		32	Text		

Standard Attribute Mappings Table

Column	Description	Editable
PLM Attribute	Name of attribute in PLM. This column is populated automatically based on the part classes that have been added to the Custom Attributes Mapping table (see Working with Custom Attribute Mappings on page 48).	Y
Database Attribute	Specifies the name of attribute in the EDA database.	Y
String Length	Specifies the string column width in the EDA database. Default value is specified by the **string.column.length** property in **config.ini**. Defaults to 32 if not specified.	Y
PLM Data Type	Displays the PLM data type of the attribute.	N
Format	Specifies the format to use if conversion to string is required by the EDA database. Default formats are specified by the ***database.date.forma****t*, ***database.money.precision***, and ***database.numeric.precisio****n* properties **config.ini**. The money and numeric formats use an integer to specify the number of digits to the right of the decimal point or “-1” if the raw floating point number is to be displayed. The date format uses standard Java date notation. See **Configuring LibSync Properties** in the **LibSync Installation Guide** for more information.	Y

Conversion Rule	Specifies an optional Unit of Measure conversion rule to be applied to the attribute value during synchronization. Unit of Measure conversion is sometimes desired for Page Three attributes for discrete components such capacitors, inductors, resistances, etc. Please see Using Unit of Measure Conversion on page 52 for more information.	Y

You may add as many Database Attribute mappings and Format specifications as desired. The Standard Attribute Mappings are applied to all part classes on the **Mappings** tab.

> **NOTE**: If a pre-1.4.2 version of a mapping is opened in LibSync, it will immediately be marked as needing to be saved due to the addition of the PLM Data Type and Format columns.
>
> **WARNING**: Depending on the database application used for the local EDAC database, if the specified "String Length" is not sufficient to accommodate a string value, the string will be truncated to fit without warning.

► To add a new Standard Attribute Mapping

1. Expand the Standard Attribute Mappings group box on the **General** tab, if not already expanded.
2. Click on the corresponding cell in the "Database Attribute" column and enter the ECAD attribute name.

> **NOTE**: Standard Attribute Mappings may also be added from the Attribute Mappings tables on the Mappings tab. See To add an Attribute Mappings attribute to the Standard Mappings table on page 51 for more information.
>
> **NOTE**: LibSync currently does not provide support multi-list attributes.

3. Specify the desired String Length if other than the default.
4. Specify the desired Format if other than the default.
5. Specify a Unit of Measure conversion rule if desired.

► To edit an existing Standard Attribute Mapping

1. Expand the Standard Attribute Mappings group box on the **General** tab, if not already expanded.

2. Click in the cell you desire to

edit. 3. Enter the new value.

4. Press **Enter** to accept the change.

► To remove a Standard Attribute Mapping

1. Expand the Standard Attribute Mappings group box on the **General** tab, if not already expanded.

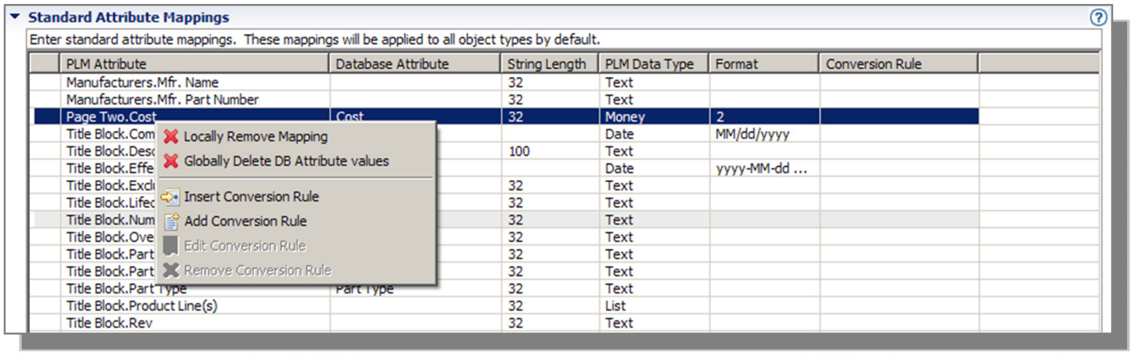
2. Click in the left-most column to select the row to be deleted.

   NOTE: Use **Shift-Click** to select contiguous rows. Use **Ctrl-Click** to select multiple, non-contiguous rows.

3. Right-click and select the desired delete option:

   A. Select **Locally Remove Mapping** to remove the mapping from the Standard Attribute Mappings table without removing the mapping from the custom Attribute Mappings tables.

   B. Select **Globally Delete DB Attribute values** to remove the mapping from both the Standard Attribute Mappings table and the custom Attribute Mappings tables.



Standard Attribute Mappings

Enter standard attribute mappings. These mappings will be applied to all object types by default.

PLM Attribute	Database Attribute	String Length	PLM Data Type	Format	Conversion Rule
Manufacturers.Mfr. Name		32	Text		
Manufacturers.Mfr. Part Number		32	Text		
Page Two.Cost	Cost	32	Money	2	
Title Block.Com			Date	MM/dd/yyyy	
Title Block.Desc		100	Text		
Title Block.Effe			Date	yyyy-MM-dd ...	
Title Block.Excl		32	Text		
Title Block.Lifec		32	Text		
Title Block.Num		32	Text		
Title Block.Ove		32	Text		
Title Block.Part		32	Text		
Title Block.Part		32	Text		
Title Block.Part Type	Part Type	32	Text		
Title Block.Product Line(s)		32	List		
Title Block.Rev		32	Text		

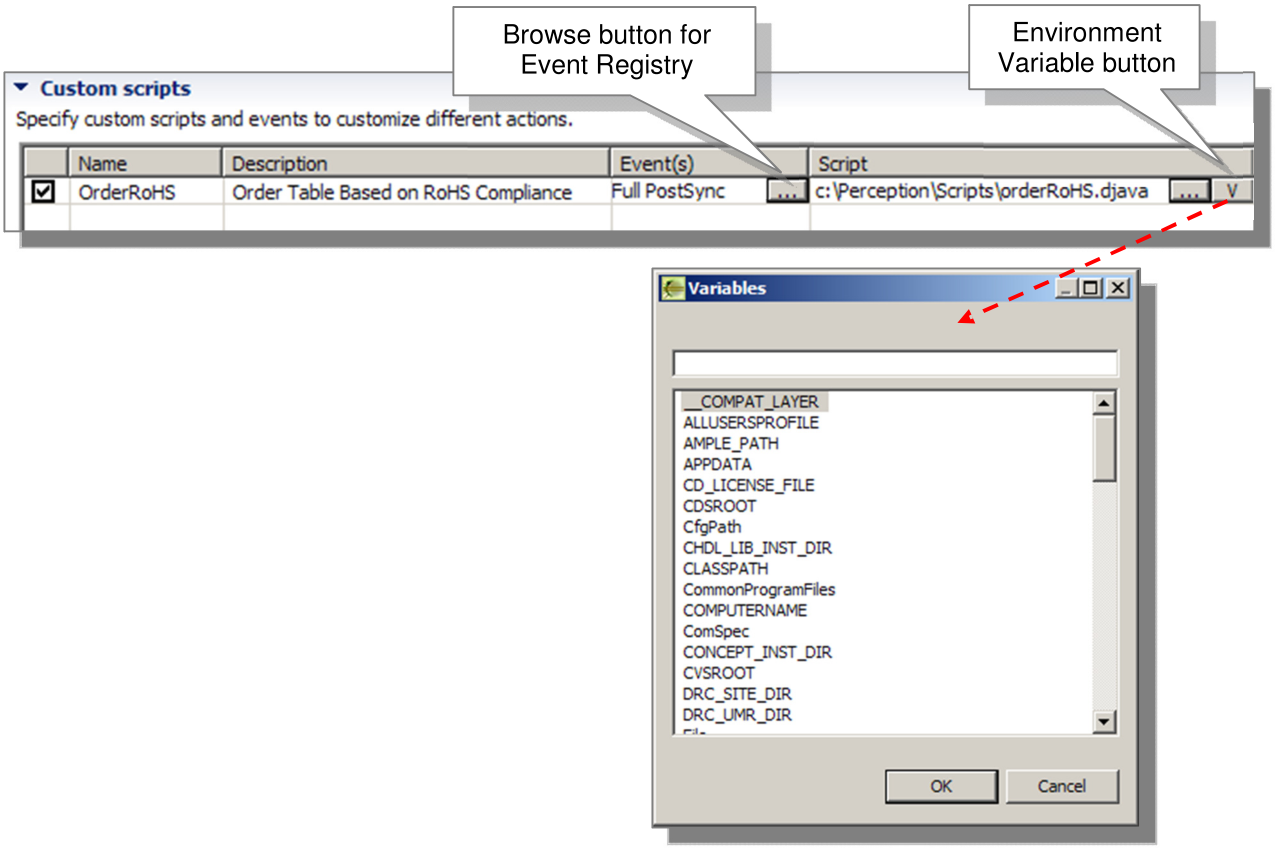
Custom Scripts

The Custom Scripts table allows you to associate a custom script with a triggering event so that customized action can be taken in response to the event. The list of available trigger events is displayed in the Event Registry which is accessed through the browse button in the Event column.

NOTE: For more information on custom scripting, consult LibSync's javadocs at:

Start→Programs→Perception Software→EDAConnect-LibSync→LibSync Java Docs

Script pathnames can be entered directly or through the script browse button. Script pathnames may contain environment variables which can be accessed via the Environment Variable button.



Browse button for Event Registry

Environment Variable button

Custom scripts

Specify custom scripts and events to customize different actions.

	Name	Description	Event(s)	Script
☑	OrderRoHS	Order Table Based on RoHS Compliance	Full PostSync	c:\Perception\Scripts\orderRoHS.djava

Variables

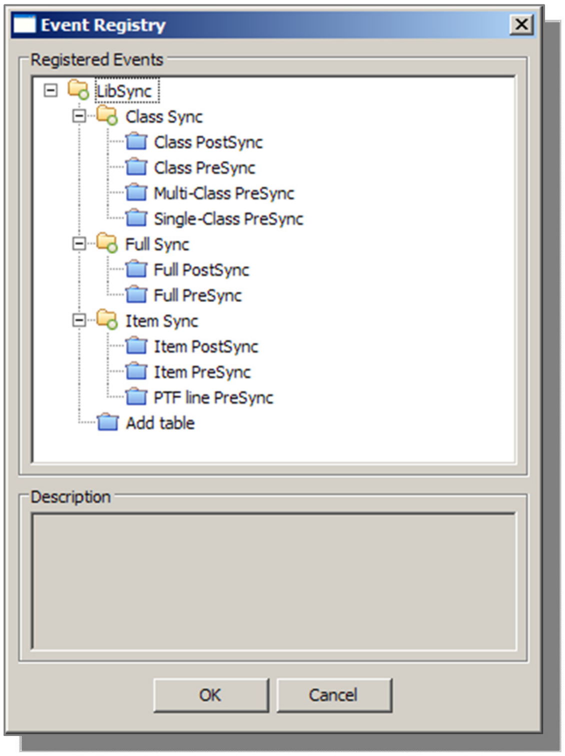
__COMPAT_LAYER
ALLUSERSPROFILE
AMPLE_PATH
APPDATA
CD_LICENSE_FILE
CDSROOT
CfgPath
CHDL_LIB_INST_DIR
CLASSPATH
CommonProgramFiles
COMPUTERNAME
ComSpec
CONCEPT_INST_DIR
CVSROOT
DRC_SITE_DIR
DRC_UMR_DIR

► To assign a trigger event to a custom script

1. Expand the Custom Scripts table on the **General** tab, if not already expanded.
2. In the “Name” column, enter a name for the script.
3. In the “Description” column, enter a description of the function that the script performs.
4. Select the desired trigger events:

   A. Click in the “Event” column to display the Event Registry browse button.

   B. Click the Event Registry browse button display the Event Registry.



   C. Click on the desired trigger event to select it. Use Ctrl-click to select multiple trigger events.

5. In the “Script” column, specify the full pathname to the script:

   A. Click the browse button to select a script.

   B. Click the Environment Variable button to select an environment variable.

Managing PLM to ECAD Mappings

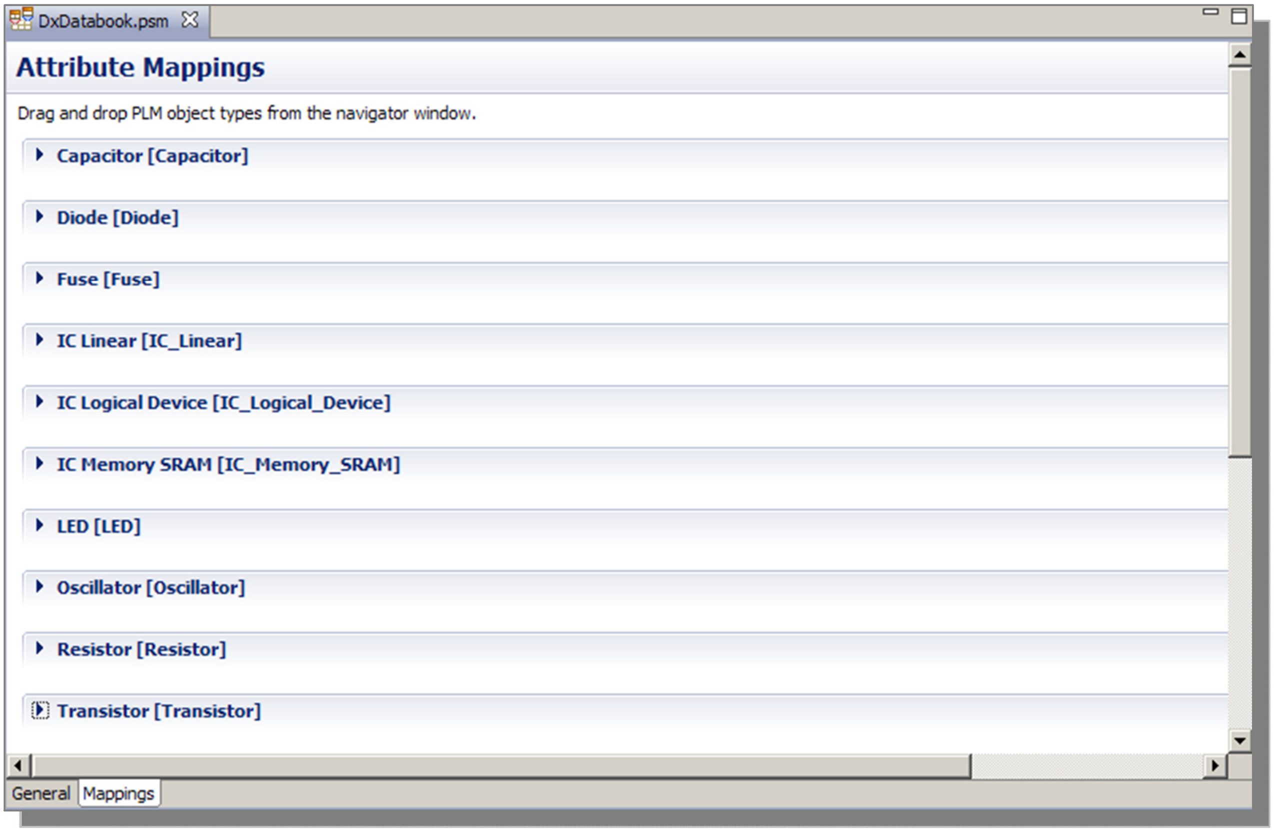
The library synchronization process involves creating a local database table for various Part Types in the PLM system. The names of the attributes in PLM are typically different than the names used in the ECAD library. To handle attribute name translation, LibSync manages the mapping between the two domains.

LibSync supports two types of mappings:

- **Standard Attribute Mappings** – Default mappings common to most Part Types
- **Custom Attribute Mappings** – Part Type specific attribute mappings

The Standard Attribute Mappings table allows you to define a default set of attribute mappings once and apply those mappings automatically in the Attribute Mapping tables on the **Mappings** tab. The Standard Attribute Mappings table is located on the **General** tab and is described in Standard Attribute Mappings on page 37.

The Attribute Mappings table is located on the **Mappings** tab and contains multiple mapping tables – one for each Part Type mapped.



Attribute Mappings

Drag and drop PLM object types from the navigator window.

- Capacitor [Capacitor]
- Diode [Diode]
- Fuse [Fuse]
- IC Linear [IC_Linear]
- IC Logical Device [IC_Logical_Device]
- IC Memory SRAM [IC_Memory_SRAM]
- LED [LED]
- Oscillator [Oscillator]
- Resistor [Resistor]
- Transistor [Transistor]

As can be seen by expanding the table for a given Part Type, a Part Type mapping table consists of the following seven columns:

Custom Attribute Mappings Table

Column	Description	Editable
PLM Attribute	Name of attribute in PLM. This column is populated automatically based on part class.	N
Database Attribute	Specifies the name of attribute in the EDA database.	Y
String Length	Specifies the string column width in the EDA database. Default value is specified by the **string.column.length** property in **config.ini**. Defaults to 32 if not specified.	Y
PLM Data Type	Displays the PLM data type of the attribute.	N
Format	Specifies the format to use if conversion to string is required by the EDA database. Default formats are specified by the ***database.date.forma***_t, **database.money.precision**,_ and ***database.numeric.precision*** properties **config.ini**. The money and numeric formats use an integer to specify the number of digits to the right of the decimal point or “-1” if the raw floating point number is to be displayed. The date format uses standard Java date notation. See **Configuring LibSync Properties** in the **LibSync Installation Guide** for more information.	Y
Standard DB Attribute	Displays the mapping from the Standard Attribute Mapping table.	N
Conversion Rule	Specifies an optional Unit of Measure conversion rule to be applied to the attribute value during synchronization. Unit of Measure conversion is sometimes desired for Page Three attributes for discrete components such capacitors, inductors, resistances, etc. Please see Using Unit of Measure Conversion on page 52 for more information.	Y

When a part class is added to the Attribute Mappings table, any mappings defined in the Standard Attribute Mappings table are inherited. These values can be overridden as desired but apply only to the current part class. You may add as many Database Attribute mappings and Format specifications as desired.

> **NOTE**: Only 18 rows are viewable in an expanded Part Type mapping table view. Therefore, it will typically be necessary to scroll within the table view to see all the attributes.
>
> **NOTE**: LibSync currently does not provide support multi-list attributes.

The process of setting up the PLM to ECAD Attribute Mappings consists of the following general steps:

1. Drag the desired Part Classes (PLM Subclasses) from the Navigation pane to the Attribute Mappings area on the **Mappings** tab. This populates the PLM Attribute column with the superset of the Title Bock properties from the selected part classes.
2. On the General tab, create Standard Attribute Mappings that are common to all Part Types.

   NOTE: This step is optional, but highly recommended. It is a timesaver and helps ensure consistency in mapping of common attributes.

3. For each attribute to be mapped, enter the ECAD attribute name, string length, and format as appropriate.
4. Specify a Unit of Measure conversion rule if desired. The following sections describe these steps in greater detail.

Working with the Attribute Mappings Table

Initial set up of the Standard Attribute Mappings table is described in Standard Attribute Mappings on page 37. Mappings defined in the Standard Attribute Mappings table are applied automatically by LibSync to the individual Part Type mapping tables on the **Mappings** tab. Also, you can readily add a mapping from one of the Part Type tables to the Standard Attribute Mappings table.

Viewing Available Subclasses

Available Part subclasses are displayed in the Navigator pane. Any number of Part subclasses can be added to the **Attribute Mappings** table on the **Mappings** tab of a PSM file.

► To view the available PLM Part subclasses for mapping

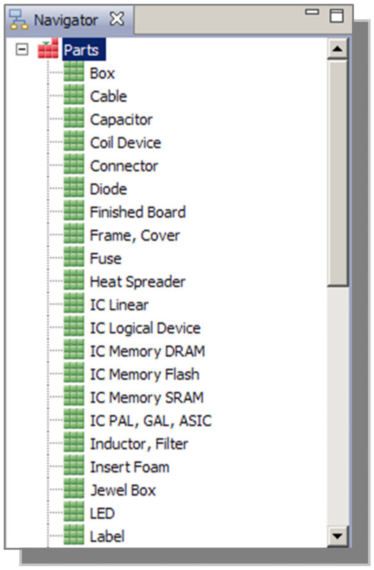
1. If LibSync is not already logged in to the PLM system:

   A. Right-click on the **Parts** node in the Navigator pane and select **Login**.

   B. Provide your PLM system credentials and login.

2. Expand the tree by clicking the → (plus-sign) in front of the **Parts** node.

   The Parts tree expands and shows the available objects from the PLM system per the specifications in the **Window→Preferences: Library Synchronization** (see LibSync Preferences on page 22).



3. If the objects are incorrect, do the following:

   A. Change the “PLM Server Host” or “Part Types” in **Window>Preferences: Library Synchronization**.

   B. Right-click on the corresponding **Parts** node in the Navigator pane and select **Login** to re-login to the PLM system.

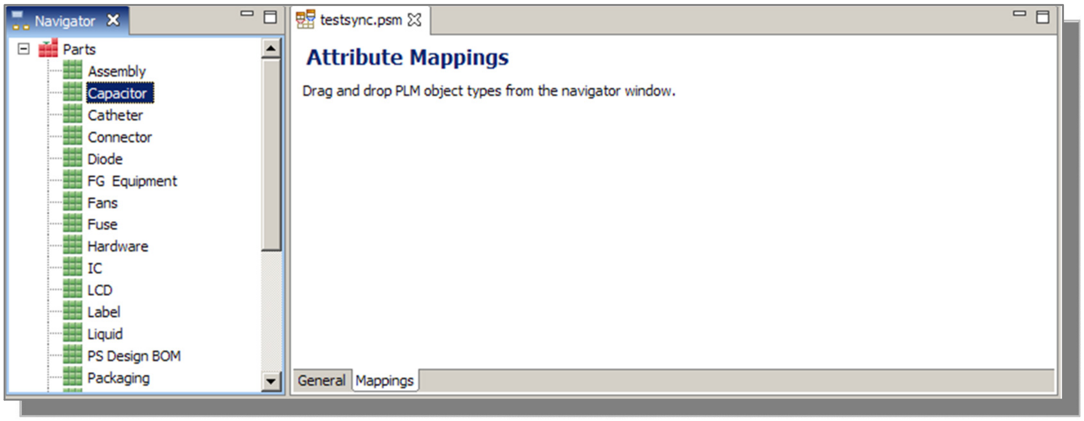
Working with Custom Attribute Mappings

Creating a Part Type Mapping Table

A Part Type Mapping table can be created for any object in the Parts tree. This is done by dragging the part node from the Parts tree in the Navigator window to the Attribute Mappings pane on the Mappings tab in the Editor pane.

► To add a new Part Type mapping table

1. Select the **Mappings** tab in the Editor pane.
2. Expand the **Parts** node in the Navigator pane.
3. Drag-and-drop a node from the Parts tree into the **Mappings** tab.



Attribute Mappings

Drag and drop PLM object types from the navigator window.

LibSync creates a new Part Type mapping table corresponding to the Part Type selected, queries the PLM system for the available attributes, applies the Standard Attribute Mappings, and displays the resultant table.

NOTE: LibSync creates a default table name that is the same as the class/type name from the PLM system. For particularly long names, this default name may exceed the table name length limit of the database being used. If you are using Oracle for your local ECAD database, a best practice recommendation is that all table names and column names be all caps.

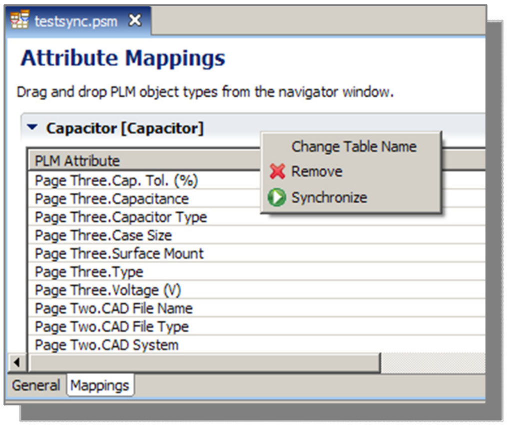
Please consult your database application documentation if you encounter difficulties. If you need to change the default name of a table, see Renaming a Part Type Mapping Table on page 49.

Renaming a Part Type Mapping Table

By default LibSync uses the Part Type name as the Part Type mapping table name for the Local EDA Database. Most databases do not allow special characters other than underscore in a table name. If the default name is not suitable, it may be changed.

► To change the name of a Part Type mapping table

1. Right-click on the header of the Part Type mapping table to be changed and select **Change Table Name**.



Attribute Mappings

Drag and drop PLM object types from the navigator window.

▼ Capacitor [Capacitor]

PLM Attribute
Page Three.Cap. Tol. (%)
Page Three.Capacitance
Page Three.Capacitor Type
Page Three.Case Size
Page Three.Surface Mount
Page Three.Type
Page Three.Voltage (V)
Page Two.CAD File Name
Page Two.CAD File Type
Page Two.CAD System

2. Enter a new table name.

   NOTE: There may be restrictions on the name of the table depending on the target database. Table names should not conflict with SQL reserved words. A list of SQL reserved words can be found in Appendix B – SQL Reserved Words on page 97. If you are using Oracle for your local ECAD database, a best practice recommendation that all table names and column names be all caps.

   NOTE: Each database application also has restrictions on the length of table and column names and the use of reserved keywords. If a table name or column name that is too long for the target database, no warning is given until a synchronization is performed. This is due to the fact that there is no way to determine the maximum table name or column name size through the ODBC interface. Please consult your database application documentation if you encounter difficulties.

Removing a Part Type Mapping Table

► To delete a Part Type mapping table

1. Right-click on the Part Type mapping table header.
2. Select **Remove**.

NOTE: This action permanently removes the Part Type mapping table and associated unique PLM to ECAD mappings. Default mappings defined in the Standard Attribute Mappings table are unaffected by this operation.

Editing Part Type Attribute Mappings

► To edit an existing Part Type attribute mapping

1. Expand the Part Type mapping table on the **Mappings** tab, if not

already expanded.

2. Click in the “Database Attribute” cell you desire to edit.
3. Enter the new value.
4. Press **Enter** to accept the change.

Deleting Part Type Attribute Mappings

► To delete an existing Part Type attribute mapping

1. Expand the Part Type mapping table on the **Mappings** tab, if not already expanded.
2. Click in the left-most column to select the desired row.

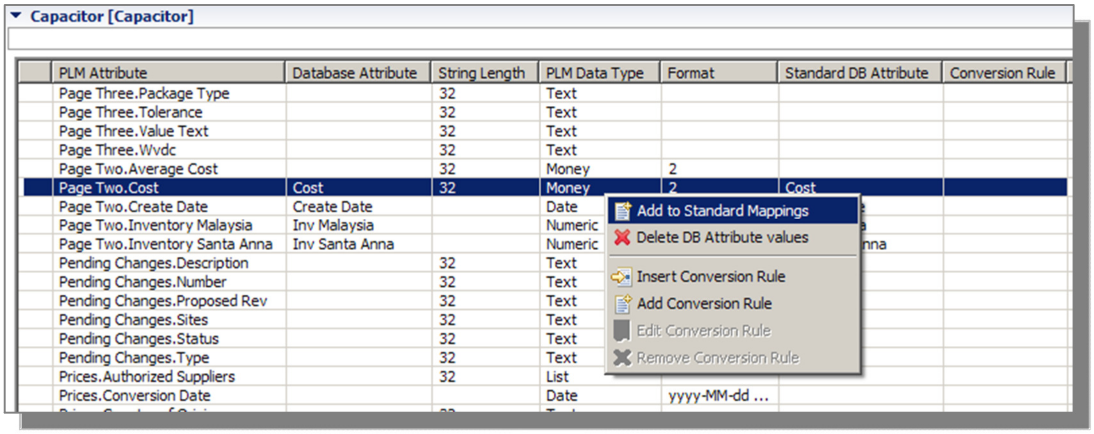
   NOTE: Use **Shift-Click** to select contiguous rows. Use **Ctrl-Click** to select multiple, non-contiguous rows.
3. Right-click and select **Delete DB Attribute values**.

   The Database Attribute mappings are deleted for the current part type only.

Adding Attribute Mappings attributes to Standard Attribute Mappings

► To add an Attribute Mappings attribute to the Standard Mappings table

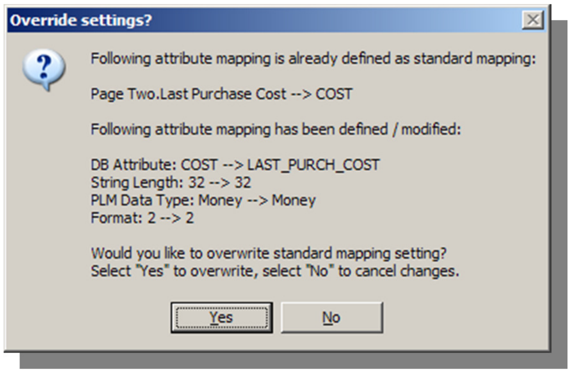
1. On the **Mappings** tab, select the row containing the attribute to be added.
2. Right-click and select **Add to Standard Mappings**.



Capacitor [Capacitor]

PLM Attribute	Database Attribute	String Length	PLM Data Type	Format	Standard DB Attribute	Conversion Rule
Page Three.Package Type		32	Text			
Page Three.Tolerance		32	Text			
Page Three.Value Text		32	Text			
Page Three.Wvdc		32	Text			
Page Two.Average Cost		32	Money	2		
Page Two.Cost	Cost	32	Money	2	Cost	
Page Two.Create Date	Create Date		Date		e	
Page Two.Inventory Malaysia	Inv Malaysia		Numeric		a	
Page Two.Inventory Santa Anna	Inv Santa Anna		Numeric		nna	
Pending Changes.Description		32	Text			
Pending Changes.Number		32	Text			
Pending Changes.Proposed Rev		32	Text			
Pending Changes.Sites		32	Text			
Pending Changes.Status		32	Text			
Pending Changes.Type		32	Text			
Prices.Authorized Suppliers		32	List			
Prices.Conversion Date			Date	yyyy-MM-dd ...		

If a different mapping for the attribute already exists, a confirmation dialog is displayed:



Override settings?

Following attribute mapping is already defined as standard mapping:

Page Two.Last Purchase Cost --> COST

Following attribute mapping has been defined / modified:

DB Attribute: COST --> LAST_PURCH_COST
String Length: 32 --> 32
PLM Data Type: Money --> Money
Format: 2 --> 2

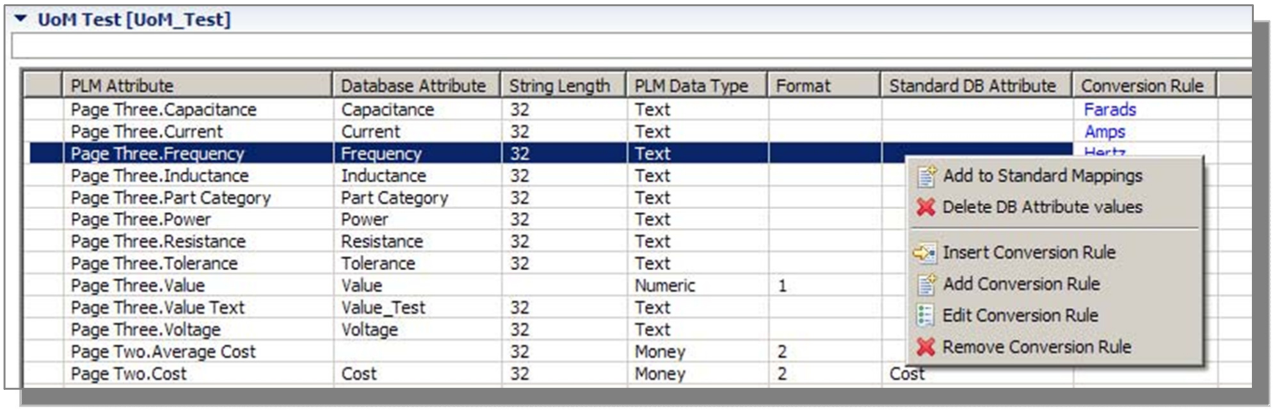
Would you like to overwrite standard mapping setting?
Select "Yes" to overwrite, select "No" to cancel changes.

3. Click **Yes** if you want add the mapping to the Standard Attribute Mappings and apply it globally to ALL custom mapping tables. Otherwise, click **No**.

Using Unit of Measure Conversion

Due to the that fact that component values may have been entered into PLM using inconsistent standards, it may be desirable to scale an attribute's value when synchronizing certain part classes such as those for passive components. This is easily accomplished through LibSync's Unit of Measure (UoM) conversion rules.

UoM conversion rules are specified in the Conversion Rule column of an Attribute Mapping table. Use the Attribute Mappings table context menu to Add, Insert, Edit, or Remove a rule. Each of these commands are described in the table below.



UoM Test [UoM_Test]

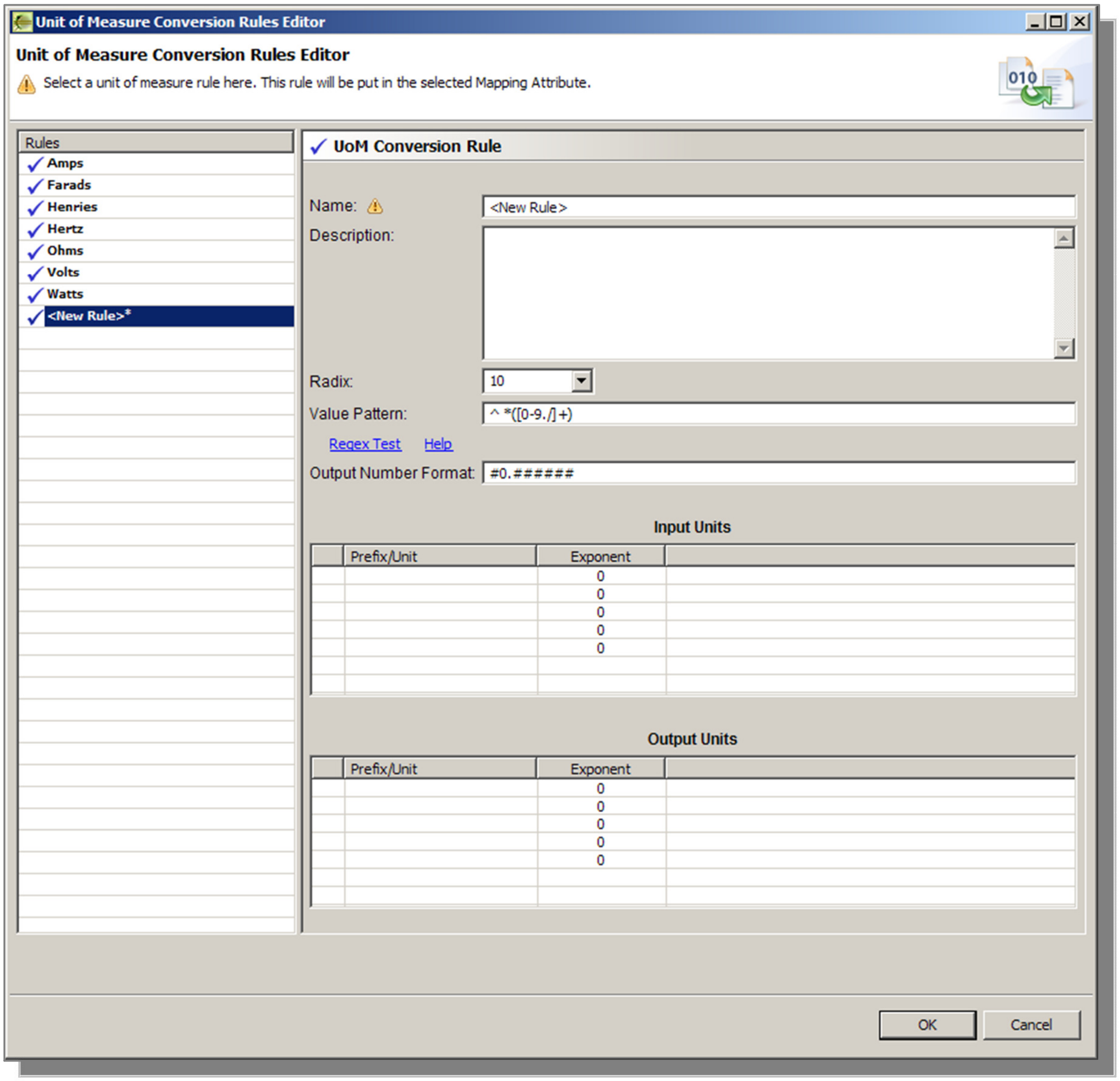
PLM Attribute	Database Attribute	String Length	PLM Data Type	Format	Standard DB Attribute	Conversion Rule
Page Three.Capacitance	Capacitance	32	Text			Farads
Page Three.Current	Current	32	Text			Amps
Page Three.Frequency	Frequency	32	Text			Hertz
Page Three.Inductance	Inductance	32	Text			
Page Three.Part Category	Part Category	32	Text			
Page Three.Power	Power	32	Text			
Page Three.Resistance	Resistance	32	Text			
Page Three.Tolerance	Tolerance	32	Text			
Page Three.Value	Value		Numeric	1		
Page Three.Value Text	Value_Test	32	Text			
Page Three.Voltage	Voltage	32	Text			
Page Two.Average Cost		32	Money	2		
Page Two.Cost	Cost	32	Money	2	Cost	

Attribute Mappings Table - Conversion Rule Menu	
Menu	Action
Insert Conversion Rule	Allows the user to select from a list of existing rules and inserts the rule in the Conversion Rule column of the selected row as a hyperlink. If an existing rule is present, the user is queried for overwrite confirmation.

Add Conversion Rule	Opens the Conversion Rules Editor to a new rule which will be inserted into the Conversion Rule column of the selected row as a hyperlink. If an existing rule is present, the user is queried for overwrite confirmation.
Edit Conversion Rule	Opens the rule in the Conversion Rules Editor. The same thing can be accomplished by double-clicking a conversion rule hyperlink.
Remove Conversion Rule	Queries the user for confirmation and removes the conversion rule from the selected attribute.

Unit of Measure Conversion Rules Editor

The UoM Conversion Rules Editor allows the user to create as many conversion rules as desired. Typically, there would be a rule for each of the base units to be processed (e.g., farads, henries, ohms, hertz, etc.).



Unit of Measure Conversion Rules Editor

Unit of Measure Conversion Rules Editor

⚠ Select a unit of measure rule here. This rule will be put in the selected Mapping Attribute.

Rules
✓ Amps
✓ Farads
✓ Henries
✓ Hertz
✓ Ohms
✓ Volts
✓ Watts
✓ <New Rule>*

✓ UoM Conversion Rule

Name: ⚠ <New Rule>

Description:

Radix: 10

Value Pattern: ^ *([0-9./]+)

Regex Test Help

Output Number Format: #0.######

Input Units

	Prefix/Unit	Exponent
		0
		0
		0
		0
		0

Output Units

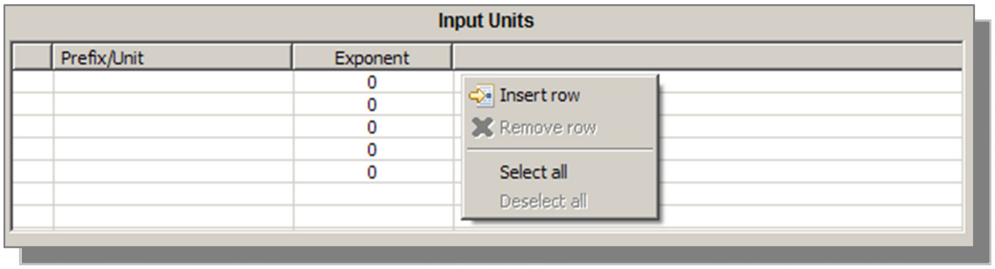
	Prefix/Unit	Exponent
		0
		0
		0
		0
		0

The following table describes the fields of the Unit of Measure Conversion Rules Editor:

Unit of Measure Conversion Rules Editor Fields	
Item	Description
Rules	Table of defined rules. Select a rule from this table when executing the Insert Conversion Rule command in an Attribute Mappings table.
Name	Rule name
Description	Rule description
Radix	Radix for base unit calculations. Typically, this will be base 10.
Value Pattern	Regular expression to extract the numeric portion of an incoming string value. The expression *must* define the numeric portion of the string as the first parenthetical group in the regular expression. Defaults to "^ (*[0-9./]+)" which interprets a string containing digits 0-9, a decimal point, or "/" as a number.
Regex Test	Displays a Regex Test dialog in which the user can verify the parsing results for a Value Pattern.
Help	Displays a table of commonly used regular expression meta-characters.
Output Number Format	Specifies the Java number format for the numeric portion of the output string. Default value is "#0.######".
Input Units Table	Table of input Prefix/Unit strings and associated exponent to be applied to the Radix. Entries are case sensitive.
Output Units Table	Table of output Prefix/Unit strings and associated exponent to be applied to the Radix. Entries are case sensitive. Output values are auto-scaled to the "best match" of the enumerated output Prefix/Units.

NOTE: Since the default Value Pattern contains "/", strings like "1/4" will be converted to the decimal equivalent "0.25". If this is not desired, remove the "/" from the Value Pattern regular expression.

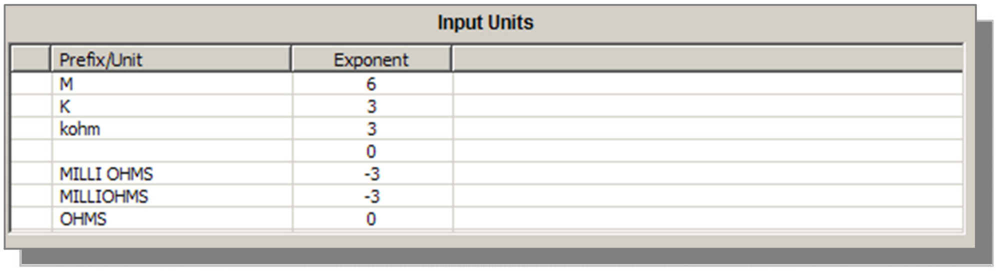
Whenever a new rule is opened in the Conversion Rules Editor, the Input Units table and the Output Units table are populated with five rows with no Prefix/Unit value and Exponent set to 0. The table context menu can be used to insert or remove rows in the table.



Input Units

Prefix/Unit	Exponent
	0
	0
	0
	0
	0

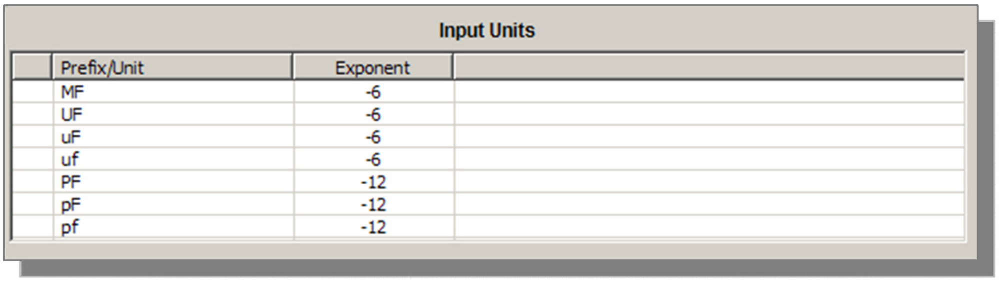
The Input Units table enumerates each of the Prefix/Unit strings to be processed. For example, the Input Units table for resistance might be:



Input Units

	Prefix/Unit	Exponent	
	M	6	
	K	3	
	kohm	3	
		0	
	MILLI OHMS	-3	
	MILLIOHMS	-3	
	OHMS	0	

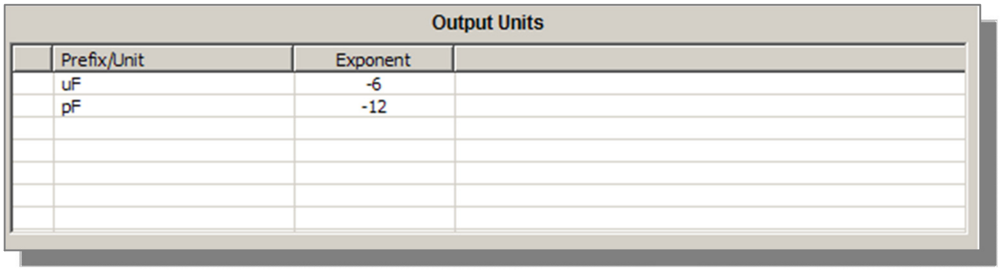
The Input Units table for resistance might be:



Input Units

	Prefix/Unit	Exponent	
	MF	-6	
	UF	-6	
	uF	-6	
	uf	-6	
	PF	-12	
	pF	-12	
	pf	-12	

The Output Units table enumerates each of the Prefix/Units allowed for a given output unit. Since, capacitance is typically expressed in either micro-farads or pico-farads, the Output Units table for capacitance might be:

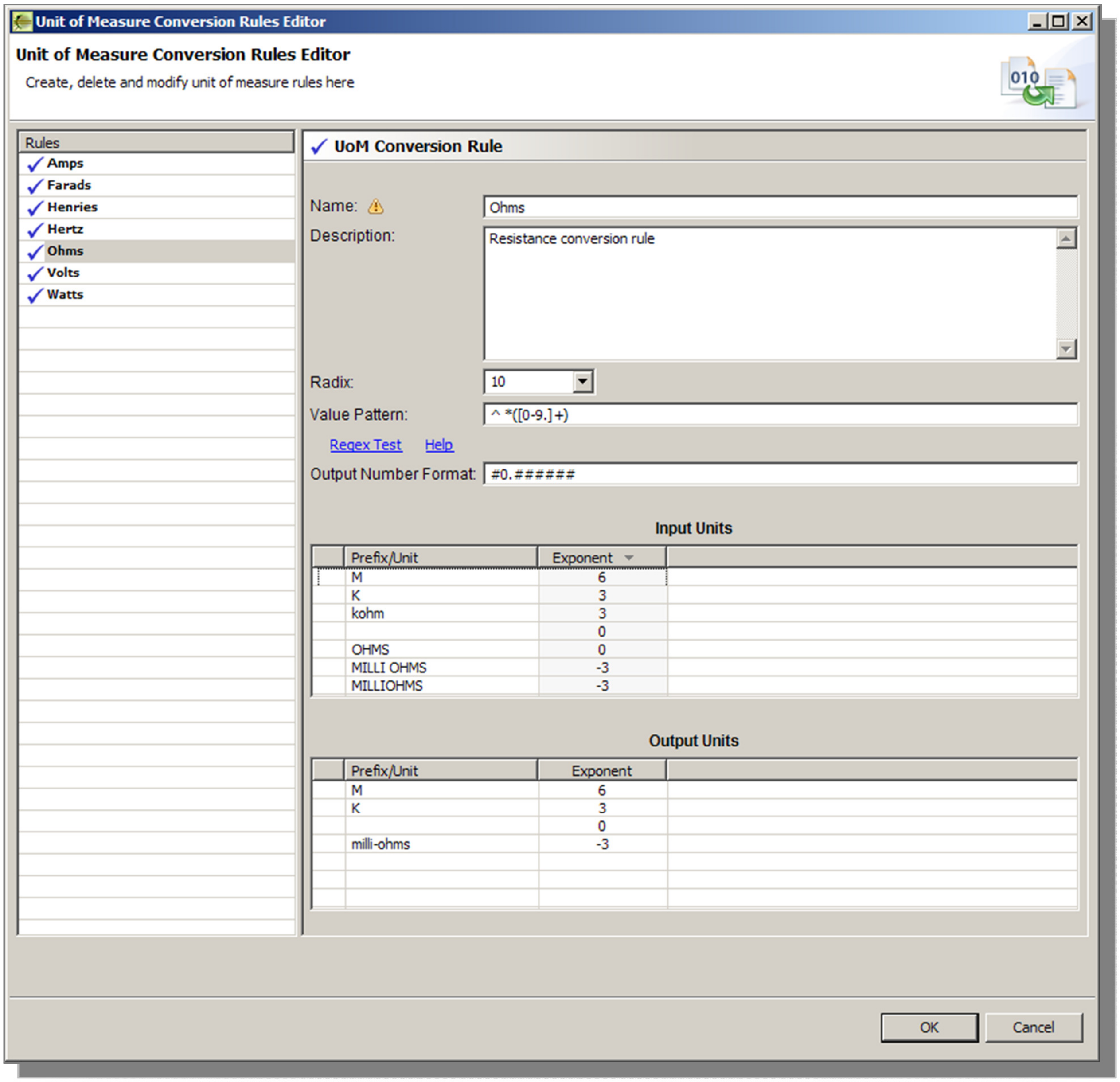


Output Units

	Prefix/Unit	Exponent	
	uF	-6	
	pF	-12	

> **NOTE**: If uppercase is desired for the prefix, then the Prefix/Unit should be entered in uppercase (e.g., “UF”, “PF”).

A completed Unit of Measure rule for resistance might appear as follows:



Unit of Measure Conversion Rules Editor

Create, delete and modify unit of measure rules here

Rules

- ✓ Amps
- ✓ Farads
- ✓ Henries
- ✓ Hertz
- ✓ Ohms
- ✓ Volts
- ✓ Watts

✓ UoM Conversion Rule

Name: ⚠ Ohms

Description: Resistance conversion rule

Radix: 10

Value Pattern: ^ *([0-9.]+)

Regex Test Help

Output Number Format: #0.######

Input Units

Prefix/Unit	Exponent
M	6
K	3
kohm	3
	0
OHMS	0
MILLI OHMS	-3
MILLIOHMS	-3

Output Units

Prefix/Unit	Exponent
M	6
K	3
	0
milli-ohms	-3

► To create a Unit of Measure conversion rule

1. Right-click in an Attribute Mappings table and select **Add Conversion Rule**.
2. Enter the rule Name and Description.
3. Select the desired Radix if other than the default.
4. Enter the desired Value Pattern if other than the default.

   NOTE: Click on **Regex Test** to validate a Value Pattern against sample value strings. Click on **Help** for a listing of commonly used regular expression meta-characters.

5. Enter the Output Number Format if other than the default.

> **NOTE**: Append a space to the Output Number Format if you desire to separate the numeric value from the prefix/unit in the output (e.g., “100 pF” vs. “100pF”).

6. Enter the input Prefix/Unit strings and associated exponents in the Input Units table. Entries are case sensitive.
7. Enter the output Prefix/Unit strings and associated exponents in the Output Units table. Entries are case sensitive.
8. Click the **OK** button to save the rule to the Rules table and exit the Conversion Rules Editor.

Click on an existing rule in the Rules table to save the current rule and open the selected rule in the Conversion Rules Editor.

Right-click in the Rules table and select **New Rule** to add another rule to the Rules table.

Unit of Measure Processing

Unit of Measure conversion consists of the following steps:

1. Parsing of the incoming value into a numeric value and a

prefix/unit value 2. Scaling of the numeric value to the base unit

3. Scaling and formatting of the output value

If the incoming attribute value is string, LibSync uses the Value Pattern to parse the string into a Value String and a Unit String. The Value String is set to the group 1 content of the Value Pattern and the Unit String is set to the *trimmed* remainder of the incoming string.

The Unit String is then used to locate the corresponding exponent E_{IN} in the Input Unit table. If a match is found, Value String is converted to a numeric value N and scaled to the base unit using the specified radix R:

$$N = \text{Number(Value String)} * R\hat{}\ E_{IN}$$

If a match is not found, the incoming string value is passed through as-is and an appropriate error message is written to the log file if *uom.logging* is enabled in the config.ini file. If the incoming property is numeric, it is assumed to be in the appropriate base unit and no scaling is performed.

LibSync then auto-scales the numeric value N to the most appropriate range available in the Output Units table. To be consistent with typical value ranges for discrete components such as capacitors, inductors, and resistors, the following algorithm is used to determine the output Prefix/Unit:

Output Units Auto-scaling Algorithm
(Radix = 10)

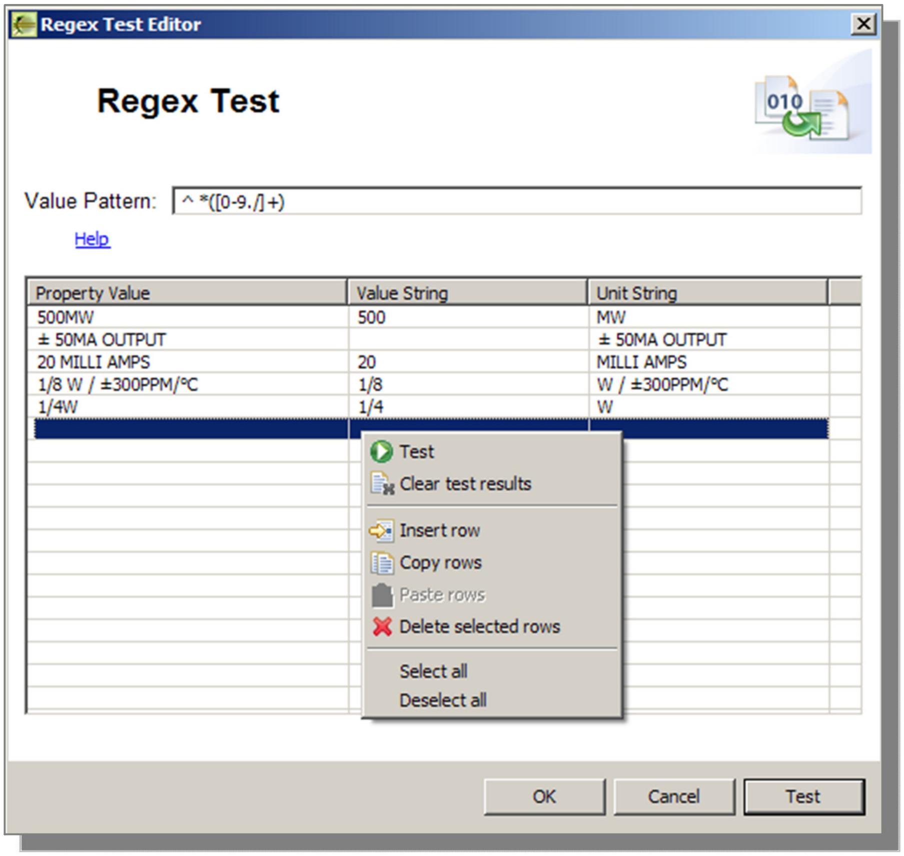
Primary Condition	Secondary Condition	*Tertiary Condition*	Then E_{OUT} Is
$N \geq R\hat{}\ E_{MAX}$			E_{MAX}
$R\hat{}\ E_2 > N \geq R\hat{}\ E_1$	$E_2 - E_1 = 3$		E_1
	$E_2 - E_1 > 3$	$N < R\hat{}((E_2 + E_1) / 2)$	E_1
		$N \geq R\hat{}((E_2 + E_1) / 2)$	E_2
$N < R\hat{}E_{MIN}$			E_{MIN}

If R = 2, the algorithm is the same but an exponent difference of 10 is used.

If the incoming value is string, LibSync uses appropriate exponent EOUT to scale the numeric value N:

$$N = N / R\hat{}\ EOUT$$

LibSync then applies the specified Output Number Format and concatenates the appropriate Unit String to form the output Value string. If the incoming property is numeric, it is simply scaled as described above.



Regex Test Editor

Regex Test

Value Pattern: ^ *([0-9./]+)

Help

Property Value	Value String	Unit String
500MW	500	MW
± 50MA OUTPUT		± 50MA OUTPUT
20 MILLI AMPS	20	MILLI AMPS
1/8 W / ±300PPM/°C	1/8	W / ±300PPM/°C
1/4W	1/4	W

Regex Test

The **Regex Test** dialog provides an easy way to validate a Value Pattern against sample data. A Value Pattern and multiple Property Values can be entered manually or via copy/paste.

► To validate a Value Pattern

1. Click on the **Regex Test** hyperlink to open the **Regex Test** dialog.
2. Enter a Value Pattern if a pattern other than the default is desired.
3. Enter or copy/paste as many sample Property Values as desired.

   NOTE: Multiple rows may be selected by Ctrl-click. Right-click in the table and select **Paste rows** to paste multiple cells.

4. Click the **Test** button.

   LibSync parses the Property Value strings into Value String and Unit String.

5. If the parsing results are satisfactory, click the **OK** button to save the Value Pattern to the “Value Pattern” field in the Conversion Rules Editor. Otherwise, go to step 2 to modify the Value Pattern and test again.

Regex Help

The **Help** hyperlink in the Conversion Rules Editor or the Regex Test dialog displays the following information:

Performing Database Synchronization

LibSync supports two modes of synchronization:

- Immediate (manual) synchronization
- Scheduled background synchronization

Immediate synchronization is useful for testing an initial synchronization configuration, while scheduled background is intended for the day-to-day update of the ECAD library attributes.

NOTE: Running multiple instances of LibSync is not recommended. Care should be taken to insure that only a single instance of LibSync is active.

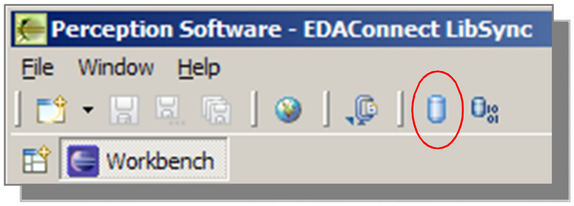
Initiating Synchronization

Use the following procedure to initiate library synchronization.

WARNING: LibSync deletes all synchronized tables in the Local EDA Database, one at a time, as the Part Types are synchronized. It is best to schedule library synchronizations to occur during non-production hours for the engineers.

► To execute an immediate synchronization

Click the **Synchronize** button on the LibSync toolbar. The synchronization operation begins immediately.



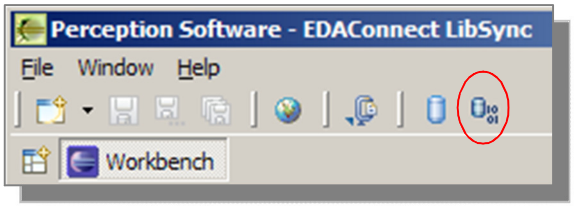
NOTE: This function has no effect on a background synchronization already scheduled.

NOTE: For efficiency, LibSync processes table in batches of 2000 rows.

► To execute a background synchronization

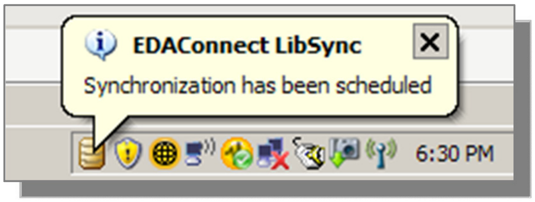
1. If you have not already done so, set up a library synchronization schedule. See To execute a background synchronization on page 63.
2. Click the **Synchronize in Background** button in the LibSync toolbar.

   LibSync minimizes itself to an icon in your Windows system tray. Library synchronization occurs based on the specified schedule.



Controlling Synchronization

When LibSync is operating in the background, an icon is placed into the system tray on Windows platforms. This icon can be used to control and obtain status of an existing synchronization.


EDAConnect LibSync

Synchronization has been scheduled

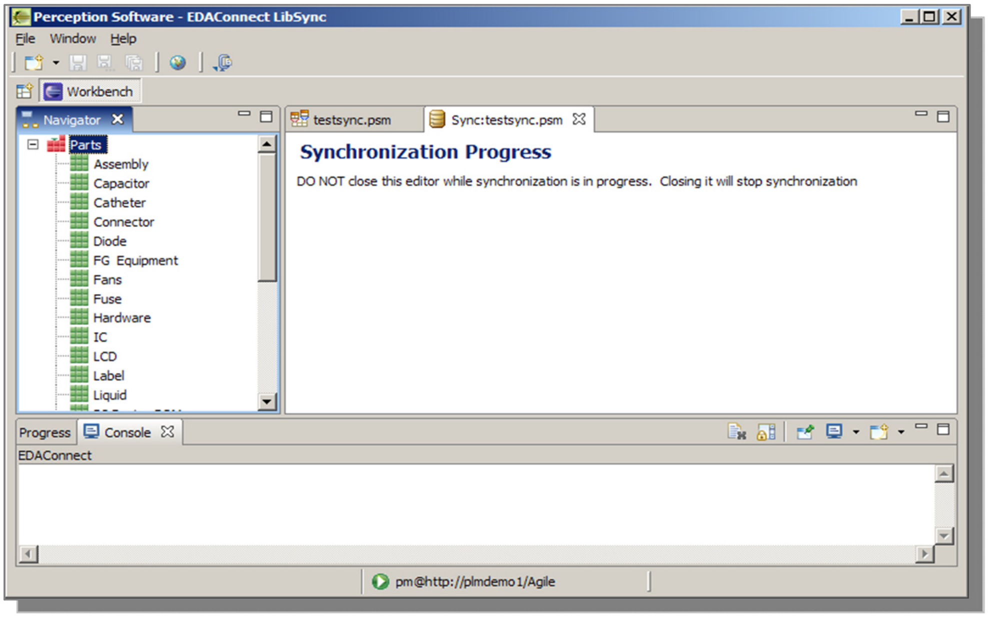
► To control and monitor library synchronization

1. Right-click the LibSync icon in the system tray to display the LibSync Synchronization menu:

2. Select the desired operation:

   - Select **Show Workbench** to reopen the main LibSync main window.
   - Select **Hide Workbench** to minimize the application to the system tray.
   - Select **Show Progress** to display synchronization status.
   - Select **Cancel Synchronization** to stop the current synchronization.
   - Select **Quit** to exit the LibSync application.

Show Progress opens LibSync's main window. The current synchronization status is displayed in the Editor Pane as shown below:

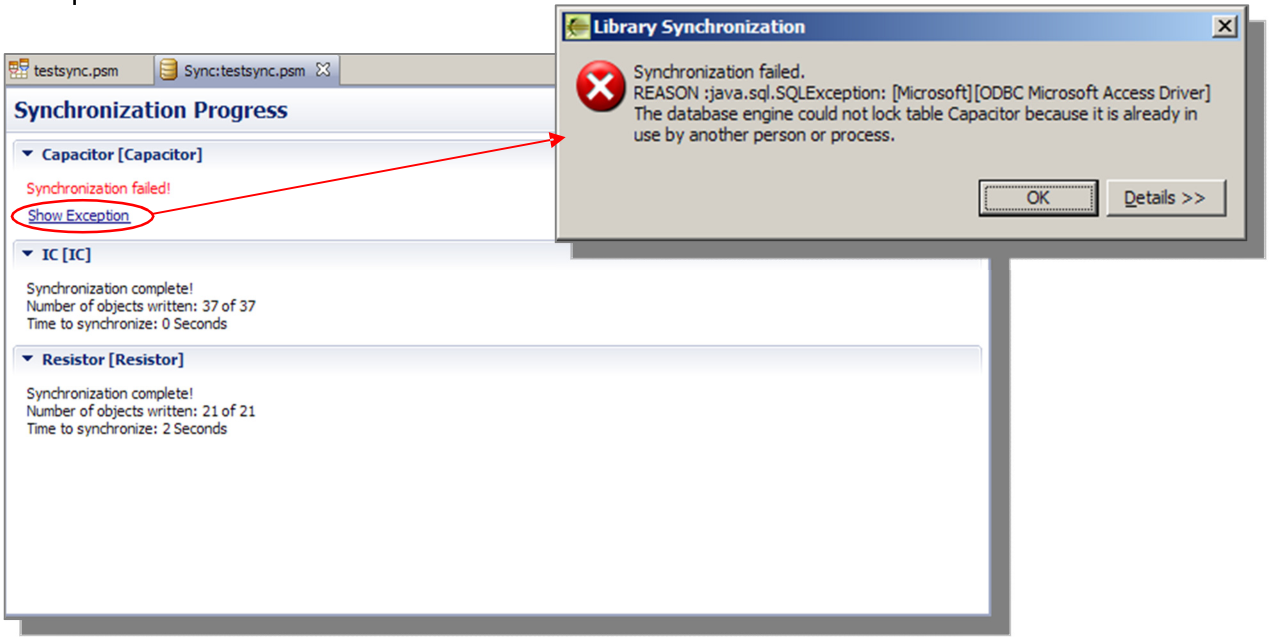


Synchronization Progress

DO NOT close this editor while synchronization is in progress. Closing it will stop synchronization

NOTE: Closing the Synchronization Progress tab in the Editor pane will terminate synchronization.

As synchronization progresses, the status for each Part Type synchronized is displayed. If an error is encountered for a given Part Type, a **Show Exception** link is created and may be used to view information regarding the cause of the exception.



Synchronization failed!

Synchronization complete!
Number of objects written: 37 of 37
Time to synchronize: 0 Seconds

Synchronization complete!
Number of objects written: 21 of 21
Time to synchronize: 2 Seconds

Synchronization failed.
REASON :java.sql.SQLException: [Microsoft][ODBC Microsoft Access Driver] The database engine could not lock table Capacitor because it is already in use by another person or process.

If Email Notification has been set up, the LibSync log is sent to the specified email addresses upon completion of synchronization.

Performing PTF File Synchronization

Cadence Concept HDL libraries specify various part properties in a text file known as the Physical Part Table or PTF file. The PTF file has an extension of '.ptf' and stores packaging properties and other data such as ERP properties for a part. A typical PTF file may contain part package types, manufacturers, part numbers, and other data such as Status, Value, Tolerance and Cost for a part. Each physical part must have an entry in the PTF file in order to package properly. The PTF files are stored either at the library level (library level PTF) or at the part level (cell level PTF).

PTF files are used for:

- Component browsing
- Component instantiation (placement)
- Packaging
- Deriving BOM content

After a PTF has been created and released to the central library, the data that is contained in the PTF can become outdated as updates are made in PLM. An example of this is when a part status changes from Approved (A) to non-approved (NA) in the ERP/PLM system. If the customer is using Status as a property in their PTF files, then this change needs to be reflected in the PTF so that design engineers are discouraged from placing the non-approved part on new designs, but legacy designs can continue using the part without issue.

Keeping PTF's synchronized manually is a time consuming and error prone process. LibSync provides a way to automate PTF updates and ensure that the content of the PTF files are in sync with information in the PLM system.

PTF File Format Overview

PTF files are plain text files and can contain information for one or more part types. The physical part table file has the following general format:

PTF File Format

```
FILE_TYPE = MULTI_PHYS_TABLE;
PART 'part_name'
[ part_type_ prop_list ]
part_table_format_definition
part_table_data_entries
END_PART
PART 'part_name'
        .
        .
        .
END_PART
END.
```

Between these two lines can be any number of part tables. Each table begins with a line with the keyword PART followed by the name of the part type being redefined by the table entries, and ends with the keyword END_PART. Part tables must have a Part Table Format Definition record and may have as many Part Table Data records as necessary. The Part Table Format Definition defines the property name for each property column used in the Part Table Data record. The Part Table Data has the following format:

instance_value = part_type_value [(name_spec)] [:added_property = 'added_value']

The instance value specifies the *key* property values that are used by the packager select the *injected* part type values to use to create a new part type to be instantiated on a design. The properties in a Part Table Data record are separated by a delimiter which is typically a vertical bar.

Each PTF record type is summarized in the following table.

PTF Record Types	
Type	**Description**
FILE_TYPE	Initiates a PTF file and specifies the file type (.e.g, MULTI_PHYS_TABLE or LIBRARY_PARTS)
PART	Identifies the physical part name of the component being redefined by the table entries. Must be enclosed in single quotes.
Part Type Property List	Used to add new properties to all instances of a part type without having to modify the physical information files or library drawings. property_name = property_value There can be any number of property name/value entries, but there can be only one entry per line. If a property does not fit on one line, the tilde (~) can be used as a continuation character.
Part Table Format Definition	Defines the property name for each column in the Part Table Data Record. : prop_name [(**OPT**='def')] [separator prop_name ...] = prop_name [separator prop_name ...] ;
Part Table Data	Specifies the *key* properties (left-hand side of the first “=” sign) used to select the *injected* properties (right-hand side of the first “=” sign) to be used by the packager to create a new type part. instance_value = part_type_value [(name_spec)] ~ [:added_property = 'added_value'] **NOTE**: LibSync does not currently support the :added_property = 'added_value' syntax and skips PTF files containing such entries.
END_PART	Terminates a part table.
END.	Terminates a PTF file.

Example PTF Files

Here is a sample PTF file for an IC:

```
FILE_TYPE = MULTI_PHYS_TABLE;
PART '100307'
CLASS=IC
PHYS_DES_PREFIX=U
{============================================================================}
:PACK_TYPE | PART_NUMBER = BODY_VERSION | DESCRIPTION | VENDOR | REV | CAGE_CODE |
STATUS  |  DATA_SHEET  |  JEDEC_TYPE ;
{===========================================================================}
'QFLAT24’ | '100307FMQB'(!) = '1' | 'LOW POWER QUINT EXCLUSIVE OR/NOR GATE' |  'NATIONAL
SEMICONDUCTOR CORP' | '0' | '27014' | 'A' | '100307-data.pdf' | 'A4013A'
END_PART
END.
```

Here is a typical PTF file for a discrete component:

```
FILE_TYPE=MULTI_PHYS_TABLE;
PART 'CAP'
CLASS=DISCRETE
:VALUE | TOLERANCE | VOLTAGE | PACK_TYPE | PART_NUMBER=ROHS | CLASS | DESCRIPTION |
JEDEC_TYPE | ALT_SYMBOLS | MFG | MFG_PN | PART_NUMBER;
'10UF' | '20%' | '6.3V' | '1263' | '0161-0591'='TBD' | 'IC' | 'CAP' | 'CDXD-R0002_254C-L032W016H019A' |
'(CDXD-R0002_254C-L032W016H019A)' | '-' | '-' | '0161-0591'
'10UF' | '20%' | '6.3V' | '1263A' | '0161-0591'='TBD' | 'IC' | 'CAP' | 'CDXD-R0002_254C-
L032W016H019A_A' | '(CDXD-R0002_254C-L032W016H019A)' | '-' | '-' | '0161-0591'
'0.01UF' | '10%' | '50V' | '0805' | '0160-5945'='TBD' | 'IC' | 'CAP' | 'CDXD-R0002_175C-L020W012H013A' |
'(CDXD-R0002_175C-L020W012H013A)' | '-' | '-' | '0160-5945'
END_PART
END.
```

LibSync PTF Synchronization Process

LibSync can be used to automatically synchronize PTF files with PLM system data by using the customer part number (CPN) property as the unique key in both the PTF and PLM data. LibSync provides the ability to configure how *key* and *injected* properties are updated based on a customer's internal business rules. Since changing *key* properties in a PTF requires a commensurate schematic update, *key* properties typically are not included in a synchronization. On the other hand, synchronizing *injected* or 'pass through' properties in a PTF is acceptable since no corresponding changes to the schematic are required.

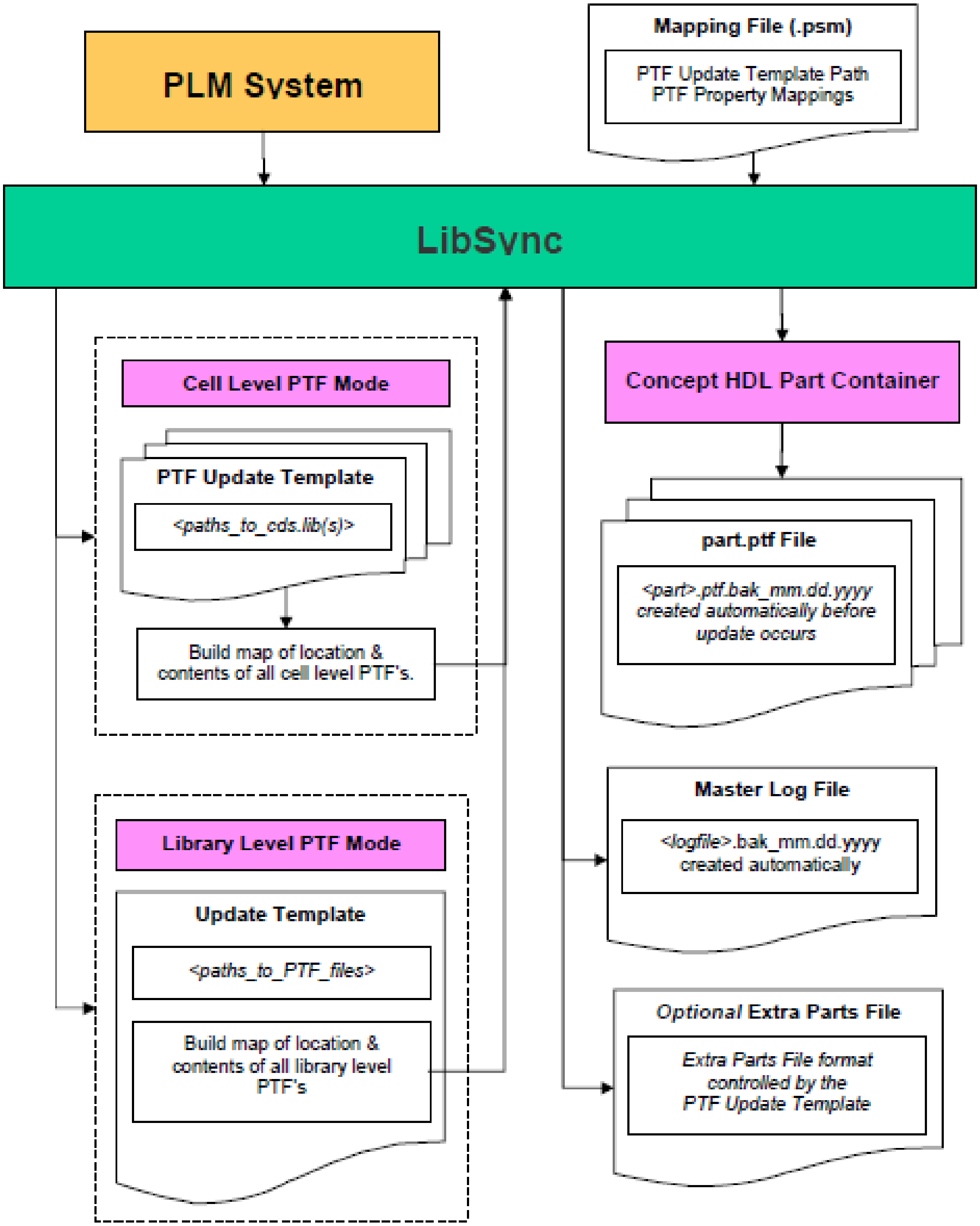
Whether updates are permitted or not for a given property is specified by a PTF Update Template. If a property update is specified as "*allowed"*, LibSync updates the property in the PTF and records the before and after state of the PTF record in the PTF log file. If a property update is specified as *"blocked"*, LibSync stores the update in the PTF log file so a use implement the change manually if appropriate.

LibSync can process PTF files with standard names such as *part_table.ptf* or with customized names such as *digital_ic.ptf* or *1234567.ptf*. If Unit of Measurement conversion is required for PTF updates, the conversion is handled through scripting so that it can be customized to specific syntax required by the customer. Backup files are created for each PTF file synchronized. The backup file resides in the same part table folder as the PTF file. If necessary, an update can be rolled back by referencing the master logfile in the restore option. See PTF Command Line Interface on page 81.

> **WARNING**: A PTF file will not be included in the synchronization if any of the following are TRUE:
>
> - The number of properties in the Part Table Data record does not equal the number of properties defined in the Part Table Format Definition.
> - Any Part Table Data record in a PTF file contains an added property:
>
> :added_property = 'added_value' entry

A data flow diagram of the PTF synchronization process is shown on the next page.

PTF Synchronization Dataflow Diagram


PLM System
Mapping File (.psm)
PTF Update Template Path
PTF Property Mappings
LibSync
Cell Level PTF Mode
PTF Update Template
<paths_to_cds.lib(s)>
Build map of location & contents of all cell level PTF's.
Library Level PTF Mode
Update Template
<paths_to_PTF_files>
Build map of location & contents of all library level PTF's
Concept HDL Part Container
part.ptf File
<part>.ptf.bak_mm.dd.yyyy created automatically before update occurs
Master Log File
<logfile>.bak_mm.dd.yyyy created automatically
Optional Extra Parts File
Extra Parts File format controlled by the PTF Update Template

Key PTF Synchronization Files

PTF synchronization utilizes the following key files:

PTF Synchronization File Usage	
File	**Description**
Extra Parts File	Logs all parts in the PLM that are not currently in an existing PTF file in the library.
Mapping File	Standard LibSync mapping file (.psm). Defines the PLM connexion, location of the PTF update template, update schedule, email notifications, PLM to PTF property mappings, and custom script execution.
PTF Log File	Logs detailed information regarding the updates to each PTF file.
Update Template	Specifies the PTF file set to be processed and the PTF property update rules.

These files can be located anywhere in the file system. The pathnames to these files are specified in the PTF Update Template.

Extra Parts File

Parts that are in the PLM system but not in the PTF files are considered extra parts and are recorded in the Extra Parts file. Through the Update Template, a user can configure LibSync to build a new, properly formatted PTF file containing the extra parts data. Properties that are not stored in the PLM (e.g., footprint name, symbol name, etc.) can be identified as '*unknown'* in the resultant PTF by specifying "*unknown=true*" for the corresponding property.

NOTE: The '*unknown*' option defaults to "*true*" if not specified.

Parts that are in the PLM system but not in existing PTF files are logged first followed by parts that are in the PTF files but not in the PLM system. This information is useful for identifying mismatches between CAD libraries and PLM content.

Mapping File

The mapping file is a standard .psm file. It provides the necessary information to access the PLM system, locate the PTF Update Template, specify the synchronization schedule, and configure email notifications. The mappings between PLM properties and PTF properties are specified in the mapping as well as the trigger events for any custom scripts to be executed.

PTF Log File

The log file records detailed information regarding each PTF file update. This includes the following:

- PTF file pathname
- Date and time of the update
- Properties updated
- Properties blocked from being updated
- Original PTF data record
- Updated PTF data record

A sample PTF log file is shown below:

```
 PTF Updated:
C:\customer_libs\hdl\100k\100307\part_table\part.ptf
Date & time Updated:
01/20/2007 01:30:23
Properties updated:
CAGE_CODE=27077
Properties Blocked from Update:
PTF entry was:
'QFLAT24’|'100307FMQB'(!) = '1'|'LOW POWER QUINT EXCLUSIVE OR/NOR GATE'| 'NATIONAL
SEMICONDUCTOR CORP'|'0'|'27014'|'A'|'100307-data.pdf'|'A4013A'
Updated PTF entry:
'QFLAT24’|'100307FMQB'(!) = '1'|'LOW POWER QUINT EXCLUSIVE OR/NOR GATE'| 'NATIONAL
SEMICONDUCTOR CORP'|'0'|'27077'|'A'|'100307-data.pdf'|'A4013A'
```

PTF Update Template File

The Update Template is an XML file that governs all aspects of PTF file synchronization and is managed outside of the LibSync application. The pathname to the PTF Update Template is specified in the “Connection String” field of the Database Connection table on the **General** tab of the .psm mapping file. It is through the Update Template that all PTF-file-specific information required for synchronization is specified. The parameters in the Update Templates are described in the following table.

PTF Update Template Properties	
Tag	**Description**
Update Mode	
<ptf-mode name=”*mode”* />	Sets the PTF update mode to either ‘cell’ or ‘library’.
Library Paths	
<input-files> : </input-files>	Specifies paths to the PTF files (.ptf) for library mode or to the PTF library files (.lib) for cell mode.
<file>*path*</file>	Specifies path name to an individual .lib file.
Exclusion File Rules	
<exclusion-rules case_sensitive="true” | “false" type="wildcard” | “regexp” > : </exclusion-rules>	Specifies PTF files to be excluded from processing. Exclusion rules may be case sensitive and may use either wildcards or regular expressions.
<rule>*path*</rule>	Specifies single path exclusion rule.
<rules-list> : </rules-list>	Specifies multiple path exclusion rule with one pathname per line.
Field Delimiters	
<delimiters> : </delimiters>	Specifies the column delimiters used for each PTF record. Multiple delimiters may be specified. The default delimiter is '|'.
<delimiter type=”*scope*”> : </delimiter>	Specifies a delimiter and its scope. Set scope to “all” to apply to both key and injected properties, to “key” to apply to key properties only, and to “injected” to apply injected properties only.
Character Exclusion List	
<ignore-characters> : </ignore-characters>	Specifies the set of characters to be ignored in PLM property values. This allows characters that are legal in PLM values but not in PTF values to be filtered out.
<char>@</char>	Specifies the character to be ignored.

PTF Update Template Properties	
Tag	**Description**
Property Update Specification	
<parts> : </parts>	Delineates the part update specification.
<global-properties> : </global-properties>	Delineates the global property processing specification which is applied to all part classes unless specifically overridden by a part-class specification.
<key-properties> : </key-properties>	Specifies key property processing. Settings can be overridden by a part-class specification.

<property
name=”*property_name”*
update_allowed=”true” | “false” />
	Specifies a PTF property name and whether updates are allowed or blocked. ‘update_allowed’ defaults to “false” if not specified.

<injected-properties> : <injected-properties>	Specifies injected property processing. Settings can be overridden by a part-class specification.

<property
name=”*property_name”*
update_allowed=”true” | “false” />
	Specifies a PTF property name and whether updates are allowed or blocked. ‘update_allowed’ defaults to “false” if not specified.

<part-class name=”*class_name*”> : </part-class>	Delineates a class-specific property processing specification which overrides a global property specification.
<key-properties> : </key-properties>	Specifies key property processing. Settings can be overridden by a part-class specification.

<property
name="property_name"
update_allowed="true” | “false" />
	Specifies a PTF property name and whether updates are allowed or blocked. ‘update_allowed’ defaults to the value specified by ‘def_update_allowed’ or to “false” if not specified.

<injected-properties def_update_allowed="true" | “false”> : <injected-properties>	Specifies injected property processing. Settings can be overridden by a part-class specification. Optional ‘def_update_allowed’ property can be used to set the default ‘update_allowed’ value.

<property
name="property_name"
update_allowed="true” | “false" />
	Specifies a PTF property name and whether updates are allowed or blocked. ‘update_allowed’ defaults to the value specified by ‘def_update_allowed’ or to “false” if not specified.

PTF Update Template Properties	
Tag	**Description**
Extra Part Generation Scheme	
<extra-parts enabled=”true” | “false”> : </extra-parts>	Initiates the extra parts specification section.
<output-dir>*directory_path*</output-dir>	Specifies the output directory for the PTF Extra Parts file.
<properties-order> : </properties-order>	Delineates the property ordering specification for the Extra Parts file.
<key-properties> : </key-properties>	Delineates the key property ordering specification for the Extra Parts file.

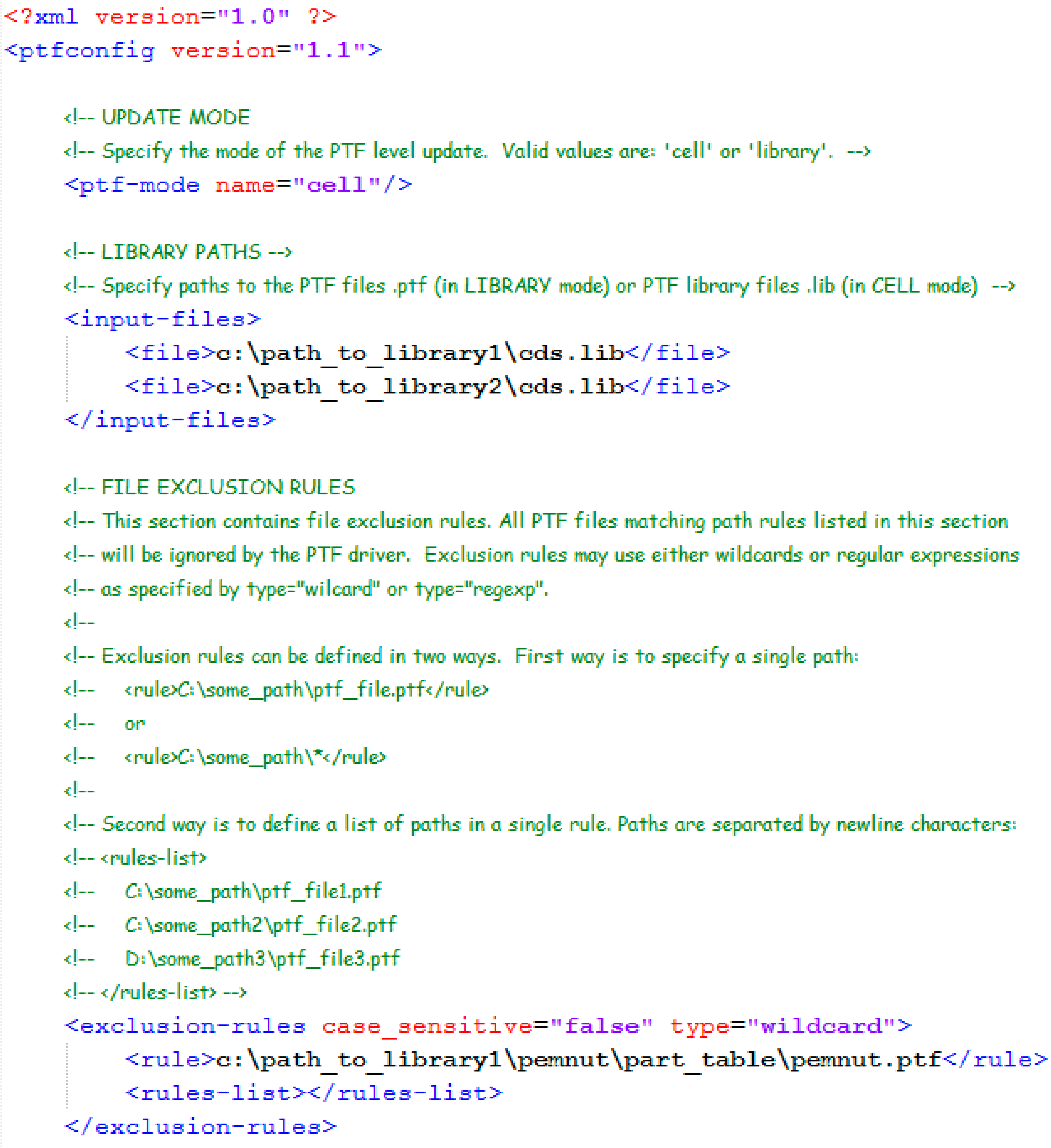
<property
name="*property_name*"
unknown="true” | “false" />
	Specifies a key property name. Set 'unknown' to “true” (the default value) for properties that don't exist in PLM. The property will be included in the PTF Extra Parts file with value set to 'UNKNOWN'.

<injected-properties> : </injected-properties>	Delineates the injected property ordering specification for the Extra Parts file.

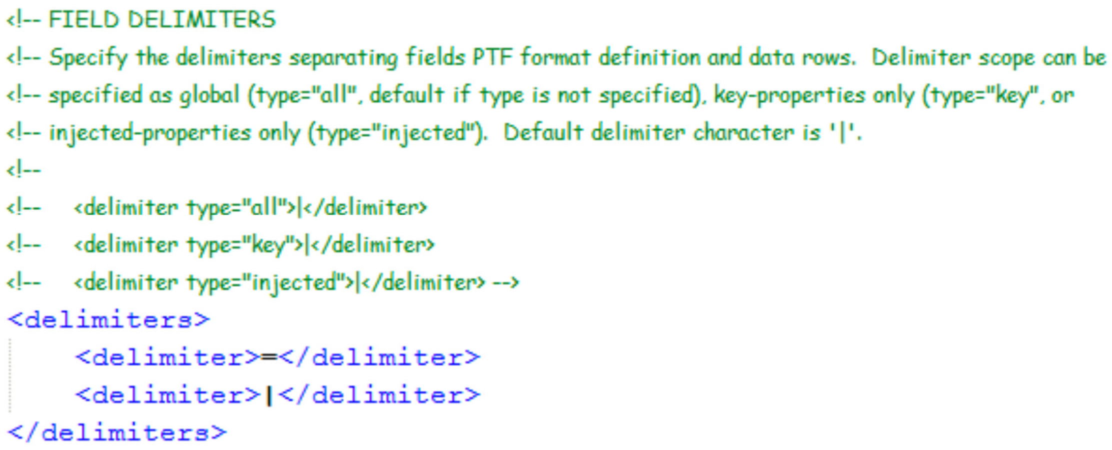
<property
name="*property_name*"
unknown="true” | “false"
value=”*value*” />
	Specifies an injected property name. Set 'unknown' to “true” (the default value) for properties that don't exist in PLM. The property will be included in the PTF Extra Parts file with value set to “*value*” or to ‘UNKOWN” if value is not specified.

Example Update Template

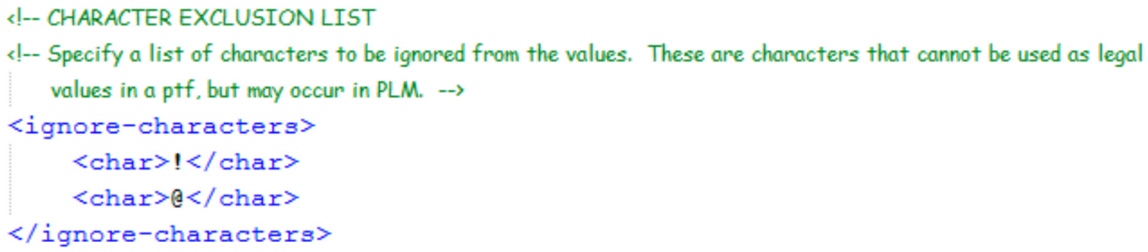
A sample Update Template is shown on the next four pages.



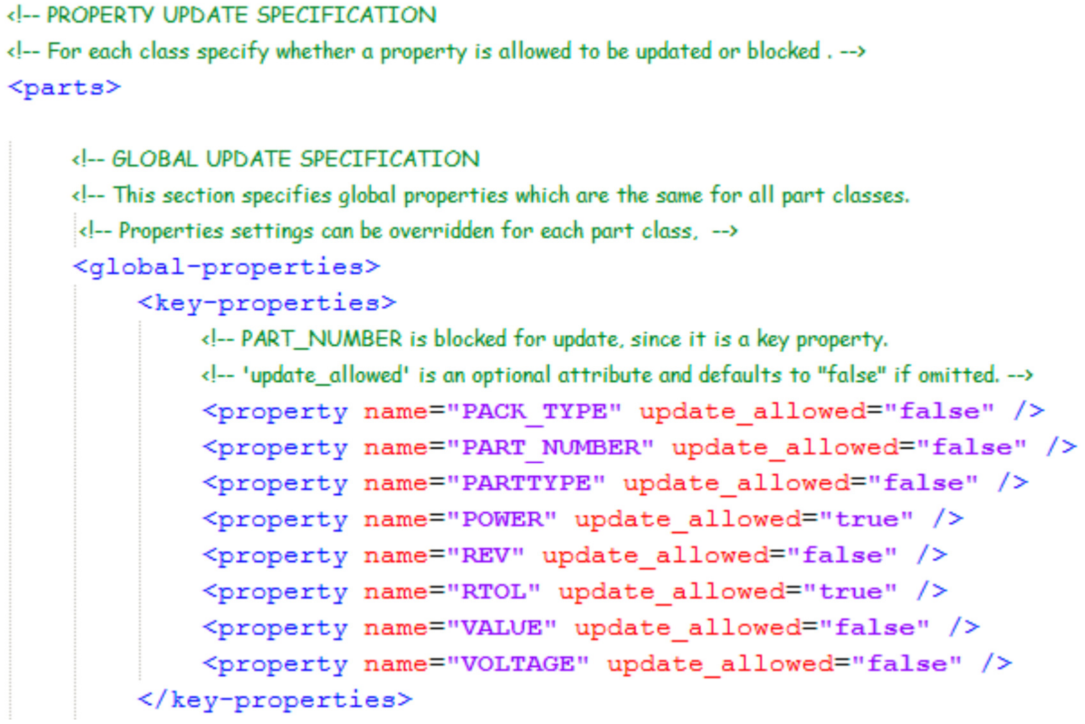
```
<?xml version="1.0" ?>
<ptfconfig version="1.1">

    <!-- UPDATE MODE
    <!-- Specify the mode of the PTF level update.  Valid values are: 'cell' or 'library'.  -->
    <ptf-mode name="cell"/>

    <!-- LIBRARY PATHS -->
    <!-- Specify paths to the PTF files .ptf (in LIBRARY mode) or PTF library files .lib (in CELL mode)  -->
    <input-files>
        <file>c:\path_to_library1\cds.lib</file>
        <file>c:\path_to_library2\cds.lib</file>
    </input-files>

    <!-- FILE EXCLUSION RULES
    <!-- This section contains file exclusion rules. All PTF files matching path rules listed in this section
    <!-- will be ignored by the PTF driver.  Exclusion rules may use either wildcards or regular expressions
    <!-- as specified by type="wilcard" or type="regexp".
    <!--
    <!-- Exclusion rules can be defined in two ways.  First way is to specify a single path:
    <!--    <rule>C:\some_path\ptf_file.ptf</rule>
    <!--    or
    <!--    <rule>C:\some_path\*</rule>
    <!--
    <!-- Second way is to define a list of paths in a single rule. Paths are separated by newline characters:
    <!-- <rules-list>
    <!--    C:\some_path\ptf_file1.ptf
    <!--    C:\some_path2\ptf_file2.ptf
    <!--    D:\some_path3\ptf_file3.ptf
    <!-- </rules-list> -->
    <exclusion-rules case_sensitive="false" type="wildcard">
        <rule>c:\path_to_library1\pemnut\part_table\pemnut.ptf</rule>
        <rules-list></rules-list>
    </exclusion-rules>
```



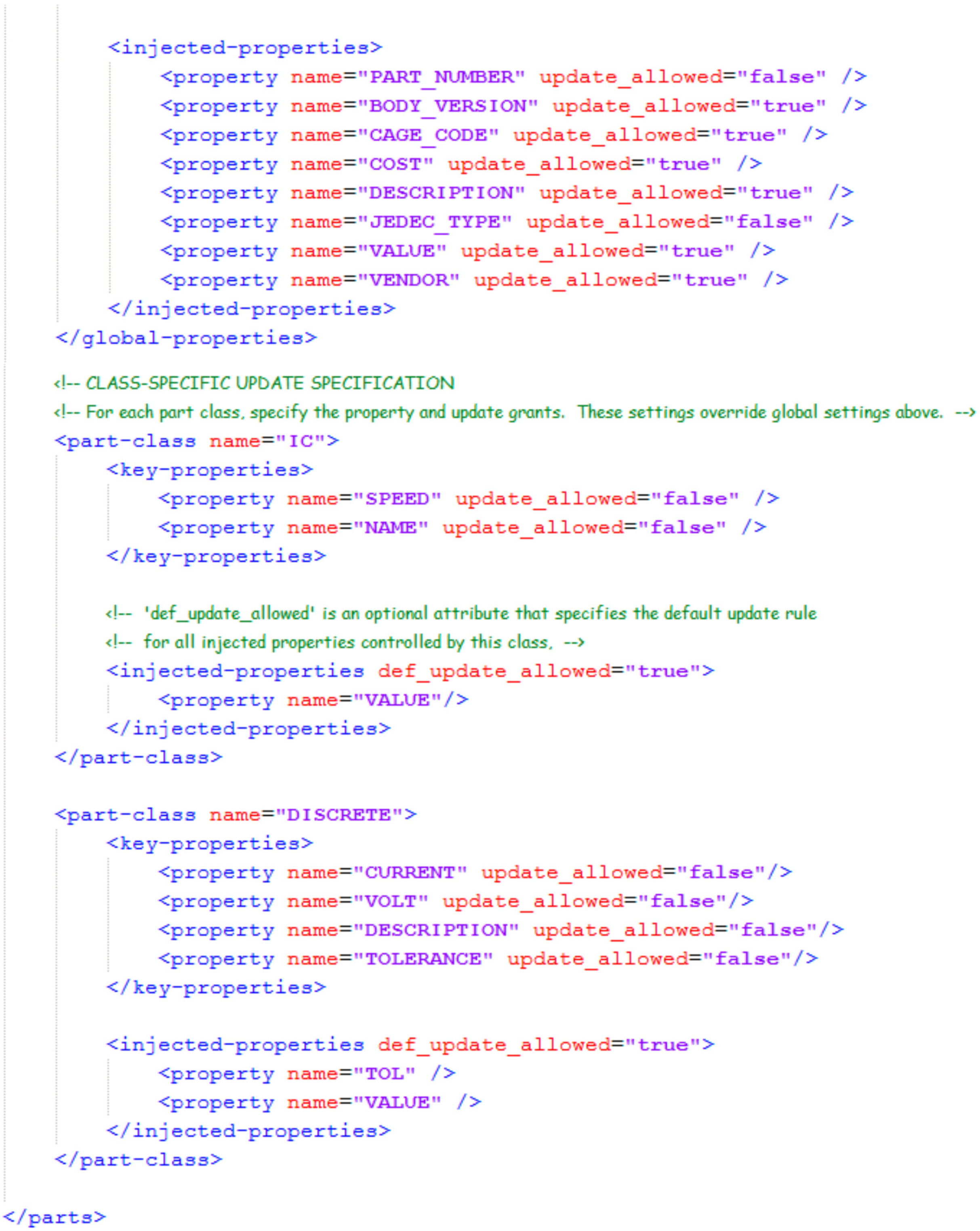
```
<!-- FIELD DELIMITERS
<!-- Specify the delimiters separating fields PTF format definition and data rows.  Delimiter scope can be
<!-- specified as global (type="all", default if type is not specified), key-properties only (type="key", or
<!-- injected-properties only (type="injected").  Default delimiter character is '|'.
<!--
<!--    <delimiter type="all">|</delimiter>
<!--    <delimiter type="key">|</delimiter>
<!--    <delimiter type="injected">|</delimiter> -->
<delimiters>
    <delimiter>=</delimiter>
    <delimiter>|</delimiter>
</delimiters>
```



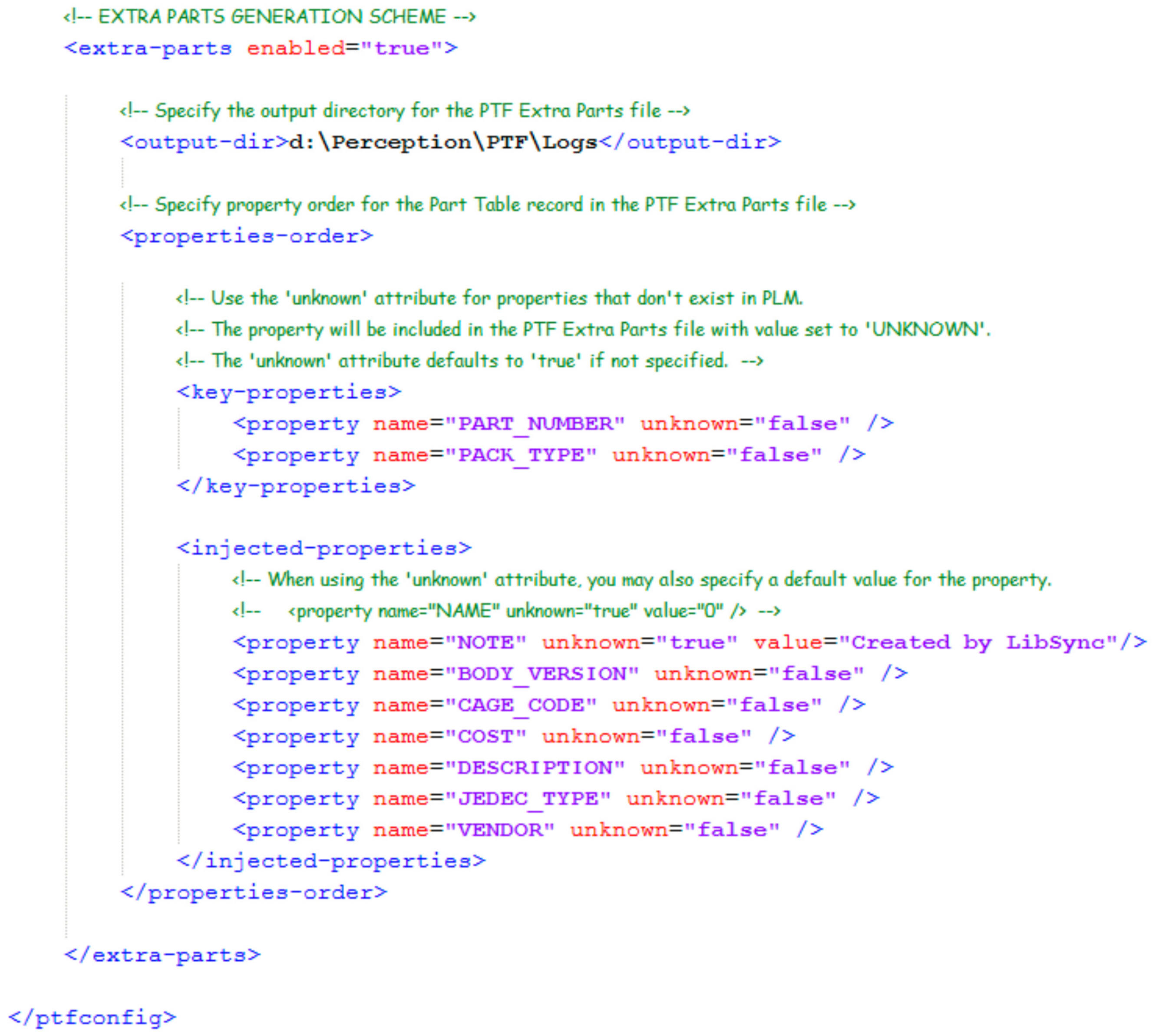
```
<!-- CHARACTER EXCLUSION LIST
<!-- Specify a list of characters to be ignored from the values.  These are characters that cannot be used as legal
     values in a ptf, but may occur in PLM.  -->
<ignore-characters>
    <char>!</char>
    <char>@</char>
</ignore-characters>
```



```
<!-- PROPERTY UPDATE SPECIFICATION
<!-- For each class specify whether a property is allowed to be updated or blocked . -->
<parts>

    <!-- GLOBAL UPDATE SPECIFICATION
    <!-- This section specifies global properties which are the same for all part classes.
    <!-- Properties settings can be overridden for each part class,  -->
    <global-properties>
        <key-properties>
            <!-- PART_NUMBER is blocked for update, since it is a key property.
            <!-- 'update_allowed' is an optional attribute and defaults to "false" if omitted. -->
            <property name="PACK_TYPE" update_allowed="false" />
            <property name="PART_NUMBER" update_allowed="false" />
            <property name="PARTTYPE" update_allowed="false" />
            <property name="POWER" update_allowed="true" />
            <property name="REV" update_allowed="false" />
            <property name="RTOL" update_allowed="true" />
            <property name="VALUE" update_allowed="false" />
            <property name="VOLTAGE" update_allowed="false" />
        </key-properties>
```



```
        <injected-properties>
            <property name="PART_NUMBER" update_allowed="false" />
            <property name="BODY_VERSION" update_allowed="true" />
            <property name="CAGE_CODE" update_allowed="true" />
            <property name="COST" update_allowed="true" />
            <property name="DESCRIPTION" update_allowed="true" />
            <property name="JEDEC_TYPE" update_allowed="false" />
            <property name="VALUE" update_allowed="true" />
            <property name="VENDOR" update_allowed="true" />
        </injected-properties>
    </global-properties>

    <!-- CLASS-SPECIFIC UPDATE SPECIFICATION
    <!-- For each part class, specify the property and update grants.  These settings override global settings above.  -->
    <part-class name="IC">
        <key-properties>
            <property name="SPEED" update_allowed="false" />
            <property name="NAME" update_allowed="false" />
        </key-properties>

        <!--  'def_update_allowed' is an optional attribute that specifies the default update rule
        <!--  for all injected properties controlled by this class.  -->
        <injected-properties def_update_allowed="true">
            <property name="VALUE"/>
        </injected-properties>
    </part-class>

    <part-class name="DISCRETE">
        <key-properties>
            <property name="CURRENT" update_allowed="false"/>
            <property name="VOLT" update_allowed="false"/>
            <property name="DESCRIPTION" update_allowed="false"/>
            <property name="TOLERANCE" update_allowed="false"/>
        </key-properties>

        <injected-properties def_update_allowed="true">
            <property name="TOL" />
            <property name="VALUE" />
        </injected-properties>
    </part-class>

</parts>
```



```
    <!-- EXTRA PARTS GENERATION SCHEME -->
    <extra-parts enabled="true">

        <!-- Specify the output directory for the PTF Extra Parts file -->
        <output-dir>d:\Perception\PTF\Logs</output-dir>

        <!-- Specify property order for the Part Table record in the PTF Extra Parts file -->
        <properties-order>

            <!-- Use the 'unknown' attribute for properties that don't exist in PLM.
            <!-- The property will be included in the PTF Extra Parts file with value set to 'UNKNOWN'.
            <!-- The 'unknown' attribute defaults to 'true' if not specified.  -->
            <key-properties>
                <property name="PART_NUMBER" unknown="false" />
                <property name="PACK_TYPE" unknown="false" />
            </key-properties>

            <injected-properties>
                <!-- When using the 'unknown' attribute, you may also specify a default value for the property.
                <!--    <property name="NAME" unknown="true" value="0" /> -->
                <property name="NOTE" unknown="true" value="Created by LibSync"/>
                <property name="BODY_VERSION" unknown="false" />
                <property name="CAGE_CODE" unknown="false" />
                <property name="COST" unknown="false" />
                <property name="DESCRIPTION" unknown="false" />
                <property name="JEDEC_TYPE" unknown="false" />
                <property name="VENDOR" unknown="false" />
            </injected-properties>
        </properties-order>

    </extra-parts>

</ptfconfig>
```

PTF Command Line Interface

Since PTF processing is typically implemented through command scripts, LibSync provides an easy to use command line interface (CLI) for PTF synchronization. LibSync's standard mapping file (.psm) editor is first used to configure the login to the PLM system, set the synchronization schedule, create the property mappings, and select the PTF update mode. Once that is complete, the CLI is used to perform PTF synchronization as desired.

The following tables summarize the PTF commands. Command options are indented.

LibSync PTF CLI Commands	
Command / Option	**Description**
-cleanlib *<update_template_file>*	Deletes all but the latest PTF back-up (.bak) file from the file system.
-create *<update_template_file>*	Creates a PTF Update Template from the selected base library.
-extra < *extra_parts_file>*	**Optional:** Specifies the pathname to the extra parts file.
-files *<ptf_or_lib_file_list>*	**Required:** Used in conjunction with the *–create* command to specify a comma-delimited list of pathnames to Cadence *.ptf* or *.lib* files.
-item_number *<pn_col_number>*	**Optional:** Specifies the PTF file part number column name if other than the default name of "PART_NUMBER'. All PTF files processed MUST use the same part number column name.
-mode *<cell | library>*	**Required:** Used in conjunction with the *–create* command to specify whether to perform CELL or LIBRARY updates.
-help	Displays a description of the PTF CLI command options.
-restore *<logfile>*	Uses the specified logfile to restore a PTF file set to its state prior to synchronization.
-sync *<psm_file_list>*	Initiates the PTF synchronization process using the list of comma-delimited .psm files.
-bg	**Optional:** Performs synchronization in background
-blocksize *<N>*	**Optional:** Specifies the number of parts to process before pausing.
-logfile *<logfile>*	**Optional:** Specifies the pathname to the master log file.
-sim	**Optional:** Simulation mode: foregoes PTF property updates and saves the update information in the master log file.

NOTE: The LibSync PTF interface is pre-configured to use a Java maximum heap size of 128 MB. Depending upon the size of the part classes being synchronized, you may encounter Java "Out of memory" errors. Should this occur, you may increase the maximum heap size by editing the MAX_HEAP_SIZE parameter in:

<install_path>\EDAConnect-LibSync\EDAConnect-CLILibSync.bat

► To synchronize PTF files for the first time

1. Launch LibSync, if not already open:

   Start→Programs→Perception Software→EDAConnect-LibSync→LibSync

2. Create a PTF Update Template:

   A. In Windows, open a command window.

   B. Navigate to the installation folder for EDAConnect-LibSync.

   C. Execute the CLI command to create a new Update Template:

      clilibsync.bat -ptf -create *<update_template_file>* -extra *<extra_parts_file>* -files *<ptf_or_lib_file_list>* -mode cell

   D. Review the Update Template.

   E. Use a text editor to make any desired changes in the Update Template.

   F. Save the Update Template.

3. Create a mapping file:

   A. Launch LibSync.

   B. Select **File→New→Mapping File** to open the mapping file editor.

   C. Complete the appropriate sections of the mapping file.

   D. Save the mapping file: *<psm_file>*

4. Perform the synchronization:

   A. In Windows, open a command window.

   B. Navigate to the installation folder for EDAConnect-LibSync.

   C. Execute the CLI command to sync the PTF files:

      clilibsync.bat -ptf -sync *<psm_file>* -logfile *<logfile>*

► **To perform a PTF synchronization in background mode**

1. In Windows, open a command window.
2. Navigate to the installation folder for EDAConnect-LibSync.
3. Execute the CLI command to sync the PTF files:

   clilibsync.bat -ptf -sync <*psm_file*> -bg -logfile <*logfile*>

► **To restore PTF files after a synchronization**

1. In Windows, open a command window.
2. Navigate to the installation folder for EDAConnect-LibSync.
3. Execute the CLI command to restore the PTF files:

   clilibsync.bat -ptf -restore <*logfile*>

► **To cleanse the file system of PTF back-up files**

1. In Windows, open a command window.
2. Navigate to the installation folder for EDAConnect-LibSync.
3. Execute the CLI command to remove all but the latest PTF back-up file: clilibsync.bat -ptf -cleanlib <*update_template_file*>

Problem Reporting

If it appears that LibSync is not operating as expected and you require technical support assistance, you will need to report the problem to Perception Software. You may submit problem reports or product enhancement suggestions by logging to your customer account at www.perceptionsoftware.com and then clicking the **Submit Issue** button in the navigation panel.

Please provide as much detail as possible about the error.:

1. Description of the problem
2. Steps to reproduce
3. EDAConnect LibSync Version and Build ID
4. Copy of Plug-in Details report (if appropriate) 5. Copy of the Configuration Details report
6. Copy of the Perception log file:

   <*install_path*>\EDAConnect-LibSync\Logs\perception.log

7. Screenshots and other additional files/information as appropriate

A Customer Support representative will review the issue and contact you regarding the course of action. If you need to speak directly to a Customer Support representative, you may dial: 888-828-1444 x118.

> **NOTE**: Before submitting a Problem Report, please consult the Known Issues and Workarounds on page 90.

Information Gathering

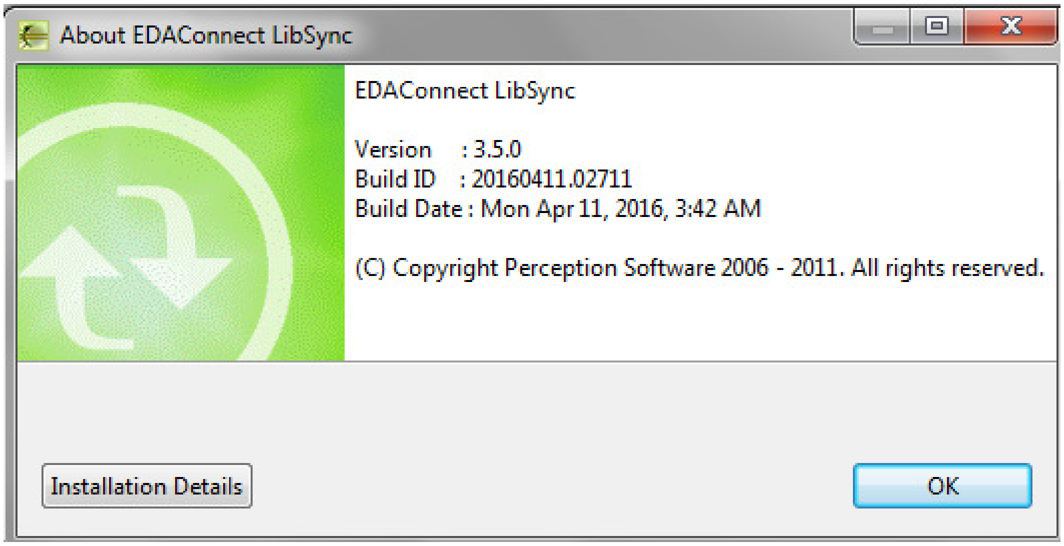
Providing the right information in a problem report is important. The following sections describes where to find relative information.

Version Number and Build ID

► **To obtain the EDAConnect LibSync Version and Build**

ID 1. Click **Help→About EDAConnect LibSync**.

2. Copy/paste the Version number and Build ID into your Problem Report email.



About EDAConnect LibSync

EDAConnect LibSync

Version : 3.5.0
Build ID : 20160411.02711
Build Date : Mon Apr 11, 2016, 3:42 AM

(C) Copyright Perception Software 2006 - 2011. All rights reserved.

Plug-in Details

This Plug-ins Details list may be useful if a recent upgrade or patch was made to the installation.

► **To obtain a list of EDAConnect LibSync plug-ins** 1. Click the **Plug-in Details** button.

2. Use **Ctrl-Alt-Print Screen** to capture screenshots of the Plug-in Details list and attach the screenshots to your Problem Report email.

Configuration Details

When reporting an issue, it is helpful to include the system Configuration Details report. Please include this information with your Problem Report.

► **To obtain system configuration details** 1.

Click the **Configuration Details** button.

2. Click the **Copy to Clipboard** button and paste this information into your Problem Report email or include it as a file attachment.

Perception Log

LibSync maintains an information and error log that records transaction information and exceptions. The log file is located at:

<install_path>\EDAConnect-LibSync\logs\perception.log

► **To access the Perception log file**

1. Open ***<install_path>*\EDAConnect-LibSync\logs\perception.log** in a text editor.
2. Be sure to attach the **perception.log** file to any Problem Reports you submit.

> **NOTE**: The **perception.log** logging level for Designer scripts can be set via the **scripts.logging.level** property in the **config.ini** file. Please see the *EDAConnect LibSync Installation Guide* for more information.

Appendix A – Glossary

Connection String – A field in the Database Connection group box on the **General** tab of a PSM file. The Connection String is the Data Source Name of the Local EDA Database.

Local EDA Database – A database the holds the resultant part table of a library synchronization. The database is local to the librarian's workstation and is accessed by EDA tools via an ODBC connection.

Mapping File – An alternate name for PSM file.

Part Type – A class of parts such as capacitor, resistor, IC, switch, power supply, etc.

Part Type Mapping table – A Part Type specific attribute mapping table located on the Mappings tab of a PSM file. The Part Type Mapping table specifies the PLM to Local EDA Database attribute mappings for the corresponding Part Type.

PSM File – The PLM System Mapping file specifies all information needed for library synchronization. This includes connection information for the PLM system and Local EDA Database, the synchronization schedule, an email notification list, the Part Types to process and the PLM to EDA Database mappings. A PSM file has a file extension of ".psm".

PTF File – A Part Table File is a text-based file used in the standard Cadence ConceptHDL-Allegro flow to store part information such as part_number, footprint, symbol and manufacturers part_number, and additional information such as parametric and ERP data such as Status, Value, Tolerance, and Cost. PTF Files are used for component browsing, component instantiation (placement), packaging, and ultimately providing BOM content.

PTF Update Template - The Update Template is an XML file that governs all aspects of PTF file synchronization and is managed outside of the LibSync application. The template specifies the location of the libraries or PTF files to be processed, the PTF file delimiters, property updated permissions, and Extra Parts file formatting.

Standard Attribute Mappings table – An attribute mapping table that specifies the default PLM to EDA Database mappings for attribute that are commonly shared by multiple Part Types. Typically, these attributes will include Part Number, Revision, Part Type, Part Family, Lifecycle Phase, and so on. The Standard Attribute Mappings table is located on the General tab of a PSM file.

Appendix B – SQL Reserved Words

The following table lists all words reserved in the SQL standard.. These words MUST NOT be used for table names or column names in the Local EDA Database. Additionally, since each database application has its own set of Reserved Words, please check the documentation for your database application.

SQL-92	SQL-99	SQL-2003
ABSOLUTE	ABSOLUTE	
ACTION	ACTION	
ADD	ADD	ADD
	AFTER	
ALL	ALL	ALL
ALLOCATE	ALLOCATE	ALLOCATE
ALTER	ALTER	ALTER
AND	AND	AND
ANY	ANY	ANY
ARE	ARE	ARE
	ARRAY	ARRAY
AS	AS	AS
ASC	ASC	
	ASENSITIVE	ASENSITIVE
ASSERTION	ASSERTION	
	ASYMMETRIC	ASYMMETRIC
AT	AT	AT
	ATOMIC	ATOMIC
AUTHORIZATION	AUTHORIZATION	AUTHORIZATION
AVG		
	BEFORE	
BEGIN	BEGIN	BEGIN
BETWEEN	BETWEEN	BETWEEN
		BIGINT
	BINARY	BINARY
BIT	BIT	
BIT_LENGTH		
	BLOB	BLOB
	BOOLEAN	BOOLEAN

SQL-92	SQL-99	SQL-2003
BOTH	BOTH	BOTH
	BREADTH	
BY	BY	BY
CALL	CALL	CALL
	CALLED	CALLED
CASCADE	CASCADE	
CASCADED	CASCADED	CASCADED
CASE	CASE	CASE
CAST	CAST	CAST
CATALOG	CATALOG	
CHAR	CHAR	CHAR
CHAR_LENGTH		
CHARACTER	CHARACTER	CHARACTER
CHARACTER_LENGTH		
CHECK	CHECK	CHECK
	CLOB	CLOB
CLOSE	CLOSE	CLOSE
COALESCE		
COLLATE	COLLATE	COLLATE
COLLATION	COLLATION	
COLUMN	COLUMN	COLUMN
COMMIT	COMMIT	COMMIT
CONDITION	CONDITION	CONDITION
CONNECT	CONNECT	CONNECT
CONNECTION	CONNECTION	
CONSTRAINT	CONSTRAINT	CONSTRAINT
CONSTRAINTS	CONSTRAINTS	
	CONSTRUCTOR	
CONTAINS		
CONTINUE	CONTINUE	CONTINUE
CONVERT		
CORRESPONDING	CORRESPONDING	CORRESPONDING
COUNT		
CREATE	CREATE	CREATE
CROSS	CROSS	CROSS
	CUBE	CUBE
CURRENT	CURRENT	CURRENT

SQL-92	SQL-99	SQL-2003
CURRENT_DATE	CURRENT_DATE	CURRENT_DATE
	CURRENT_DEFAULT_TRANSFORM_GROUP	CURRENT_DEFAULT_TRANSFORM_GROUP
CURRENT_PATH	CURRENT_PATH	CURRENT_PATH
	CURRENT_ROLE	CURRENT_ROLE
CURRENT_TIME	CURRENT_TIME	CURRENT_TIME
CURRENT_TIMESTAMP	CURRENT_TIMESTAMP	CURRENT_TIMESTAMP
	CURRENT_TRANSFORM_GROUP_FOR_TYPE	CURRENT_TRANSFORM_GROUP_FOR_TYPE
CURRENT_USER	CURRENT_USER	CURRENT_USER
CURSOR	CURSOR	CURSOR
	CYCLE	CYCLE
	DATA	
DATE	DATE	DATE
DAY	DAY	DAY
DEALLOCATE	DEALLOCATE	DEALLOCATE
DEC	DEC	DEC
DECIMAL	DECIMAL	DECIMAL
DECLARE	DECLARE	DECLARE
DEFAULT	DEFAULT	DEFAULT
DEFERRABLE	DEFERRABLE	
DEFERRED	DEFERRED	
DELETE	DELETE	DELETE
	DEPTH	
	DEREF	DEREF
DESC	DESC	
DESCRIBE	DESCRIBE	DESCRIBE
DESCRIPTOR	DESCRIPTOR	
DETERMINISTIC	DETERMINISTIC	DETERMINISTIC
DIAGNOSTICS	DIAGNOSTICS	
DISCONNECT	DISCONNECT	DISCONNECT
DISTINCT	DISTINCT	DISTINCT
DO	DO	DO
DOMAIN	DOMAIN	
DOUBLE	DOUBLE	DOUBLE
DROP	DROP	DROP
	DYNAMIC	DYNAMIC

SQL-92	SQL-99	SQL-2003
	EACH	EACH
		ELEMENT
ELSE	ELSE	ELSE
ELSEIF	ELSEIF	ELSEIF
END	END	END
	EQUALS	
ESCAPE	ESCAPE	ESCAPE
EXCEPT	EXCEPT	EXCEPT
EXCEPTION	EXCEPTION	
EXEC	EXEC	EXEC
EXECUTE	EXECUTE	EXECUTE
EXISTS	EXISTS	EXISTS
EXIT	EXIT	EXIT
EXTERNAL	EXTERNAL	EXTERNAL
EXTRACT		
FALSE	FALSE	FALSE
FETCH	FETCH	FETCH
	FILTER	FILTER
FIRST	FIRST	
FLOAT	FLOAT	FLOAT
FOR	FOR	FOR
FOREIGN	FOREIGN	FOREIGN
FOUND	FOUND	
	FREE	FREE
FROM	FROM	FROM
FULL	FULL	FULL
FUNCTION	FUNCTION	FUNCTION
	GENERAL	
GET	GET	GET
GLOBAL	GLOBAL	GLOBAL
GO	GO	
GOTO	GOTO	
GRANT	GRANT	GRANT
GROUP	GROUP	GROUP
	GROUPING	GROUPING
HANDLER	HANDLER	HANDLER
HAVING	HAVING	HAVING

SQL-92	SQL-99	SQL-2003
	HOLD	HOLD
HOUR	HOUR	HOUR
IDENTITY	IDENTITY	IDENTITY
IF	IF	IF
IMMEDIATE	IMMEDIATE	IMMEDIATE
IN	IN	IN
INDICATOR	INDICATOR	INDICATOR
INITIALLY	INITIALLY	
INNER	INNER	INNER
INOUT	INOUT	INOUT
INPUT	INPUT	INPUT
INSENSITIVE	INSENSITIVE	INSENSITIVE
INSERT	INSERT	INSERT
INT	INT	INT
INTEGER	INTEGER	INTEGER
INTERSECT	INTERSECT	INTERSECT
INTERVAL	INTERVAL	INTERVAL
INTO	INTO	INTO
IS	IS	IS
ISOLATION	ISOLATION	
	ITERATE	ITERATE
JOIN	JOIN	JOIN
KEY	KEY	
LANGUAGE	LANGUAGE	LANGUAGE
	LARGE	LARGE
LAST	LAST	
	LATERAL	LATERAL
LEADING	LEADING	LEADING
LEAVE	LEAVE	LEAVE
LEFT	LEFT	LEFT
LEVEL	LEVEL	
LIKE	LIKE	LIKE
LOCAL	LOCAL	LOCAL
	LOCALTIME	LOCALTIME
	LOCALTIMESTAMP	LOCALTIMESTAMP
	LOCATOR	
LOOP	LOOP	LOOP

SQL-92	SQL-99	SQL-2003
LOWER		
	MAP	
MATCH	MATCH	MATCH
MAX		
		MEMBER
		MERGE
	METHOD	METHOD
MIN		
MINUTE	MINUTE	MINUTE
	MODIFIES	MODIFIES
MODULE	MODULE	MODULE
MONTH	MONTH	MONTH
		MULTISET
NAMES	NAMES	
NATIONAL	NATIONAL	NATIONAL
NATURAL	NATURAL	NATURAL
NCHAR	NCHAR	NCHAR
	NCLOB	NCLOB
	NEW	NEW
NEXT	NEXT	
NO	NO	NO
	NONE	NONE
NOT	NOT	NOT
NULL	NULL	NULL
NULLIF		
NUMERIC	NUMERIC	NUMERIC
	OBJECT	
OCTET_LENGTH		
OF	OF	OF
	OLD	OLD
ON	ON	ON
ONLY	ONLY	ONLY
OPEN	OPEN	OPEN
OPTION	OPTION	
OR	OR	OR
ORDER	ORDER	ORDER
	ORDINALITY	

SQL-92	SQL-99	SQL-2003
OUT	OUT	OUT
OUTER	OUTER	OUTER
OUTPUT	OUTPUT	OUTPUT
	OVER	OVER
OVERLAPS	OVERLAPS	OVERLAPS
PAD	PAD	
PARAMETER	PARAMETER	PARAMETER
PARTIAL	PARTIAL	
	PARTITION	PARTITION
PATH	PATH	
POSITION		
PRECISION	PRECISION	PRECISION
PREPARE	PREPARE	PREPARE
PRESERVE	PRESERVE	
PRIMARY	PRIMARY	PRIMARY
PRIOR	PRIOR	
PRIVILEGES	PRIVILEGES	
PROCEDURE	PROCEDURE	PROCEDURE
PUBLIC	PUBLIC	
	RANGE	RANGE
READ	READ	
	READS	READS
REAL	REAL	REAL
	RECURSIVE	RECURSIVE
	REF	REF
REFERENCES	REFERENCES	REFERENCES
	REFERENCING	REFERENCING
RELATIVE	RELATIVE	
	RELEASE	RELEASE
REPEAT	REPEAT	REPEAT
RESIGNAL	RESIGNAL	RESIGNAL
RESTRICT	RESTRICT	
	RESULT	RESULT
RETURN	RETURN	RETURN
RETURNS	RETURNS	RETURNS
REVOKE	REVOKE	REVOKE
RIGHT	RIGHT	RIGHT

SQL-92	SQL-99	SQL-2003
	ROLE	
ROLLBACK	ROLLBACK	ROLLBACK
	ROLLUP	ROLLUP
ROUTINE	ROUTINE	
	ROW	ROW
ROWS	ROWS	ROWS
	SAVEPOINT	SAVEPOINT
SCHEMA	SCHEMA	
	SCOPE	SCOPE
SCROLL	SCROLL	SCROLL
	SEARCH	SEARCH
SECOND	SECOND	SECOND
SECTION	SECTION	
SELECT	SELECT	SELECT
	SENSITIVE	SENSITIVE
SESSION	SESSION	
SESSION_USER	SESSION_USER	SESSION_USER
SET	SET	SET
	SETS	
SIGNAL	SIGNAL	SIGNAL
	SIMILAR	SIMILAR
SIZE	SIZE	
SMALLINT	SMALLINT	SMALLINT
SOME	SOME	SOME
SPACE	SPACE	
SPECIFIC	SPECIFIC	SPECIFIC
	SPECIFICTYPE	SPECIFICTYPE
SQL	SQL	SQL
SQLCODE		
SQLERROR		
SQLEXCEPTION	SQLEXCEPTION	SQLEXCEPTION
SQLSTATE	SQLSTATE	SQLSTATE
SQLWARNING	SQLWARNING	SQLWARNING
	START	START
	STATE	
	STATIC	STATIC
		SUBMULTISET

SQL-92	SQL-99	SQL-2003
SUBSTRING		
SUM		
	SYMMETRIC	SYMMETRIC
	SYSTEM	SYSTEM
SYSTEM_USER	SYSTEM_USER	SYSTEM_USER
TABLE	TABLE	TABLE
		TABLESAMPLE
TEMPORARY	TEMPORARY	
THEN	THEN	THEN
TIME	TIME	TIME
TIMESTAMP	TIMESTAMP	TIMESTAMP
TIMEZONE_HOUR	TIMEZONE_HOUR	TIMEZONE_HOUR
TIMEZONE_MINUTE	TIMEZONE_MINUTE	TIMEZONE_MINUTE
TO	TO	TO
TRAILING	TRAILING	TRAILING
TRANSACTION	TRANSACTION	
TRANSLATE		
TRANSLATION	TRANSLATION	TRANSLATION
	TREAT	TREAT
	TRIGGER	TRIGGER
TRIM		
TRUE	TRUE	TRUE
	UNDER	
UNDO	UNDO	UNDO
UNION	UNION	UNION
UNIQUE	UNIQUE	UNIQUE
UNKNOWN	UNKNOWN	UNKNOWN
	UNNEST	UNNEST
UNTIL	UNTIL	UNTIL
UPDATE	UPDATE	UPDATE
UPPER		
USAGE	USAGE	
USER	USER	USER
USING	USING	USING
VALUE	VALUE	VALUE
VALUES	VALUES	VALUES
VARCHAR	VARCHAR	VARCHAR

SQL-92	SQL-99	SQL-2003
VARYING	VARYING	VARYING
VIEW	VIEW	
WHEN	WHEN	WHEN
WHENEVER	WHENEVER	WHENEVER
WHERE	WHERE	WHERE
WHILE	WHILE	WHILE
	WINDOW	WINDOW
WITH	WITH	WITH
	WITHIN	WITHIN
	WITHOUT	WITHOUT
WORK	WORK	
WRITE	WRITE	
YEAR	YEAR	YEAR
ZONE	ZONE	