cadence®

Allegro® EDM - Frequently Asked Questions

Product Version 23.1
September 2023

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

Contents

Frequently Asked Questions (FAQs)

This document contains answers to some frequently asked questions about Allegro EDM 17.2-2016 and prior releases. Allegro EDM administrators can use the information in this document to accomplish Allegro EDM tasks. These tasks require Allegro EDM administrator rights and are not part of a librarian's or designer's routine tasks.

The document is periodically updated to list the issues that our customers face while working with Allegro EDM 17.2-2016 and prior versions.

Important

To understand the answers to the questions in this document, first familiarize yourself with the following terms:

- Allegro EDM Server - Allegro EDM uses proprietary methodology known as the Allegro EDM Server to enable various applications to access the Allegro EDM Component Database.
- Allegro EDM Component Database - an EDM database that stores components.
- Librarians - responsible for creating and maintaining component symbols and footprint libraries.
- Master Library Server - The Master Library Server is part of the Allegro EDM server and publishes librarian data, such as changes to components, libraries, and so on.
- Library client/application - The library client refers to all the applications such as Database Editor, Data Administration, Data Exchange, Library Import, Library Distribution which edit, modify, and store library data in Allegro EDM.
- Designers - schematic and layout designers
- Designer Server - The Designer Server is a replication of the Master Library Server for design purposes, and does not contain the vault. It is part of the Allegro EDM server and only design applications can access this server. Changes in the Master Library Server are reflected in Designer Servers after library distribution.
- Designer client/application - refers to Part Information Manager, Allegro Design Management, also referred to as TDO.

Questions for the current release are in the following pages.

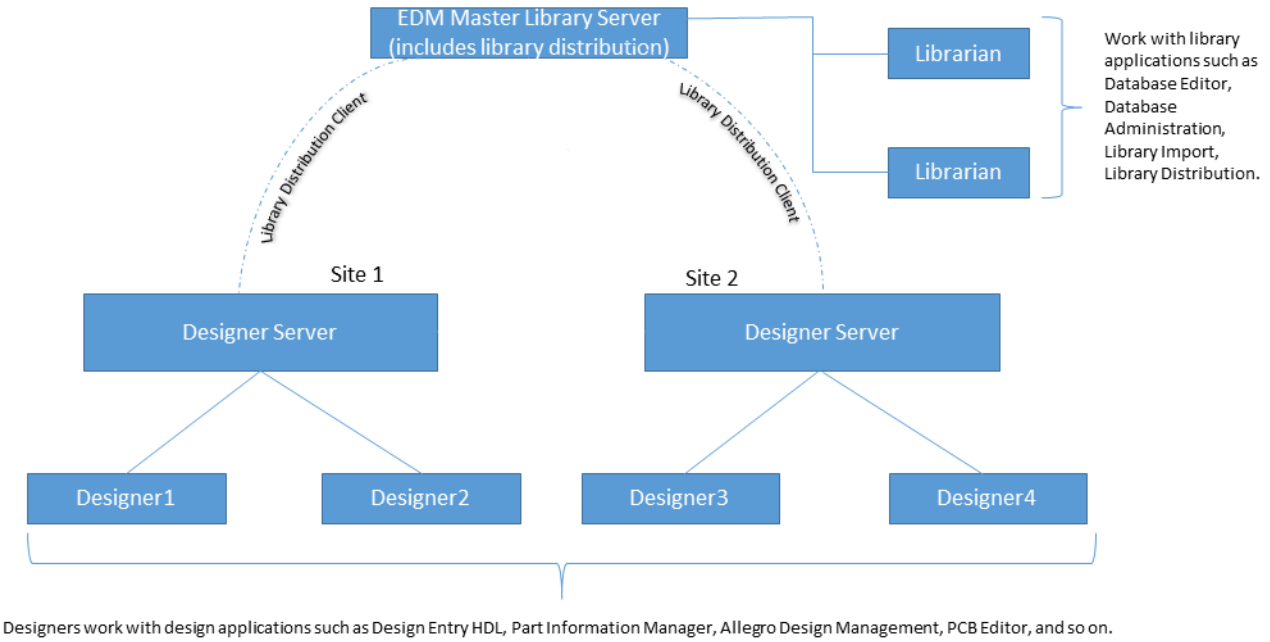
Setup

What is the recommended EDM deployment architecture?

Ideally, all Designer Servers across your sites should be on the same release and hotfix as your Master Library Server. For example, if the Master Library Server is in 17.2.031, all your Designer Servers should be on 17.2.031.

With this ideal configuration, the entire company works with the same database, data, and applications. Library data is authored in a particular EDM version, and is shared with designers who also work with the same EDM version. As a result, a database uprev is not required by either librarians or designers.

This means that the Master Library Server, the library applications (Database Editor, Database Administration, Library Import, Library Distribution, and so on), the Designer Servers, and design applications such as Part Information Manager, Allegro Design Management, Design Entry HDL, PCB Editor should all be on the same release and hotfix - for example, 17.2.031.



The ideal scenario is when everyone is on the same release and hotfix. In this figure, it is assumed that all librarians and designers have their own local EDM installation.

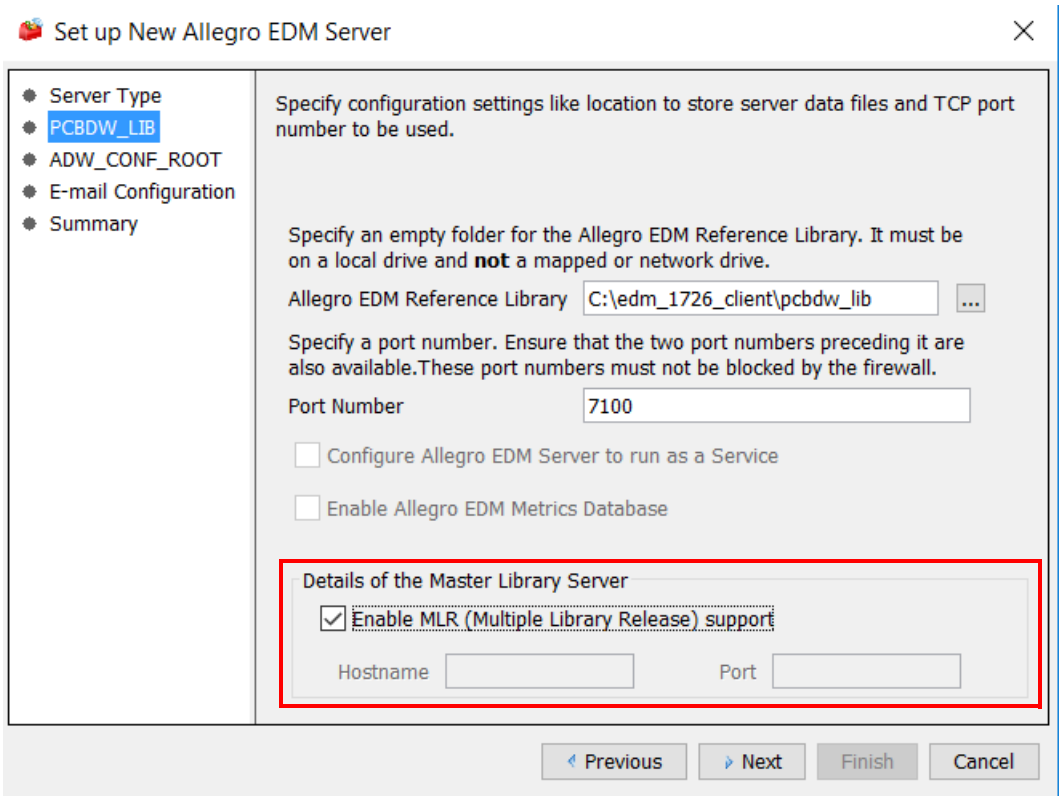
However, sometimes, designers and librarians might not be in sync, or designers might be on different releases or hotfixes. For example, one of the divisions/business units in your company wants to upgrade to the latest EDM release. This means that some Designer Servers will be on a different release or different hotfix from that of the Master Library Server. In this case, as an administrator, you will need to enable multiple library revisions (MLR).

Tip

If your company wants to work with different releases or hotfixes, it is recommended that you enable multiple library revision (MLR) support by default.

When MLR is enabled, EDM uprevs the database schema; design data is not modified. Depending on the amount of data you work with, the uprev process may take some time.

To enable MLR, first stop the Master Library Server. Then run Allegro EDM Configuration Manager and select the *Enable MLR (Multiple Library Release) support* check box.



For details on specific server scenarios, go through the case applicable to your company/site to understand the required server configuration.

- Scenario 1: Some designers at a site want to move to a higher hotfix.

Scenario 1: Some designers at a site want to move to a higher hotfix.

In this case, set the following variable on the client where Part Information Manager will be run:

```
CDS_UCBCLIENT_PATH=<installation directory>/tools/bin
```

The value of this variable, that is, the installation directory, will be that of the Master Library Server.

UNC paths are not supported with this variable. It is recommended that you run your front-end tools from a local or a mapped network drive.

After you set this variable, when you start Design Entry HDL in the Allegro EDM (online) mode or launch the standalone Part Information Manager, Part Information Manager is launched from the path defined in the variable.

Note: This variable is only supported for DE-HDL designs. It is not supported in the Allegro EDM-System Connectivity Manager (SCM) flow.

Scenario 2: An entire site, that is, all the designers at a site want to move to a higher hotfix.

In this case, update all the Designer Servers in the site to the higher hotfix. Enable multiple library revision (MLR) using Allegro EDM Configuration Manager.

Scenario 3: Librarians want to move library applications to a higher hotfix.

In this case, librarians should update the Master Library Server and library applications to the lowest version of the Designer Servers and enable multiple library revision (MLR) using Allegro EDM Configuration Manager. When the Master Library Servers are updated, EDM will uprev the database schema. Design data will not be modified.

Example of lowest version of the Designer Servers: Consider a scenario where you have three designers servers on different hotfixes such as 16.6.101, 17.2.031, 17.2.035. In this case, the Master Library Server should be 16.6.101 or lower, and the libraries should be authored in 16.6.101 or lower.

Although the Master Library Server and the libraries should be in the lowest version, you can keep upgrading the Designer Servers to the latest EDM version as required.

Scenario 4: Some designers at a site want to move from one release to another, say 16.6 to 17.2.

Because the Master Library Server should be on the lowest version of all Designer Servers, this would mean that the Master Library Server will still be on 16.6, and some designers will have moved to a 17.2 Designer Server.

In such cases, do the following:

1. Set the following variable on the client where Part Information Manager will be run:

   `CDS_UCBCLIENT_PATH=<17.2 installation directory>/tools/bin`

   Note: This variable works only the Allegro EDM component database is on 17.2 and Part Information Manager on 16.6.

   UNC paths are not supported with this variable. It is recommended that you run your front-end tools from a local or a mapped network drive.

2. Update the *<PCBDW_LIB>* setting in the *<startworkbench>* script depending on the version. For example, designers working with the 16.6 release should point to the 16.6 *<PCBDW_LIB>*.

3. For a smooth front to back flow, ensure that the path for the footprints in your PCB Editor setup points to the 16.6 or 17.2 footprints, depending on the release you are working with.

After you set this variable, when you start Design Entry HDL in the Allegro EDM (online) mode or launch the standalone Part Information Manager, Part Information Manager is launched from the path defined in the variable.

Note: When working with Part Information Manager (online mode) across releases, you cannot add database blocks. Design Entry HDL does not support the use of 17.2 design blocks in 16.6 DE-HDL designs.

Note: Only front-end models are available across releases. As the PSMPATH set in the 16.6 and 17.2 designers' environment needs to point to 16.6 and 17.2 models respectively for the front-to-back flow, Part Information Manager might not be able to show footprints when you work across releases. This is because the utility runs from 17.2 and the back-end models are from 16.6.

Scenario 5: An entire site, that is, all the designers of a site want to move from one release to another, such as from 16.6 to 17.2.

All Designer Servers and the design applications should move to the new release and be on the same hotfix. Also enable multiple library revision (MLR) using Allegro EDM Configuration Manager.

Scenario 6: Designers or librarians of a 16.6 site want to move to the next hotfix in 16.6.

In this case, whoever is moving to the next hotfix, that is, designers or librarians, should upgrade their SPB and ADW installation. To know the compatibility between the ADW and SPB hotfixes in 16.6, see Cadence Online Support.

What are the <PCBDW_LIB> folder permissions for Master Library Servers and Designer Servers?

ECAD library administrators, or anyone responsible for distributing libraries to Designer Servers should have read and write permissions for *<PCBDW_LIB>*, *<adw_conf_root>*, and EDM logs.

All librarians should have read and write permissions for *<PCBDW_LIB>* on an EDM server.

All designers should have:

- read permission for *<PCBDW_LIB>*, *<adw_conf_root>*
- read and write permission for their local <startworkbench>.bat
- read and write permission for file system-based team design

  Note: If Microsoft SharePoint is used for team design, users and permissions need to be handled in SharePoint.

Can <PCBDW_LIB> be on a different system than the EDM server?

<PCBDW_LIB> must be on a local drive and not mapped or network drive.

What are the rules for setting up Library Distribution server and client?

Master libraries developed and maintained by librarians are published to the reference library on the Master Library Server, and are fetched and utilized by client sites using `lib_dist_client`.

Administrators usually configure and run the `lib_dist` utility to automatically run the relevant library distribution processes at master and client sites to distribute the latest libraries to designers.

If you have a multi-site (separate master and client sites) environment, master site commands can be run collectively using the `lib_dist_server` command. Client-site commands can be run collectively using the `lib_dist_client` command. You can run these commands as a cron job (on Unix) or as a scheduled task (on Windows) at regular intervals.

Important

Ensure that the client site cron job or the scheduled task starts after the master site cron jobs have been completed. Ordinarily, scheduling the client site cron job 10 minutes after the master site cron job should suffice. This will ensure that the Master Library Server has created all the required library data before beginning the library distribution process.

Are UNC paths supported in the Allegro EDM environment?

The `ADW_CONF_ROOT` environment variable supports UNC paths. The following is a list of EDM environment variables for which UNC paths are not supported.

- Cadence (including EDM) installation - `PCBDW_CDSROOT`
- Allegro EDM Configuration Manager (`conf.bat`/`conf.sh`)
- Allegro EDM Server - the `adwstart`, `adwstop`, `adwservices`, and `adwserver` batch files
- EDM projects directory - `PCBDW_PROJECTS_DIR`
- Reference libraries - `PCBDW_LIB`
- Master library for multi-library support - `MLR_PCBDW_LIB`

User-defined variables, which can be used to configure Allegro EDM Flow Manager for pre-execution or post-execution, can have UNC paths. However, it is strongly recommended that you check your user-defined variables with your EDM setup.

Important

You can specify either UNC paths or mapped drives for shared files or directories when using Allegro EDM. To ensure better performance, Cadence recommends mapped drives.

If using mapped drives, be aware that the mapped drive might be disconnected after a regular interval of inactivity. Trying to access or browse to the mapped drive quickly reconnects the drive.

This behavior occurs because the systems can drop idle connections after a specified time-out period to prevent wasting server resources on unused sessions. If required, you can contact your IT division and consider modifying the default time-out period on the shared network computer.

Scalability and Growth

How do I scale up the EDM Library and Designer Servers for increased memory/storage/processing requirements?

There is no simple answer for this as it depends on many variables such as the number of relations, attributes, classifications, as well as the number of library elements and number of revisions for objects.

However, when you set up the EDM Library and Designer Servers, by default, EDM sets 2GB as the maximum memory. If EDM logs an error related to insufficient memory, you can modify the memory to a maximum of 3.5 GB. Memory errors can be checked in <*PCBDW_LIB*>/server/log/adwserver.out.

If Allegro EDM continues to log errors related to memory even with a setting of 3.5GB for libraries, contact Cadence Customer Support. For details on storage and processing requirements, see: How many servers do I need to ensure room for growth?

How many servers do I need to ensure room for growth?

The number of servers you need for your EDM setup will depend on a number of factors, such as the following:

- Does your company work from a single site or multiple sites?
- Do you intend to work across releases? For example, your X server is in 16.6 and your Y server is in 17.2.
- What is the average Designer Server's load at a site?
- Do you want redundant Designer Servers?

In general, since any changes to a server can impact multiple users who work on the EDM server, Cadence recommends that you maintain two environments: a production and a test environment.

Test Environment

A test environment can be used to test process changes that you intend in your Allegro EDM setup, such as modifying the flow steps in Flow Manager, or adding verification rules in the library or design flow. You can also use a test environment to evaluate software updates before you update the production environment.

To run a test server, you can use the PDW705 Allegro EDM Library Test Server suite.

Important

A test server cannot be run as a service.

In the test environment, you can install Allegro EDM on a single machine. In this case:

- A single instance of the master component database is installed and can serve as a library and a Designer Server combined. Or, you can have one instance each of the EDM Master Library Server and EDM Designer Server. The Designer Server instance is updated by the library distribution process.
- Designers can access library data from a Master Library Server and manage design data using a data management server such as a shared file system or SharePoint server.

  If you want to configure a SharePoint server as a data management platform, refer to the *Using Allegro Design Management with SharePoint* chapter of *Allegro Design Management User Guide*.

Recommended hardware and operating system requirements for EDM Library and Designer Servers

Test Environment	Production Environment **Note:** This assumes an average of 30 users concurrently accessing the Allegro EDM component database.
Intel® Core™ i5-2540M CPU @2.60 GHz 2.60 GHz or higher	Intel® Xeon® CPU X568 @3.33 GHz 3.33 GHz or higher

Windows 2008 Server R2 or Microsoft Windows 2012 Server (All Service Packs)	Windows 2008 Server R2 or Microsoft Windows 2012 Server (All Service Packs) **Note:** On Linux systems, the supported operating system is the same as listed for the Cadence® Allegro® and OrCAD® (including EDM1) tools in the Cadence document, *Allegro Platform System Requirements*.
RAM 8.0GB	RAM 16.0 GB
Cache Memory 3 MB	Cache Memory 6 MB
System Type 64-bit Operating System	System Type 64-bit Operating System
Minimum 50 GB free space on the hard drive (IDE, SCSI, SSD)	Minimum 80 GB free space on the hard drive (IDE, SCSI, SSD)

Master Library Server

You can only have one Master Library Server for each unique library database in your company when working with Allegro EDM. For example, if all the libraries that your designers access are in a single, central repository, it is recommended that you import and manage all these libraries in one instance of the EDM Master Library Server.

Provided library data is sourced from the central Master Library Server, other sites in your company can work with partially distributed library data using site-specific library distribution.

Designer Servers

- For every site, we recommend one Designer Server.
- If a site uses multiple versions of EDM, then for every version, we recommend an additional Designer Server. For example, if you work with 16.6 and 17.2, you need two Designer Servers - one for each version.
- For critical sites, it is recommended that you have multiple Designer Servers instead of a single server.

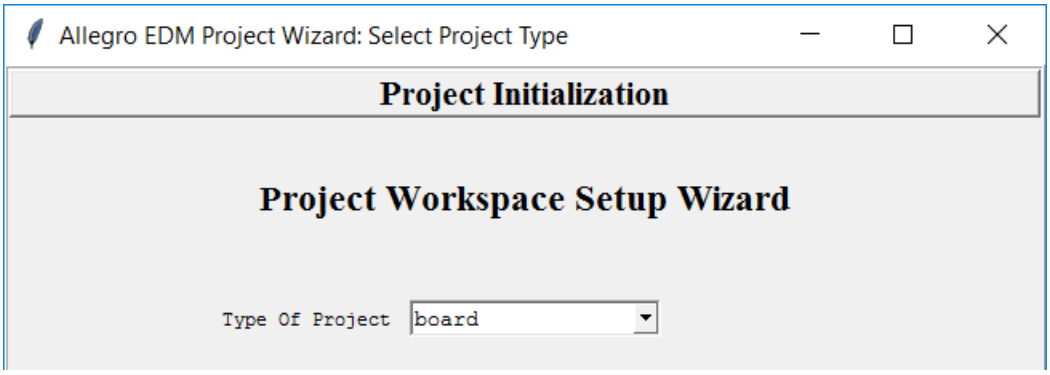
Recommended Ping Time Between Servers and Clients

- Master Library Server and Library Client - 25ms
- Designer Servers and Master Library Server - can be high, such as 200 ms. However, the network is expected to be glitch free.
- Designer Servers and the designer client - 25 to 40ms

Allegro EDM Configuration

How do I define the order of project types in the Allegro EDM Project Wizard?

Cadence provides various default project types, such as `systemdesign, highspeed, generic, libflow-condensed, library`. These project types are displayed in the Type of Project drop-down list.



To modify the flow order in this drop-down list, modify the `flow_type_order` variable in the *<ADW_CONF_ROOT>*`\`*<company>*`\`*<site>*`\design_init\15.5\design_init.ini` file.

For example, `flow_type_order=systemdesign,highspeed,generic,libflow-condensed,library`.

If you do not specify all the out-of-the-box project types in this file, the ones not specified appear at the end of the list. If you specify an incorrect project type name, that type will not appear in the drop-down list.

Version Management

Are multiple Master Library Servers supported in Allegro EDM?

No, Allegro EDM does not support multiple Master Library Servers in a company.

Why must the Master Library Server be at the lowest version of all EDM software?

When librarians want to move library applications to a higher hotfix, they should update the Master Library Server and library applications to the lowest version of the Designer Servers, since database schema can be upreved, but cannot be downreved.

For example, you have three designers servers on different hotfixes such as 16.6.101, 17.2.031, 17.2.035. In this case, the Master Library Server should be 16.6.101 or lower, and the libraries need to be authored in 16.6.101 or lower.

When should I move a Master Library Server to the latest EDM release or hotfix?

For example, my Master Library Server is currently in 16.6. When should I move the server to 17.2? Move only when all the designers servers in your company have moved to the latest EDM version.

For example, the latest EDM version is 17.2.035 and you have three designers servers on different hotfixes such as 16.6.101, 17.2.031, 17.2.035. Your Master Library Server is on 16.6.100 or lower. Move the Master Library Server to 17.2.035 only when all the Designer Servers are on 17.2.035.

Can librarians work in different releases or hotfixes?

No, all librarians should be in the same base release and the same hotfix. For example, if there are five librarians in your company, all five should be on the same release, such as 16.6 or 17.2, and should be on the same hotfix, such as 17.2.035.

The library applications that the librarians work with should be on the same hotfix as the Master Library Server.

Must the Master Library Server and Designer Server be in the same release or hotfix?

Ideally, yes, all Designer Servers across your sites should be on the same release and hotfix as your Master Library Server. For example, if the Master Library Server is in 17.2.031, all your Designer Servers should ideally be on 17.2.031.

However, if one of the divisions/business units in your company upgrades to the latest EDM release, then some Designer Servers will be on a different release or different hotfix from that of the Master Library Server. In this case, you will need to work with multiple library revisions (MLR).

For example, consider a scenario where your Master Library Server and Designer Servers are within the same release such as 17.2. Assume that you have three designers servers on different hotfixes such as 17.2.026, 17.2.031, 17.2.035, and a Master Library Server on 17.2.026 or lower. In this case, the Master Library Server should be in 17.2.026, and the libraries should be authored in 17.2.026.

In another example, assume that the division that is upgrading to the latest EDM release has five Designer Servers, all on different versions (e.g., 16.6 and 17.2) or on different hotfixes (16.6.101 and 17.2.035). In this case, all the designers servers should be served by one Master Library Server only. This Master Library Server should be on the lowest version. In our example, that would be 16.6.101.

When working with multiple library revisions, libraries should be authored in the lowest version of Allegro EDM. For example, if you have three designers servers on different hotfixes such as 16.6.101, 17.2.031, 17.2.035, and a Master Library Server on 16.6.101 or lower, the libraries should be authored in 16.6.101.

Although the libraries should be authored in the lowest version and the Master Library Server should be in the lowest version, you can keep upgrading the Designer Servers to the latest EDM version, as required.

For details on working with multiple library revisions, see the *Supporting Multiple Library Releases* in section in *Allegro EDM Configuration Guide*.

Can Designer Servers be on different releases or hotfixes?

Yes, Designer Servers can be on different releases or hotfixes.

Ideally, all Designer Servers across your sites should be on the same hotfix as your Master Library Server, and all the Designer Servers should ideally be on the same hotfix.

If you want your Designer Servers to be in different releases or hotfixes, refer to your specific scenario in: What is the recommended EDM deployment architecture?

Can my design applications, such as Design Entry HDL, PCB Editor, be on a different release from the EDM server?

Yes, design applications can be on a different release from that of the EDM server. Refer to the Scenario 4: Some designers at a site want to move from one release to another, say 16.6 to 17.2. for details.

General

What are the supported units and multipliers in Allegro EDM?

Units and multipliers in Allegro EDM use the following syntax:

```
n[Multiplier][Unit]
```

Where:

- `n` is a numerical value that indicates the magnitude of the value.
- `Multiplier` is optional and is a value from the Multipliers table.
- `Unit` is a value from the Units table. Units are case insensitive. For example, both Ohm and OHM are valid units.

Note: There must not be any space in between the magnitude, multiplier, or unit. For example, 10KOHM is valid; 10K OHM is not.

Multipliers

`K`	1000	Kilo
`k`	1000	Kilo
`T`	1.000000E12	Tera
`G`	1.000000E9	Giga
`g`	1.000000E9	Giga
`u`	1.000000E-6	Micro
`U`	1.000000E-6	Micro
`n`	1.000000E-9	Nano
`N`	1.000000E-9	Nano
`p`	1.000000E-12	Pico
`P`	1.000000E-12	Pico
`f`	1.000000E-15	Femto
`F`	1.000000E-15	Femto
`Mega`	1.000000E6	Mega

M	1.000000E6	Mega
m	1.000000E-3	Milli

Units

%	1 Percent
"	0.0254 Meters
ACRE	4046.856422400 0006 square meters
AMP	1 Ampere
AMPERE	1 Ampere
AMPS	1 Ampere
ATM	Atmospheric Unit, 101325.2002622 0624 Pa
BTU	British thermal unit, 1055.05585262 Joules
C	1 Celsius
CC	1 cubic centimeter, .000001 cubic meters
CENTIMETERS	0.01 Meters
CM	0.01 Meters
CPS	1 Cps
CYCLE	1 Cycle
D	1 Degree, 0.017453292519 943295 radians
DAY	86400 Seconds

DEG	1 Degree, 0.017453292519 943295 radians
DEGC	1 Degree Celsius
DEGF	1 Degree Fahrenheit, 0.555555555555 5556 Celsius
DEGREE	1 Degree, 0.017453292519 943295 radians
DYNE	1 Dyne, 0.00001 Kg·m/s2
F	1 Farad
FAILURE	1 Failure
FARAD	1 Farad
FEET	0.3048 Meter
FOOT	0.3048 Meter
FORTNIGHT	1209600 Seconds
FPMH	1 FPMH
FT	Feet - 0.3048 Meters
FURLONG	201.168 Meters
G	1 Gram, 0.001 Kg
GAL	1 Gallon, 3.78541 liter
GALLON	1 Gallon, 3.78541 liter
GF	1 Gram-Force, 0.00980665 newton
GRAM	0.001 Kg

H	1 Henry
HENRY	1 Henry
HORSEPOWER	1 Horsepower, 745.6987 Watts
HOUR	3600 Seconds
HP	1 Horsepower, 745.6987 Watts
HR	3600 Seconds
HZ	1 Hz
IN	0.0254 Meters
INCH	0.0254 Meters
INCHES	0.0254 Meters
INS	0.0254 Meters
J	1 Joule
JOULE	1 Joule
L	1 Liter, 0.001 cubic meter
LB	1 Pound, 0.453592 Kg
LBF	Pound force, 4.448214902093 425 Kg-m/s2
LBS	1 Pound, 0.453592 Kg
LINE	1 Line
LITRE	1 Liter, 0.001 cubic meter
M	1 Meter
METER	1 Meter
MHO	1 Mho
MHOS	1 Mho

MHZ	1000000 Hz
MICRON	0.000001 Meters
MICRONS	0.000001 Meters
MIL	0.0000254 Meters
MILE	1609.344 Meters
MILLIMETERS	0.001 Meters
MILS	0.0000254 Meters
MIN	60 Seconds
MINUTE	60 Seconds
MPH	Miles per Hour, 0.44704 meters per second
NEWTON	1 Newton
NS	1.0E-9 S
OHM	1 Ohm
OUNCE	028349523125 Kgs
OZ.CU	3.6576E-5
OZ.FL	1 fluid ounce, 29.5735296875 grams
OZF	Ounce-Force, 0278013431380 83906 Kg-m/s2
PA	1 Pascal
PASCAL	1 Pascal
PERCENT	1 Percent
POUND	0.453592 Kg

PPHH	Pound Per Hour. 0.027777777777 777776 Kg/sec
PS	1.0E-12 S
PSI	1 Pound per sq. inch
RAD	1 Radian
RADIAN	1 Radian
S	1 Second
SEC	1 Second
SECOND	1 Second
SEIMEN	1 Seimen
SMOOT	1.7018 Meters
SMOOTS	1.7018 Meters
V	1 Volt
VOLT	1 Volt
VOLTS	1 Volt
W	1 Watt
WATT	1 Watt
WATTS	1 Watt
WEBER	1 Weber

Does Part Information Manager convert between units (for example, metric and imperial)?

Yes, it does. Consider the following example: The *Height* field of Part A has a value of 50mil but the other fields for the same part are defined in metric, say 1.270mm or 1270micron. Can Part Information Manager search for values lower than 60mil in any unit? Can Part Information Manager convert between imperial and metric units?

Yes, Part Information Manager converts between imperial and metrics units but only for searches. Units are not converted in free-text search or filters.

Important

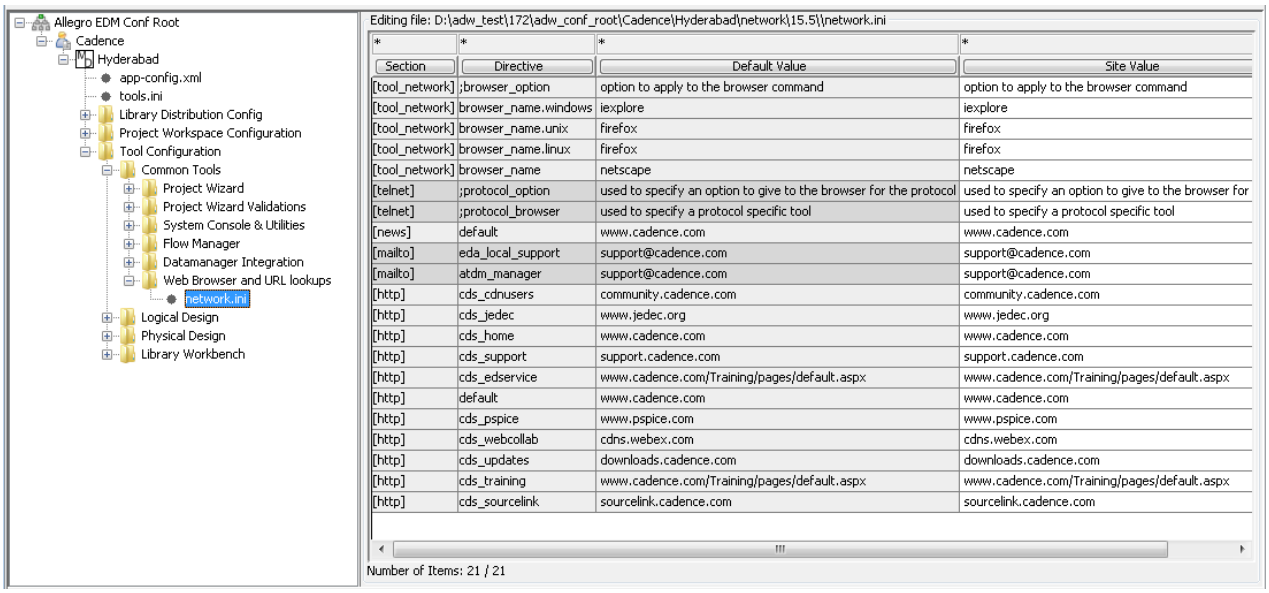
The unit ***mm*** is not supported; mils and millimeters are.

How do I configure a default browser for the Allegro EDM environment?

You can configure default browsers by doing the following:

1. Open Configuration Manager.
2. Click *Set up or Manage Company & Site*.
3. Select *Allegro EDM Conf Root – <company> – <site> – Tool Configuration – Common Tools – Web Browser and URL lookups – network.ini*.

Note that the default browser configurations are port specific. For example, the directive name that ends with `.linux` indicates that it is for Linux.



Section	Directive	Default Value	Site Value
[tool_network]	;browser_option	option to apply to the browser command	option to apply to the browser command
[tool_network]	browser_name.windows	iexplore	iexplore
[tool_network]	browser_name.unix	firefox	firefox
[tool_network]	browser_name.linux	firefox	firefox
[tool_network]	browser_name	netscape	netscape
[telnet]	;protocol_option	used to specify an option to give to the browser for the protocol	used to specify an option to give to the browser for
[telnet]	;protocol_browser	used to specify a protocol specific tool	used to specify a protocol specific tool
[news]	default	www.cadence.com	www.cadence.com
[mailto]	eda_local_support	support@cadence.com	support@cadence.com
[mailto]	atdm_manager	support@cadence.com	support@cadence.com
[http]	cds_cdnusers	community.cadence.com	community.cadence.com
[http]	cds_jedec	www.jedec.org	www.jedec.org
[http]	cds_home	www.cadence.com	www.cadence.com
[http]	cds_support	support.cadence.com	support.cadence.com
[http]	cds_edservice	www.cadence.com/Training/pages/default.aspx	www.cadence.com/Training/pages/default.aspx
[http]	default	www.cadence.com	www.cadence.com
[http]	cds_pspice	www.pspice.com	www.pspice.com
[http]	cds_webcollab	cdns.webex.com	cdns.webex.com
[http]	cds_updates	downloads.cadence.com	downloads.cadence.com
[http]	cds_training	www.cadence.com/Training/pages/default.aspx	www.cadence.com/Training/pages/default.aspx
[http]	cds_sourcelink	sourcelink.cadence.com	sourcelink.cadence.com

After you specify the browser in the *network.ini* file, you can also use the `openweb` command to check and open the required Web pages from the Allegro EDM system console.

Examples

- The following command opens the Web page defined for the *Directive* field in the `network.ini` file:

  ```
  openweb -protocol http -address <directive>
  ```

For example:

```
openweb -protocol http -address cds_home
```

- The following command opens Outlook with the *To* field populated with the following e-mail address: support@cadence.com

```
openweb -protocol mailto -address eda_local_support
```

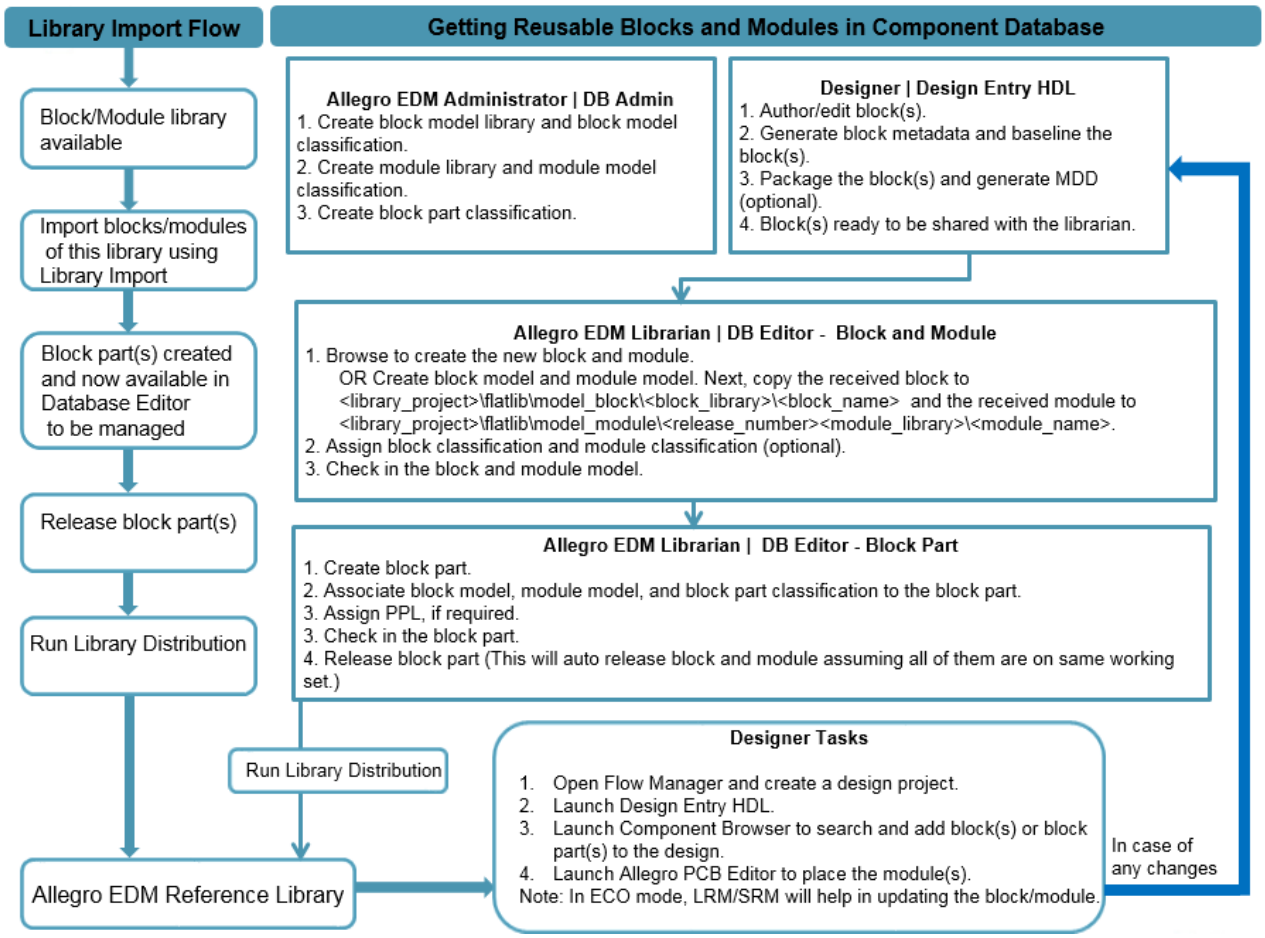
What is the block flow?

Front-end applications, such as Design Entry HDL and Allegro System Capture, support blocks in a reference library so that designers can reuse them like cells. Similarly, Allegro PCB Editor supports the reuse of partitioned designs and PCB Editor Module Definition File (.MDD) modules.

The process of turning blocks in reference libraries and partitioned designs modules into reusable blocks is referred to as the *block flow*.

Allegro EDM supports the block flow for both Library Manager as well as Data Manager. A designer defines and creates a block. The librarian acts as a facilitator who manages the block, and then releases and distributes it for use by designers.

The following diagram describes the block flow:



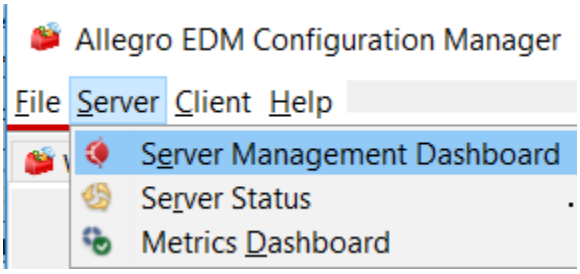
Database Administration

What is the adwadmin/adwmetrics vault and what kind of data is stored in them?

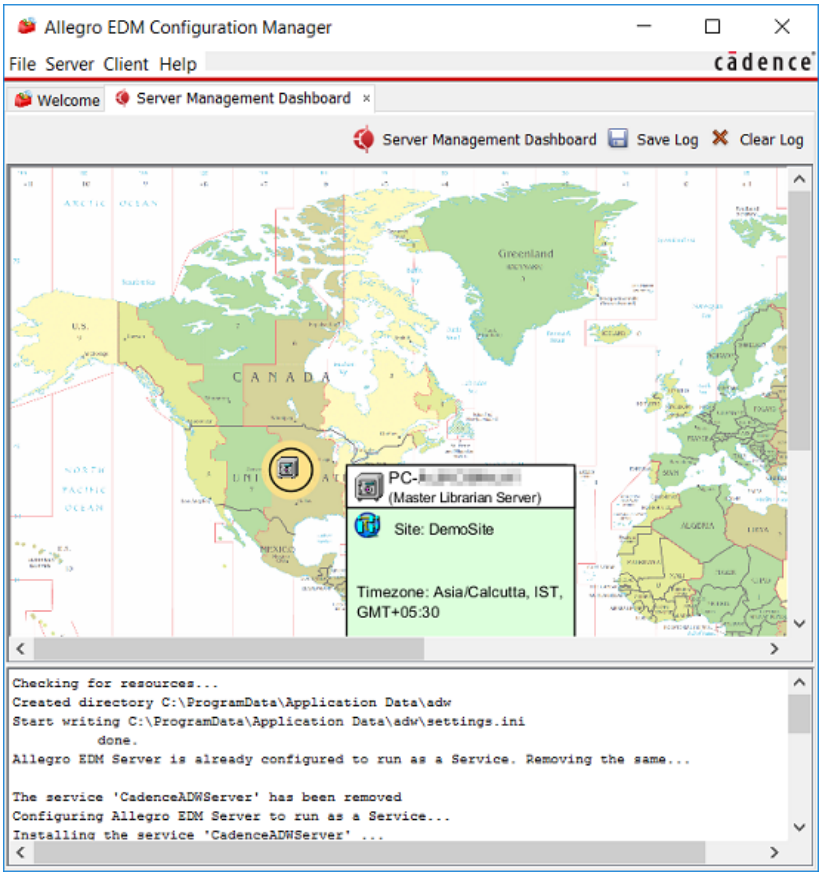
The `adwadmin` and `adwmetrics` databases are available in *<PCBDW_LIB>*`\databases`. Both folders store Server Management Dashboard information in XML format.

Server Management Dashboard

You can access the Server Management Dashboard in Allegro EDM Configuration Manager:



The Server Management Dashboard provides a high-level graphical view of the Master Library Server, all the Designer Servers that point to this Master Library Server, and the linkages between all the Designer Servers that point to a Master Library Server.



The information in the Server Management Dashboard is updated according to the values in `linked_servers_refresh_interval` and `server_status_refresh_interval` in

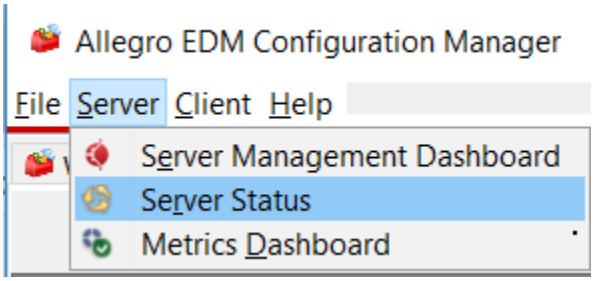
the `adw_dashboard.ini` file located in *<ADW_CONF_ROOT>*. The default values for these are 10 seconds and 1 second respectively, unless you modify them.

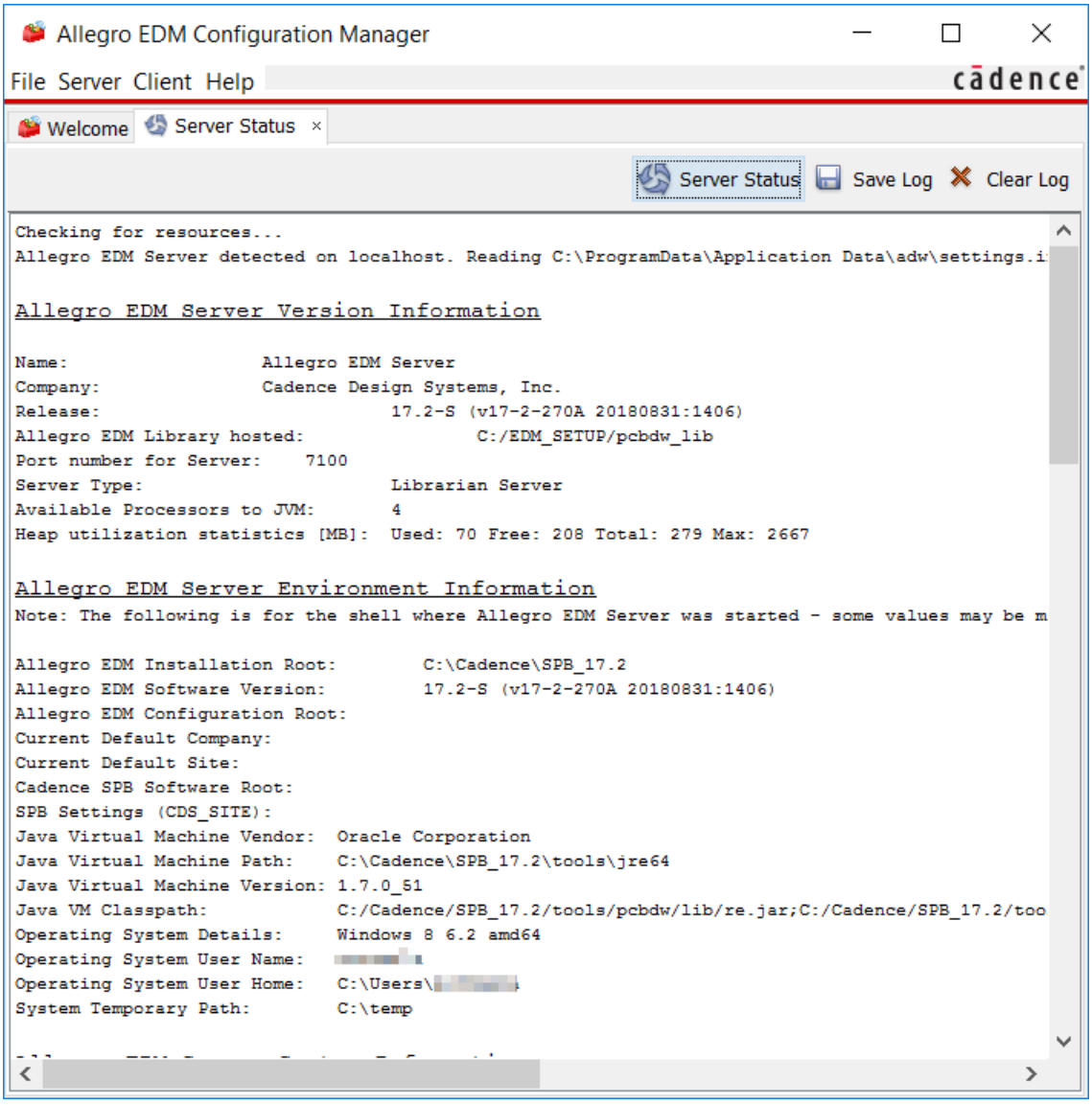
- `linked_servers_refresh_interval`: governs the refresh rate for linked server nodes
- `server_status_refresh_interval`: governs the refresh rate of the server node

adwadmin

The `adwadmin` database contains information such as the name of the company and the site that were configured during the Allegro EDM server setup, relations between various entities such as Company Site, Client, and Site Server, the master-slave relation between the Master Library Server and all the Designer Servers that point to this Master Library Server.

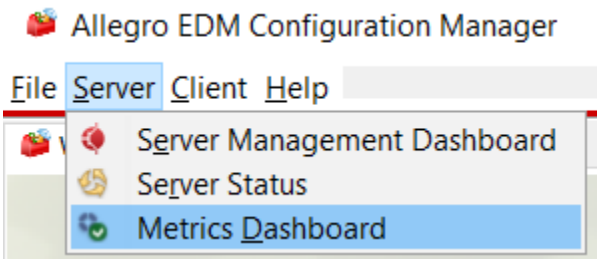
You can view the information that is maintained in the `adwadmin` database using:





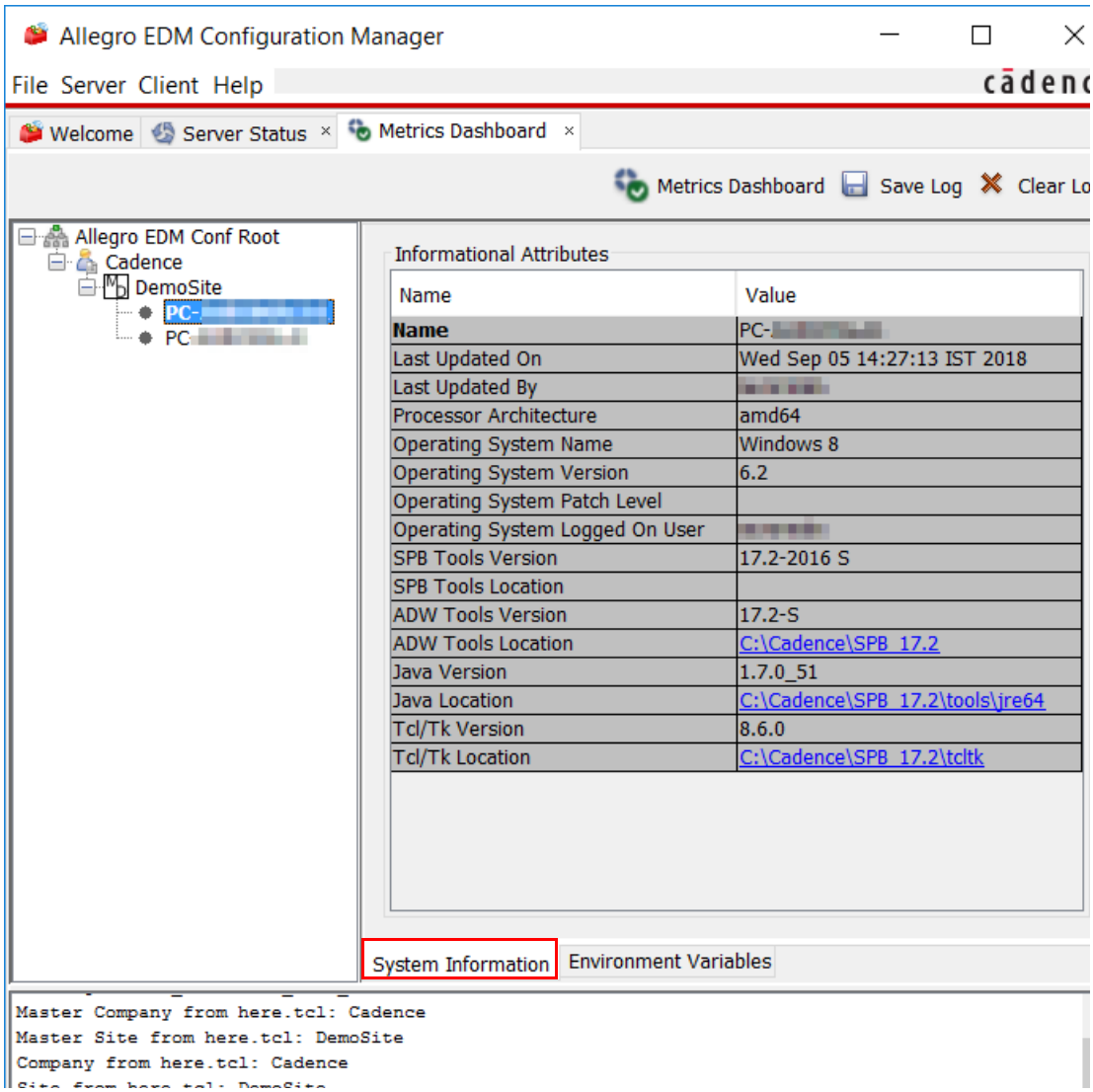
adwmetrics

The `adwmetrics` database contains information about the local computer on which Allegro EDM is being run. You can view the information that is maintained in the `adwmetrics` database using:

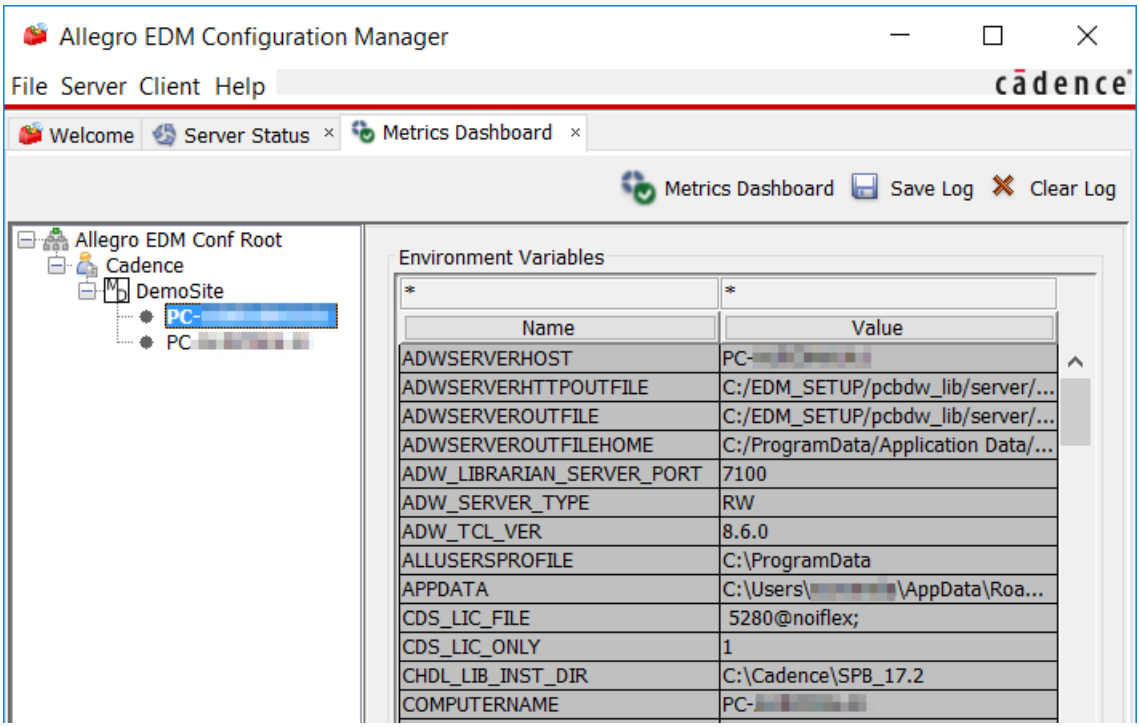


The information in the `adwmetrics` database is split into two categories:

- machine details, which can be viewed in the *System Information* tab, highlighted in the following image



- list of environment variables that have been set on the local computer on which Allegro EDM is being run



Model Related

What are models in EDM?

To understand models in EDM, let us first understand what EDM considers a part.

Parts in EDM

Off-the-shelf physical components, such as resistors, diodes, capacitors, integrated circuits, and so on, are called *parts* in Allegro EDM. From the perspective of Cadence applications, the part number in a front-end `.ptf` file (illustrated below) is defined as a part in Allegro EDM.

```
FILE_TYPE = MULTI_PHYS_TABLE;

PART 'VGA'

{=================================================================================
:PART_NUMBER     | PACK_TYPE     = JEDEC_TYPE | HEIGHT     | ROHS         | STATUS
{=================================================================================
 'CDN-CON-0015' | 'VGA'(!)      = 'DSUB15'    | '12.55MM' | 'COMPLIANT' | 'Active'

END_PART

END.
```

From this point on, the term *part* will be used to refer to off-the-shelf physical components available in the industry.

Models in EDM

Applications such as Design Entry HDL, Allegro System Capture, OrCAD Capture, each have different abstract representations of a part, depending on the way the application is structured. To ensure that Allegro EDM understands the authoring tool of a part as well as the use of the part, Allegro EDM categorizes these abstract representations as database models. Each part can be linked to one or more models. For example, EDM can categorize an electrical part as a Schematic Model-Footprint Model-Datasheet Model. A part is constructed by first creating the various models and then referencing/linking those models to define the part.

Each model in Allegro EDM contains ECAD data and metadata information that describes the model — name, description, creator, creation date, authoring tool (for example, Design Entry HDL, OrCAD Capture), application information, version, characteristics.

Allegro EDM supports the following models out of the box:

Out-of-the-Box Allegro EDM Models and Use in Allegro EDM

Model Name	Model Use	Authoring Tool	File Extension
Block	A block in front-end schematic entry applications In EDM, Block Models are used as reuse blocks.	Design Entry HDL (DE-HDL), Allegro System Capture	Not applicable
Board	Mechanical symbols can be card outlines, mechanical parts, or mounting holes.	Allegro PCB Editor	Not applicable
Capture	Combination of part aliases and packages in an OrCAD Capture library (`.olb` file) In OrCAD Capture, a single library part can be assigned aliases to represent multiple parts that vary - for example, they have different speed ratings or are made by several manufacturers - but all have a common graphic and PCB footprint. A package consists of one or more parts. Packages are classified differently depending on the type of parts in the package. For example, a package can be heterogeneous or homogeneous.	OrCAD Capture	Not applicable
Capture Standard	A part (Capture Model) without pins in OrCAD Capture, used for graphical purposes only, such as a page border, symbol origin, and so on	OrCAD Capture	Not applicable
Datasheet	Regardless of the file format (for example, PDF), the specifications sheet or document that summarizes the performance and other technical characteristics of a part		

Out-of-the-Box Allegro EDM Models and Use in Allegro EDM

Flash	Flash symbols are pads for photoplotting, using standard apertures.	Allegro PCB Editor	`.fsm`
Footprint	An Allegro PCB Editor symbol, which is a physical representation of a logical part in a schematic design (front-end symbol), such as a dual in-line package (DIP), resistor, capacitor, or edge connector.	Allegro PCB Editor	`.psm`
Format	Format symbols are page-size formats, graphics, logos, assembly/ fab notes, cross section diagrams and so on.	Allegro PCB Editor	`.osm`
Mechanical	Mechanical parts are those defined in the MECH CLASS property in front-end applications such as Design Entry HDL, Allegro System Capture ,in the `chips.prt` file. Pin information such as pin names, types, loading, and physical numbers is stored in the `chips.prt` file.	DE-HDL, Allegro System Capture	`.bsm`
Module	A module file contains a selected portion of a board (with the routes, components, vias, layers, and so on) that is saved in a manner that allows it to be placed again in its entirety on a board. A module is similar to a part in that you can place, delete, and move it multiple times with or without the logic that represents it.	Allegro PCB Editor	`.mdd`
Padstack	A geometrical description of each pin in a PCB, a padstack is a collection of a pad and holes that define the graphical representation of a pin. A padstack may range from 1 (surface mounted) to n-layers.	Allegro PCB Editor	`.pad`

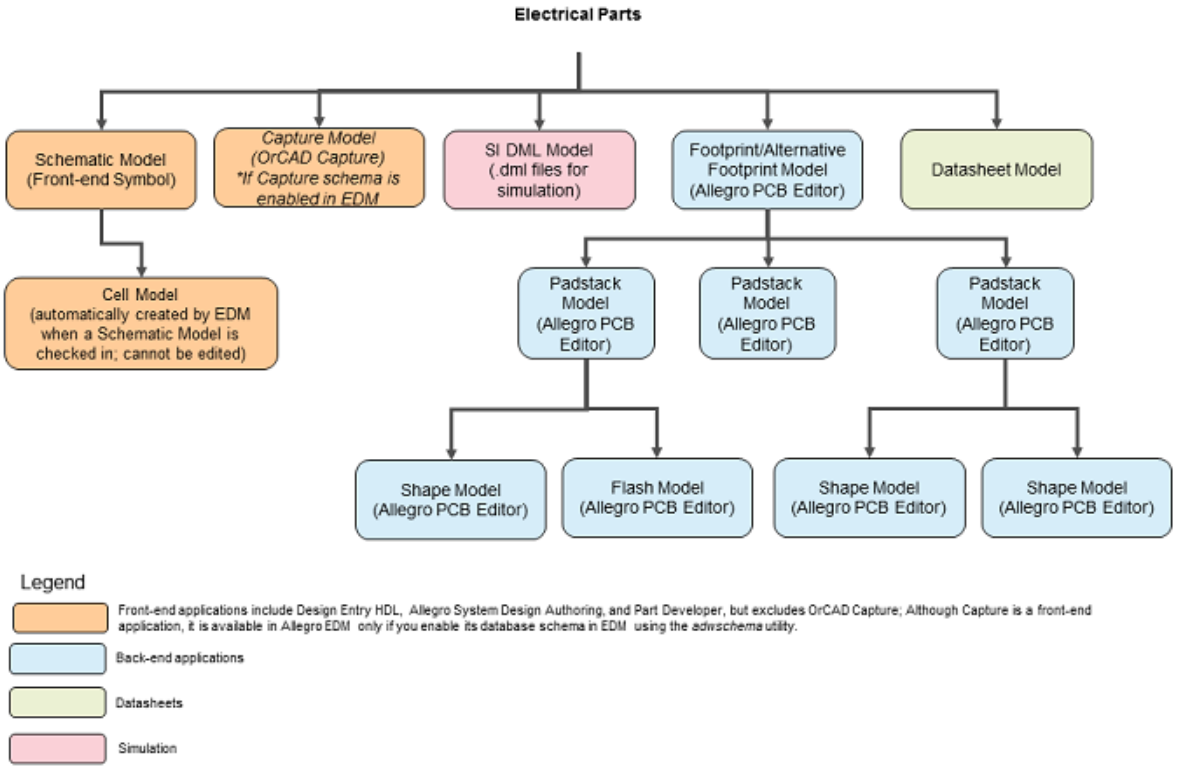
Out-of-the-Box Allegro EDM Models and Use in Allegro EDM

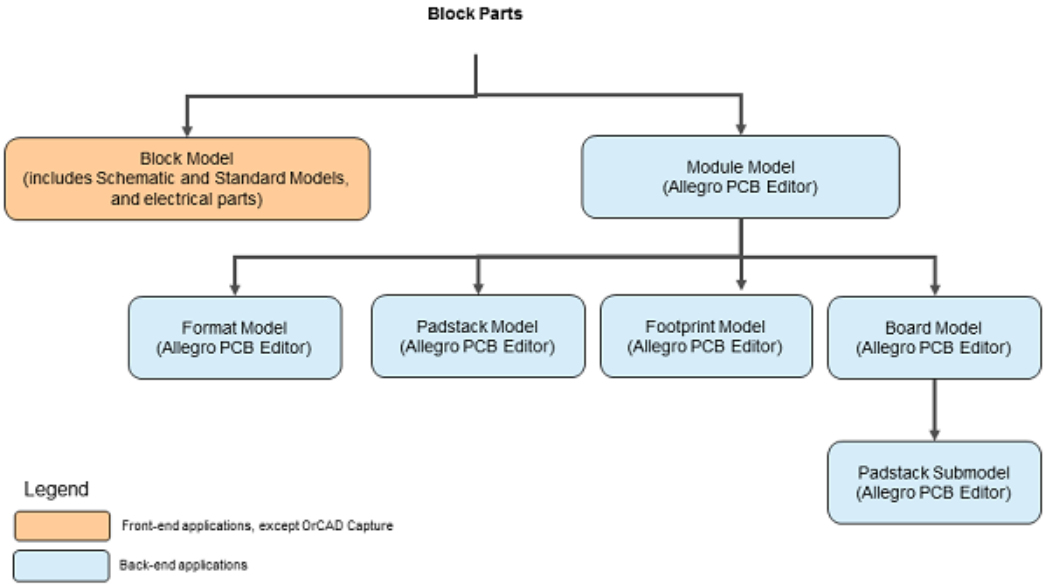
Schematic	The part name in a Part Table File (`.ptf`) file in DE-HDL is a Schematic Model in EDM. The PTF file stores packaging properties for a part in a library, such as package types, manufacturer names, part numbers, custom properties, and so on. Because each part in front-end applications is a cell (which is a collection of views, such as symbol, chips, entity, each of which describes the part in a unique way), EDM automatically creates a Cell Model when you check in a Schematic Model.	DE-HDL, Allegro System Capture	Not applicable
Shape	Shape symbols are filled polygons, closed circular shapes, and so on used for custom pads	Allegro PCB Editor	`.ssm`
SI DML	All device models stored in Device Modeling Language (DML) format. A DML file contains one or more device models written in the DML language, and is used in circuit simulation using analysis tools, such as Allegro PCB SI and SigXplorer. DML files can be translated IBIS files, library files of discrete components or of package models to represent package parasitics, or they can be several DML files grouped together	Any text editor	`.dml`

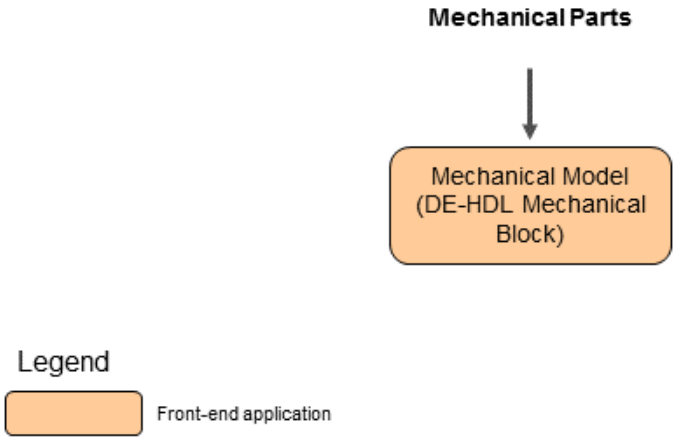
Out-of-the-Box Allegro EDM Models and Use in Allegro EDM

Standard	For Allegro EDM, a cell without a `chips.prt` file is a Standard Model in EDM. These symbols are used for graphical purposes only, such as a page border, symbol origin, and so on. A cell in DE-HDL or Allegro System Capture is a collection of views that describes an individual building block of a chip or system.	DE-HDL, Allegro System Capture	Not applicable

The links between parts and models in Allegro EDM are as follows:





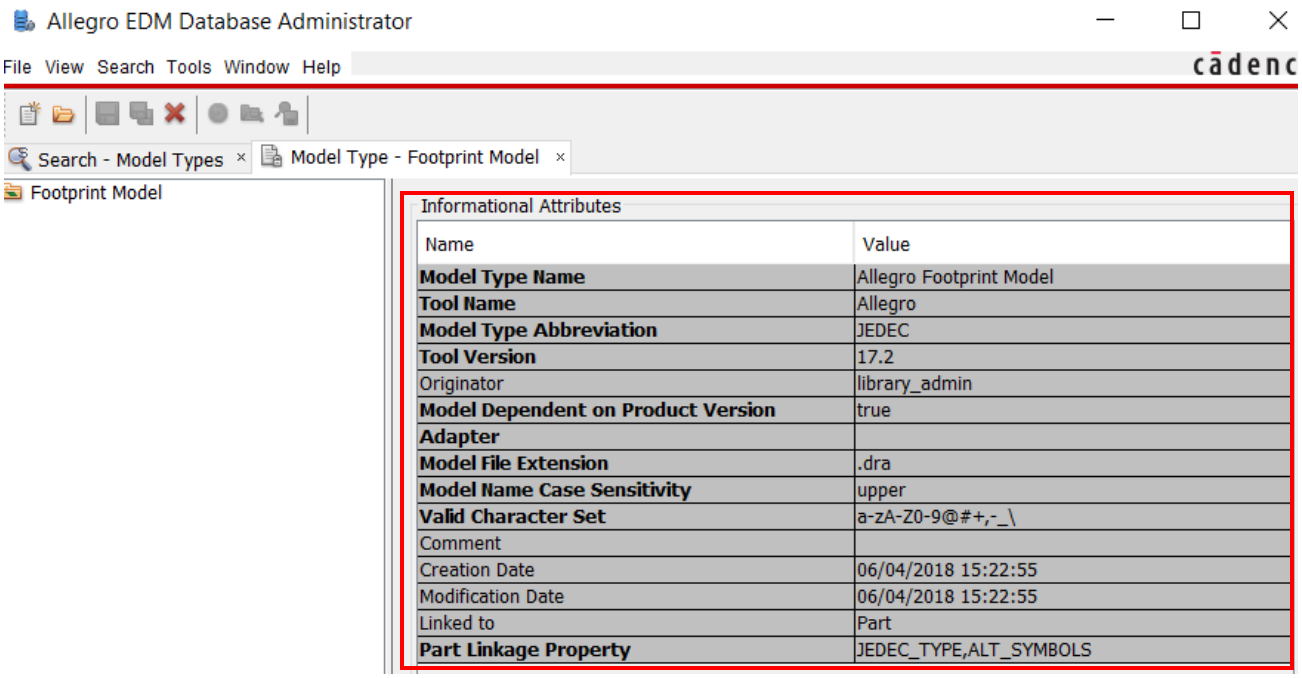


What are model types in Allegro EDM?

To understand model types, you must first understand models. See *What are models in EDM?*

Model types are objects in the Allegro EDM database, which share a set of characteristics that defines a model. Default model types cannot be modified.

Allegro EDM has a default set of model types, each of which is defined by a default set of characteristics. For example, a Footprint Model is defined by the set of characteristics illustrated in the Informational Attributes panel:



Model types are useful when searching the Allegro EDM Component Database. For example, when trying to find a footprint, as a designer, you can narrow your search to only Footprint Models.

Can I create a new model type in EDM?

Yes, you can create a new model type. You will have to create a model type only if you need to work with model types that are not provided out of the box in Allegro EDM.

For example, if you work with 3D CAD models in `.step` format and you want to manage them in Allegro EDM, you can create a new model type called STEP Model.

See also

Out-of-the-Box Allegro EDM Models and Use in Allegro EDM

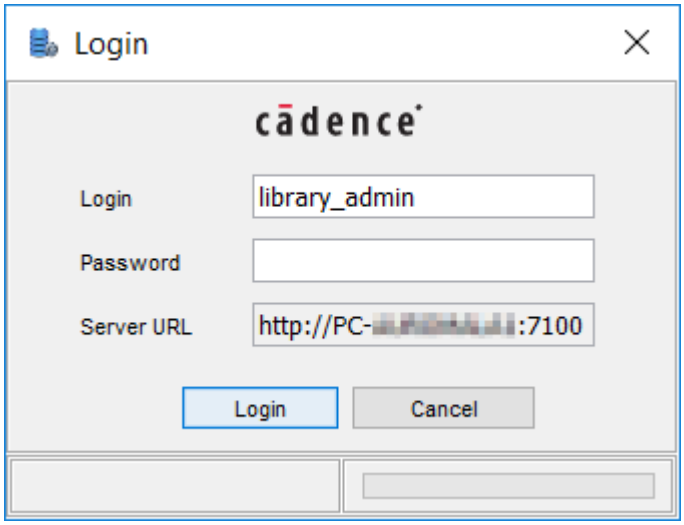
How do I create a custom model type in EDM?

You create a new model type using Database Administrator. To work with Database Administrator, you must have the ECAD Library Administrator role, as defined in Database Administrator.

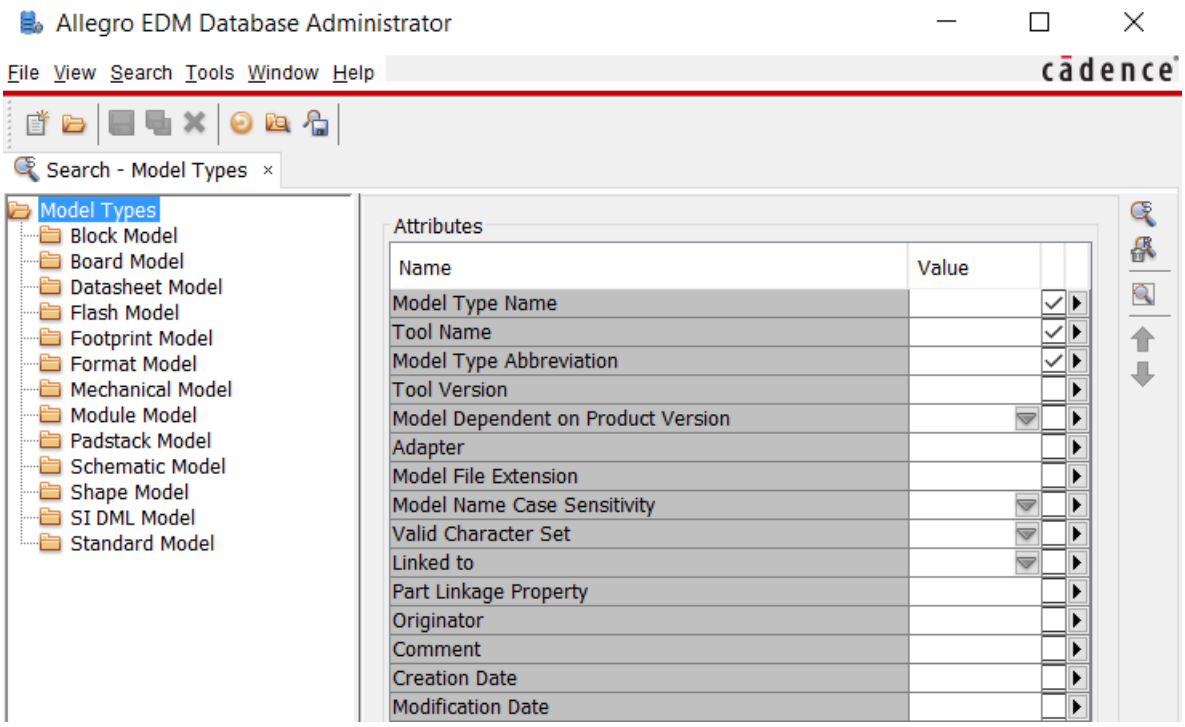
Creating a model type

To create a new model type, do the following:

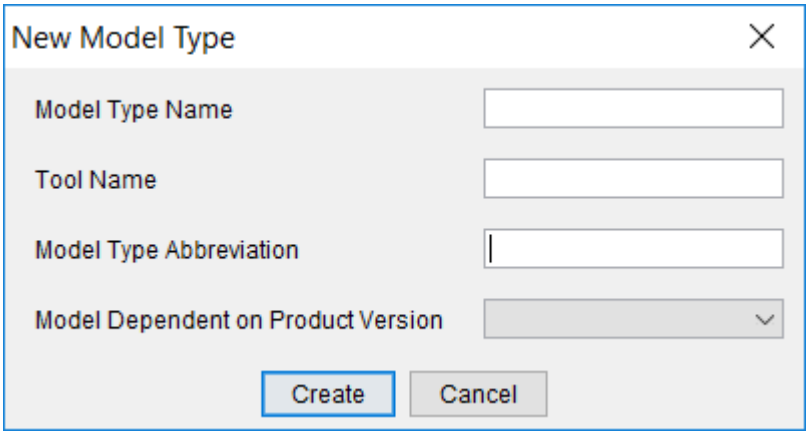
1. Launch Database Administrator either from Flow Manager or through the command line prompt:
   - In Flow Manager:
     - Choose *Tools – Database Administration*.
     - In the flow-specific tools section, choose *Tools Admin – Database Administration*.
   - In the Allegro EDM system console, type `dbadmin`, and press the `Return` key.
2. Provide your login details in the Login dialog box.



The Database Administrator interface appears.



3. In the Database Administrator window, choose *File – New – Model Type*.



4. Specify a name for the model type. For example, in this example, we specified *STEP Model*.

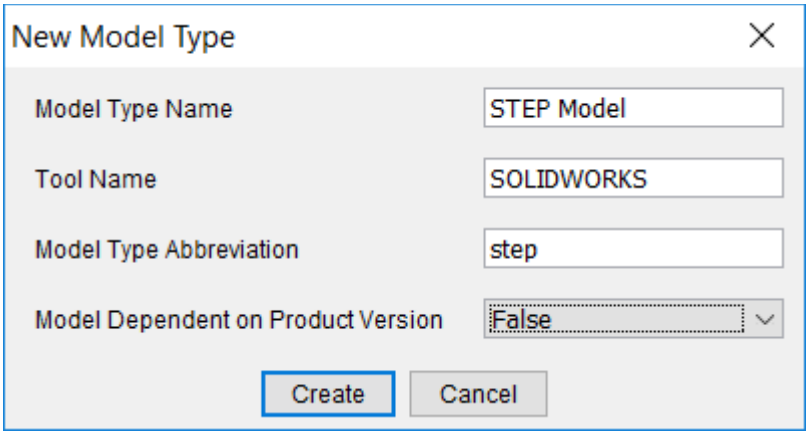
5. Specify the name of the application in which the 3D CAD model is authored. For example, *SOLIDWORKS*.

6. Provide an abbreviation for the model type name. For example, *step*. To ensure compatibility across EDM utilities and operating systems, it is recommended that you use all lowercase letters.

Important

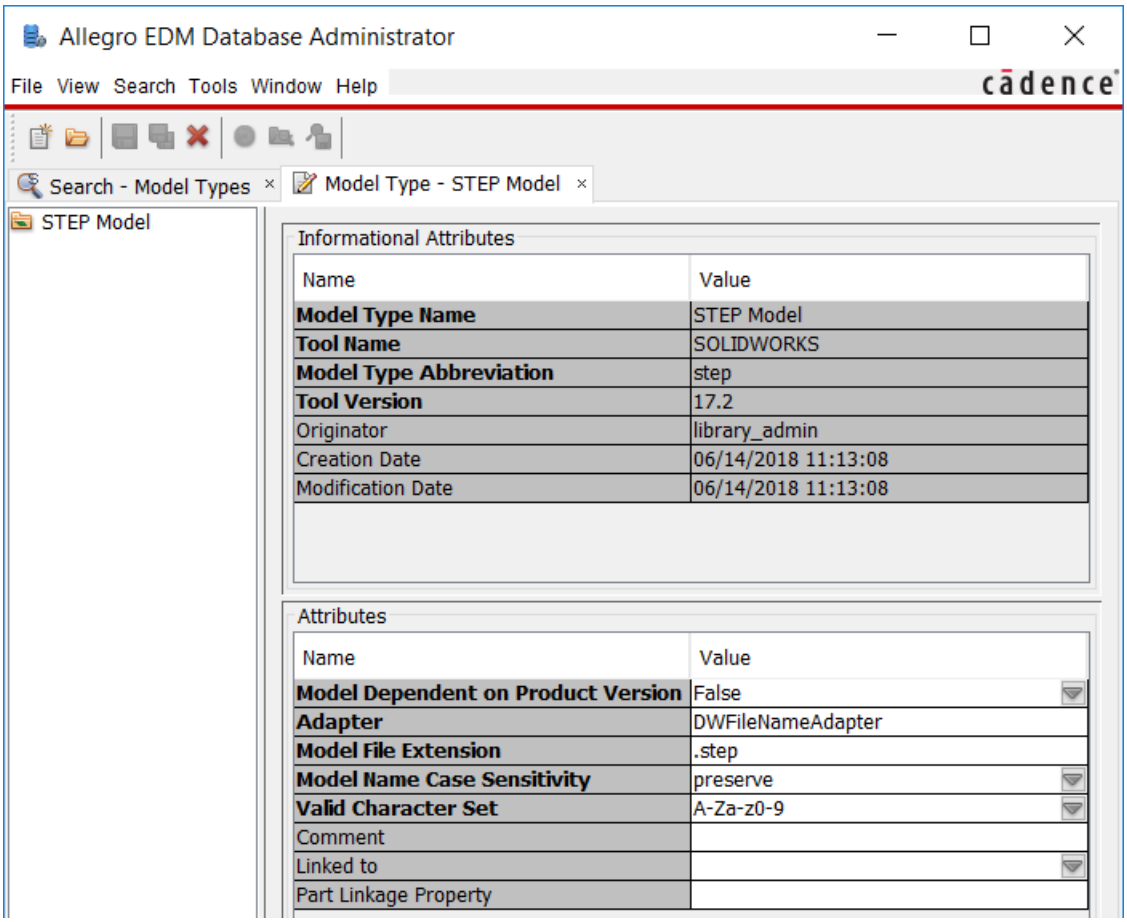
Be aware that this abbreviation is used as the file type. For example, if you specify `stepmodel` as the abbreviation, the Model File Extension will be `.stepmodel`.

7. Specify whether the model is depending on the product version. If you define the model as dependent on the product version, then moving to a newer version of the model authoring software will require you to update the model too.



8. Click *Create*.

The new model type is added to the database and Allegro EDM creates an object, which defines all the metadata related to the new model.



In the *Attributes* panel, you will notice that EDM automatically associated the default EDM adapter, `DWFileNameAdapter`, to the model you created.

This adapter reads the file name of the model. For example, if you create or already have a STEP model file with the name *3M_961401-9040704-AR.step*, Allegro EDM will only read the file name so as to store the name in the metadata of the model.

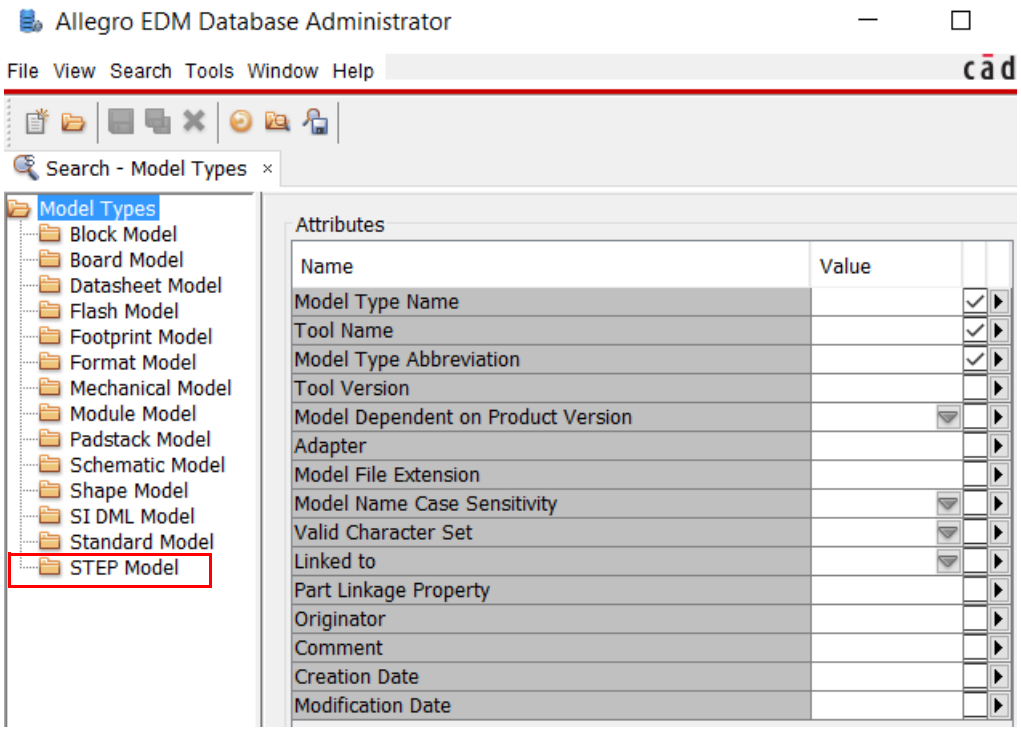
With the use of the default adapter, even if *3M_961401-9040704-AR.step* has submodels within it, that is, even if it is a compound model, EDM will assume that the model type has only one model whose name is *3M_961401-9040704-AR.step*.

Important

If you want Allegro EDM to extract and read the data or content of custom model types, you must create a custom adapter. For help in creating a custom adapter that works with Allegro EDM, contact Cadence Customer Support.

9. Click *File – Save* or click the *Save* tool bar button.

As you can see, the new model type you created appears in Database Administrator.

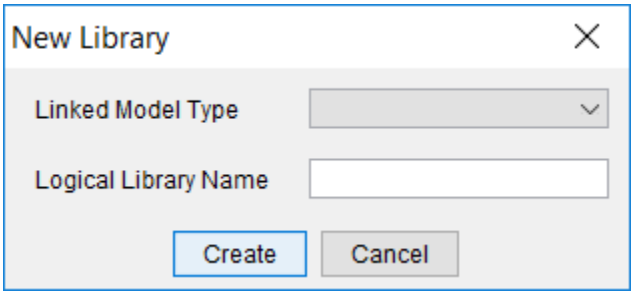


Now that you have created a model type and added it to the Allegro EDM database, you need to either:

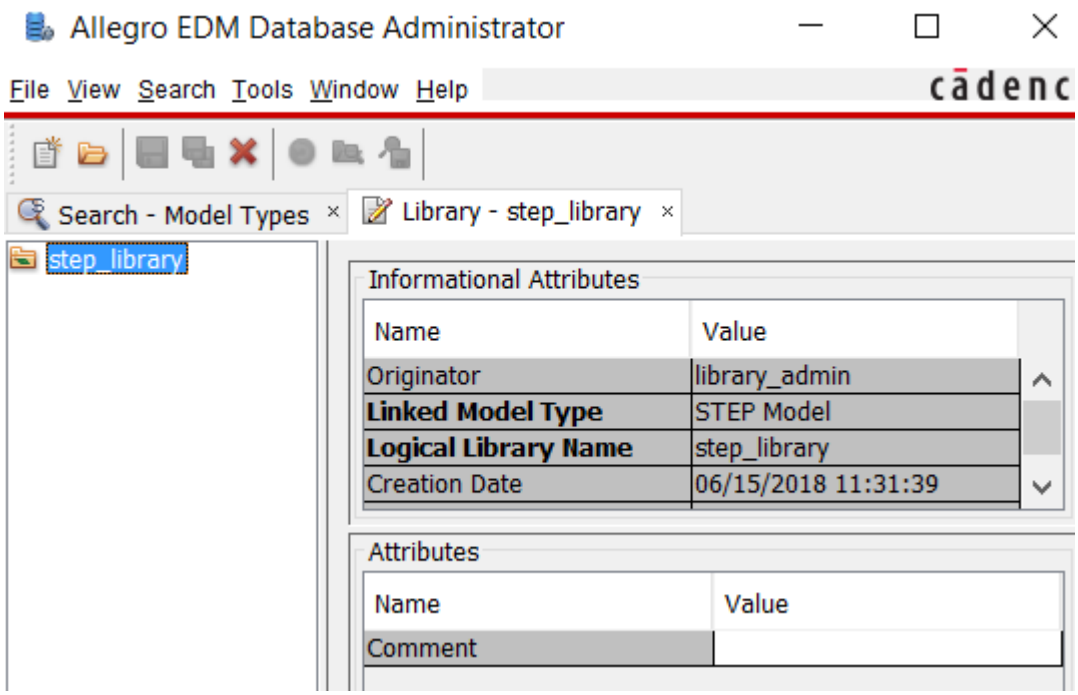
- create a library, which contains the 3D CAD drawings in `.step` format that you want to manage in Allegro EDM.
- import already existing libraries with `.step` files into Allegro EDM. To import already existing libraries into EDM, use the Library Import utility in EDM. For details about importing libraries into EDM, refer to *Allegro EDM Library Import User Guide*.

Creating Libraries for Custom Model Types

1. To create a new library of `.step` files, choose *File – New – Library*.



2. Select the model type to which this library will be associated. In our example, we will choose *STEP Model*.
3. Enter a name for the library. For example, `step_library`. Only lowercase alphanumeric characters and some special characters are allowed. Allegro EDM prompts you if you specify an unsupported character.
4. Click *Create*.

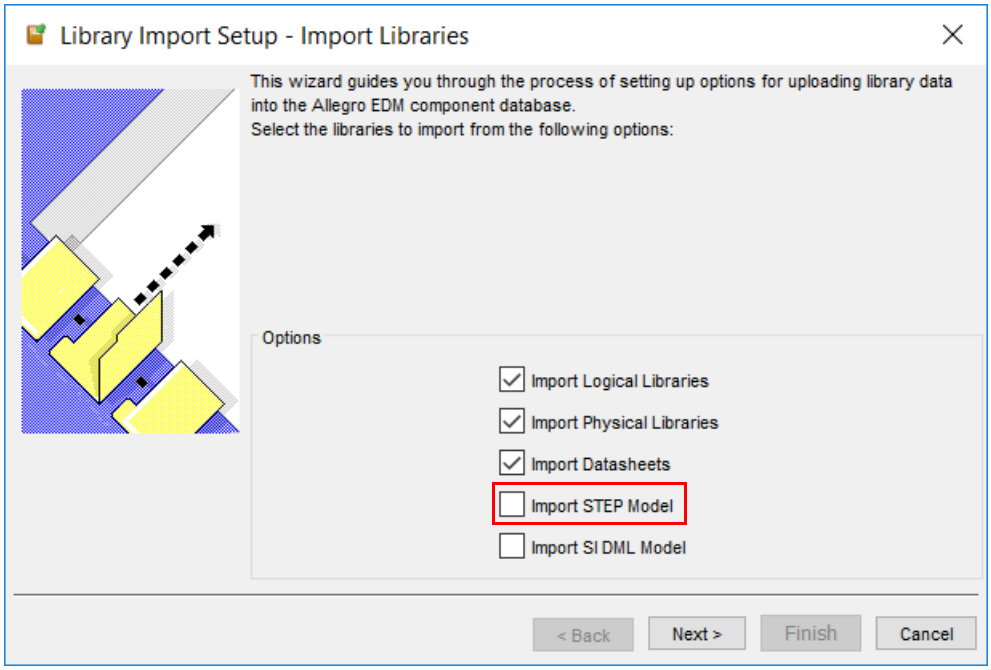


As you can see, EDM creates a new library.

Importing libraries for a custom model type

When you add a custom model type to the Allegro EDM database, EDM automatically adds an option to import its libraries in the EDM Library Import wizard.

For example, in this section, we added a STEP Model to the Allegro EDM database. As you can see, in the Library Import wizard, EDM added an option to import its libraries.



Why and when do I need to create an adapter in EDM?

Allegro EDM has one, default adapter, `DWFileNameAdapter`, for all out-of-the-box model types in EDM. This adapter reads the file name of a model. For example, if you have a model file with the name *3M_961401-9040704-AR.step*, EDM will only read the file name so as to store the name in the metadata of the model.

If you need to work with model types that are not provided out of the box in Allegro EDM, you can create a new model type. For example, if you work with 3D CAD models in `.step` format, and you want to manage them in Allegro EDM, you can create a new model type called STEP Model.

Any custom model you create is associated with `DWFileNameAdapter` by default. However, if you want Allegro EDM to extract and read the data or content of a custom model type, you must create a custom adapter. For help in creating a custom adapter that works with Allegro

EDM and associating a custom model type with the custom adapter, contact Cadence Customer Support.

See also

Out-of-the-Box Allegro EDM Models and Use in Allegro EDM

Allegro EDM Flow Manager

Why is Rules Checker not launching in the Design Flow of Flow Manager?

This can happen because of an unavailable license - `Concept_HDL_Studio`. In this case, you will need to provide another license.

When you choose *Verification (Graphical) - Interactive mode - Run ConceptHDL and Checkplus*, Allegro EDM Flow Manager looks for the `Concept_HDL_Studio` license.

The *Run ConceptHDL and Checkplus* button calls a JavaScript function called `Launch_ConceptHDL_Checkplus()`

To change the product that should be launched on clicking this button, modify this call: `Launch_ConceptHDL_Checkplus("pcb_expert")` or any valid license string. This is passed to the ConceptHDL command using the `"-product"` command line argument.

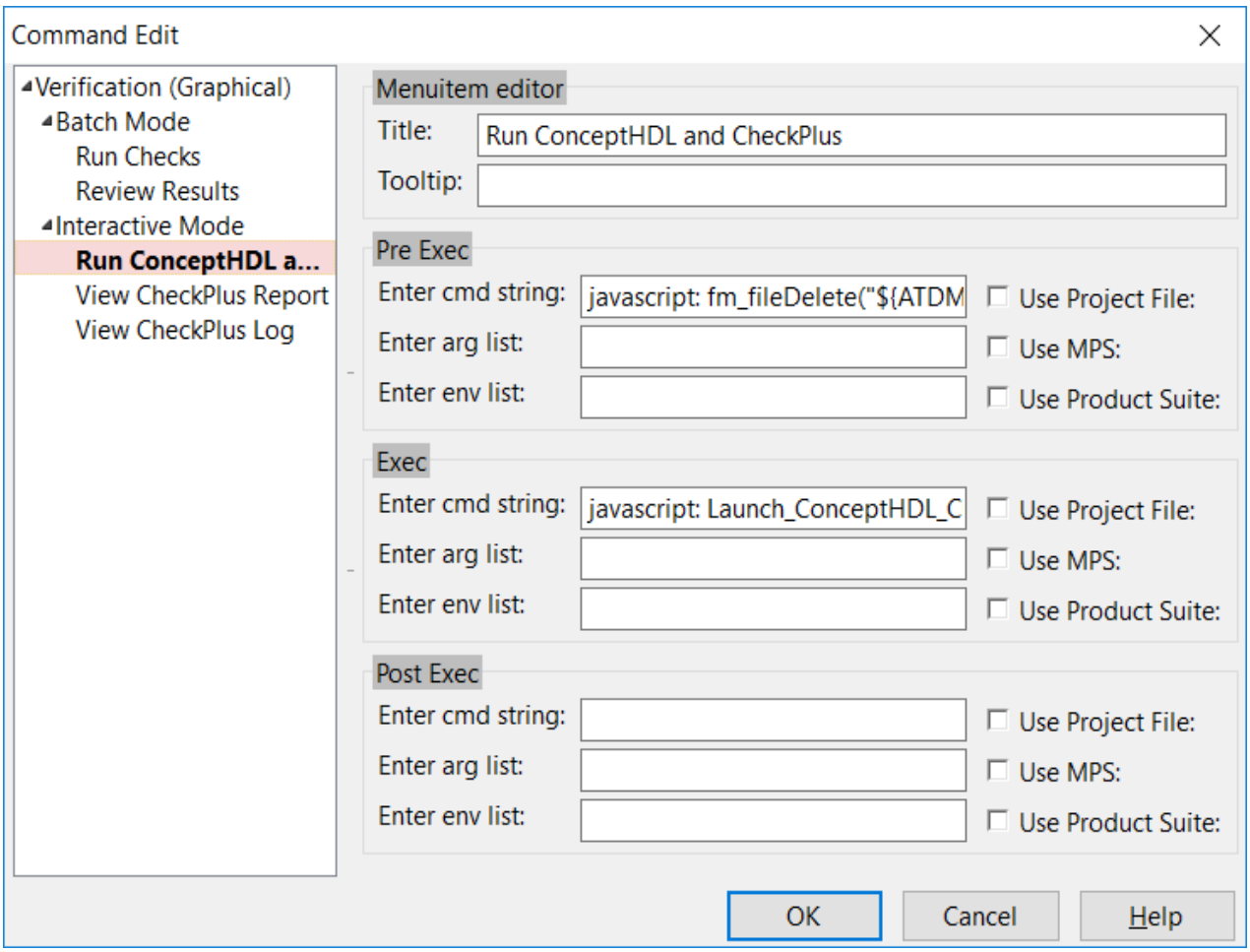
To change the default license being called by Allegro EDM Flow Manager, do the following:

1. Ensure that you are the flow administrator and open a design project in Flow Manager.

   To check whether you are the flow administrator, see How do I assign someone as a flow administrator in Allegro EDM Flow Manager?

2. Right-click the *Verification (Graphical)* flow button and choose *Properties*.
3. Choose *Interactive Mode - Run ConceptHDL and CheckPlus*.

4. Modify the *Enter cmd string* field in the *Exec* section. For example, use `"PCB_Librarian_Expert"` and ensure that you use the double quote marks.



5. Click *OK*.
6. To check whether you an now launch Flow Manager, choose *Verification (Graphical) - Interactive mode - Run ConceptHDL and Checkplus*.

   Allegro EDM now checks out the `PCB_Librarian_Expert` license.

Why is Flow Manager coming up blank - with only the title bar visible?

On high-latency networks, such as over 25 `ms`, and if the `PCBDW_FM_JVMLOAD_MAXRETRIES` and `PCBDW_FM_JVMLOAD_RETRY_INTERVAL` variables are set in the EDM client startup script, that is, `<startworkbench>.bat`, Flow Manager might sometimes come up blank. In this case, only the title bar is visible.

To address this, close Flow Manager, delete the `.adw_firefox_profile` folder from your user home directory and launch Flow Manager again.

How do I assign someone as a flow administrator in Allegro EDM Flow Manager?

To be a flow administrator, your login name must be in the `flowmanager.properties` file.

To add your login name to this file, do the following:

1. Browse to the `flows` directory at
2. `<Allegro EDM >\<company>\<site>\cdssetup\projmgr\flows.`
3. Open the `flowmanager.properties` file in a text editor.
4. Add your login ID to the list of administrators.

   This list is specified by the `admin` variable in the file.

   For example, if your login ID is `smithd`, after entering your login details, the value of the `admin` variable will be:

   ```
   admin= user1 user2 user3 smithd
   ```

Library Management

How to force Library Import to use an existing padstack model for a new footprint being imported?

Keep the PADSTACKS as input forLlibrary Importand ensure that the PADSTACK library name is the same as the reference library to get new footprints linked to existing PADSTACKS in the Allegro EDM database.

What are the commandline options for Library Import?

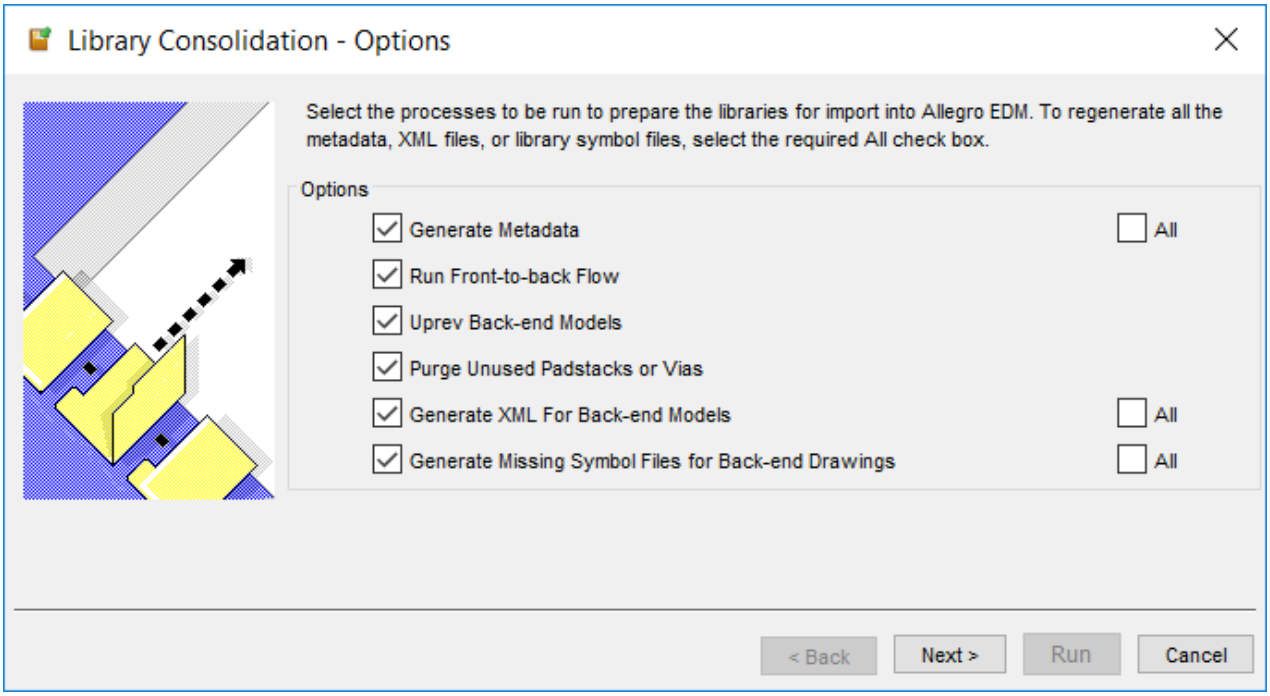
To	Use the Command	Example

Generate an XML file	`libimport -xmlgen -proj <PROJECT> -conf <CONF FILENAME>`	`libimport -xmlgen -proj "d:\atp_libimp\atplibimp.cpm" -conf default`
Pre-analyze a project	`libimport -preanalyse -proj <PROJECT> <CONF FILENAME>`	`libimport -preanalyse -proj "d:\atp_libimp\atplibimp.cpm" -conf default`
Upload data in the user interface	`libimport -upload -proj <PROJECT> -conf <CONF FILENAME>`	`libimport -upload -proj "d:\atp_libimp\atplibimp.cpm" -conf default`
Upload data using commandline options	`libimport -upload -nogui -proj <PROJECT> -conf <CONF FILENAME>`	`libimport -upload -nogui -proj "d:\atp_libimp\atplibimp.cpm" -conf default`
Post-analyze project	*`libimport -postanalyse`*	*`libimport -postanalyse`*

Why do cells need to be baselined before running Library Import?

If cells are not baselined, no metadata and version information is available in the symbols. When you try and use such symbols in a design, Library Revision Manager is unable to check the symbol version in the Allegro EDM component database versus the symbol being cached in the `flatlib` folder.

To baseline cells, select the *Generate Metadata* option during library import.



How do I correct missing PCB Editor Model information during Library Import?

While importing a PCB Editor model, Library Import uses the folder name in which the model exists as the library name for the model.

Sometimes, some models might be used in other models. For example, some padstacks might be used in a footprint, but do not exist in any library. Library Import reports these models that are not available in a library in the Pre-Analyze report.

When you generate an XML of the source library data, for all such models that are not available physically in a folder, Allegro EDM creates a dummy folder and the name of the dummy folder is used as the library name for all the models that do not exist in any library.

The naming convention for the dummy folder is `TMP_<model_type>`. For example, for a footprint, the dummy folder created is `TMP_FOOTPRINT`, and for a shape the dummy folder created is `TMP_SHAPE`. The same names will be used as the library names when the models are uploaded to Allegro EDM.

To ensure that all the models and library information is available for the flows, as the librarian do the following:

1. Create or add the missing PCB Editor models
2. Places them in the correct libraries
3. Generate the XML
4. Upload the models into the Allegro EDM component database

This corrects the status of the model and the model-library relationship in Allegro EDM. In the next (or any later) run of Library Import, the new model and library information is added to Allegro EDM and the previously existing dummy libraries information is removed.

Example

A footprint, `fp_abc`, is used in a part called `part123`. The `fp_abc` footprint is not available in any library. When importing a library, Library Import creates a placeholder footprint with the same name as the part, which is `abc`, and places the footprint in the `TMP_FOOTPRINT` library.

In this run of Library Import, this information is added to Allegro EDM. Later, the librarian creates a footprint called `abc` and places it under a folder for footprint model libraries.

As the librarian, generate the XML file again by selecting the Extract Library Data option.

Allegro EDM now modifies the library information for the footprint and adds the new model-library relationship. The `abc` footprint is placed under the footprint library.

How do I use a CSV file for part classifications?

Here is an example of a CSV file for part classifications.

Serial No.	Part Class. 1	Part Class. 2	Part Class. 3	Part Class. 4	Properties	Part Number
1	GATE	NAND	BUFFER	BUFFER_1	BODY_VERSION	N74F38D
2	GATE	NAND	BUFFER	BUFFER_1	DESCRIPTION	74F38SC
3	GATE	NAND	BUFFER	BUFFER_1	VENDOR	M38510/35202BCA
4	GATE	NAND	BUFFER	BUFFER_1	CAGE_CODE	54F38FM
5	GATE	NAND	BUFFER	BUFFER_1	DATA_SHEET	SN74F38D
6	GATE	NAND	BUFFER	BUFFER_1	REV	
7						
8	GATE	NAND	BUFFER	BUFFER_2	BODY_VERSION	SNJ5438J
9					DESCRIPTION	SN54LS38J
10					VENDOR	SN74LS38N

11				CAGE_CODE	SN74LS38D
12				DATA_SHEET	M38510/30203BCA
13				REV	
14					
15	GATE	BUFFER	BUFFER_1	BODY_VERSION	SN74LVC1G126DBVR
16	GATE	BUFFER	BUFFER_1	DESCRIPTION	74LVC1G125GW
17	GATE	BUFFER	BUFFER_1	VENDOR	SN74LVC1G126YZPR
18	GATE	BUFFER	BUFFER_1	CAGE_CODE	SN74AUC2G34YZPR
19	GATE	BUFFER	BUFFER_1	DATA_SHEET	SN74LVC1G126DBVT
20	GATE	BUFFER	BUFFER_1	REV	SN74LVC1G126DRYR
21					
22	AMPLIFIER	BUFFER	BUFFER_1	BODY_VERSION	A-0020EZ
23				DESCRIPTION	ECG008B-G
24				VENDOR	ECG008
25				CAGE_CODE	
26				DATA_SHEET	
27				REV	
28					
29	AMPLIFIER	BUFFER	GENERAL PURPOSE	BODY_VERSION	AG302-63

Here is an explanation of this table:

- The first row is the header and has the maximum levels of classification depth required by a part in your CSV. For example, if some parts in the CSV have three levels, some two, and some four, specify the number of levels required by the part with the highest classification depth.
- In the data rows, classification levels are filled from left to right leaving unused levels empty. For example, line 16 onwards, *Part Classification 4* does not apply.
- Allegro EDM expects the complete classification depth to be provided for each part number, but supports some level of sparse data. For example, in rows 10-14, the classification hierarchy would be derived as `GATE.NAND.BUFFER.BUFFER_2`. That is, empty cells pick the values of the preceding cell with data in the CSV.

- When the row with the list of searchable properties is completed, you can have a blank value in the Properties column in the subsequent rows and continue to list remaining parts for the classification tree.

Note: All empty lines, and lines without any values, are ignored; there are some in the CSV for visual relief and improved readability, such as rows 7, 14, and 28.

How do I correct mismatched padstack names during Library Import?

When importing libraries into the EDM environment, Library Import might report a padstack as missing when, in fact, it exists. This is caused by an internal padstack name that does not match the padstack file name.

Padstacks have external names which are the file names. They also have internal names in the `*.pad` files. When a padstack, `a.pad`, is copied to `b.pad`, its internal name is left as "`a`". Allegro PCB Editor and Pad Designer are not affected because they only use the file name (`a.pad`).

However, Allegro EDM Library Import reads the internal name when checking which padstacks are available.

To successfully import libraries, the internal names of padstacks must match their file names. To ensure this, open a padstack in Pad Designer and save it again. Repeat this task for every padstack in your library.

Here is a handy Windows script you can use for this task. To run it, do the following:

1. `cd` to your directory of `*.pad` files.
2. Check that only `*.pad` files are listed by running `dir *.pad`

   If you run this in the reference library, `*.pad` on Windows also matches `*.padstack`
3. Now run the following script:

```
echo Building Script File

set PADSTACK_NOWARNING_DISPLAY=1

%ECHO OFF

echo version 16.2 > pad_name_fix.scr

@echo set noconfirm >> pad_name_fix.scr

echo set window form.padedit >> pad_name_fix.scr
```

```
for %%i in ( *.pad ) do (
if %%~xi EQU .pad (
@echo pse_open >> pad_name_fix.scr
@echo fillin "%%i" >> pad_name_fix.scr
@echo pse_save >> pad_name_fix.scr
)
)
echo pse_exit >> pad_name_fix.scr
%ECHO ON
echo Running Script
pad_designer -s pad_name_fix.scr
echo operation completed.
```

Now run Library Import again. The padstack names error should be resolved.

How do I upload only archives during Library Import?

When a library is uploaded to the Allegro EDM database, the following is added to the database:

- Classifications
- Objects
- Relationships
- Archives

You can control what is added to the database using the default `SKIP_UPLOAD` directive in the `default.conf` file, which is in *<library_import_project_directory>*`\migration`.

By default, this directive is set to `FALSE`, which means that all of these are added to the database. Setting the value to `TRUE` will add only archives; the remaining three items are skipped.

How do I restore the last state of the database if Library Import upload fails?

Every time you run Library Import, Allegro EDM creates a backup of the Allegro EDM database (`mkdump`) before uploading data to the database. This backup also has the vault archives. This is the last state of the database.

The vault is a hierarchical file folder where library model files are archived/stored with entries of these models available in the component database. The model files are stored in the vault in a compressed format (as a tar ball). Model files are archived after library models in the integration area are verified and released.

The integration area is an intermediate area where archived models are stored until they pass through the verification process.

In case of errors, you can restore your database to the last safe state by doing the following:

1. Run the `fetch_dump` utility. This utility finds models to import by identifying differences between the models in the client site and master site. Once identified, the data is fetched from the master site to the client site.

   To run the utility, type the `fetch_dump` command at the Allegro EDM System Console. Errors, if any, during this process can be viewed in the *<Allegro_EDM_reference_library>*`\log\fetch_dump.log` file.

2. After you run `fetch_dump`, run the `install_dump` utility. This updates the database with the data that has been fetched from the master site (using the `fetch_dump` utility) to the client site database server.

   To run the `install_dump` utility, type the `install_dump` command at the Allegro EDM System Console.

   Errors, if any, during this process can be viewed in the *<Allegro_EDM_reference_library>*`\log\install_dump.log` file.

The `install_dump` utility might not always be able to fetch the vault archives in their previous state depending on tar ball creation.

Example: Say there were 100 models in the Allegro EDM component database. Then you ran Library Import for 20 models. EDM creates tar balls for these 20 models, and there are now 120 models. However, if you want to retrieve the 100 you had earlier, you cannot identify which of the 100 out of 120 you now want to retrieve.

As a result, it is always recommended after you run Library Import, always preserve the vault archives before you run Library Import again.

How do I use Data Exchange to update identical part numbers?

You can use Data Exchange to update parts if the part numbers in an XML being imported into Allegro EDM, that is the incoming input XML, has corresponding duplicate part numbers in the Allegro EDM component database.

To identify and update the correct part in the database, do the following:

1. Open `settings.sync` from `.../pcbdw_lib/exchange/sync/adw`.
2. Add the following in `settings.sync`:

   ```
   <parameter name="part.mapping.disable" value="true" />
   ```

   If this setting is already in the file, ensure that its value is true.
3. Remove the following setting:

   ```
   <parameter name="part.mapping.apply.all" value="false" />
   ```
4. Add the following setting:

   ```
   <parameter name="import.sync.properties" value="syncproperty1,
   syncproperty2, syncpropertyN" />
   ```

   Where `syncproperty1, syncproperty2, syncpropertyN` are the names of the sync properties. Ensure that you specify a property that can identify multiple instances of an identical part.

   Some examples of this entry are:

   ```
   <parameter name="import.sync.properties" value="PACK_TYPE" />
   ```

   ```
   <parameter name="import.sync.properties" value="MPN" />
   ```

   ```
   <parameter name="import.sync.properties" value="MANUFACTURER" />
   ```

   ```
   <parameter name="import.sync.properties" value="MPN,
   MANUFACTURER" />
   ```

   Note: The sync property name is case sensitive. Ensure that you use the sync property name correctly.
5. Create a CSV file of the data you want to import and name the file. For example, `import.csv`.

The column headers in the CSV file should be:

```
Part Number, syncproperty1, syncproperty2, syncpropertyN,
part_property_to_be_updated
```

For example:

```
Part Number     PACK_TYPE      RoHS
CDN-CON-0007    SMD            Compliant
CDN-CON-0007    2SHLD_PINS     Non-compliant
```

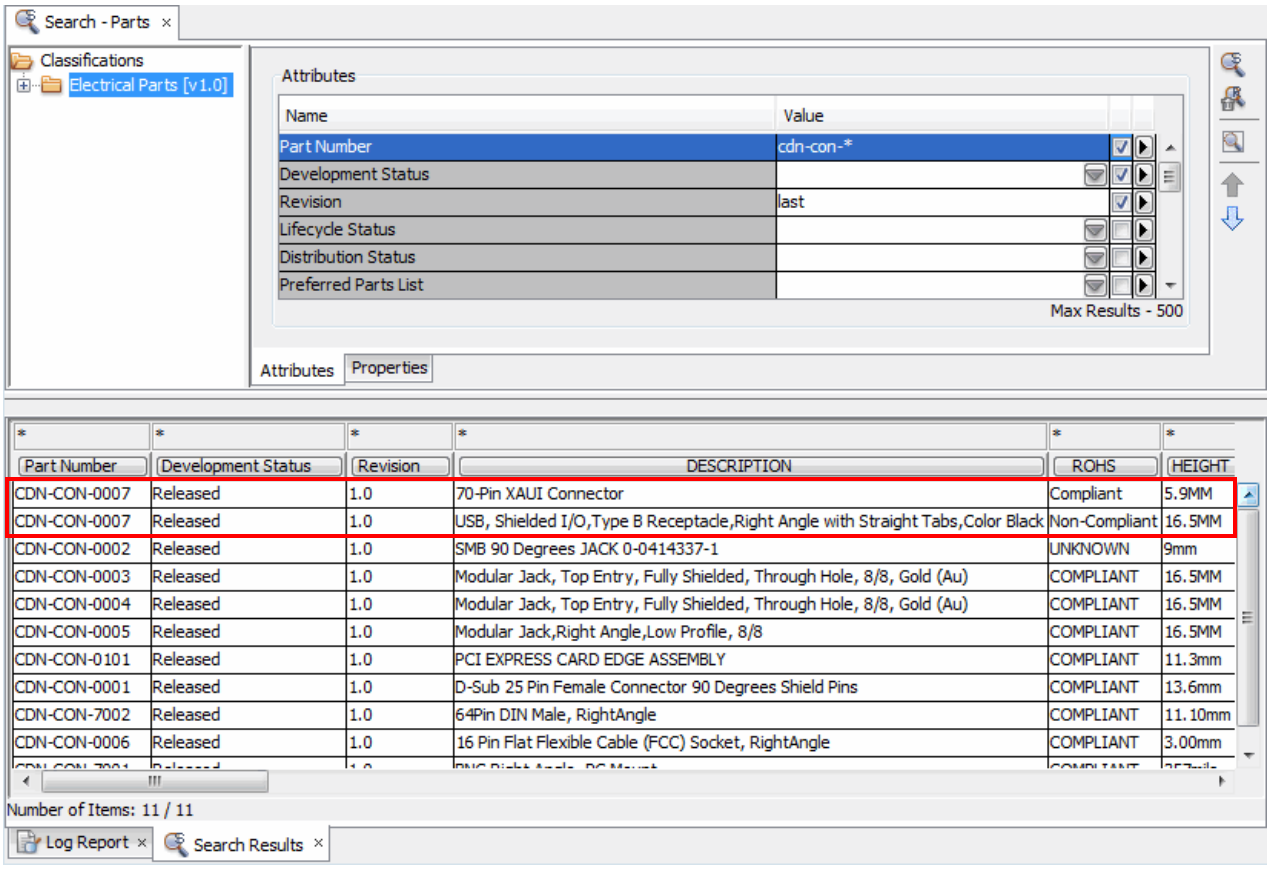
In this example, the combination of the attribute Part Number and sync property `PACK_TYPE` will be used to identify multiple instances of identical part numbers in the component database. The identified instances will be updated with the corresponding incoming RoHS value.

6. Modify `sync.xml` to specify the rules to be applied to the target objects. In this example, `sync.xml` is modified, as shown in the following figure, to update the target objects that are parts with the following part number: `CDN-CON-0007`.

```
<ObjectRules id="object_id">
    <Condition name="TargetNotExist" action="create" />
    <Condition name="TargetExist" action="update">
    </Condition>
</ObjectRules>
```

7. Run the command: `dataexchange -import adw`

   In this example, the two instances of `CDN-CON-0007` now have the value of `RoHS` as shown in the following figure:



8. Open `dxreport.csv` from `.../pcbdw_lib/exchange/sync/adw/reports`.

   The report shows the modified property values for each instance of the identical parts. In this example, the following report shows the updated value of the ROHS property:

Type	TypeOfChange	Name/Number	PACK_TYPE	Version		Development Status		ROHS	
				Previous	Current	Previous	Current	Previous	Current
Part	Update	CDN-CON-0007	SMD	v1.0	v1.0	Released	Released	R1	Compliant
Part	Update	CDN-CON-0007	2SHLD_PINS	v1.0	v1.0	Released	Released	R2	Non-Compliant

Updating sync property value

If you need to update the value of the sync property itself for each instance of an identical part, the column headers of `import.csv` should be as follows:

Part Number	#syncproperty1	syncproperty1	syncproperty2	syncpropertyN	part_property_to_be_updated

In this example, if the value of `PACK_TYPE` is also to be updated, then `import.csv` will be as shown in the following figure:

Part Number	#PACK_TYPE	PACK_TYPE	ROHS
CDN-CON-0007	SMD	SMD_R	Compliant
CDN-CON-0007	2SHLD_PINS	2SHLD_PINS_R	Non-Compliant

Updating an identical part with one or more sync properties missing

This sections explains how to modify `import.csv` if you need to update an identical part that does not have a particular sync property.

In the following example, the first and third instances of `CDN-CON-0007` are identified by the `PACK_TYPE` sync property and the second instance is identified by the `STATUS` sync property. The second instance of `CDN-CON-0007` with `PACK_TYPE`, `2SHLD_PINS` does not have the `STATUS` property.

To update such a part, modify `import.csv` as shown in the following figure:

Part Number	PACK_TYPE	STATUS	ROHS
CDN-CON-0007	SMD	OBSOLETE	Compliant
CDN-CON-0007	2SHLD_PINS	<<null>>	Non-Compliant
CDN-CON-0007	SMD	ACTIVE	Compliant

How do I rename multiple electrical part numbers (PART_NUMBER) at a time?

You can use the import feature of Allegro EDM Data Exchange to rename multiple electrical parts by importing a CSV file into Allegro EDM.

To rename the part numbers, create a CSV file with the data that needs to be updated or imported into Allegro EDM. For example, `data.csv`. Have three column headers in this file:

- `Name` - the part to be renamed
- `Part Number` - new part number
- `Description` - of the part

Sample CSV file (data.csv)

Name	OLD_PART NUMBER
723644-013	723644-013
809380-000	809380-000
602433-019	602433-019

After you create the CSV file, do the following:

1. Ensure that `settings.sync` has the following values:

   `part.mapping.apply.all = true` (to rename duplicate parts; else `false`)

2. Ensure that `sync.xml` has the following lines:

   `<Condition name="TargetExist" action="update">`

   `<Condition name="TargetNotExist" action="ignore" />` (so as not to create a new part if the source part does not exist)

3. Import the CSV file, in this case, `data.csv`, by doing either of the following:

   - Use the Data Exchange user interface included in the Condensed Library Flow.
   - Use the command line by doing the following:

     a. Rename the CSV file (in this example, `data.csv`) to `import.csv`.

     b. Place the `import.csv` file in the following location: `<PCBDW_LIB>/exchange/sync/csv/import`.

     c. Run `dataexchange -import csv` from the EDM console.

4. Run `lib_dist` after renaming the parts.

   Note: When you rename parts using a CSV file without revisioning the parts, the part table node on the schematic of the part is updated only after the `lib_dist` command is run.

How do I know whether the import of objects using Data Exchange has succeeded or failed?

Currently, Data Exchange does not display any message to indicate whether object import is a success or not. However, Data Exchange is considered a success when it works according to the rules specified in the `sync.xml` file.

Any failures or errors during the import process are listed in the log files that are at *<PCBDW_LIB>*`/exchange/sync/adw/logs`.

Why is Data Exchange exporting data with all revisions of relations?

Check the query XML file and ensure that the line in boldface is configured to export the latest versions of the relations, instead of all versions.

```
<Relation name="Alternate Footprint" fromtype="ECAD Component"
totype="Allegro Footprint Model">
```

<Attribute name="revision" value="last" />

```
</Relation>
```

Why are the library flow steps not visible in Allegro EDM Database Editor?

Library flow steps are not visible in Database Editor if a board, design, or any project type other than library is set as the active project.

Check that a library project is set as the active project.

How do I view the searchable and injected properties of a part in the Shopping Cart?

1. To view the searchable and injected properties of parts in the shopping cart, add a directive, `SHOPCART_DISPLAY_PROPS`, in the `START_COMPBROWSER` section of the *<project>*`.cpm` file. For example:

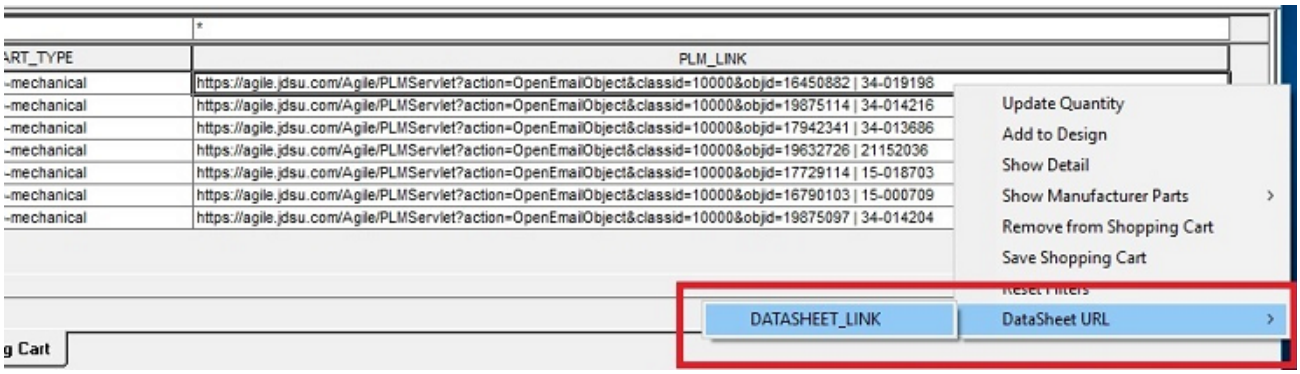
   ```
   START_COMPBROWSER

   SHOPCART_DISPLAY_PROPS 'PLM_LINK'

   END_COMPBROWSER
   ```

If you want to display a property as a URL in the shopping cart, use two more directives in the `START_COMPBROWSER` section of the CPM file. For example:

```
START_COMPBROWSER

SHOPCART_DISPLAY_PROPS 'PLM_LINK'

DataSheet_Url '@PLM_LINK@'

Display_URL 'DATASHEET_LINK'

END_COMPBROWSER
```

Note: The value of the column name you specify within the ‘at the symbol’ signs will be appended to the property display. The `Display_URL` directive is used to specify how you want Part Information Manager to display the option in the user interface.

2. Launch Part Information Manager and right-click on any part.
3. You will view an option, *DataSheet URL*, with a sub-option called *DATASHEET_LINK*.



4. Click on *DATASHEET_LINK*.

   This takes you to the link specified as the property value of PLM_LINK.

How is the classification property order for OrCAD Capture and DE-HDL models defined in the Part Details tab of Database Editor?

If the database administrator has not defined the order in which classification properties should be displayed, in the Part Details tab of Database Editor, the order for ECAD properties is defined on the basis of the OrCAD Capture Model, and Schematic and Part Classifications; searchable properties are presented in random order.

Can I delete a Schematic Model that is in the Preliminary State?

No, you cannot directly delete or rename a schematic model that is in the preliminary state.

To delete a schematic model that is in the preliminary state, add the model to a working set and then remove it from the working set. The model will be permanently deleted from the Allegro EDM component database.

Why is 'Use value in XML' unavailable in Database Editor while importing XML data?

When you import XML data and there is a difference in the values between the incoming XML data and what is already stored in the Allegro EDM component database, the *Use Value in XML* option is grayed out.

The *Use Value in XML* option is enabled when you use classification features or properties that are editable.

Note: Creation Date, Modification Date, and Development Status are also system-generated values that cannot be selected by users.

Why does Library Distribution display errors such as "ERROR: Matchfile 0" while running install_model?

If the length of the `<PCBDW_LIB>` path length is longer than 100 characters, then, when running the `install_model` utility, errors such as the following are seen:

- `INFO (LIBDIST-1033): X:/PSD_PV2/ADW_reg/libflow/dbeditor/blk_flow/distribute_checkin_release_block_model_block5/outputs/pcbdw_lib/exchange/receive/model_concept installation starts`
- `MATCH-002 : Can't open file Y:/adw/WINT/adwcds/16.63/v16-63-1/tools/pcbdw/database/db_install/archindep/par/match_files.block5/outputs/pcbdw_lib/exchange/receive/model_concept.sym`
- `ERROR: Matchfile 0 !!!`

To avoid such errors, shorten the length of the `PCBDW_LIB` path.

How do I preserve/control the order of injected properties in a PTF file during Library Distribution?

To preserve/control the order of injected properties, the PTFGen utility reads the `injprops.ini` file, which Allegro EDM creates during the process of importing a library. The file is created in *<PCBDW_LIB>*`/distribution/env` location, during library distribution.

The file has entries in the following format:

```
PART_NAME1,PART_NAME2,PART_NAME3=Prop1,prop2,prop3,prop4

PART_NAME4=Prop2,Prop1,Prop3,Prop4
```

The text to the left of the equal to (=) sign in the `ini` file entry indicates the comma-separated list of part names.

The text to the right of the equal to (=) sign in the `ini` file entry indicates the property order with property names separated by commas for injected properties only.

PTFGen now reads this entry from the `.ini` file for the part names and maintains the property order while generating the PTF.

In existing Allegro EDM component databases, if the `injprops.ini` file is not available, Allegro EDM maintains the property order based on the current search order. As a result, existing databases and users are not impacted.

rewriteproporderini

A PTFGen option, `-rewriteproporderini`, can be used to generate this file from the property order in the database in the following two ways:

- `ptfgen -global -rewriteproporderini`

  This generates the property order `.ini` file based on the cached PTF and backs up the existing `.ini` file.

- `ptfgen -global -rewriteproporderini -rewrite2db`

  This regenerates the PTF cache, and generates the property order `.ini` file based on a new PTF and backs up the existing `.ini` file.

As the librarian, you can also add a default property order using the following syntax:

```
CDN_ADW_DEFAULT= prop1, prop2, prop3
```

If no property order exists for a part name, the default order will be checked to order the injected properties. You will need to manually sync the `.ini` file across multiple libraries.

Examples

If the `.ini` file has the following entry:

```
A=1,2,3
```

and

- Sch model `A` has injected properties `1,2,4`, the PTF will have `1,2,4`.
- Sch model `A` has injected properties 2,4,3,5, the PTF will have `2,3,4,5`.
- Sch model `A` has injected properties `4,5,` the PTF will have `4,5`.

Migration

When a company migrates from a lower release to a higher release, what is the recommended methodology to set up EDM?

In this case, first migrate all the Designer Servers at your site to the higher release, then migrate the Master Library Server. If you first migrate the library server, any data published by the server cannot be read by the Designer Servers.

While migrating, you can do the following:

- For Designer Servers: Migrate your existing system to the next release while adhering to the operating system and hardware requirements of the new release. Ensure that *<PCBDW_LIB>*, *<adw_conf_root>* are backed up before moving to the new release.
- For Library Servers, you can do either of the following:
  - Maintain your existing server, configure a new Master Library Server in the new release on a new system, migrate the library data to the new system, then retire the old system. This method is recommended since it allows you to verify and compare all your data.
  - Or, take a backup of *<PCBDW_LIB>*, *<adw_conf_root>* and migrate your existing Master Library Server to the new release.

You can have only one EDM server (whether Master Library or Designer) hosted on a system.

For more information on setting up EDM, refer to What is the recommended EDM deployment architecture?

Backup and Restore

What is the recommended EDM Master Server backup policy?

<PCBDW_LIB> and *<adw_conf_root>* of the Master Library Server should be backed up on separate systems. Take a backup of the server during server idle time.

Designer server libraries do not need to be backed as they are read-only versions of the Master Library Server.

What are the recommended steps to restore the EDM Master Server from a backup?

There might be times when you need to restore a backup of a server. For example, you want to migrate your current server to a new server or need to retrieve backed up data due to hardware failure or obsolescence.

To restore a backed-up server, do the following for both - Master and Designer Servers:

1. Copy *<adw_conf_root>*, *<PCBDW_LIB>*, and the Allegro EDM project folder from the existing, or backup system to the new system.
2. Install the latest Allegro EDM software on the new server.
3. On the new server, ensure that you can launch Cadence applications, such as Part Developer, Part Information Manager, and so on.
4. Set the Allegro EDM environment by doing the following:
    a. Launch Allegro Configuration Manager and set up the server.
    b. Choose the *<PCBDW_LIB>* and *<adw_conf_root>* folders that you copied from in step 1.
    c. Ensure that the server host name and port are modified in *<adw_conf_root>* (workbench.ini) using Allegro EDM Configuration Manager.
    d. Set up the Allegro EDM client, that is, *<startworkbench>*.bat.
5. On the new server, set up a task scheduler for library distribution.

6. If you need to adjust the new server memory settings, make the required changes.
7. Verify that the new server is set by doing the following:
    a. Start the new Allegro EDM server.
    b. Ensure that the EDM license is successfully checked out.
    c. Run `adw_uprev` to update the database and database schema.
    d. In the new server, execute the `lib_dist` command to distribute libraries.
    e. Check that utilities such as Allegro EDM Database Editor, Part Information Manager, Allegro EDM Flow Manager, and so on, have all the required data.
    f. Make sure that all the library applications are on the same version as that of the Master Library Server.

Note: The steps detailed above will work to move a server from an older release to a newer release, except that any customization in *<adw_conf_root>* should be manually updated in the new *<adw_conf_root>*.

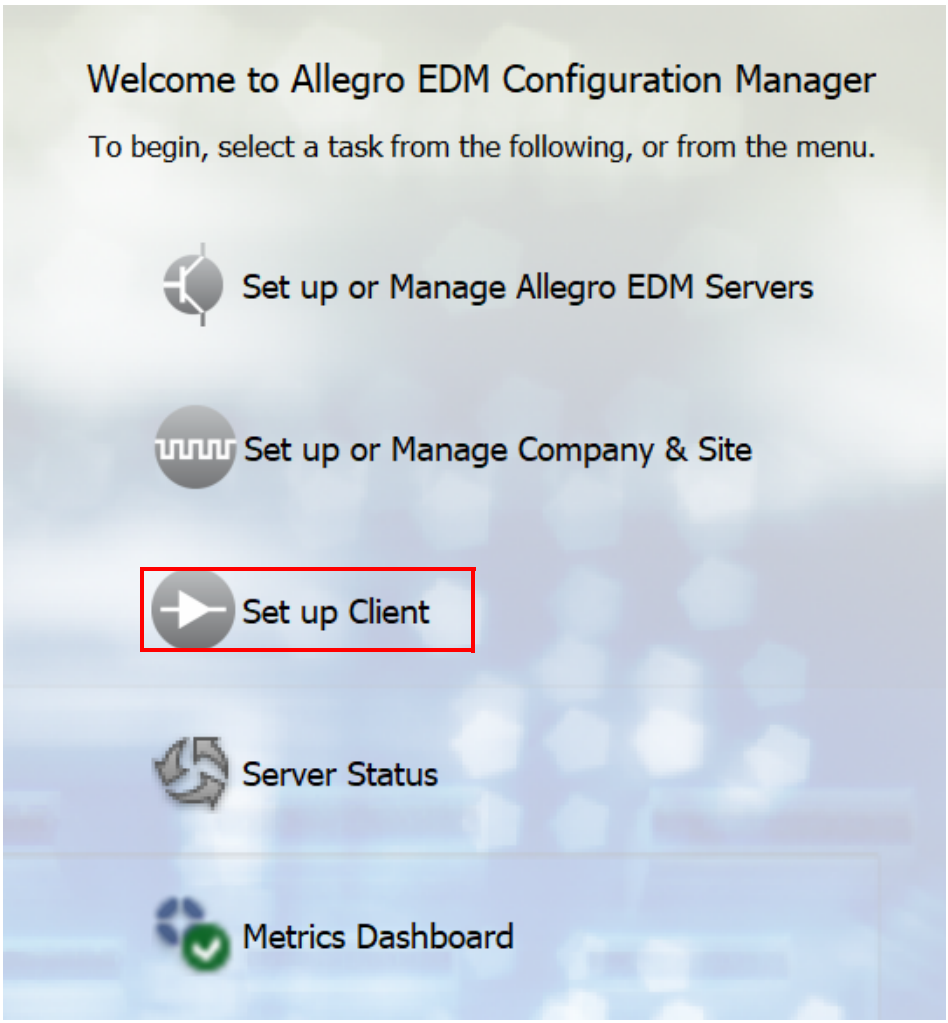
Troubleshooting

After copying a project using Allegro EDM Copy Project, why has the top-level design name retained the old project name?

A possible reason is if the `ADW_INST_DIR` variable in *<startworkbench>*`.bat` is either not set or points to an old installation.

```
@ECHO OFF
setlocal
echo Allegro EDM startup script for cadence
echo .
SET TEST_4_DW=C:\Cadence\SPB_17.2\tools\pcbdw\bin
REM Test Access to Allegro EDM software
REM -----------------------------------------------
set TH_TARGET=%TEST_4_DW%
dir "%TH_TARGET%" > NUL: 2>NUL:
IF %ERRORLEVEL% LSS 1 goto test_dw_ok
echo ERROR: Unable to find %TH_TARGET%
goto :no_start

:test_dw_ok
echo Path %TH_TARGET% located for Allegro EDM


REM Testing complete

REM set the other variables
if not defined ADW_INST_DIR set ADW_INST_DIR=C:\Cadence\SPB_17.2
if not defined ADW_CONF_ROOT set ADW_CONF_ROOT=C:\EDM_SETUP\adw_conf_root
REM This is the workbench reference library path
if not defined PCBDW_LIB set PCBDW_LIB=C:\EDM_SETUP\pcbdw_lib
```

To ensure that the top-level design name is updated, re-configure the EDM client. This will update `<startworkbench>.bat`, which is the Allegro EDM startup script.



If the EDM startup script is provided or managed by your ECAD Administrator, contact them to correct the location for `ADW_INST_DIR` in the script.

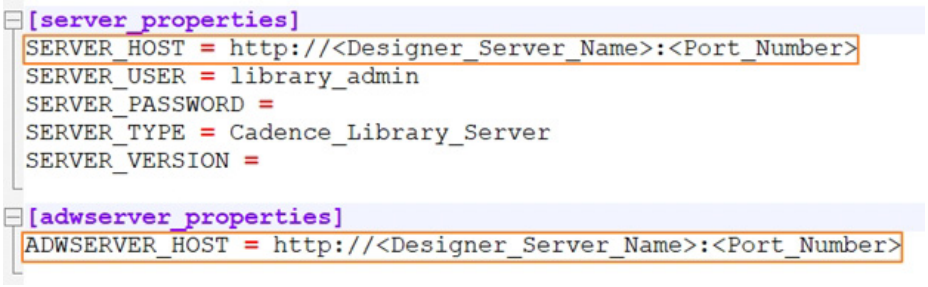
How do I address server connectivity problems when using Part Information Manager?

If you are a Design Entry HDL user, you might sometimes see either ERROR(SPDWUB-59) or ERROR(SPDWUB-6) when opening Part Information Manager to place components. These errors mean that Part Information Manager cannot communicate with the Designer Server.

First, ensure that the hotfix installed on the client machine and the Designer Server is the same, unless the client machine is intentionally on a higher hotfix and the server administrator has configured it accordingly.

Server connectivity problems can happen for various reasons, such as the following:

- **Case 1**: The server is not running or is not responding to the client machines
  - **a.** If not already stopped, the server administrator needs to stop the Designer Server from Windows Services or by using `adwstop.bat`.
  - **b.** Restart the Designer Server from Windows Services or by using `adwstart.bat`.
- **Case 2**: Allegro EDM configuration issue
  - **a.** Confirm whether the mapped drive path to the Designer Server `PCBDW_LIB` on your machine is configured correctly and can connect to the Designer Server.



  - **b.** Edit the `workbench.ini` file on the Designer Server and make sure that it points to the same Designer Server.
- **Case 3**: Change the `http` protocol to `https` in the `workbench.ini` file on the Designer Server.
  - **a.** Edit `workbench.ini` at `../<Allegro EDM Conf Root>/<company>/<site>/workbench.ini`.

b. Change the protocol from `http` to `https`.

```
[server_properties]
SERVER_HOST = https://<Designer_Server_Name>:<Port_Number>
SERVER_USER = library_admin
SERVER_PASSWORD =
SERVER_TYPE = Cadence_Library_Server
SERVER_VERSION =

[adwserver_properties]
ADWSERVER_HOST = https://<Designer_Server_Name>:<Port_Number>
```

- **Case 4**: Firewall issue

  Check whether the Windows firewall is blocking any of the following:

  - Port 7100 or the assigned port on the EDM Designer Server.
  - java.exe and/or port 7100 or the assigned port on the client machine.

- **Case 5**: Request is timing out

  Increase the timeout by adding the "`DAO_Timeout`" directive with a value greater than `300ms` in the project `CPM` file, as follows:

```
START_COMPBROWSER

DAO_Timeout '500'

END_COMPBROWSER
```