cadence®

Allegro® EDM
Library Distribution User Guide

Product Version 23.1
September 2023

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

Contents

Preface

About This Guide

The *Allegro® EDM Library Distribution User Guide* explains the methodology and procedures related to library distribution between master and client sites in the Allegro Engineering Data Management (EDM) environment. This guide is for site administrators and library administrators.

Related Documentation

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

- For learning how to configure Allegro EDM, see *Cadence Pulse and Allegro EDM Configuration Guide*.
- For information on Data Exchange import and export functionality, see *Allegro EDM Data Exchange Reference Guide*.

Typographic and Syntax Conventions

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

`literal`	Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names.
`argument`	Words in italics indicate user-defined arguments for which you must substitute a name or a value.
|	Vertical bars (OR-bars) separate possible choices for a single argument. They take precedence over any other character.

`[ ]`	Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices. You can choose one argument from the list.
`{ }`	Braces are used with OR-bars and enclose a list of choices. You must choose one argument from the list.

1

Getting Started with Library Distribution

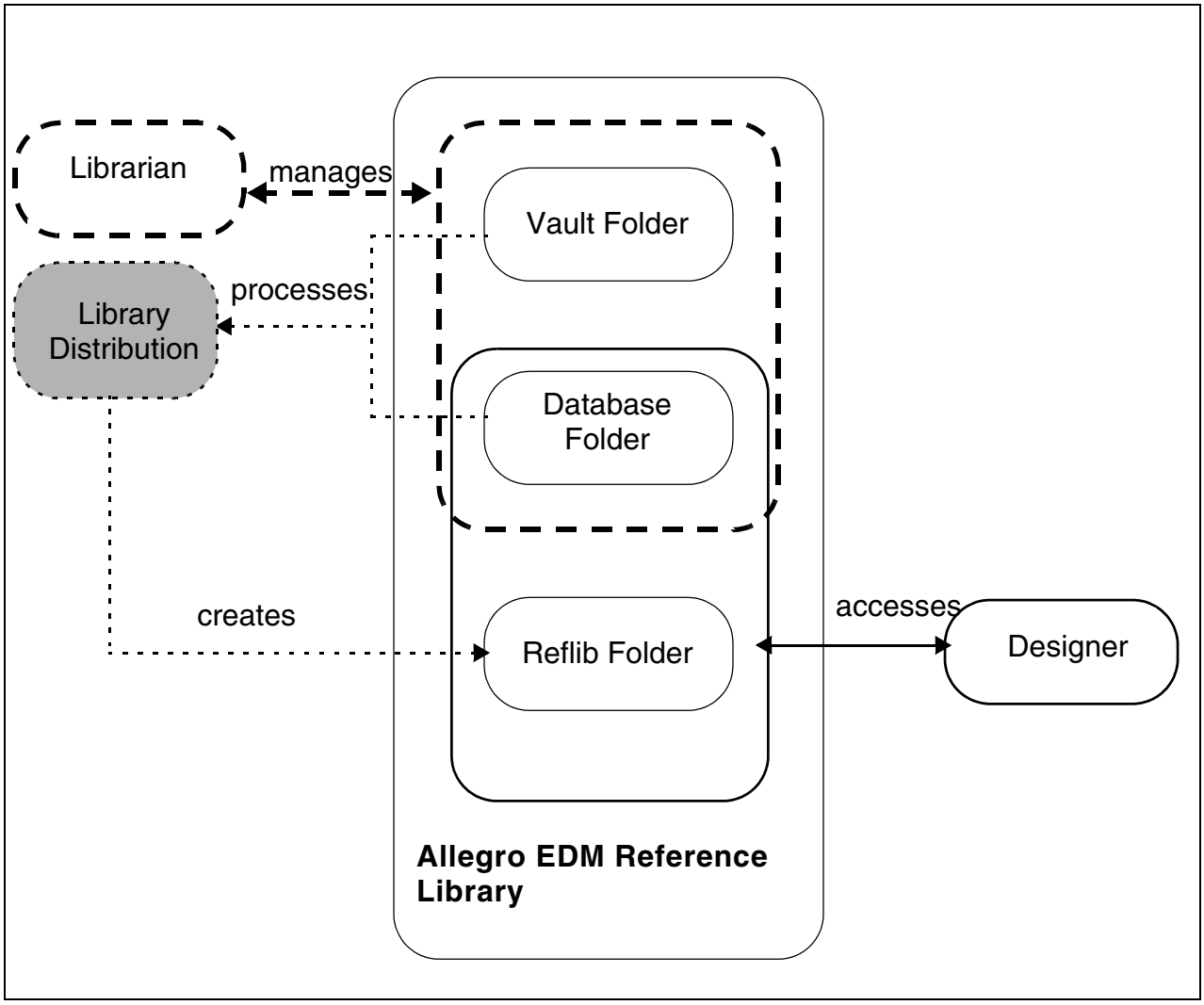
A librarian creates and maintains a repository of component libraries. A designer uses these libraries to create various schematic or board designs. When a librarian makes changes to any of the enterprise libraries, it is critical that these changes reach the designers.

Library administrators manage this repository of component libraries at a master (or global) site. All librarians should point to this master site for any library authoring and management activity. Designers can also connect to this master site to access component data. All other remote sites, considered client (or local) sites, are read-only copies and are used by designers. The data on these sites is a replica of the data on the master site.

This practice of maintaining a repository at the master site is vital for corporate library development. It also ensures that when you create new library components or modify the existing ones, all the enterprise design sites receive the components, and use these latest library components in their designs.

Component libraries are developed and published at the master site, and these are fetched and utilized by the client sites. This process of making component libraries available to the designers is called library distribution. This mechanism allows component changes (initiated by the master site) to be available and synchronized with all other client sites in the enterprise.

The following diagram explains the library distribution process:



This chapter introduces the library distribution process in Allegro EDM that helps you synchronize and distribute library components across multiple design sites.

The library distribution process runs on both the sites and distributes libraries from master to client sites in a phased manner. The library distribution process consists of two parts:

- Server Process: Runs on the master site and involves:
  - retrieving the data related to the latest versions of parts and models, which are ready to be shared between sites
  - publishing the same for the client sites

- Client Process: Involves connecting to the master site and fetching published libraries to the client site, and then creating a reference library at the client site for the designers to access.

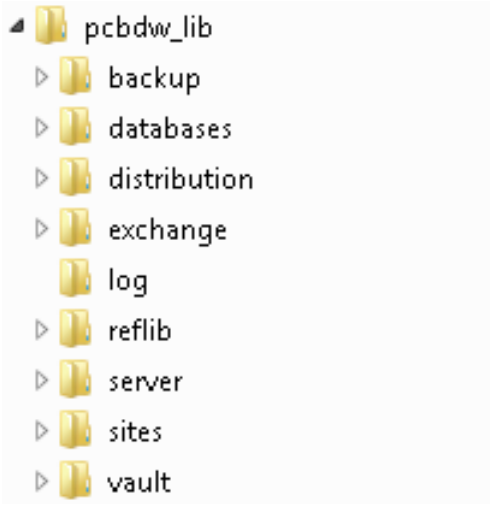
Important

The server process runs on the master site. However, if the designer is also using the same site (that is, in case of single site configuration), then the client processes are also run on the master site.

Understanding Reference Library Directory Structure

Before starting the library distribution process, it is important to revisit the directory structure that represents the Allegro EDM Reference Library. The Allegro EDM configuration process creates this folder structure and its respective content, and it is used by the library distribution process. For example, in the following screen shot, *<pcbdw_lib>* is the folder that contains these reference libraries. These libraries are linked to tool-models. Also, Allegro EDM refers to the location of this folder using the `PCBDW_LIB` variable.

The information related to component libraries is captured according to the following directory structure:



The following table explains the contents of the Allegro EDM Reference Library:

Table 1-1 Contents of Allegro EDM Reference Library

Folder Name	Description
<pcbdw_lib>	This is the name of the master folder that contains the Allegro EDM Reference Library.
databases	This folder contains the database folders of the Allegro EDM server running on various hosts. server: Contains the library database. conf: Contains the server configuration files. data: Contains files related to Allegro EDM server database. meta: Contains information about the files used by the Allegro EDM server internally. adwadmin: Contains data for the server dashboard. This folder is created when you select the *Enable Allegro EDM Metrics Database* option in the server setup wizard. adwmetrics: Contains the client-related data for all the clients. It also contains information of the software versions on the clients, which are accessing the Allegro EDM server. This folder is created when a client is launched from Flow Manager and accesses the Allegro EDM server.

Table 1-1 Contents of Allegro EDM Reference Library

Folder Name	Description
distribution	The distribution directory is used by library distribution to get all the setup information. The subdirectories include: backup: Contains the backup of the PTFs. You can specify another location (or folder) using the ptf.ini file. env: Contains the configuration files (.ini) such as lib_dist.ini, genhtml.ini, ptf.ini, category.ini, and so on required for the various library distribution utilities. html: Contains HTML files generated by the genmodelhtml utility. These files contain link to the data (models) being published by the master site. The client sites connect to these HTML files to fetch the data. list: Each list file (.lis) contains the list of models already available on the client site for a tool, tool version, and model type combination. These models are not imported again when you run the library distribution process the next time. xml: Contains intermediate XML files of the models to generate the HTML files.
exchange	The exchange directory lists all the directories for model exchange between the master and client sites. export: This folder contains the parts and models that are exported in compressed format. This compressed file is generated either using the Data Exchange export command or using the Database Editor export functionality. receive: This folder contains the models received from the master site in form of *<model_toolname>* subfolders. For example, model_allegro. sync: This folder is used when running the Data Exchange command. The Data Exchange export and import functionality can be configured using the files listed in the sync folder. For more information about this, see *Allegro EDM Data Exchange Reference Guide*. transmit: This folder contains the models to be sent to the master site in form of *<model_toolname>* subfolders. For example, model_concept.

Table 1-1 Contents of Allegro EDM Reference Library

Folder Name	Description
`log`	This directory contains the log files generated when you run the library distribution utilities.
`reflib`	This directory contains uncompressed models in the respective model libraries, to be used by designers.
server	This directory contains files specific to Allegro EDM Server. `conf`: This directory contains the logging configuration file customized to include e-mail settings, which enables you to send server messages based on the severity level (error or warning) set in the file. `log`: This directory contains the server log file: `adwserver.out`.
`vault`	The vault directory contains all the archive files for the models. `dump`: Contains the dump of the master site database in the `principal.jar` file. Contains the models that are released to the client sites in *<model_toolname>* directories. For example, `model_concept`.

2

Running Library Distribution

This chapter describes the library distribution process.

Library Distribution Setup

Before you run the library distribution process and set up the various library distribution options, ensure that you defined a company, designated a site as the master site, and specified client sites. For more information on how to do this, see the Cadence Pulse Configuration Guide.

Configuring Library Distribution Settings

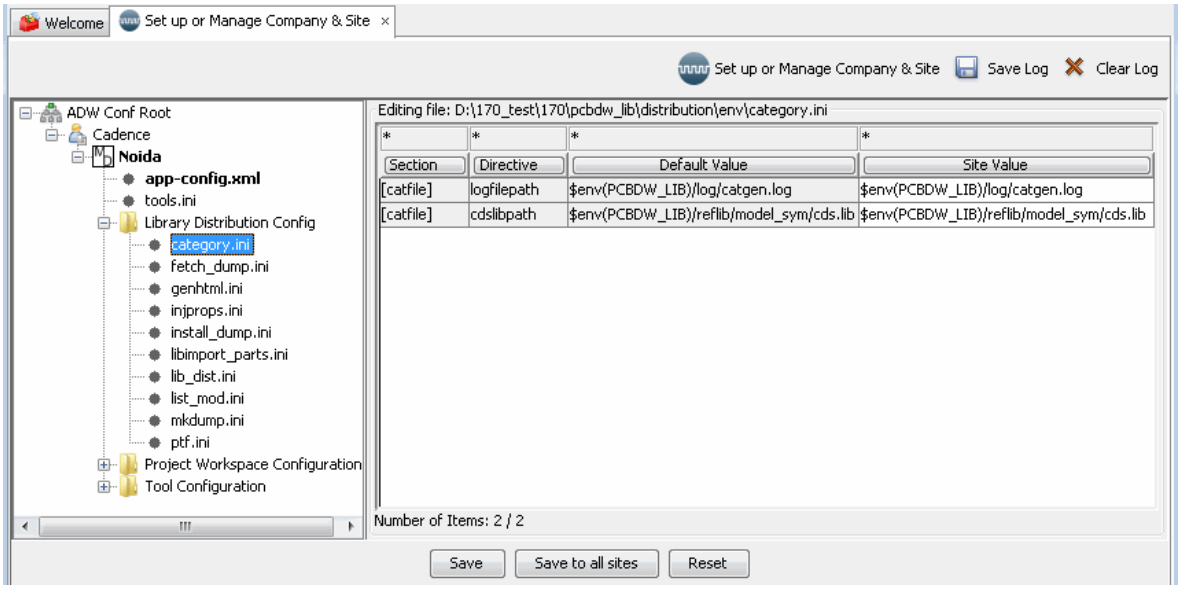
To configure the library distribution settings, do the following:

1. Launch Configuration Manager by doing the following:
   a. Navigate to: `<installation_directory>\conf`
      - Unix: Run the `conf.sh` script.
      - Windows: Double-click the `conf.bat` file.

      Allegro EDM Configuration Manager is displayed.
   b. Click *Set up or Manage Company & Site*.

      The Set up Allegro EDM Conf Root Directory wizard starts.
2. Click the browse button to specify the location of the existing *Allegro EDM Conf Root* folder and then click *Next*.
3. Navigate to `<company>` – `<site>` – *Library Distribution Config* in the *Set up or Manage Company & Site* tab.

You will see entries for the configuration files for each of the library distribution utilities.



The following table lists the function of the configuration files:

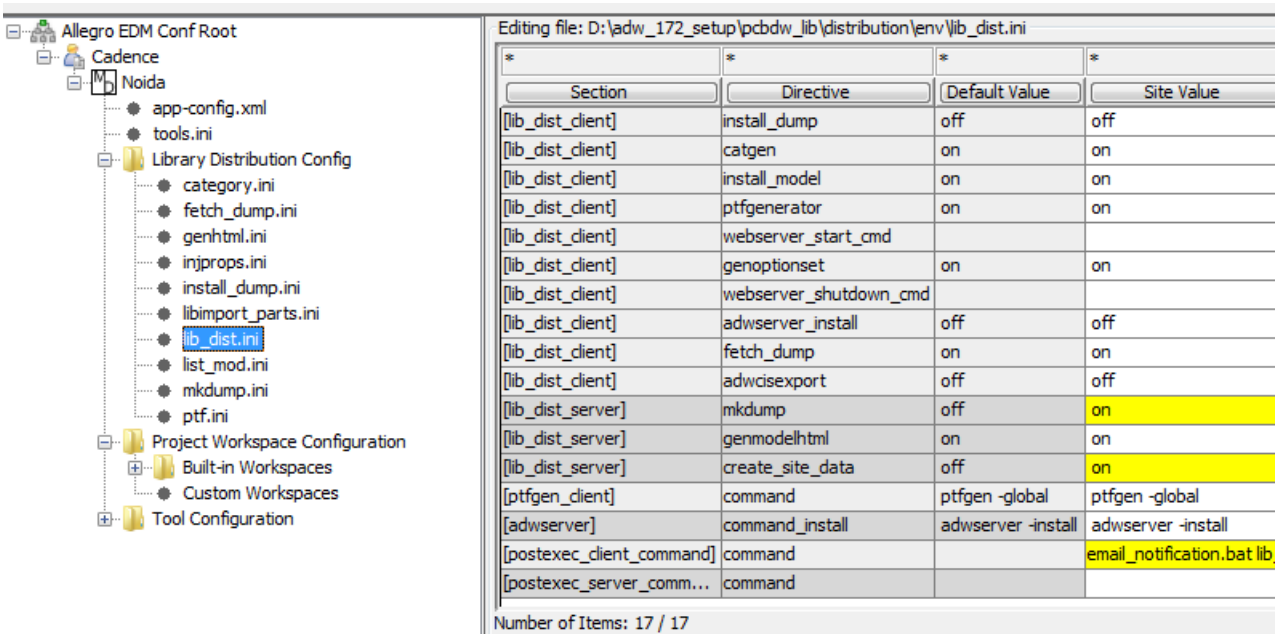
Table 2-1 Function of Configuration Files

Entry	Specifies the...	To know more, see...
`lib_dist.ini`	Default settings for master- and client-site library distribution utilities to be run as a Cron job or scheduled task	lib_dist.ini
`genhtml.ini`	Physical location for storing HTML files	genmodelhtml
`fetch_dump.ini`	Settings to connect to the server URL containing the HTML files on the master site	fetch_dump
`ptf.ini`	PTF generation setup properties, such as physical location for `cds.lib`, global PTF, and block PTF	ptfgen
`category.ini`	Category generation information for the libraries	catgen

Note: These files are located at *<Pulse Master Home>*`\distribution\env`. This guide refers to the reference library by its location *<Pulse Master Home>* or as `$PCBDW_LIB`.

lib_dist.ini

You need to configure *lib_dist.ini* before you run the library distribution process according to the Site Configuration Options.



Section	Directive	Default Value	Site Value
[lib_dist_client]	install_dump	off	off
[lib_dist_client]	catgen	on	on
[lib_dist_client]	install_model	on	on
[lib_dist_client]	ptfgenerator	on	on
[lib_dist_client]	webserver_start_cmd		
[lib_dist_client]	genoptionset	on	on
[lib_dist_client]	webserver_shutdown_cmd		
[lib_dist_client]	adwserver_install	off	off
[lib_dist_client]	fetch_dump	on	on
[lib_dist_client]	adwcisexport	off	off
[lib_dist_server]	mkdump	off	on
[lib_dist_server]	genmodelhtml	on	on
[lib_dist_server]	create_site_data	off	on
[ptfgen_client]	command	ptfgen -global	ptfgen -global
[adwserver]	command_install	adwserver -install	adwserver -install
[postexec_client_command]	command		email_notification.bat lib
[postexec_server_comm...	command		

Each utility has a corresponding directive that can be set to `on` or `off`. Setting the site value allows you to specify whether to run the respective utility. You need to specify the desired value in *Site Value* and click *Save*.

- The *lib_dist_client* section contains the following processes:
  - *fetch_dump*
  - *install_dump*
  - *webserver_shutdown_cmd*
  - *catgen*
  - adwcisexport

    The default setting for this utility is `off`. You must set it to `on` if you want to export OrCAD Capture CIS data to a database file and generate the CIS database configuration file (`.dbc`).
  - genoptionset
  - *ptfgenerator*

- *adwserver_install*

  For details on when to use this directive, see Site Configuration Options.

- *webserver_start_cmd*
- install_model

- The *lib_dist_server* section contains the following processes:
  - *mkdump*
  - *genmodelhtml*
  - *create_site_data*
- The *ptfgen_client* section contains:
  - *jvmargs*: This directive (when added in `lib_dist.ini`) allows you to adjust the memory requirements for library distribution.
  - *command*: This directive allows you to specify the PTF generator command with arguments. For example, `ptfgen -global`.
- The *adwserver* section contains the *command_install* directive, which is used to configure the command that needs to be run when *adwserver_install* (in the *lib_dist_client* section) is set to on.

You must not change this command.

- The postexec command section contains:
  - *postexec_server_command*: Contains commands that will run after the server process.
  - *postexec_client_command*: Contains commands that will run after the client process.

  Following are the rules for writing the postexec command:
  - You specify the postexec command in *Site Value of* `lib_dist.ini`.
  - Any command that contains spaces should be enclosed in double quotes.
  - A forward slash should be used to indicate a path.
  - Syntax for using the environment variables is `$env(VAR_NAME)`, where `VAR_NAME` is the name of the environment variable.

For example:

```
"mycommand -o "$env(PCBDW_LIB)/reflib/model.ndx" "$env(PCBDW_LIB)/reflib/
model_dml""
```

- If there is more than one command, create a script that invokes those commands and specify the script name in *Site Value*.

Site Configuration Options

Before you distribute libraries, it is important to do the following:

- Check the site configuration that is best suited for your enterprise.
- Identify and configure the client sites which require partial library distribution.
- Check if the client site is at a higher version than the client application.

  When you have to distribute database and model information to client sites that are at a higher version than the software, the `install_dump` utility, which is the default mechanism to install the database, does not work.

  In this case, set the value of *adwserver_install to* `on` and *install_dump to* `off`. Running the `lib_dist_client` command using this setting upgrades the database to a higher version (as used by the client) and also installs the upgraded database at the client site.

This section describes some of the popular configurations.

Configuration for Single-User Environment

Considered the default configuration, this configuration setting implies that you have a single site. This is the master and client site and you are the designer and librarian.

Table 2-2 Configuration Settings for Single Site

File Name	Field Name	Default Value
lib_dist.ini		
	[lib_dist_server]	
	genmodelhtml	`on`
	mkdump	`off`
	create_site_data	`off`
	[lib_dist_client]	
	fetch_dump	`on`
	install_dump	`off`
	install_model	`on`
	ptfgenerator	`on`

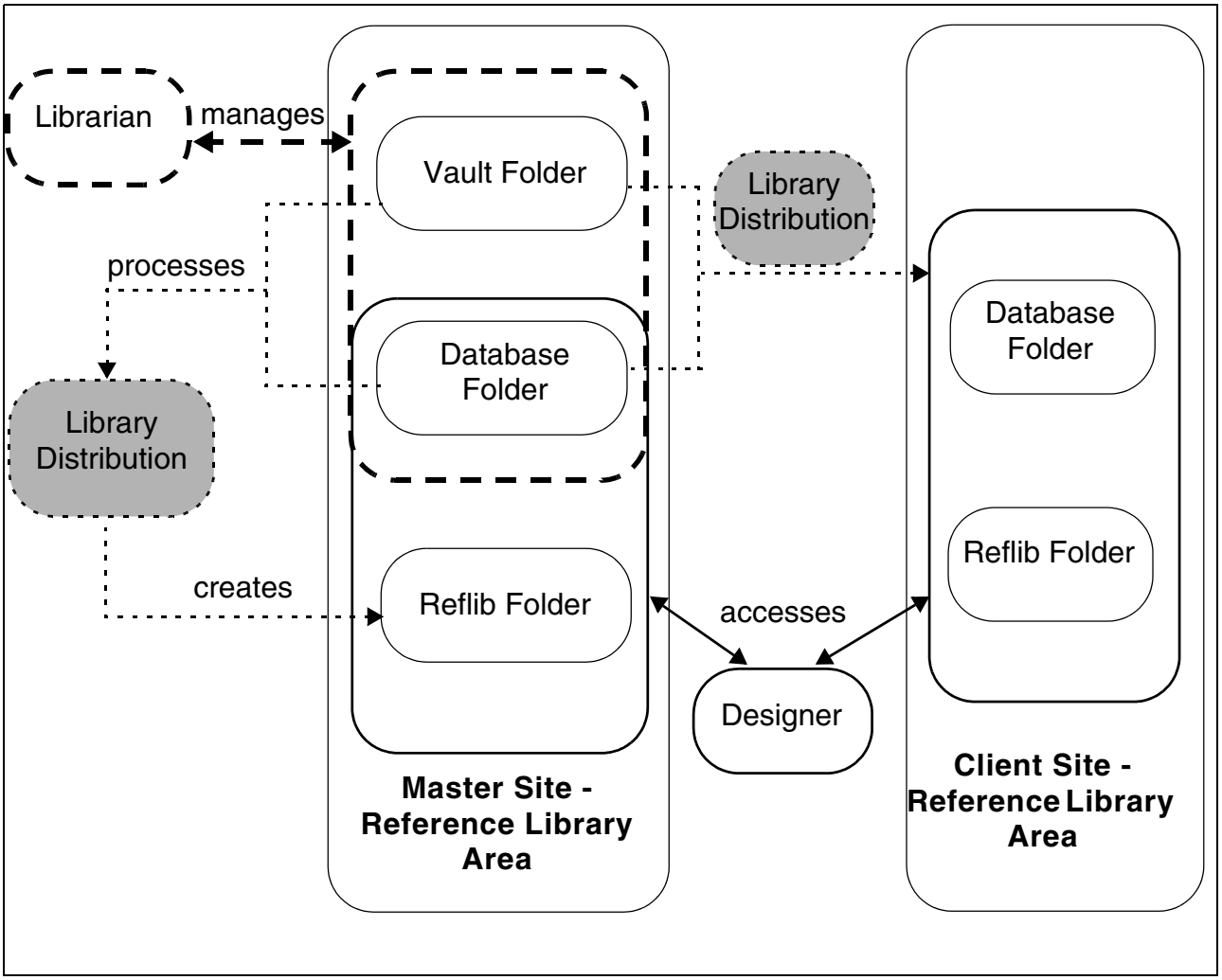
Table 2-2 Configuration Settings for Single Site

File Name	Field Name	Default Value
	catgen	`on`
	adwcisexport	off
	genoptionset	on
fetch_dump.ini	No changes are required.	

Configuration for Multiple Sites

This configuration setting implies that you have a separate master site and client sites. In this configuration, client sites replicate the designer data from the master site in the client site's reference library area.

The following diagram explains the configuration for multiple sites:



The configuration settings in this case can be classified into the following two:

Case 1, where:

- Master site contains both the librarian and designer data. In this case, both server and client processes are run on the master site. Thus, designers can work using this site's reference library area.
- Client site represents the designer site and the designer data is created by running the client processes.

The following table displays the configuration settings at the master and client site:

Table 2-3 Configuration Settings for Multiple Sites: Case 1

File Name	Field Name	Site Value at Master Site	Site Value at Client Site
lib_dist.ini			
	[lib_dist_server]		
	genmodelhtml	`on`	`off`
	mkdump	`on`	`off`
	create_site_data	off	off
	[lib_dist_client]		
	fetch_dump	`on`	`on`
	install_dump	`off`	`on`
	install_model	`on`	`on`
	ptfgenerator	`on`	`on`
	catgen	`on`	`on`
	adwcisexport	off	off
	genoptionset	on	on
fetch_dump.ini			
	urlRoot	No changes are required	Change *Site Value* to point to the `index.html` file in the *`<master_reference_library>`*`\distribution\html` folder.

Case 2, where:

- Master site contains only librarian data. Designers cannot connect to or work on the master site. You cannot run any of the client processes on this site.
- Client site represents the designer site, which implies that all client processes are run on this site, thus creating the data to be accessed by the designers.

The following table displays the configuration settings at the master and client site:

Table 2-4 Configuration Settings for Multiple Sites: Case 2

File Name	Field Name	Site Value at Master Site	Site Value at Client Site
lib_dist.ini			
	[lib_dist_server]		
	genmodelhtml	`on`	`off`
	mkdump	`on`	`off`
	create_site_data	off	off
	[lib_dist_client]		
	fetch_dump	`off`	`on`
	install_dump	`off`	`on`
	install_model	`off`	`on`
	ptfgenerator	`off`	`on`
	catgen	`off`	`on`
	adwcisexport	off	off
	genoptionset	off	on
fetch_dump.ini			
	urlRoot	No changes are required	Change *Site Value* to point to the `index.html` file in the *`<master_reference_library>`*`\distribution\html` folder.

Note: No changes are required in other configuration files.

Configuration for Site-Specific Library Distribution

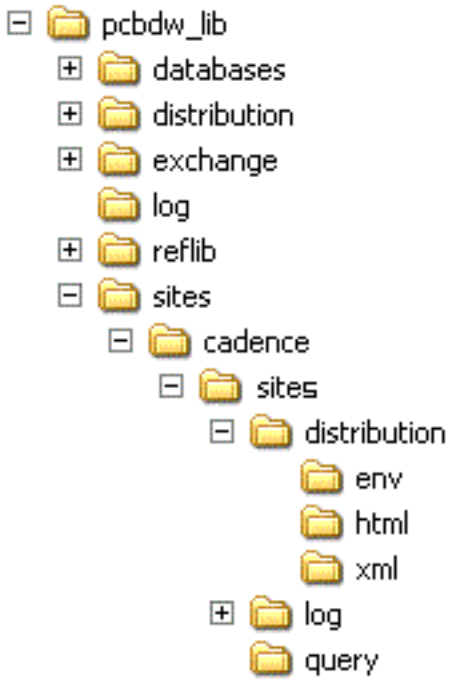
In a multi-site (separate master and client sites) environment, some client sites might not require all the component libraries that are published at the master site; you can choose to export only a partial library to such sites. You might also want to develop and publish site-specific parts and models.

With site-specific library distribution, you can publish selected parts, models, and/or libraries.

To configure the master site, do the following:

1. Navigate to the *<master_reference_library>* location.

   The following figure shows an example of a *<master_reference_library>* called *<pcbdw_lib>*.



   In the *<master_reference_library>*, create a new folder, `sites`, and its subfolders based on the folder structure shown in the following example, where:

   - *cadence* is the name of *<company>*.
   - *site5* is the name of *<restricted_client_site>*.
   - `distribution`, `log`, and `query` are subfolders within the *<restricted_client_site>* folder.

- `env`, `html`, and `xml` are subfolders within the `distribution` folder.

Important

The *<company>* and the *<restricted_client_site>* created in the *<master_reference_library>* must be identical to the name of the restricted client site.

2. Copy all the files from *<installation_directory>*\pcbdw_lib\distribution\env\sites to *<master_reference_library>*\sites*<company>**<restricted_client_site>*\distribution\env.
3. Open Allegro EDM Flow Manager.
4. Open any library project.
5. Launch Database Editor.
6. Choose *Search – Parts* or *Search – Model – <Model Type>*.
7. Specify the search criteria using the *Attributes* and/or *Properties* tab.
8. Search for the parts and/or models to be selectively distributed.
9. Choose *Search – Save Search Criteria* to save the query.

   You will find the saved criteria as a *<saved_query_name>*.xml file at the following location:

   *<current_project_directory>**<project_name>*\atdmdir\search

   Note: You can save multiple query files.
10. Copy these .xml files to the following location in the `sites` folder:

    <master_reference_library>\sites*<company>**<restricted_client_site>*\query

Changes in Configuration Files

Modify the configuration files at both master and client sites according to the following two cases:

Case 1, where:

1. Master site contains both the librarian and designer data. In this case, both server and client processes are run on the master site. Thus, the designers can work using this site's reference library area.
2. Client site and/or restricted client site represents the designer site and the designer data is created by running the client processes.

The following table displays the settings required at both the sites:

Table 2-5 Configuration Settings for Case 1

File Name	Field Name	Site Value at Master Site	Site Value at Client Site
lib_dist.ini			
	[lib_dist_server]		
	genmodelhtml	on	off
	mkdump	on	off
	create_site_data	on	off
	[lib_dist_client]		
	fetch_dump	on	on
	install_dump	off	on
	install_model	on	on
	ptfgenerator	on	on
	catgen	on	on
	adwcisexport	off	off
	genoptionset	on	on
fetch_dump.ini			

Table 2-5 Configuration Settings for Case 1

File Name	Field Name	Site Value at Master Site	Site Value at Client Site

	urlRoot	No changes are required	For complete library distribution:
			`urlRoot = file:///<master_reference_library>/distribution/html/index.html`
			For site-specific library distribution:
			`urlRoot = file:///<master_reference_library>/sites/<company>/<restricted_client_site>/distribution/html/index.html`

Case 2, where:

1. Master site contains only the librarian data. The designer cannot connect or work on the master site. You cannot run any of the client processes on this site.
2. Client site and/or the restricted client site represents the designer site, which implies that all client processes are run on this site, thus creating the data to be accessed by the designers.

The following table displays the settings required at both the sites:

Table 2-6 Configuration Settings for Case 2

File Name	Field Name	Site Value at Master Site	Site Value at Client Site
lib_dist.ini			
	[lib_dist_server]		
	genmodelhtml	`on`	`off`

Table 2-6 Configuration Settings for Case 2

File Name	Field Name	Site Value at Master Site	Site Value at Client Site
	mkdump	on	off
	create_site_data	on	off
	[lib_dist_client]		
	fetch_dump	off	on
	install_dump	off	on
	install_model	off	on
	ptfgenerator	off	on
	catgen	off	on
	adwcisexport	off	off
	genoptionset	off	on
fetch_dump.ini			
	urlRoot	No changes are required	For complete library distribution: urlRoot = file:///*<master_reference_library>*/distribution/html/index.html For site-specific library distribution: urlRoot = file:///*<master_reference_library>*/sites/*<company>*/*<restricted_client_site>*/distribution/html/index.html

Note: No changes are required in other configuration files.

Distributing Libraries

As the Pulse Master node administrator, you can distribute libraries in various ways depending on your setup.

Single-user environment (design and library data managed on the same machine)

Run the `lib_dist` utility, which automatically distributes libraries at the master and client sites at the same time.

1. Open the Allegro EDM System Console in any of the following ways:
   - Choose *Start – All Programs – Cadence Release 17.4-2019 – Allegro EDM Products – Allegro EDM System Console*.
   - Choose *Start – All Programs – Cadence Release 17.4-2019 – Allegro EDM Products – Allegro EDM Flow Manager –* `<any_library_project>` *– Tools – Allegro EDM System Console*.
   - Open a Windows command prompt, navigate to the folder containing the `<startworkbench>` script, type `<startworkbench> prompt` and press Enter.
2. Type the `lib_dist` command in Allegro EDM System Console.

   To know about the configuration settings and options, see, Library Distribution Setup and Site Configuration Options, respectively.

Multi-Site Environment

Run `lib_dist_server` to distribute libraries at the master site and the `lib_dist_client` for the client sites.

You can run both commands as a Cron job (on Unix) or as a scheduled task (on Windows) at regular intervals.

Ensure that the client site Cron job or scheduled task start after the master site Cron jobs are completed.

Distributing Libraries to the Master and Client Sites Separately

Running the individual utilities manually involves two sets of processes:

- Server Process: This includes running the `genmodelhtml, mkdump,` and `create_site_data` (only for site-specific library distribution) commands at the Allegro EDM System Console.
- Client Process: This includes running the `fetch_dump, install_dump, install_model, ptfgen -global, catgen, and genoptionset` commands at the Allegro EDM System Console.

Note: If you are running the individual utilities manually, the configuration settings in the `lib_dist.ini` file are ignored.

Important

Running these utilities individually is not advised unless really required. Instead of running these utilities individually, it is recommended that the `lib_dist, lib_dist_server,` or `lib_dist_client` utilities be used.

Adjusting Memory Requirements for Library Distribution

If you are prompted about insufficient memory when running any of these procedures, adjust the memory requirements for library distribution. Add or modify the following row in the `ptfgen_client` section of the `<PCBDW_LIB>\distribution\env\lib_dist.ini` file:

```
jvmargs=-Xmx3500m
```

The `jvmargs` directive value must be the same as that of `JVM_ARGS` in the `settings.ini` file of the Pulse Master node.

Library Distribution Utilities

The `lib_dist` utility is a batch file that executes various utilities one by one. These utilities have specific functions and properties that can be modified through their configuration (`.ini`) files.

Table 2-7 List of Library Distribution Utilities

Step	To...	Run...	At the...	To know more, see...
1	Identify the latest version of models ready to be published to the client sites and publishing them in the form of HTML files to be accessed by the client sites.	`genmodelhtml`	Master Site	genmodelhtml
2	Create a dump of the database.	`mkdump`	Master Site	mkdump
3	Identify and create a dump and HTML files for parts and/or models to be published for a restricted client site.	create_site_data	Master Site	create_site_data
3	Identify the models to be imported to the client site and import the data.	`fetch_dump`	Client Site	fetch_dump
4	Install the imported database dump from the master site.	`install_dump`	Client Site	install_dump
5	Install the models imported from the master site to the client site's reference library area (`reflib`).	`install_model`	Client Site	install_model
6	Generate and update the Part Table File (PTF) based on the part information available in the database.	`ptfgen -global`	Client Site	ptfgen
7	Update the library category files.	`catgen`	Client Site	catgen

Table 2-7 List of Library Distribution Utilities

Step	To...	Run...	At the...	To know more, see...
	Create a Capture CIS database configuration file and the database (`.db`) file of Capture CIS part data.	adwcisexport	Client Site	adwcisexport
8	Generate the `ppt_optionset.dat` file.	`genoptionset`	Client Site	genoptionset

Site-Specific Library Distribution

To distribute a partial library to client sites, complete the configuration for site-specific library distribution (see Configuration for Site-Specific Library Distribution) then run the following procedure:

1. Run the `lib_dist` or `lib_dist_server` command at the Allegro EDM System Console of the master site.
2. Run the `lib_dist_client` command at the Allegro EDM System Console of the client site.

   The `reflib` folder at the client site contains only the selected parts and/or models (as saved in the query files).

Master Site Utilities

To distribute libraries at the PUlse Master node, the following utilities are run in sequence:

1. `genmodelhtml`
2. `mkdump`
3. `create_site_data`

genmodelhtml

1. Modifies the *Distribution Status* of all objects in the component database from `Pending Distribution` to `Distributed`
2. Creates HTML files for all available distributed models and the database dump. One HTML file is created for each tool, tool version, and model type combination. Also creates an `index.html` file that contains links to the tool-model specific HTML files. The client sites connect to this `index.html` file for fetching the data from the master site.
3. Generates HTML files:
   - only for model types that have associated archive files.
   - for models whose development state is defined in the `genhtml.ini` file.

The fields in the *genhtml.ini* entry are:

- *model_status*: Lets you specify the development states of the models to be considered for library distribution. The valid values are: `Preliminary, Checked-in / To be Verified, Verified, Flow Verified, Pre Released,` and `Released`. The default value is: `Released, Pre Released, Deleted`

  For example, if the value of the *model_status* field is defined as `Released`, then the HTML links are generated for models which are in the Released state.

Modifying these values is not recommended because incorrect information can be published to the client sites.

- *dir_http_html*: Lets you specify a direct file-system-based location for HTML model files on the local server. Make sure that the master site has write permission if this is a mapped network location.
- *dir_default_html*: Lets you specify a default file-system-based location for the HTML model files if not specified in the *dir_http_html* field.

- *max_mem:* Lets you specify the maximum memory size to be utilized by the `genmodelhtml` utility. This value should be greater than 64MB. The default value is 900MB.

After the tasks run by the `genmodelhtml` command are successfully completed, you are prompted to run the next utility. See *<Pulse Master Home>*`\log\genmodelhtml.log` for errors, if any, during the `genmodelhtml` process.

mkdump

Creates a dump of the master site database and the `dump.html` file in the *<Pulse Master Home>*`\distribution\html` directory, which contains a link to the physical location of the database file.

After the tasks run by the `mkdump` command are successfully completed, you are prompted to run the next utility.

See *<Pulse Master Home>*`\log\mkdump.log` for errors, if any, during this process.

create_site_data

You need to run this utility only for Site-Specific Library Distribution. This utility, on the basis of data selected and saved in the query file, does the following:

1. Creates a dump of the master site database.
2. Creates the `dump.html` file in the *<master_reference_library>*`\sites\`*<company>*`\`*<restricted_client_site>*`\distribution\html` directory, which contains a link to the physical location of the database file..
3. Generates HTML files for each tool, tool version, and model type combination.
4. Creates an `index.html` file that contains links to the tool-model specific HTML files at:

   <master_reference_library>`\sites\`*<company>*`\`*<restricted_client_site>*`\distribution\html\index.html`

When you type `create_site_data` at the Allegro EDM System Console, the steps to configure site-specific library distribution are listed if no query files are found at: *<master_reference_library>*`\sites\`*<company>*`\`*<restricted_client_site>*`\query`

To check for any errors during this process, see the following files:

- *<master_reference_library>*\log\create_site_data.log
- *<master_reference_library>*\sites*<company>**<restricted_client_site>*\log\genmodelhtml.log
- *<master_reference_library>*\sites*<company>**<restricted_client_site>*\log\sitedump.log

To learn how to configure and run this utility, see Configuration for Site-Specific Library Distribution and Site-Specific Library Distribution respectively.

Client Site Utilities

Client-site library distribution utilities are run in the following sequence:

1. `fetch_dump`
2. `install_dump`
3. `install_model`
4. `ptfgen`
5. `catgen`
6. `adwcisexport`
7. `genoptionset`

fetch_dump

This utility finds models to import by identifying differences between the models present on the client site and master site. Once identified, the data is fetched from the master site to the client site.

In case of configuration changes to be made for multiple sites, as described in Configuration for Multiple Sites and Configuration for Site-Specific Library Distribution, each client site is required to configure the *urlRoot* entry in the `fetch_dump.ini` file to point to the master site `index.html` file.

The various fields in the *fetch_dump.ini* entry are:

- *nbLinkMax*: Indicates the maximum number of models that can be fetched from the client site in one library distribution run.
- *default_nbLinkMax*: Indicates the default number of models to distribute in a session.
- *default_urlRoot*: Allows you to specify the default URL, if not specified in the *urlRoot* field.
- *urlRoot*: Allows you to specify the URL for the physical location containing the `index.html` file. For example, `file:///$env(PCBDW_LIB)/distribution/html/index.html`

Type the `fetch_dump` command at the Allegro EDM System Console.

To check for any errors during the process, see the *<Pulse Master Home>*`\log\fetch_dump.log` file.

For information on various modes of connecting to the master site, see Modes of Establishing Connection between Master and Client Sites.

adwcisexport

This utility exports the OrCAD Capture CIS database for designers to work with parts in Capture. On running this command, a CIS database configuration file (`CISParts.dbc`) and a database file (`CISParts.db`) will be created.

To check for any errors during the process, see the `<Pulse Master Home>\log\adwcisexport.log` file.

install_dump

This utility updates the database with the data that has been fetched from the master site (using the `fetch_dump` utility) onto the client-site database server.

After the fetch_dump process is run, type the `install_dump` command at the Allegro EDM System Console.

To check for any errors during the process, see the *`<Pulse Master Home>`*`\log\install_dump.log` file.

install_model

The models to install are available in archived form in the client site's integration area. This utility installs the models received on the client site specifically in the `reflib` folder of the reference library structure.

After the install_dump process is run, type the `install_model` command at the Allegro EDM System Console.

To check for any errors during the process, see the *`<Pulse Master Home>`*`\log\install_model.log` file.

ptfgen

A PTF contains the physical properties of the parts. A PTF can have part tables for one or multiple part names. There can be multiple parts associated with each part name with different physical property values. For each part, a PTF row is created in the PTF.

In your database, this PTF data is stored in the form of key-value pairs for various parts. To start using this data, you need to create PTF files from the database. The PTF generator utility allows you to generate PTF files for the latest versions of parts available in the database. This process can be performed by the librarian or site administrator.

After the install_model process is run, type the `ptfgen -global` command at the Allegro EDM System Console.

The PTF generator utility generates a global PTF file by reading part information from the database. It also caches the PTF in database on parts to optimize subsequent run of the `ptfgen` command. When you run the `ptfgen -global` command, it finds the modified parts in the database and generates the PTF for modified parts based on key-value pair entries in database while for rest of the parts, it reads the PTF from cached entries in the database.

Note: If you do not add a value to a newly added property for a part, it does not appear in the PTF when you run `ptfgen`.

If you have made any changes to the physical part table, and want to regenerate the PTF from property information in database and recreate the PTF cache in the component database, then run the `ptfgen -global -rewrite2db` command at the Allegro EDM System Console. The time required by this activity depends on the number of parts in the component database.

If the `ptfgen` command fails because of insufficient memory, use the `-memoptimizedmode` argument with the `ptfgen` command. The time required to generate the PTF file will be slightly higher.

If the `ptfgen -global -memoptimizedmode` command fails, then perform the following steps:

1. Add or modify following row in the `ptfgen_client` section of `<PCBDW_LIB>\distribution\env\lib_dist.ini` file:

   ```
   jvmargs=-Xmx3500m
   ```

   Ensure that the value of the `jvmargs` directive is same as that of `JVM_ARGS`, specified in the `settings.ini` file for Allegro EDM Server.

2. Run the `ptfgen -global` command.

Even after adding this directive, if you get out-of-memory error, perform the following steps:

1. Add the `jvmargs` directive, if not already done.
2. Run the `ptfgen -global -memoptimizedmode` command.

3. If you do not get any error, then modify the `ptfgen_client` section in `lib_dist.ini` as:

```
command = ptfgen -global -memoptimizedmode
```

To check for any errors or warnings encountered during the PTF generation process, see the *<Pulse Master Home>*`\log\ptf.log` file. The ptfgen command has been enhanced to compare `part_table.ptf` and `part_table_mech.ptf` against parts/block parts and mechanical parts available in the database respectively. The differences in terms of missing parts, block parts, and mechanical parts are reported in the `ptf.log` file. These differences also include parts and schematic models that have any error and already reported in the log file.

To specify information such as the PTF backup location, delimiting character to be used in the PTF files, the `cds.lib` location, and the target location of the PTF files, you need to configure the PTF generation process, that is, modify the `ptf.ini` file.

The fields in the *ptf.ini* entry are:

- *logfilepath*: Lets you specify the location where the log file will be stored. The default value is: `$env(PCBDW_LIB)/log/ptf.log`
- *separatorcharacter*: Lets you specify the delimiting character in PTF files. The default character is `|`.

It is recommended that you change this character with caution.

If you change this value, then run the `ptfgen -global -rewrite2db` command to generate the PTF afresh and recreate the cached entries in database for PTF information.

- *objectstatus*: Lets you specify the development state (`Pre Released, Released`) of the parts for which PTF file is generated.

Changing this directive is not recommended because it can lead to generating an incorrect PTF, which may not be usable by designers.

- *blockptflocation*: Lets you specify the location of the block PTF file for the libraries. The default value is: `$env(PCBDW_LIB)/reflib/model_block/part_table.ptf`
- *cdslibpath*: Lets you specify the location of the `cds.lib` file. The default value is: `$env(PCBDW_LIB)/reflib/model_sym/cds.lib`

- *ptfbackuplocation*: Lets you specify the location of the backup folder where previous versions of global PTFs are backed up. The default value is: `$env(PCBDW_LIB)/distribution/backup`

  You will find the previous versions of the PTF files at the location, *<PTF_backup_folder>*`/global`. For example, in this case you will find the files at `$env(PCBDW_LIB)/distribution/backup`.

- *globalptflocation*: Lets you specify the location of the global PTF file for the libraries. The default value is: `$env(PCBDW_LIB)/reflib/model_sym/part_table.ptf`

- *mechptflocation*: Lets you specify the location of the PTF for mechanical parts. The default location is: `$env(PCBDW_LIB)/reflib/model_sym/part_table_mech.ptf`

catgen

This utility helps you to generate and update the library-level category (`.cat`) files for schematic and block models in the database. These category files are based on the classification of models in the database.

To ensure that you have the up-to-date library categories, type the `catgen` command at the Allegro EDM System Console.

To check for any errors during the process, see the *<Pulse Master Home>*`\log\catgen.log` file.

To modify the log file path, you need to modify the c*ategory.ini* entry for a site. The following table contains the fields in the c*ategory.ini* entry.

Table 2-8 Settings in category.ini File

Field Name	Description	Default Value
logfilepath	Location of the log file to be created	`$env(PCBDW_LIB)/log/catgen.log`
cdslibpath	Path to the `cds.lib`	`$env(PCBDW_LIB)/reflib/model_sym/cds.lib`

In this case, `$env(PCBDW_LIB)` is the environment variable `PCBDW_LIB`.

genoptionset

This utility helps you to generate the ppt_optionset.dat file. This file controls the default display settings, such as format and visibility, for physical properties in the schematic. The file defines the display settings of key and injected properties in the schematic by setting the Annotate to Design and Visibility fields for a particular property.

To generate the file, type the `genoptionset` command at the Allegro EDM System Console.

To check for any errors during the process, see the *<Pulse Master Home>*`\log\genoptionset.log` file.

To modify the log file path, you need to modify the c*ategory.ini* entry for a site. The following table contains the fields in the c*ategory.ini* entry.

Table 2-9 Settings in category.ini File

Field Name	Description	Default Value
logfilepath	Location of the log file to be created	`$env(PCBDW_LIB)/log/genoptionset.log`
cdslibpath	Path to the `cds.lib`	`$env(PCBDW_LIB)/reflib/model_sym/cds.lib`

In this case, `$env(PCBDW_LIB)` is the environment variable `PCBDW_LIB`.

Modes of Establishing Connection between Master and Client Sites

For multi-site configuration, you connect the master and client site in one of the following ways:

- File System
- HTTP/HTTPS
- FTP

The entry in the `fetch_dump.ini` file to point to the master site HTML file is based on the connection method agreed on between the master and client sites.

Using File System

The configuration file protocol is used to establish a connection between the master site and client site for fetching data in two ways:

- Mapped Drive: The client system can map the *<master_reference_library>* folder to a local drive and create a corresponding entry in the `urlRoot` entry of the `fetch_dump.ini` file.

  For example:

  ```
  [fetch_dump]

  urlRoot = file:///<mapped drive name of master reference
  library>/distribution/html/index.html
  ```

- UNC Path: create an entry with a UNC path in the `urlRoot` entry of the `fetch_dump.ini` file; the client will not connect the *<master_reference_library>* folder as a local mapped drive.

  For example:

  ```
  [fetch_dump]

  urlRoot = file://///<master_library_server_hostname>/
  <shared_reference_library_folder>/distribution/html/
  index.html
  ```

Using HTTP

In this method, you do not require a direct file system connection between the clients and the server; files are transferred over HTTP. However, a Web server such as Apache or Tomcat is needed to serve the HTML files on the server so that clients can locate them.

For library distribution using the HTTP method, `fetch_dump.ini` must have:

```
[fetch_dump]

urlRoot = http://<master_server:port number>/libs/distribution/
html/index.html
```

Ensure that:

- In Windows, `PULSE_LIBDIST_TOKEN` is set as a system environment variable in all the machines that will run `lib_dist` or `lib_dist_ client`. On Linux, set `PULSE_LIBDIST_TOKEN` as a variable. The value of the variable is the access key generated in Pulse Service Manager. See the *Defining Library Management Settings for Pulse Master Node* section of *Cadence Pulse Configuration Guide* for details.
- The HTTP Web server on the master site must publish the top-level folder of its *<Pulse Master Home>* (parent of `distribution` folder), often known as `document root`, `htdocs`, or `wwwroot`.
- The Web server alias (for example, `libs)` must be defined to point to the correct file system location at the master location. In the above example, `libs` must be defined to point to *<Pulse Master Home>*.

Note: You must configure the same connection method for all client sites rather than attempt to have some sites using the file system and some using HTTP.

Using FTP

You can also establish the connection between the master and client site is using File Transfer Protocol (FTP). In this method, a FTP server is started on the master site publishing the *<Pulse Master Home>* area of the master site and the client site can then connect to this FTP server by configuring `fetch_dump.ini` as follows:

```
[fetch_dump]

urlRoot = ftp://<master_server:port number>/libs/distribution/
html/index.html
```

Ensure that:

- The FTP server on the master site must publish the top-level folder of its `<Pulse Master Home>` (parent of `distribution` folder), often known as `ftp root`.
- The FTP server alias (for example, `libs)` must be defined to point to the correct file system location at the master location. In the above example, `libs` must be defined to point to `<Pulse Master Home>`.

Index

Symbols

B

C

F

G

I

K

L

M

O

P

R

S

V