EDA – Gateway – Server and Client - Components

Installation Document



Document Details

|  |  |
| --- | --- |
| Installation Instruction Document | |
| Project Owner |  |
| Process Title |  |
| Document Author |  |
| Document Version | **1.0 (DRAFT)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document Edit History | | | | |
| Version | **Description** | **Name** | **Role** | **Date** |
| V1.0 | Initial draft | **Sanjeev Beemidi** | **EDA Integrator** | **9 Jun 2022** |

Contents

[1. Introduction 4](#_Toc105989935)

[1.1. Purpose 4](#_Toc105989936)

[1.2. Scope 4](#_Toc105989937)

[1.3. Pre-Installation Requirements 5](#_Toc105989938)

[1.4. Pre-Installation Checklist 5](#_Toc105989939)

[1.5. Installing EDA – Server Components 5](#_Toc105989940)

[1.6. Installing – EDA – Client Components 33](#_Toc105989941)

[1.7. Installing – Cadence – CIS and Allegro Components 33](#_Toc105989942)

[1.8. Post-Installation Checklist 33](#_Toc105989943)

[1.9. Troubleshooting 33](#_Toc105989944)

[1.10. Reference Documents 34](#_Toc105989945)

1. **Introduction**

## Purpose

Currently, MHP is assisting Moog with ECAD Integration efforts to their PLM system. The goal of this initiative is to establish and implement a new way of working in accordance with industry best practice that aligns with a working system architecture allowing warranted data flow between Moog’s ECAD tools, warranted inputs, derived datasets, and the PLM system. These processes and architecture shall appease identified criticalities and requirements established during the current and proceeding efforts.

Reoccurring challenges, gaps, and issues consisted of heavy data chase, lack of reuse, process conformance, and non-coordinated sets of information between tools. Overall, Moog lacks a true close looped, automated, end to end system structure that could give them value added time back to their employees and to market process, reduce overall cost, and enhance information traceability.

This document will detail the technical process that shall align with architecture and business processes necessary to appease requirements and adhere to best practices. MHP will suggest the utilization of Master Data Management methods with the platform of Teamcenter hosting such master data. As we have identified current gaps within the current system and processes, a new suggested procedural framework is detailed. Considering, the complexity, capabilities, and efforts required of systems and resources explained below are three options with foreseen dependencies, challenges, and goals. Such options shall give the implementation team the ability to continue on the designated path established and refine a detailed package for implementation as external prerequisites are satisfied, thus not pigeonholing our approach to a single solution avoiding single points of failures and unidentified bottlenecks at this time.

The technical design and solution implementation document provide the technical specifications for the design and the solution implementation details as part of ECAD systems integrations with the Moog’s current PLM systems.

This document is also created as part of the initial architecture review and the solution implementation details before the development of the solution implementation.

This document also serves as a blueprint for development and maintenance of the ECAD systems with PLM systems and should be modified at any time to reflect approved changes made to the process and solution implementation code.

## Scope

|  |
| --- |
| Description |
| These procedures are to lay the framework for the new way of working at Moog in regard to the ECAD PLM integrations considering the PWB development process. |
| Insights gained based on several interviews, questionnaires, meetings, and confirmations from various representatives of the roles associated with this development. |
| Propose a technical process that shall align with architecture and business processes necessary to appease requirements and adhere to best practices. |
| Proposed to implement the utilization of Master Data Management methods with the platform of Teamcenter hosting such master data |
| As we have identified current gaps within the current system and processes, a new suggested procedural framework is detailed. |
| Considering, the complexity, capabilities, and efforts required of systems and resources explained below are three options with foreseen dependencies, challenges, and goals. Such options shall give the implementation team the ability to continue on the designated path established and refine a detailed package for implementation as external prerequisites are satisfied, thus not pigeonholing our approach to a single solution avoiding single points of failures and unidentified bottlenecks at this time. |

Apart from the above activities, any other activity is out of scope for this work.

## Pre-Installation Requirements

The following prerequisites and requirements must be satisfied in order for the to install successfully.

## Pre-Installation Checklist

The following prerequisites and requirements must be satisfied in order for the to install successfully.

Use the following checklist to ensure that your servers are ready for the components to be installed.

# Checklist Item

## Installing EDA – Server Components

The following prerequisites and requirements

#### Add EDA support on the Teamcenter server for design management

EDA server support adds the EDA data model to the existing Teamcenter data model. Before installing the EDA server support, you must install the Teamcenter server.

1. From your Teamcenter environment, start Teamcenter Environment Manager (TEM).
2. Select the **Configuration Manager** option to perform maintenance on an existing installation and click **Next** until the **Select Features** dialog box appears.
3. In the **Select Features** dialog box:
   1. Select the following options:
      * Choose **Extensions**→**Mechatronics Process Management** and select **EDA Server Support**.

* Choose **Extensions**→**Mechatronics Process Management** and select **EMPS-Foundation**.
* Specify **Active Workspace** server extensions features for EDA. These are available in the **Features** panel in **Teamcenter Environment Manager (TEM),** under **Base Install→Active Workspace→Server Extensions→EDA Server Support** for Active Workspace.
* Specify **Active Workspace** client extensions features for EDA. These are available in the **Features** panel in **Teamcenter Environment Manager (TEM),** under **Base Install→Active Workspace→Client→Electronic Design Automation** for Active Workspace.
  1. In the **Installation Directory** box, type the location where you want to install Teamcenter (TC\_ROOT).
  2. Click **Next**.

1. Enter information as needed in the subsequent panes.
2. In the **Confirm Selections** dialog box, click **Next**. The EDA server components are installed.

The **Add/Remove Components** dialog box confirms if the installation is successful. If the installation is unsuccessful, click **Show Details** and proceed as needed.

1. Obtain the license file that includes EDA licensing and EDA library license (for library support) and install it on the server.

After adding the EDA server support to your Teamcenter environment, you must verify your installation by checking whether the EDA data types are added to Teamcenter.

#### Verify your installation

Verify the addition of the EDA data types to the server:

1. Run the Teamcenter rich client.
2. In My Teamcenter, choose **File**→**New**→**Item**. The **New Item** dialog box is displayed.
3. Verify that the EDA, EDAComp, EDASchem, and EDACCABase item types are added to the list of types you can create.
4. Select one of the EDA item types and create an instance.
5. Verify that the item instance is created.

#### Install and set up Common Integration Services

Teamcenter EDA uses *Common Integration Services* (*CIS*) to manage ECAD application data in Teamcenter.

The CIS framework uses an *Integration Definition file* to define the default Teamcenter EDA objects that are created during the save-as, save, check-in, and revise operations. These EDA objects are PCA, schematic, PWB, variants, derived items, derived datasets, design dataset, viewable datasets, and BOM components. This file is imported to Teamcenter when CIS services are deployed on the Teamcenter server from the EDA install kit.

The CIS services require an *integration definition file* to be imported to the Teamcenter server. This file is used to define the relation between these EDA objects whether they are related with each other using *Generic Relationship Management* (GRM) relationship or *BOM View Revision* BVR relationship. The relationship behavior is defined OOTB for all the EDA objects with the default Teamcenter object types.

To install and set up common integration services on the Teamcenter server, you must:

1. Update the Teamcenter installer to include the CIS feature.
2. Install Common Integration Services using TEM.

#### Update the Teamcenter installer to include Common Integration Services

1. Copy the EDA kit to a temporary location.
2. Run Teamcenter Environment Manager (TEM) from the *%TC\_ROOT%\install* folder on the Teamcenter server as an administrator.
3. Select **Updates Manager**, click **Next**.
4. For the **Update kit location** field, browse to the *<EDA\_KIT>\server\_updates\cif\<PLATFORM>* folder. Click **Next**.
5. Proceed through the remaining panels in TEM, entering the required information for the features you selected.
6. When TEM displays the **Confirmation** panel, click **Start** to begin the installation.
7. When the installation is complete, close TEM.

#### Install the Common Integration Services

1. Run Teamcenter Environment Manager (TEM) from the *%TC\_ROOT%\install* folder on the Teamcenter server as an administrator.
2. In the **Maintenance** panel, choose **Configuration Manager**.
3. In the **Configuration Maintenance** panel, choose **Perform maintenance on an existing configuration**.
4. In the **Old Configuration** panel, select the configuration you want to modify.
5. In the **Feature Maintenance** panel, select **Add/Remove Features**.
6. In the **Features** panel, select the following features:

##### Extensions→Model Management→Server→Common Integration Framework

1. Proceed through the remaining panels in TEM, entering the required information for the features you select.
2. When TEM displays the **Confirmation** panel, click **Start** to begin the installation.

When the installation is complete, close TEM.

#### Populate the Teamcenter database with ECAD components

Many companies control the ECAD component library items that can be used to create BOMs by checking in the approved components for use to the database. Thereafter, you can create BOMs that use these components.

You can configure how to:

##### Control ECAD component library items for BOM creation.

To control component library items for BOM creation, when EDA is first installed, the **EDA\_CheckComponentExistence** is set to **1** (true). With this setting, you cannot save design BOMs unless all the necessary components are already in Teamcenter.

##### Populate the database with ECAD components before the designer can save designs and before creating BOMs in Teamcenter using EDA.

You can initially populate the database with components by changing the **EDA\_CheckComponentExistence** preference to **0** (false). If this preference is set to **0**, any component that does not already exist in Teamcenter is automatically created when saving the BOM.

Caution:

Do not set this preference to **0** in a production environment.

##### To populate the database with ECAD components:

1. Set the **EDA\_CheckComponentExistence** preference:
   1. Run the rich client on the Teamcenter server and log on as an administrator (for example,

**infodba**).

* 1. In the My Teamcenter application, choose **Edit**→**Options**.
  2. At the bottom of the **Options** dialog box, click **Filters**.
  3. In the **Preferences** dialog box, search for the **EDA\_CheckComponentExistence** preference.
  4. Click **Edit** and change the value of the preference from **1** (true) to **0** (false).
  5. Click **Save** to apply the changes.

1. Locate or create designs that use the approved library components. Use the EDA **Save As** command to save the designs to the Teamcenter server and create BOMs. When the BOMs are created, the components are automatically added to the database.
2. Reset the **EDA\_CheckComponentExistence** preference to **1** (true) if you want to restrict users to using only the approved (and checked-in) components. This is the recommended setting in production environments.

However, if you wish to allow new components to be automatically placed in the database when users create BOMs, leave the **EDA\_CheckComponentExistence** preference as is, set to **0**.

This action bypasses business processes for component evaluation and approval.

#### Import new or updated EDA preferences in Teamcenter to support the latest EDA version

The EDA client requires the latest preferences to be imported in the Teamcenter environment for the client to work correctly. Before installing or updating the client, you must import preferences that are specified in an xml file included at the following EDA kit path:

*<EDA\_KIT>\server\_updates\edaserver\install\eda\_preferences.xml*

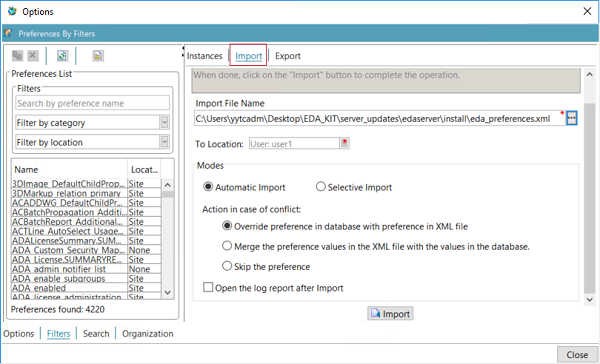
If the preferences are not updated since the Teamcenter base install, this file is not available in the EDA kit.

You can import these preferences in Teamcenter from the rich client or using the Teamcenter command prompt.

* + - Importing preferences in Teamcenter from the rich client:

1. Log on to Teamcenter and start the rich client.
2. Choose **Edit**→**Options** to display the **Options** dialog box.
3. Click **Filters** to open the preferences view.

Click **Import**.



1. Browse to the EDA kit location where the preferences file exists and select the *eda\_preferences.xml*

file for import.

1. Select **Site** from the **To Location** list.
2. Choose **Automatic Import**.
3. If the preference exists in Teamcenter, you can select any one of the **Action in case of conflict**

options to override, merge, or skip that preference.

1. Click **Import**.

#### Importing preferences in Teamcenter from the command prompt:

1. Open a Teamcenter command prompt.
2. Run the following command:

##### preferences\_manager -u=<user id> -p=<password> -g=dba -mode=import -scope=SITE - action=SKIP -file=<EDA\_KIT>\server\_updates\edaserver\install\eda\_preferences.xml

#### Import new or updated Active Workspace preferences in Teamcenter to support latest EDA version

The EDA client requires the latest EDA and Active Workspace preferences to be imported in the Teamcenter environment for the client to work correctly. Before installing or updating the client, you must import preferences that are specified in an xml file included at the following EDA kit path:

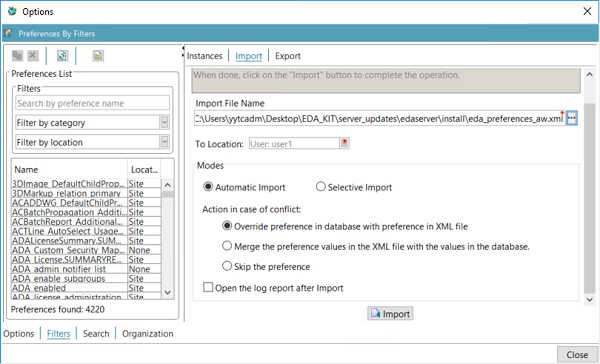
*<EDA\_KIT>\server\_updates\edaserver\install\eda\_preferences\_aw.xml*

If the Active Workspace preferences are not updated since the Teamcenter base install, this file is not available in the EDA kit.

You can import these preferences in Teamcenter from the rich client or using the Teamcenter command prompt.

#### Importing preferences in Teamcenter from the rich client:

1. Log on to Teamcenter and start the rich client.
2. Choose **Edit**→**Options** to display the **Options** dialog box.
3. Click **Filters** to open the preferences view.
4. Click **Import**.



1. Browse to the EDA kit location where the preferences file exists and select the

*eda\_preferences\_aw.xml* file for import.

1. Select **Site** from the **To Location** list.
2. Choose **Automatic Import**.
3. If the preference exists in Teamcenter, you can select any one of the **Action in case of conflict**

options to override, merge, or skip the preference.

1. Click **Import**.

#### Importing preferences in Teamcenter from the command prompt:

1. Open a Teamcenter command prompt.
2. Run the following command:

**preferences\_manager -u=<user id> -p=<password> -g=dba -mode=import -scope=SITE - action=SKIP -file=<EDA\_KIT>\server\_updates\edaserver\install\eda\_preferences\_aw.xml**

#### Modify the Active Workspace style sheet to include EDA derived data relation

The *commercial-off-the-shelf (COTS)* value of the EDA preferences used by Active Workspace to display the EDA item revision types is defined in the *Awp0ItemRevisionSummary* style sheet.

To allow EDA derived datasets to be displayed as ItemRevision attachments in Active Workspace, you must modify the *Awp0ItemRevisionSummary* style sheet on the Teamcenter server.

1. Log on to Teamcenter and start the rich client.
2. In the navigation pane, click **My Teamcenter**.
   1. Search for the *Awp0ItemRevisionSummary* dataset.
   2. Double-click the dataset.

The dataset is automatically checked out and the xml file is launched with the dataset.

1. In the *Awp0ItemRevisionSummary* style sheet, modify the **XRT\_files** page by appending the **EDAHasDerivedDataset** relation to the attachments page objectSet source attribute value as follows. The modified line is marked with a plus (+) sign on the left most column.

<page titleKey="attachments">

<section titleKey="tc\_xrt\_Files">

+. <objectSet

source="IMAN\_specification.Dataset,IMAN\_reference.Dataset,

+. IMAN\_manifestation.Dataset,IMAN\_Rendering.Dataset

,TC\_Attaches.Dataset,

+. IMAN\_UG\_altrep.Dataset,IMAN\_UG\_scenario.Dataset,I

MAN\_Simulation.Dataset,

+. EDAHasDerivedDataset.Dataset"

defaultdisplay="listDisplay"

+. sortby="object\_string"

sortdirection="ascending"><tableDisplay>

<property name="object\_string"/>

<property name="object\_type"/>

#### Map ECAD attributes with Teamcenter properties

#### How are ECAD attributes mapped with Teamcenter properties

Using mapping definition files, the ECAD part attributes can be stored in the Teamcenter database and displayed and modified both in the ECAD design tool and Teamcenter.

For the ECAD design tool, choose entries for design, project, and variant attributes such that they apply to the data types that represent the corresponding objects. Mapping definitions for variant objects must be associated with the specific object type that represents the variant BOM.

For details on how to map ECAD attributes, see the *Syntax for mapping attributes with Teamcenter Integration for NX* topic in the Application Administration documentation.

#### Customize ECAD attribute mapping

You can customize the ECAD attribute mappings in Teamcenter by exporting the attribute mappings to a file, editing the file, and importing the file back into Teamcenter.

1. Open a Teamcenter command prompt.
2. Change to the *bin* directory within TC\_ROOT. (TC\_ROOT is the installed location of Teamcenter.)
3. Run the **export\_attr\_mappings** command to export the mappings to a file, for example:

##### export\_attr\_mappings —file=test\_attribute\_mapping —u=infodba —p=infodba

1. Edit the mappings file to add EDA tool attributes.
2. Import the edited mapping file back into Teamcenter by running the **import\_attr\_mappings**

command, for example:

##### import\_attr\_mappings —file=test\_attribute\_mapping —u=infodba —p=infodba

The attribute mapping import and export process uses the PLM XML import/export mechanism. For more information, see the *PLM XML/TC XML Export Import Administration* documentation.

#### Sample ECAD design attribute mapping definitions

The below example defines dataset attribute mappings and variant attribute mappings. In this mapping definition, the Item level attribute mapping definition defines the ECAD variant attribute mappings and it is part of the dataset level attribute mapping definition.

ECAD variant mapping for Altium tool:

{ Dataset type="EDADesAltiumBrd"

ORGNAME : Item.GRM(IMAN\_master\_form,Item Master).user\_data\_1 ORGADDR1 : Item.GRM(IMAN\_master\_form,Item Master).user\_data\_2 "DES\_NAME" : ItemRevision.object\_name

"DES\_DESC" : ItemRevision.object\_desc

Doc : Item.GRM(IMAN\_master\_form,Item Master).user\_data\_3

{ Item type="Item"

"variant\_def\_list\_tc\_description" : ItemRevision.object\_desc / description="Variant Item Revision Description"

}

}

#### Configure derived data for ECAD design objects

#### Understanding how derived data works

Derived data contains information that is derived from an ECAD design and comprises derived items and datasets. Derived items represent parts, subassemblies, and tools. Derived datasets manage data files created by ECAD applications.

You can configure how derived data is created in Teamcenter EDA by using the *EDA Derived Data* folder in Business Modeler IDE, creating an EDA derived data configuration, and configuring how derived data files are named in Teamcenter. For example, a configuration can specify that when a schematic design is saved in Teamcenter EDA, a schematic drawing can be automatically generated from the schematic design and saved along with the schematic item

EDA business objects define the different types of derived data you can generate. To locate EDA business objects, use the **Find** button in the BMIDE view to search for all business objects containing the **EDA** string.

#### The following item types are children of the EDA business object:

|  |  |
| --- | --- |
| **Item Types** | **Description** |
| **EDACCABase** | Represents the common electrical CAD (ECAD) design data that is shared between variant circuit card assemblies (CCAs). It is used only for multiple CCA representations. |
| **EDAComp** | Represents electrical components contained in the CCA bill of materials (BOM). |
| **EDASchem** | Represents the electrical schematic item. |

**The following relationships are children of the ImanRelation business object:**

|  |  |
| --- | --- |
| **ImanRelation** | **Description** |
| **EDAHasDerivedDataset** | Identifies the associated dataset as a derived dataset. |
| **EDAHasDerivedItem** | Identifies the associated item as a derived item. |

#### How to use the EDA derived data editor to configure derived data in Teamcenter

Use the EDA Derived Data editor in the BMIDE to work with derived data configurations used by the Teamcenter EDA application. Teamcenter EDA integrates Teamcenter with ECAD applications that are used to design electronic components, such as circuit boards.

1. To access the **EDA Derived Data** editor, open the **Extensions**→**EDA Derived Data** folder, right-click an EDA derived data object, and choose **Open**.
2. To add an item to the EDA derived data object, click the **Add** button next to the **Configure Items**

table.

1. To add a dataset to the EDA derived data object, click the **Add** button next to the **Configure Dataset** table.

#### How to create an EDA derived data configuration

You can configure how derived data is created in Teamcenter by using the **EDA Derived Data** folder in Business Modeler IDE. After creating a derived data configuration, you can specify the name of the configuration in the **EDA\_DerivedDataConfigDefault** preference.

1. In BMIDE, open the **Extensions** folder.
2. In the **Extensions** folder, right-click **EDA Derived Data** and choose **New EDA Derived Data**. The **New EDA Derived Data** wizard is displayed.

In the **EDA Derived Data** dialog box, enter the following information and click **Next**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Name** | Specify a name you want to assign to the new derived data configuration.  This is the name used in the  **EDA\_DerivedDataConfigDefault** preference. |
| **Description** | Type a description for the new configuration. |

The **EDA Derived Data Configuration** dialog box in the wizard is displayed.

1. In the **EDA Derived Data Configuration** dialog box, set up how all EDA item and dataset types are to be handled for all contexts.
   1. Click the **Add** button next to the **Configure Items** table.

The **Add/Edit EDA Derived Item Configuration** dialog box is displayed.

In the **Add/Edit EDA Derived Item Configuration** dialog box, configure the derived items to be generated. For example, create separate rows for contexts such as schematic, PCB, simulation, and so on, including variations based on the what the parent is, such as **Schematic**, **CCA**, and **CCAVariant**. In this way, you set up how derived data is generated for all combinations of items.

In the **Add/Edit EDA Derived Item Configuration** dialog box, enter the following information and click **Finish**:

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Name** | Type the name that you want to assign to the derived item configuration.  This is the name displayed to the user on the Teamcenter EDA **Derived Item** dialog box during save operations. |
| **Context** | **Browse** to select a specific Teamcenter EDA application context from the contexts listed below. When users in Teamcenter EDA save derived data for the following specified data types, derived items are generated according to this configuration.   * all * pcb * pcb/simulation * schematic * schematic/pcb * schematic/simulation * simulation |
| **Prefix** | (Optional) Type a file name string to be attached to the beginning of the parent item ID to distinguish it as being generated by this configuration.  The resulting string, including the prefix and postfix, is used in the derived item user interface in Teamcenter EDA as the initial value for the **Derived Item ID** box and **Name** box. This can be overridden by the user. |
| **Postfix** | (Optional) Type a file name string to be attached to the end of the item ID to distinguish it as being generated by this configuration.  The resulting string, including the prefix and postfix, is used in the derived item user interface in Teamcenter EDA as the initial value for the **Derived Item ID** box and **Name** box. This can be overridden by the user. |
| **EDA Parent** | **Browse** to select the derived parent EDA business object to which the derived item is related. (Teamcenter EDA does not support attaching derived items under other derived items). |

|  |  |
| --- | --- |
| **Field** | **Description** |
|  | * **CCA** represents a circuit card assembly (CCA). * **CCABase** represents the common design data that is shared between variant circuit card assemblies (CCAs). It is used only for multiple CCA representations. * **CCAVariant** represents the variant design data for a circuit card assembly (CCA). This is the data that is used on top of the CCABase business object. * **PWB** represents a printed wire board (PWB). A PWB is the product of a schematic design and printed circuit board (PCB) layout design and holds all the printed wire board production data created by those designs. * **Schematic** represents the electrical schematic item. |
| **Relation** | **Browse** to select the relationship between the derived dataset and the parent item revision.  The **EDAHasDerivedDataset** business object and its children are displayed in the selection dialog box. |
| **Dataset Business Object** | **Browse** to select the parent dataset business object type in Teamcenter to represent the derived item, for example, PDF. |
| **Dataset Reference** | **Browse** to select the kind of file reference to use for the derived dataset.  If the derived data instance comprises more than one file, this field must either be specified as a *ZIPFILE* type or must be specified using a separate derived dataset configuration entry with the same derived data name. |
| **Pathname** | Type the path where the derived dataset is to be saved on the user's machine.  Path names are evaluated at run time and must be the fully qualified path of the dataset that is to be saved. Path names can be explicitly specified (for example, *D:\EDA*  *\Datasets\readme.txt*) or formed using the variables or file name filters. Derived datasets can contain multiple files. Path names are case sensitive, and the directory delimiters of / or \ are used interchangeably. |
| **Callback Name** | Type the EDA callback name to execute.  This name is used to identify the configured callback in the EDA configuration file to determine what script to execute. The script is responsible for creating or placing |
| **Field** | Description |
|  | the corresponding derived files to be uploaded as specified by the configured source path name. |

The derived item configuration is added to the **Configure Dataset** table.

* 1. Click **Finish**.

The derived data configuration is added under the **EDA Derived Data** folder

1. To save the changes to the data model, choose **BMIDE**→**Save Data Model**, or click the **Save Data Model** button  on the main toolbar.
2. Deploy your changes to the test server. Choose **BMIDE**→**Deploy Template** on the menu bar, or select the project and click the **Deploy Template** button  on the main toolbar.
3. In the rich client, set the **EDA\_DerivedDataConfigDefault** preference to point to the EDA derived data configuration you just created.
4. Choose **Edit**→**Options**, click the **Search** link at the bottom of the **Options** dialog box, locate the **EDA\_DerivedDataConfigDefault** preference, and change its value to the new configuration. There may be multiple configurations created in the Business Modeler IDE, but an administrator can point to only one of them through this preference.
5. After deployment, test your new configuration in Teamcenter EDA.

For example, in your ECAD design tool, choose **Teamcenter**→**Save Derived Data**. (You can also select the **Generate Derived Data** check box in the **Save As**, **Save**, or **Check In** dialog box.)

To verify that the derived data is generated, in Teamcenter, expand the item that contains the derived data (for example, a CCA item). The derived dataset entries appear as you expand the tree structure. To see the contents of the derived dataset, right-click the dataset and choose **Named References**. A dialog box appears that shows the files that are contained in the derived dataset.

#### Configure how derived data files are named in Teamcenter

Update the value of the **EDA\_UseDerivedDataSubDirs** preference based on how you want the derived data files to be named in Teamcenter. The values are as follows:

##### True

The derived dataset name is the same as the Derived Dataset Configuration Name.

##### False

The derived dataset name consists of the parent item ID and an underscore (character) followed by the Derived Dataset Configuration Name.

#### Configure design variants

#### What are design variants?

To design and develop a product with several *variations* of that product, where each variation has different options or capabilities, you create design variants. Creating design variants avoids the need to create a unique version of the design for each variation. A variant uses the same base design, but the PCB assembly is loaded with the set of components specified by the variation. A variation may then be nominated when generating the design’s manufacturing output (such as BOM, Pick and Place (P&P), and assembly drawings), which will in turn determine how the product is assembled.

When you are working with ECAD design files, you can create any number of variations of the same base design, where each component may be configured differently as follows:

##### Fitted

This is the default state of a component if the component is fitted. It does not have variations. When you create a new variant, all components default to the state *fitted*.

##### Not Fitted

If a component is set to *not fitted*, it still exists on the schematic and is transferred to the PCB, but it is removed from the appropriate output documentation, such as the BOM.

##### Fitted with Varied Parameters

A component can have variations of any of its parameters, as part of the variant definition process. Modifying the value of a parameter is a local variation, only affecting the output documentation. The original schematic and the component whose parameter is being varied are not modified in any way.

##### Alternate Part

It is also possible to select an entirely different component as an *alternate part*. Since the alternate part is a different component, only one component is presented on the compiled schematic sheet. The alternate part must also share the same set of pins placed in the same locations as the base part. This is an essential requirement to ensure that the connectivity remains valid when the design is compiled.

An ECAD design is saved as a non-variant design for the first time in Teamcenter. Later, the variants of the ECAD design are added or created in Teamcenter as required. There are two scenarios for how variants are added.

* In the first scenario, you save the ECAD design as a non-variant design in Teamcenter. When new variants of the designs are created in the ECAD tool, you update the non-variant design in Teamcenter with the variants.
* In the second scenario, you save the non-variant ECAD design as a variant design in Teamcenter by creating dummy variants. When the variant designs are created in the ECAD tool, you replace the dummy variants in Teamcenter with the actual variant designs.

##### 

##### You can [configure how to save a nonvariant ECAD design to Teamcenter as a variant design](#_bookmark22)

using the EDA variant object model.

#### Configure ECAD design variants in Teamcenter

You can save a nonvariant ECAD design to Teamcenter using the EDA variant object model. To configure the saving of new nonvariant ECAD designs in Teamcenter as variant designs:

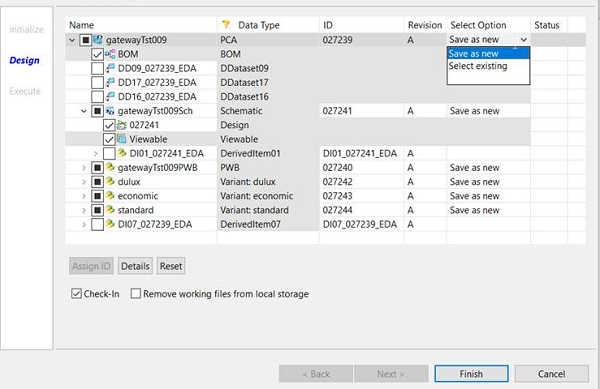
1. Update the value of the **EDA\_SaveAsForceVariant** preference to **true**. This value specifies that nonvariant ECAD designs not previously saved in Teamcenter are to be saved in Teamcenter, using the EDA variant object model.
2. Update the value of the **EDA\_FutureVariantName** preference. The default value is **futureVariant**. When you save a nonvariant design to Teamcenter and the value of the **EDA\_SaveAsForceVariant** is **true**, the value of the **EDA\_FutureVariantName** preference is used as the **Variant Name** in the **PCA Variants** dialog box. The value of the **Variant Name** in the **PCA Variants** dialog box cannot be changed.

Nonvariant ECAD designs already saved to Teamcenter using the EDA nonvariant object model are converted to use the EDA variant object model by the existing nonvariant-to-variant conversion functionality. This conversion occurs automatically when an ECAD design, previously saved to Teamcenter as a nonvariant design, is saved to Teamcenter after the first variant is added to the ECAD design.

#### Base BOMs do not support this functionality.

#### Manage design files in local storage

You can configure the default behavior of removing design files in the local storage option after you check in, check out, revise, or perform a save as operation on the design files. The **Remove working files from local storage** check box allows you to remove associated design files after you check in, check out, revise, or perform a save as operation on the design.



You can configure the default status of this check box by setting the following values in the

**EDA\_RemoveWorkingFilesOptionDefault** preference:

1. **unchecked**: The check box is cleared by default, and the user can select it.
2. **checked**: The check box is selected by default, and the user can clear it.
3. **Forceunchecked**: The check box is cleared, and the user cannot select it.
4. **Forcechecked**: The check box is selected, and the user cannot clear it.

#### Apply Teamcenter user's permission on the network shared ECAD design folder

During any Teamcenter EDA operation on a design in a network shared folder, the ECAD users accessing the design will have viewing or modifying rights on that shared folder based on the access permissions set for that user in Teamcenter.

To apply the Teamcenter user’s permissions on the ECAD design folder, the Teamcenter administrator must set the **EDA\_EnableFolderPermissions** preference value to **true**.

To specify which Teamcenter user roles must have permission to access the ECAD design folder, the user roles must be set in the **FolderPermissions\_ValidRolesForAccess** preference.

#### Allow ECAD designers to assign a Teamcenter project from the EDA client

While saving a new or an existing EDA design from the EDA client to Teamcenter, the ECAD designers can assign or remove a Teamcenter project for that design.

To make sure that the ECAD designer assigns a Teamcenter project while saving a design, you must set the **EDA\_TCProjectRequired** preference to **true**. If this preference is set to **false**, the ECAD designer can save the design without assigning any project. The default value of the preference is **false**. If a default project is assigned to a user in Teamcenter, that project is assigned to the design by default. However, the user can assign additional projects while saving the design.

You must also assign the ECAD users to a project in Teamcenter to allow the users to select that project during the save operation. All projects associated with the ECAD users are displayed for selection to that user during the save operation.

A Teamcenter propagation rule is used for propagating security-related property values, such as project and license assignments, from a source business object to a destination business object. The projects assigned to Printed Circuit Assembly (PCA) objects are propagated to all **ItemRevisions** and **Datasets** associated with the PCA object based on the Teamcenter propagation rule defined for that object. You can also modify the OOTB propagation rules as per your requirements. For more information on how to modify an existing propagation rule, refer to the Configure your business data model in BMIDE on Support Center.

The OOTB propagation rules defined for EDA objects are as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Directio n** | **Source business object** | **Relation** | **Destination business object** | **Prop Group** | **Operation** | **Style** | **Details** |
| Forward | **EDACCABa**  **seRevision** | **IMAN\_Ren dering** | **Dataset** | Security Group I | All | Merge | Propagation rule between |
|  |  |  |  |  |  |  | PCA |
|  |  |  |  |  |  |  | **EDACCABas** |
|  |  |  |  |  |  |  | **eRevision** |
|  |  |  |  |  |  |  | and design |
|  |  |  |  |  |  |  | **Dataset** in a |
|  |  |  |  |  |  |  | variant |
|  |  |  |  |  |  |  | design. |
| Forward | **EDACCABa**  **seRevision** | **structure\_ revisions** | **PSBOMView Revision** | Security Group I | All | Merge | Propagation rule between |
|  |  |  |  |  |  |  | PCA |
|  |  |  |  |  |  |  | **EDACCABas** |
|  |  |  |  |  |  |  | **eRevision** |
|  |  |  |  |  |  |  | and |
|  |  |  |  |  |  |  | **PSBOMView** |
|  |  |  |  |  |  |  | **Revision** in a |
|  |  |  |  |  |  |  | variant |
|  |  |  |  |  |  |  | Design. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Directio n** | **Source business object** | **Relation** | **Destination business object** | **Prop Group** | **Operation** | **Style** | **Details** |
| Forward | **ItemRevisi on** | **EDAHasSc hematic** | **EDASchem Revision** | Security Group I | All | Merge | Propagation rule between |
|  |  |  |  |  |  |  | PCA |
|  |  |  |  |  |  |  | **ItemRevisio** |
|  |  |  |  |  |  |  | **n** and |
|  |  |  |  |  |  |  | **EDASchem** |
|  |  |  |  |  |  |  | **Revision**. |
| Forward | **EDASche mRevision** | **IMAN\_Ren dering** | **Dataset** | Security Group I | All | Merge | Propagation rule between |
|  |  |  |  |  |  |  | **EDASchemR** |
|  |  |  |  |  |  |  | **evision** and |
|  |  |  |  |  |  |  | design |
|  |  |  |  |  |  |  | **Dataset**. |
| Forward | **ItemRevisi on** | **EDAHasP WB** | **ItemRevisio n** | Security Group I | All | Merge | Propagation rule between |
|  |  |  |  |  |  |  | PCA |
|  |  |  |  |  |  |  | **ItemRevisio** |
|  |  |  |  |  |  |  | **n** and |
|  |  |  |  |  |  |  | **Printed** |
|  |  |  |  |  |  |  | **Wiring** |
|  |  |  |  |  |  |  | **Board**. |
| Forward | **EDACCABa**  **seRevision** | **EDAHasVa riant** | **ItemRevisio n** | Security Group I | All | Merge | Propagation rule between |
|  |  |  |  |  |  |  | PCA |
|  |  |  |  |  |  |  | **EDACCABas** |
|  |  |  |  |  |  |  | **eRevision** |
|  |  |  |  |  |  |  | and variant |
|  |  |  |  |  |  |  | **ItemRevisio** |
|  |  |  |  |  |  |  | **n**. |
| Forward | **ItemRevisi on** | **EDAHasDe rivedItem** | **ItemRevisio n** | Security Group I | All | Merge | Propagation rule between |
|  |  |  |  |  |  |  | any |
|  |  |  |  |  |  |  | **ItemRevisio** |
|  |  |  |  |  |  |  | **n** and |
|  |  |  |  |  |  |  | derived |
|  |  |  |  |  |  |  | **ItemRevisio** |
|  |  |  |  |  |  |  | **n** in a |
|  |  |  |  |  |  |  | design. |
| Forward | **ItemRevisi on** | **EDAHasDe rivedData set** | **Dataset** | Security Group I | All | Merge | Propagation rule between any |
|  |  |  |  |  |  |  | **ItemRevisio** |
|  |  |  |  |  |  |  | **n** and |
| Directio n | Source business object | Relation | Destination business object | Prop Group | Operation | Style | **Details** |
|  |  |  |  |  |  |  | **derived Dataset in a design.** |

You can also modify the OOTB propagation rules to create custom rules for variant and non-variant designs.

#### Create a custom propagation rule for variant designs:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Directio n** | **Source business object** | **Relation** | **Destination business object** | **Prop Group** | **Operation** | **Style** | **Details** |
| Forward | **ItemRevisi** | **structure\_** | **PSBOMView** | Security | All | Merge | Propagation |
|  | **on** for the | **revisions** | **Revision** | Group I |  |  | rule between |
|  | variant |  |  |  |  |  | the variant |
|  | item type |  |  |  |  |  | **ItemRevisio** |
|  | defined in |  |  |  |  |  | **n** and |
|  | the |  |  |  |  |  | **PSBOMView** |
|  | **EDA\_CCAV** |  |  |  |  |  | **Revision**. |
|  | **ariantItem** |  |  |  |  |  |  |
|  | **TypeDefa** |  |  |  |  |  |  |
|  | **ult** |  |  |  |  |  |  |
|  | preference |  |  |  |  |  |  |

**Create a custom propagation rule for non-variant designs:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Direction** | **Source business object** | **Relation** | **Destinatio n business object** | **Prop Group** | **Operation** | **Style** | **Details** |
| Forward | **ItemRevisi on** for the PCA item type defined in the **EDA\_CCAI**  **temTypeD efault** preference | **IMAN\_Ren dering** | **Dataset** | Security Group I | All | Merge | Propagatio n rule between the PCA **ItemRevisi on** and design **Dataset** in a non- variant design. |
| Forward | **ItemRevisi on** for the | **structure\_ revisions** | **PSBOMVie**  **wRevision** | Security Group I | All | Merge | Propagatio n rule |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Direction** | **Source business object** | **Relation** | **Destinatio n business object** | **Prop Group** | **Operation** | **Style** | **Details** |
|  | PCA item type defined in the **EDA\_CCAI**  **temTypeD efault** preference |  |  |  |  |  | between the variant **ItemRevisi on** and **PSBOMVie**  **wRevision**  . |

#### Set configurations based on the ECAD applications used

You can configure which preference values should be used for a specific preference based on the connectors configured for your ECAD design tool. This configuration allows all EDA preferences to be overridden by application-specific preferences, if set. The application specific preferences can be created for all the EDA design preferences by adding a suffix to the OOTB preference name. The suffix is the name of the ECAD tool application as mentioned in the *%TCEDAECAD\_ROOT%\\*\_usrdef.xml* file.

If the preference exists with the suffix of the connector name, this preference is used. Else, the preference without a suffix is used.

Example:

Consider that you are using two different connectors, and you want to set different values for the following preference, depending on which connector is currently being used. You can set different values for the same preference by adding the application name as a suffix to the preference name.

The OOTB preference name is **EDA\_CCAVariantItemTypesAllowed**

To set the preferences with different values for both the connectors, create two different instances of the preference with the connector names suffixed to the preference.

* Preference to be used if connector 1 is used:

**EDA\_CCAVariantItemTypesAllowed\_<connectorname\_1>**

* Preference to be used if connector 2 is used:

**EDA\_CCAVariantItemTypesAllowed\_<connectorname\_2>**

#### Allow users to view the PCB board requirements for ECAD objects in Active Workspace

ECAD designers can view the PCB board requirements specifications related to the design revisions in the **Requirements** tab in Active Workspace for the *EDACCABase* and *EDASchem* object types, by default.

To allow users to configure **Requirements** tab for other Item Revisions, user can include the following in the respective stylesheet.

<page titleKey="tc\_xrt\_Requirements" visibleWhen="ActiveWorkspace:SubLocation

!= com.siemens.splm.client.occmgmt:OccurrenceManagementSubLocation">

<htmlPanel declarativeKey="Eda0RequirementsTraceability" />

</page>

Additionally, you can specify the revision rule to configure the requirements BOM structure in the **Requirements** tab by setting the **EDA\_AWRequirementBOMRevRule** preference. The default value of the revision rule is set to **Latest Working**.

#### Customize the acronyms used for ECAD design objects in Teamcenter

If your ECAD designers use acronyms different from the default acronyms provided, the Teamcenter a can override the defaults by creating custom acronyms used for ECAD designs in Teamcenter.

By default, EDA uses the following acronyms to refer to electronic design objects on the user interface:

* **PCA** for printed circuit assembly or **CCA** for circuit card assembly (in earlier versions of EDA).
* **PCABase** for printed circuit assembly base or **CCABase** for circuit card assembly base (in earlier versions of EDA).
* **PWB** for printed wire board.

If your ECAD designers use different acronyms, the Teamcenter administrator can override the defaults by creating custom text copies of the **TC\_ROOT\lang\textserver***locale***\eda0\_text\_locale.xml** file. For each *custom-name***\_text\_locale.xml** file, locate the following lines in the file, and edit the display text as required:

<key id="Eda0\_PCA">PCA</key>

<key id="Eda0\_PCAToolTip">Printed Circuit Assembly</key>

<key id="Eda0\_SCH">SCH</key>

<key id="Eda0\_SCHToolTip">Schematic</key>

<key id="Eda0\_PWB">PWB</key>

<key id="Eda0\_PWBToolTip">Printed Wire Board</key>

<key id="Eda0\_Design\_Metrics\_key">Design Metrics</key>

<key id="Eda0\_Cad\_Baselines\_key">CAD Baselines</key>

<key id="Eda0\_Edm\_Viewer\_key">EDM Viewer</key>

Note:

There is a separate locale file for each supported locale. The example shown is from the **en\_US**

locale file.

For more information about customizing textserver file text, see the *Server Customization* help.

#### Configure the EDA client to use the Teamcenter EDA panels instead of Active Workspace

By default, the EDA client uses Active Workspace panels for user actions such as open and save. You can use Teamcenter EDA panels instead, if you prefer to or in case you do not have Active Workspace installed. This can be done by setting the following preferences:

#### EDA\_Use\_ActiveWorkspace

This preference specifies whether to display Active Workspace panels instead of the older client panels in dialog boxes where available. The default value is set to **True**. To switch to Teamcenter EDA client panels, change the preference value to **False**.

The default value is valid only if one of the following preferences is also defined to point to an Active Workspace server.

##### ActiveWorkspaceHosting.URL

* **ActiveWorkspaceHosting.EDA.URL**

If neither preference is defined, then the **EDA\_Use\_ActiveWorkspace** preference is ignored, and Active Workspace is not used. If either one of these preferences is defined, **ActiveWorkspaceHosting.URL** takes precedence and the default for value for **EDA\_Use\_ActiveWorkspace** is **True**.

#### EDA\_Use\_ActiveWorkspace\_Open

This preference specifies whether to display an Active Workspace panel in the **Open** dialog box instead of the Teamcenter EDA client panel. The default value is set to **True**. To switch to Teamcenter EDA client panels, change the preference value to **False**.

The preference **EDA\_Use\_ActiveWorkspace** must also be set to **True** to use this preference.

#### Managing and configuring Teamcenter preferences for design management

You can set system-wide preferences on the Teamcenter server that apply to all EDA users. To understand how preferences work and to understand how to access a list of all supported preferences, refer to the Managing Preferences help on Support Center. All EDA preferences begin with the prefix **EDA**.

1. Run the rich client on the Teamcenter server and log on as an administrator.
2. In My Teamcenter, choose **Edit→Options**.
3. At the bottom of the **Options** dialog box, click **Index**.
4. In the **Preferences** dialog box, search for the preferences that begin with **EDA\_**.
5. Change the preference value and click the **Modify** button to save the new value.

## Installing – EDA – Client Components

The following prerequisites and requirements

## Installing – Cadence – CIS and Allegro Components

The following prerequisites and requirements

## Post-Installation Checklist

In the next section, describe the steps to install and configure the components.

#Checklist Item

1 Check the API endpoints are correct.

2 Check that the database tables are running.

3 Check that digital certificates are installed.

4 Check that you can access the web servers.

5 Check that security settings have been applied correctly.

## Troubleshooting

In the next section, describe the steps to install and configure the components.In this section, we help users fix common issues that may arise.To do this, identify known issues, common mistakes users may make when configuring the system, and any recommended settings.For example:The following installation issues may occur when installing .Problem: Warning 123. A numeric value must be provided for the port.Fix: Port numbers for services must be positive numeric values and represent valid HTTP ports.Internal Error 123.Problem: The cannot duplicate an existing file name and path.Fix: Remove the existing file name and path.To make this easier to follow, list the tasks and the steps the System Admin needed to follow:The installation procedure contains the following steps:

# Checklist Item

1 Configuration pre-requisites

2 Ensure that servers meet minimum hardware requirements.

3 Ensure that software prerequisites are installed.

4 Run the installer program on the server.

5 Run the DB installer program on the server.

6 Install the database.

7 Import data

8 Configure API endpoints

9 Install Application Server

10 Install APIs

## Reference Documents

There are multiple documents referred as part of documenting the Technical design and the solution implementation document and they are referred as below.

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Document Name | Link | Comment |
| 1 |  |  |  |
|  |  |  |  |