



# TEAMCENTER

# Manufacturing Process Planning — Deployment and Administration

Teamcenter 2412

Unpublished work. © 2025 Siemens

This Documentation contains trade secrets or otherwise confidential information owned by Siemens Industry Software Inc. or its affiliates (collectively, "Siemens"), or its licensors. Access to and use of this Documentation is strictly limited as set forth in Customer's applicable agreement(s) with Siemens. This Documentation may not be copied, distributed, or otherwise disclosed by Customer without the express written permission of Siemens, and may not be used in any way not expressly authorized by Siemens.

This Documentation is for information and instruction purposes. Siemens reserves the right to make changes in specifications and other information contained in this Documentation without prior notice, and the reader should, in all cases, consult Siemens to determine whether any changes have been made.

No representation or other affirmation of fact contained in this Documentation shall be deemed to be a warranty or give rise to any liability of Siemens whatsoever.

If you have a signed license agreement with Siemens for the product with which this Documentation will be used, your use of this Documentation is subject to the scope of license and the software protection and security provisions of that agreement. If you do not have such a signed license agreement, your use is subject to the Siemens Universal Customer Agreement, which may be viewed at <https://www.sw.siemens.com/en-US/sw-terms/base/uca/>, as supplemented by the product specific terms which may be viewed at <https://www.sw.siemens.com/en-US/sw-terms/supplements/>.

SIEMENS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS DOCUMENTATION INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. SIEMENS SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, LOST DATA OR PROFITS, EVEN IF SUCH DAMAGES WERE FORESEEABLE, ARISING OUT OF OR RELATED TO THIS DOCUMENTATION OR THE INFORMATION CONTAINED IN IT, EVEN IF SIEMENS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TRADEMARKS: The trademarks, logos, and service marks (collectively, "Marks") used herein are the property of Siemens or other parties. No one is permitted to use these Marks without the prior written consent of Siemens or the owner of the Marks, as applicable. The use herein of third party Marks is not an attempt to indicate Siemens as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A list of Siemens' Marks may be viewed at: [www.plm.automation.siemens.com/global/en/legal/trademarks.html](http://www.plm.automation.siemens.com/global/en/legal/trademarks.html). The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

## About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow's idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: [support.sw.siemens.com](http://support.sw.siemens.com)

Send Feedback on Documentation: [support.sw.siemens.com/doc\\_feedback\\_form](http://support.sw.siemens.com/doc_feedback_form)

# Contents

## Getting started with administering Manufacturing Process Planning — Deployment and Administration 1-1

### Installing and configuring the Manufacturing Resource Library

Installation overview and workflow	2-1
Install graphics builder on Linux	2-2
Configure Manufacturing Resource Library users, groups, and roles	2-5
Populating the database	2-7
Beginning the installation	2-7
Import the class hierarchy	2-9
Update the class hierarchy	2-10
Configuring the update	2-13
Import or update rules and preferences	2-14
Import MRL NX seed parts	2-15
Import 3D template parts	2-16
Import or update MRL sample resources	2-17
Import part family templates when you use custom item IDs	2-20
Install localization data for MRL	2-20
Configure NX Library using the installation wizard	2-21
Configure NX CMM library using the installation wizard	2-26
Configure NX Library	2-28
Configure the graphics builder for MRL	2-29
Configure MRL Connect	2-30

### General configuration and customizing tasks

Configure the commands in the context menu	3-1
Configuring the business object types in the New dialog boxes	3-5
Hide a custom perspective	3-5
Defining which occurrence types display in paste and assign resource commands	3-6
Example of how to copy and assign custom occurrence types to processes and operations	3-6
Configure the update of multiple product views	3-7
Set up Manufacturing Process Planner for Product Configurator	3-8
Set up 4D planning	3-9
Creating new search criteria for a broken link search	3-11
Using the ME-revise-AH action handler	3-13
Customizing the BOM line quantity	3-14
Specify a property value to be a combination of other property values	3-14
Configure incremental changes	3-15
Make manufacturing change notices mandatory for BOP structures	3-16

Configure manufacturing change notices (MCNs)	3-16
Restructure BOP product views using an upgrade utility	3-18

## Setting up the Custom Process View

Configure the Custom Process View	4-1
Modify the view using preferences	4-2
Modifying style sheets	4-3
Specify the tables and content	4-4

## Configuring 4G design in manufacturing

Configuring the comparison of design to manufacturing features	5-1
Mapping property and attribute groups between design elements and BOM lines	5-1

## Administering standard text

Standard text overview	6-1
Prerequisites to using the standard text feature	6-2
Set up standard text	6-2
Understanding the standard text template	6-5
Using the standard text template	6-6
Edit standard text templates	6-6
Remove customization	6-8
Restore customization	6-9
Editing options	6-10
Add BOM line properties	6-10
Add or modify template styles	6-14
Remove symbols section	6-14
Add symbol section	6-15
Specifying the location of the Add Symbol button	6-16
Importing standard text library structures from a legacy system	6-17
Updating standard text instances using the update_stx_elements utility	6-19

## Administering structure cloning

Creating structures from templates overview	7-1
Reusing process templates	7-1
Carrying over future incremental change when cloning	7-2
Configuring how assemblies in context are cloned and replaced	7-3
Defining cloning rules	7-4

## Administering manufacturing publish

Configuring the publish feature	8-1
Understanding roles	8-1
Example: creating a customized publish role	8-2
Set preferences and environment variables	8-3

Installing the transfer mode	8-6
Defining alias files	8-6
Building a sample alias file	8-7
Specifying end objects	8-15
Navigating through a PLM XML file	8-16
Strategies for working with alias files	8-17
Create a new publishing alias dataset	8-18
Update a publishing alias	8-18
Load or reload the alias file	8-19
Create or edit publishing pages	8-19
Creating manufacturing documentation in batch mode overview	8-20
Enabling batch processing of publishing pages	8-20
Troubleshooting the PublishBatch translator	8-21

## Setting up enterprise bill of process (BOP)

Configuring the enterprise BOP	9-1
Setting up scheduled part assignment resolution	9-4
Setting up logistics planning	9-5
Line balancing reports	9-6
Customizing a line balancing report	9-6
Customize the contents of the reports	9-6
Line balancing report parameters	9-7

## Administering time data in the Time view

Configure new categories in the Time view	10-1
MTM data card	10-3
Configuring an MTM data card	10-3
Install MTM data cards	10-4
Create your own data card	10-4
TiCon integration	10-5
Understanding the Teamcenter to TiCon mapping	10-5
Map attributes of time elements to activity properties	10-6

## Configuring studies

Enable Study Manager	11-1
Set up study synchronization	11-1

## Administering EBOM and MBOM

Reasons for creating an MBOM	12-1
Comparing the EBOM and MBOM	12-1
EBOM-MBOM preferences	12-2
Automatically creating an MBOM from an EBOM	12-4
Create an MBOM from a recipe	12-6
Creating an MBOM using the me_create_mbom utility	12-9
Reconciling EBOM and MBOM in a Multi-Site environment	12-18

EBOM to MBOM effectivity options	12-19
Delegating, overriding, or copying line properties from MBOM to BOP	12-23

## Administering CAD BOM and part BOM alignments

Part BOM requirements	13-1
Set up CAD BOM smart search criteria	13-1
CAD and part BOM action handlers	13-2
Use the CBA 2.0 to CBA 3.0 me_migrate_design_ebom migration utility	13-4

## Configuring in-context IDs (IDIC)

Understanding equivalence and in-context ID (IDIC)	14-1
Setting up the search for components in other structures	14-1
Understanding ID in context and assigned products	14-2
Configuring the generation of smart IDICs	14-5
Generate smart IDICs	14-5
Configuring the IDIC mechanism when using shared assemblies	14-6
Stamp missing IDIC strings	14-7

## Administering the import and export of manufacturing structures

Configuring PLM XML export	15-1
Configuring custom primary objects for use with DIPA	15-1
Configuring the import and export of dynamic in-process assemblies	15-1
Configuring the export of end item assemblies	15-2

## Configuring briefcase exchange of collaboration context

Key preferences for configuring briefcase exchange of collaboration context	16-1
Configuring recursive ownership transfer	16-2
Configuring automatic APN creation during export	16-3

## Administering GD&T symbols

Displaying GD&T symbols and rich text in forms	17-1
Configure GD&T symbols and rich text in forms	17-1

## Setting up rich client reports

Creating reports	18-1
Troubleshooting reports in the rich client	18-1
Configure reports in the rich client	18-4
View or print rich client reports with a Web browser	18-6
Creating reports in the rich client	18-6

## Configuring the accountability check

Setting up the accountability check	19-1
-------------------------------------	------

<b>Publish accountability check settings</b>	19-1
<b>Configuring the inclusion rules</b>	19-2
<b>Customize accountability check reports using a temporary XML file</b>	19-4
<b>Add or remove accountability check report stylesheets</b>	19-6
<b>Configure the accountability check to use net effectivity</b>	19-8
<b>Display an icon to indicate revision mismatches on BOP lines</b>	19-8
<b>Customizing the net effectivity user exit</b>	19-8
<b>Accountability check batch mode</b>	19-8
Running an accountability check in batch mode	19-8
Running an accountability check or propagation at a specified time	19-9
Running an accountability check in batch mode using the acc_check_client utility	19-9
Run the acc_check_client utility from a script	19-10
<b>Using dynamic equivalence to locate matching EBOM and MBOM lines</b>	19-10
<b>Modify accountability check closure rules to exclude dynamic in-process assemblies</b>	19-11
<b>Set up the accountability check to compare to intermediate data captures</b>	19-11
<b>Using the two-tier accountability check utility</b>	19-12



# 1. Getting started with administering Manufacturing Process Planning — Deployment and Administration

Manufacturing Process Planner allows you to design a plan that details how to manufacture a product. The manufacturing process plan includes a top-level structure of the process needed to manufacture the product, as well as a detailed design of the individual processes and activities to be included in the plan.

As you build the process structure, you can assign resources to the various processes, operations, and activities. You can also identify the specific locations within the plant where each operation and activity will be performed.

You typically use *Manufacturing Process Planner* to design and manage the manufacturing processes for large assemblies. Common tasks are:

- Assigning an engineering bill of materials (BOM) to a manufacturing BOM.
- Using enterprise bills of process (BOPs) to capture standards and best practices across the business as generic processes and operations.
- Consuming 4GD design data into manufacturing process structures.
- Performing line balancing and time way planning.
- Creating work instructions for process data.
- Comparing structures to find missing parts in source or targets or differences in properties of the two structures.

For many of these tasks, you must configure aspects of your installation, including setting various preferences. For example:

- *Configuring the enterprise BOP*
- *Configuring the accountability check*
- *Administering standard text*

Additionally, you can customize Manufacturing Process Planner features to suit your business needs. For example:

- *Add or remove accountability check report stylesheets*

- *Line balancing reports*

# 2. Installing and configuring the Manufacturing Resource Library

## Installation overview and workflow

The Manufacturing Resource Library (MRL) is a collection of data that you can import into the database. It includes a classification hierarchy for resources such as tools, machines, and fixtures. You use this data in the Classification and Resource Manager applications. In addition, you can access these resources from NX CAM. To do this, you must **configure NX Library**.

To populate the database with sample Manufacturing Resource Library (MRL) data, make sure that the Teamcenter corporate server is installed on the installation host and the Teamcenter database is configured or upgraded.

To install the Manufacturing Resource Library on a Linux server, you must use a Windows rich client. If you do not have one in your network, you must install one temporarily. After you do this, start the setup program in the Windows software kit.

Alternatively, you can run the following script in a Linux shell:

```
.../advanced_installations/resource_management/MRL/ImportFiles/mrl_all_modules/  
unix_all_modules_import_en.sh
```

There are general workflows for installing or updating the Manufacturing Resource Library.

Installing for the first time	Upgrading from an earlier MRL version
1. <b>Configure the users, groups, and roles in the database.</b>	1. <b>Begin the installation.</b>
2. <b>Begin the installation.</b>	2. <b>Update the class hierarchy.</b>
3. <b>Import the class hierarchy.</b>	3. <b>Import or update rules and preferences.</b>
4. <b>Import rules and preferences.</b>	4. <b>Import or update seed parts.</b>
5. <b>Import seed parts.</b>	5. <b>Import or update part family templates and template parts.</b>
6. <b>Import part family templates and template parts.</b>	6. <b>Import or update sample parts.</b>
7. <b>Import sample parts.</b>	

## Install graphics builder on Linux

You can install the graphics builder on machines that use Linux.

### Prerequisites

The NX version being used must be compatible with the Teamcenter version. It is recommended to use the latest maintenance update available.

### Procedure

1. Download the NX batch kit from the Siemens Support Center, and copy the file into the Linux server. For more information, see [Downloads](#).

Note:

You must use the NX batch kit for NX version 2312, or later.

- a. Download the `nxbatch_2406.6001_Inx64.tar.gz` file in **NX 2406.6000 Batch**, or a later version.

Note:

**NX 2406.6000 Batch** is used as an example.

- b. Create the following directory: `mkdir /apps/nxkit`.
  - c. Copy the `.tar.gz` file into this directory: `/apps/nxkit`.
  - d. Unzip and move the following files to the `nxkit` folder:
    - `tar -xzvf /apps/nxkit/nxbatch_2406.6001_Inx64.tar.gz -C /apps/nxkit`
    - `rm /apps/nxkit/nxbatch_2406.6001_Inx64.tar.gz`
    - `mv /apps/nxkit/nxbatch_2406.6001_Inx64 /apps/nxkit/nxbatch_Inx64`
2. Install the NX batch.
    - Select and install the `sudo ./run_install` file.
  3. Configure graphics builder.

**Note:**

All Manufacturing Resource Library (MRL) files are located in Teamcenter kit:  
`tc2412_Inx64.zip/advanced_installations/advanced_installations.zip`.

- a. **Unzip** the following files into the listed directories:
  - **unzip** `tc2412_Inx64.zip -d /tmp/TC`.
  - **unzip** `/tmp/TC/tc2412_Inx64/advanced_installations/advanced_installations.zip -d /tmp/MRL`.
  - **unzip** `/tmp/TC/tc2412_Inx64/tc/bin.zip -d /tmp/BIN`.
- b. `sudo mkdir ${TC_ROOT}/bin/nx_graph/TCL_Create_Graphics/`.
- c. **unzip** `/tmp/MRL/resource_management/MRL/GraphicsMacros/fcm.zip -d ${TC_ROOT}/bin/nx_graph/TCL_Create_Graphics/`.
- d. `sudo cp -f /tmp/BIN/bin/nx_graph/start_nx_graphicsbuilder.template ${TC_ROOT}/bin/nx_graph/start_nx_graphicsbuilder`.

**Note:**

You must adjust the following lines in file `start_nx_graphicsbuilder` to your values:

- `UGII_BASE_DIR=~ugii_base_dir~`  
 Example: `UGII_BASE_DIR=/apps/siemens/nx`
- `UGII_UGMGR_HTTP_URL=~poolmanager_url~`  
 Example: `UGII_UGMGR_HTTP_URL=http://YourTeamcenterServerName:8085/tc`
- `TC_GB_BASE_PORT=~tc_gb_base_port~`  
 Example: `TC_GB_BASE_PORT=2300`
- `TC_GB_NUMBER_OF_PORT=~tc_gb_number_of_port~`  
 Example: `TC_GB_NUMBER_OF_PORT=50`

4. Set the `NXGraphicsBuilder` preference.

**Note:**

You must adjust the `TC_ROOT` path to the actual `TC_ROOT` path of the Linux server.

- Set the `NXGraphicsBuilder` preference to the following values:

- `ActivationMethod:TCE_Activation`

- `CommandLine:/apps/siemens/Teamcenter_2412/tc_root/bin/nx_graph/start_nx_graphicsbuilder`

- `ScriptEvaluationPath:/apps/siemens/Teamcenter_2412/tc_root/bin/nx_graph/TCL_Create_Graphics`

- `evaluateScript:yes`

5. Adjust the NX/Teamcenter attribute mapping.

For more information, see the following:

- Synchronize attribute values from NX to Teamcenter.
- NX/Teamcenter attribute mapping file.

6. Adjust the NX PAX file on the Linux server, including setup sheet templates.

For more information, see the following:

- Enable setup sheet generation for resources.
- PAX file entries for setup sheets.

7. Copy the TCL and DEF files for CAM and CMM.

```
sudo cp -fr /tmp/MRL/resource_management/MRL/install/tcxmrl/nxlib/MACH/ /apps/siemens/nx/mach/
```

```
sudo cp -fr /tmp/MRL/resource_management/MRL/install/tcxmrl/nxlib/CMM_INSPECTION/ /apps/siemens/nx/cmm_inspection/
```

8. Adjust the values, as needed, in the following file: `/apps/siemens/nx/mach/resource/configuration/cam_part_planner_mrl.dat`.

## Configure Manufacturing Resource Library users, groups, and roles

If you have already defined your own database user, group, and role settings in the Organization application, you can use those existing users to populate the Manufacturing Resource Library (MRL). If not, run the MRL installation setup to populate the following user structure.



Perform the following task before installing the Manufacturing Resource Library.

1. In the Organization application, ensure that a default volume is assigned to the application user that you specify in this installation procedure.
2. In the **resource\_management** directory, click **Setup.exe** and proceed to the **Setup Type Selection** dialog box.
3. Select **Manufacturing Resources - Configure MRL Users** and click **Next**.

**Tip:**

Press the F1 key to see the help for each page in the installation wizard.

4. Select one or more of the following:

- **Create and Configure TC/MRL Users**

Creates and configures MRL-specific users:

**CAMUser01**  
**Planner01**  
**TOOLAdmin01**  
**MFGAdmin01**

- **Create Manufacturing Admin User in DBA Group**

Creates and configures the **MFGAdmin01** user in the **dba** group.

- **Assign Manufacturing Applications to MRL Users**

Assigns the created MRL users to Teamcenter applications as follows:

**CAMUser01**: Part Planner, Resource Manager

**Planner01**: Classification, Part Planner, Resource Manager

**TOOLAdmin01**: Classification, Resource Manager

**MFGAdmin01**: Organization, Classification Admin, Classification, Part Planner, Resource Manager

5. Click **Next**.

Teamcenter displays the **Choose TC\_ROOT Directory** dialog box.

6. Select the path to your Teamcenter installation and click **Next**. The path must point to an existing installation.

Teamcenter displays the **Choose TC\_DATA Directory** dialog box.

7. Select the path to the directory containing the database-specific data files and click **Next**.

The path must point to an existing **TC\_DATA** directory.

8. Enter the user name, password, and group of two valid Teamcenter logon accounts.

Teamcenter requires two types of user accounts to configure the ownership of MRL objects correctly:

- An administrative user account

Used to create the classification hierarchy.

- An application user account

Used to populate the database MRL data such as sample resources items and part family templates. This user is the owner of the content including sample data for tools, machines, and fixtures.

9. Enter the name of a valid Teamcenter volume name.

10. Click **Next** and complete the user configuration.

# Populating the database

## Beginning the installation

Before you begin installation, make sure you have the latest available version of the Manufacturing Resource Library. If a later version is available, download and extract the latest version and browse to the **advanced\_installations** directory in the location of the extracted files.

For information about later versions, see Support Center.

To populate the Manufacturing Resource Library (MRL) on the Teamcenter server:

1. In the **advanced\_installations** directory in the Teamcenter software kit, locate the **advanced\_installations.zip** file and unzip it to a local directory.
2. In the **resource\_management** directory, click **Setup.exe**.

Teamcenter displays the **Teamcenter Configuration Setup** dialog box.

3. Click **Next**.

Teamcenter displays a notification that you must check for the most recent version of the resource library.

4. Click **Next**.

Teamcenter displays the **Setup Type Selection** dialog box.

5. Select **Manufacturing Resources - Database Population**.
6. Click **Next**.

Teamcenter displays the **MRL Database Population - Type Selection** dialog box.

7. Select from the following, and then click **Next**:

- **Import Class Hierarchy**

Initially populates the Teamcenter database with MRL classification structures for tools, machines, fixtures, factory resources, machining data library, manufacturing process templates, vendor catalogs, and measuring devices.

- **Update Class Hierarchy**

Updates an existing MRL classification structure. Teamcenter analyzes the available MRL class structures and creates individual update scripts to migrate your class hierarchy to the latest MRL classes.

- **Import Rules and Preferences**

Imports and updates rules and preferences required to use the MRL in Resource Manager.

- **Import MRL NX Seed Parts**

Imports the NX part files that are used by the MRL. This includes template part files to build tool assemblies, a seed part to import GTC STEP files, and sample drawing templates to generate resource setup sheets.

- **Import 3D Template Parts**

Imports the template part files to generate 3D graphics. For tools, the system provides template part files (TPs); for factory resources, part family templates are provided (PFTs).

- **Import MRL Sample Resources**

Imports sample data for the individual modules. This includes sample tool assemblies, fixtures, and factory resources. This option also imports technology data such as feeds and speeds records for the Machining Data Library. Currently, sample machines are no longer delivered with the MRL kit. You must import your machines manually.

Click one of the links to proceed in the documentation. The installer continues with three more common steps.

8. Select the path to your Teamcenter installation in the **Choose TC\_ROOT Directory** dialog box and click **Next**. The path must point to an existing installation.

Teamcenter displays the **Choose TC\_DATA Directory** dialog box.

9. Select the path to the directory containing the database-specific data files and click **Next**. The path must point to an existing **TC\_DATA** directory.
10. Enter the user name, password, and group of two valid Teamcenter logon accounts.

Teamcenter requires two types of user accounts to configure the ownership of MRL objects correctly:

- An administrative user account

Used to create the classification hierarchy.

- An application user account

Used to populate the database MRL data such as sample resources items and part family templates. This user is the owner of the content including sample data for tools, machines, and fixtures.

## Upgrading from an earlier MRL version

In addition to the steps outlined above, it is advised that you:

- update the NX Library files. There may be enhancements to the TCL and DEF files.
- use the MRL graphics builder to copy the latest `ics_graphicsbuilder.tcl` file.

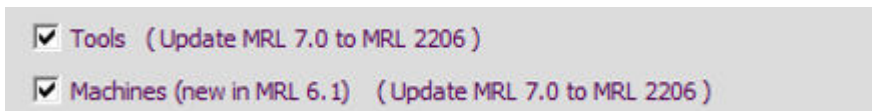
## Import the class hierarchy

Use this procedure to initially populate the Teamcenter database with MRL classification structures for tools, machines, fixtures, factory resources, machining data library, manufacturing process templates, and vendor catalogs.

The following procedure assumes you have completed the steps in *Beginning the installation*. The installation wizard should currently display the **Manufacturing Resource Library - Content Selection** dialog box.

1. Specify which types of data you want to load. The options listed indicate which modules have already been imported to the database. Depending on the information shown in brackets, you may have to return to the **MRL Database Population - Type Selection dialog box** to update the existing hierarchy for a particular module.

For example, if this is displayed:



You must return to the **MRL Database Population - Type Selection** dialog box and select **Update hierarchy** if you want to import the tool hierarchy, but you can proceed with this dialog to import the machine hierarchy.

The following modules are available:

- **Vendor Catalogs**

Imports the empty `MRM_Vendor` class into which vendor catalogs can be imported.

- **Tools (Assemblies and Components)**

Imports the groups and classes used to classify tool assemblies and components.

- **Machines** (New in MRL 6.1)

Imports the groups and classes used to classify machines and devices such as chucks, jaws, or tool holders.

- **Fixtures and Devices** (New in MRL 6.0)

Imports the groups and classes used to classify clamping fixtures.

- **Measuring Devices**

Imports the groups and classes used to gauge, measure, and probe.

- **Factory Resources**

Imports the groups and classes used to classify resources used in the NX Line Designer such as conveyors or industrial components.

- **Machining Data Library**

Imports the groups and classes used to classify technology data such as feeds and speeds or material tables.

- **Manufacturing Process Templates**

Imports the groups and classes used to classify manufacturing processes, operations, or activities.

2. Click **Next**.

Teamcenter displays the **Confirmation** dialog box.

3. Confirm that all your selections are correct and then click **Next**.

Teamcenter installs the specified classification hierarchies.

### Update the class hierarchy

When you are upgrading from an earlier Teamcenter version, Teamcenter checks whether you have previously installed all or parts of the library during the Manufacturing Resource Library (MRL) installation. If you have done so, you can now choose to update older class hierarchies. Teamcenter automatically checks whether first installation (database population) or an update is required for an existing structure.

This procedure assumes you have completed the steps in *Beginning the installation*. The installation wizard should currently display the **Choose MRL Update Work Directory** dialog box.

1. Specify the path for the MRL update files you are creating.

During the course of the update, Teamcenter creates script files for the update and stores them in this directory.

2. Click **Next**.

Teamcenter displays the **MRL Update Type Selection** dialog box, where you must take action in this sequence:

- a. Select **Export existing class hierarchy**.

This step exports the class hierarchies that you want to update to the update directory you specified.

- A. Click **Next**.

Teamcenter displays the **MRL Update Class Hierarchy for Modules** dialog box, which contains a list of the available hierarchies, with version information. For example:



The hierarchies that need to be updated are already selected.

- B. Select the hierarchies you want to update and click **Next**.
- C. Confirm your selections and click **Next**.

Teamcenter exports the selected hierarchies to an **EXPORT** directory in the update directory that you specified at the beginning of this procedure.

The installation wizard returns to the **MRL Update Type Selection** dialog box.

- b. (Optional) Select **Modify configuration file in a text editor** to specify certain objects that are not changed during the update process. Selecting this option allows you to modify the **MRL\_Update\_Configurationfile.xml**. Although it is optional, if you choose to do it, you must do this before moving to the next step. For more information see [Configuring the update](#).

The **MRL\_Update\_Configurationfile.xml** is located here:

*D:\TC\_Kits\TC2412\wntx64\advanced\_installations\resource\_management\MRL\conf\*

- c. Select **Compare your hierarchy to new class hierarchy**.

This step compares the exported hierarchies to the corresponding hierarchy in the newest MRL kit.

- A. Click **Next**.

Teamcenter displays the **MRL Update Class Hierarchy for Modules** dialog box, which contains only the hierarchies that you exported.

- B. Select the hierarchies that you want to update and click **Next**.

- C. Confirm your selections and click **Next**.

Teamcenter compares the selected hierarchies to the ones in the **EXPORT** directory and creates individual update scripts with detailed information about what has changed in the hierarchies. The scripts are stored in a new **UPDATE** directory in the update directory that you specified.

When complete, the installation wizard returns to the **MRL Update Type Selection** dialog box.

- d. Select **Update to new class hierarchy**.

This step updates the hierarchies that you compared to the new hierarchies in the MRL kit.

- A. Click **Next**.

Teamcenter displays the **MRL Update Class Hierarchy for Modules** dialog box, which contains only the hierarchies that you compared.

- B. Select the hierarchies that you want to update and click **Next**.

- C. Confirm that all your selections are correct and then click **Next**.

**Warning:**

The update requires an extended period of time. The length of time required depends on the performance of your hardware.

Upgrading the tool classification hierarchy can take up to 20 hours.

Teamcenter does the following during the update:

- Adds new classes to the hierarchy
- Adds new attributes to classes in the hierarchy
- Updates class images and hierarchy icons

3. If necessary, remove obsolete classes manually.

## Configuring the update

You can exclude some objects from the update by modifying the **MRL\_Update\_Configurationfile.xml** configuration file found in the directory where you extracted the installer:

**advanced\_installations\resource\_management\MRL\conf**

You can modify this configuration file during the MRL update by selecting **Compare your hierarchy to new class hierarchy** and then selecting **Modify configuration file in a text editor**. You must save modifications before you execute the comparison because this file will be used during the compare step.

You can add attribute IDs to the section **DoNotAddAttributes** in the configuration file to only include the attributes that you need. The **DoNotAddAttributes** section is empty by default.

```
<Configurationfile>
  <!-- This file is used for the MRL update procedure -->

  <!-- The following parameters configure what object types are processed -->
  <ProcessDictionaryAttributes value="true"/>
  <ProcessKeyLoves value="true"/>
  <ProcessAdminClasses value="true"/>
  <ProcessAdminViews value="true"/>

  <!-- The following parameters configure whether the output files are generated. -->
  <!-- Files in the INFO_FILES folder are always created. -->
  <WriteOutputTextfiles value="true"/>

  <!-- The following parameters configure keyloves that will NOT be updated. -->
  <IgnoreKeyLoves>
    <KeyLov id="-40922"/>      <!-- Tool: Material reference for NX-CAM -->
    <KeyLov id="-40928"/>    <!-- Tool: Machine Adapter to assign a pocket in NX-
```

```

CAM -->
    <KeyLov id="-45210"/>      <!-- Tool: Status (old) -->
    <KeyLov id="-45211"/>      <!-- Tool: Plant assignment (old) -->
    <KeyLov id="-45212"/>      <!-- Tool: Machine Group -->
    <KeyLov id="-46001"/>      <!-- Resource Status -->
    <KeyLov id="-46002"/>      <!-- Resource Location -->
    <KeyLov id=" ---- Add your ID here ---- "/>
</IgnoreKeyLoves>

<!-- The following attributes will not be added to the classes or views -->
<DoNotAddAttributes>
    <Attribute id="-40700"/>
    <Attribute id="-40701"/>
    <Attribute id="---- Add your ID here ----"/>
</DoNotAddAttributes>

<!-- The following parameters configure whether attribute groupings are updated -->
<Applicability1 update="true"/>      <!-- NX-CAM related attributes -->
<Applicability2 update="true"/>      <!-- Graphics creation related attributes -->
<Applicability3 update="true"/>      <!-- GCS connection attributes -->
<Applicability4 update="false"/>     <!-- User configurable attribute set -->
<Applicability5 update="false"/>     <!-- User configurable attribute set -->

<!-- The following parameters configure whether the User1 and User2 class properties
are updated -->
<ClassUser1 update="true"/>
<ClassUser2 update="true"/>

<!-- The following parameters configure whether class icons and images are updated
-->
<Icon update="true"/>
<Image update="true"/>
<ICS-ClassImage1 update="true"/>
<ICS-ClassImage2 update="true"/>
<ICS-ClassImage3 update="true"/>
<ICS-ClassImage4 update="true"/>
<ICS-ClassImage5 update="true"/>
<ICS-ClassImage6 update="true"/>
<ICS-ClassImage7 update="true"/>
<ICS-ClassImage8 update="true"/>
<ICS-ClassImage9 update="true"/>
</Configurationfile>

```

## Import or update rules and preferences

The following procedure assumes you have completed the steps in *Beginning the installation*. The installation wizard should currently display the **MRL Database Population: Import Rules and Preferences** dialog box.

1. Select the data that you want to import or update:
  - Update Teamcenter preferences for MRL

Installs all preferences required to install the Manufacturing Resource Library and configure its behavior.

- **Guided Component Search rules for tool components**

Imports or updates rules that are used by the guided component search in Structure Manager.

- **Import tool checking rules**

Imports or updates rules used by the tool checker in Resource Manager.

- **Import tool catalog vendor mapping**

Imports or updates mapping rules required to map tool vendor data from the catalog to the tool component classes.

2. Confirm that all your selections are correct and then click **Next**.

Teamcenter installs the specified rules and preferences.

## Import MRL NX seed parts

The following procedure assumes you have completed the steps in *Beginning the installation*. The installation wizard should currently display the **MRL Database Population: Import MRL NX seed parts** dialog box.

1. Select the seed parts that you want to import or update:

- **Template parts for Auto-Assembly**

Imports the template part file used to build tool assemblies in Resource Manager.

- **Template parts for STEP (GTC 3D model Import)**

Imports the seed part used to import GTC STEP files for vendor catalog objects.

- **Template parts for setup sheet creation**

Imports sample drawing templates to generate resource setup sheets.

- **Template parts for fixtures graphics creation**

Imports the template file used to create fixture graphics with the Tcl graphics method.

- **Template parts for temporary tool retrieval**

Imports the template required to extract tool holder data and system tracking points.

**Caution:**

If you have modified any of these objects, for example, created custom setup sheets, importing or updating these objects overwrites your modifications.

2. Confirm that all your selections are correct and then click **Next**.

Teamcenter installs the specified seed parts.

## Import 3D template parts

The following procedure assumes you have completed the steps in *Beginning the installation*. The installation wizard should currently display the **MRL Database Population: Import 3D Template Parts** dialog box.

1. Select one or more of the part family templates or template parts to import or update:
  - **Metric Template Parts (3D templates for Tools)**

Imports metric template part files to generate 3D tool component graphics. These template parts (TPs) are attached to the corresponding MRL tool component classes.
  - **Inch Template Parts (3D templates for Tools)**

Imports inch template part files to generate 3D tool component graphics. These template parts (TPs) are attached to the corresponding MRL tool component classes.
  - **Metric Part Family Templates (PFT for Factory Resources)**



Imports part family templates (PFTs) and attaches them to the **Factory Resources** classes.
2. Select the item type with which each of the objects is imported into the database.

The item types you enter must already exist in the database. For a list of existing item types, move the cursor into the dialog box and press F1. The item types that are requested depend on which types of templates you elected to import in the previous step.

If you enter an item type, you must be aware of the name of the revision of this item. The name of the revision consists of the item name appended by the word **Revision**, for example, **MyItem Revision**. These two terms may or may not have a space between them. You must ascertain whether the revision has a space. If so, type *"item\_name "*. Note the empty space after the item

name, for example "**MyItem**". If the item revision contains no spaces, such as **ItemRevision**, this is not necessary, and you can simply type **Item**.

You must specify the *internal* item type. This name can be different from the displayed item type name. The following table shows the default item types used by the installation.

Internal item type	Item display name	Typical usage	Symbols (item, item revision)
Mfg0MENCTool	NC Tool	Tools (components, assemblies)	
Mfg0MEEquipment	Equipment	Factory resources	

3. Click **Next**.

Teamcenter displays the **Overwrite Existing Data** dialog box.

4. Confirm that all your selections are correct and then click **Start**.

Teamcenter installs the specified classification hierarchies and data into your database.

## Import or update MRL sample resources

This procedure assumes you have completed the steps in *Beginning the installation*. The installation wizard should currently display the **Load 'Sample Resource' for Module(s)** dialog box.

1. Specify which sample data you want to load. You can choose from the following:

- **Tools**

Imports sample tool assemblies and tool components.

- **Machines**

Currently, sample machines are no longer delivered with the MRL kit.

Sample machines must be imported manually from the NX directory.

- **Fixtures** (New in MRL 6.0)

Imports sample fixture components.

- **Measuring Devices**

Imports sample CMM (Coordinate Measuring Machines) and probing tools.

- **Factory Resources**

Imports sample records used for the **Factory Resources** classification.

- **Factory Conveyors**

Imports sample records used for the **Factory Resources** classification.

- **Factory Robots**

Imports sample records used for the **Factory Resources** classification.

- **Factory Weld Guns**



Imports sample records used for the **Factory Resources** classification.








- **Machining Data Library**


Imports technology data, such as materials and feeds and speeds records, for the Machining Data Library.

2. Load the sample data for the specified modules by selecting the corresponding item type for each module. The item types you select here must already exist in the database. For a list of existing item types, move the cursor into the dialog box and press F1.

You must specify the *internal* item type. This name can be different from the displayed item type name. The following table shows the default item types used by the installation.

Item Display Name	Item-Type (Internal)	Icons
<b>Introduced in Version 8.3.0:</b>		
NC Tool	MfgOMENCTool	
Typical usage: Tools (components and assemblies) used for CAM		
<hr/>		
<b>Introduced in Version 8.3.0:</b>		
NC Machine	MfgOMENCMachine	
Typical usage: Machines and robots used for CAM and CMM		
<hr/>		

Item Display Name	Item-Type (Internal)	Icons
<b>Introduced in Version 12.4.0:</b>		
<b>CMM Probe Assembly</b>	<b>PM0CMMProbeAssy</b>	
Typical usage: Probing tool assemblies used for CMM inspection		
<hr/>		
<b>Introduced in Version 12.4.0:</b>		
<b>CMM Probe Component</b>	<b>PM0CMMProbeComp</b>	
Typical usage: Probing tool components used for CMM inspection		
<hr/>		
<b>Introduced in Version 8.3.0:</b>		
<b>Resource</b>	<b>Mfg0MEResource</b>	
Typical usage: Fixtures and devices used for CAM		
<hr/>		
<b>Introduced in Version 8.3.0:</b>		
<b>Equipment</b>	<b>Mfg0MEEquipment</b>	
Typical usage: Factory resources		
<hr/>		
<b>Introduced in Version 10.0.0:</b>		
<b>Robot</b>	<b>Mfg0MERobot</b>	
Typical usage: Factory robots		
<hr/>		
<b>Introduced in Version 10.1.3:</b>		
<b>Conveyor Resource</b>	<b>Mfg0Conveyor</b>	
Typical usage: Factory conveyors		
<hr/>		
<b>Introduced in Version 11.4:</b>		
<b>Factory Tool</b>	<b>Mfg0MEFactoryTool</b>	
Typical usage: Factory tools such as Screwdriver and Weld Gun		

Item Display Name	Item-Type (Internal)	Icons
<b>Introduced in Version 11.4:</b>		
<b>Fixture Unit</b>	<b>PM0NCFixture</b>	
Typical usage: Factory fixtures to locate an assembly in a station		

### 3. Click **Next**.

Teamcenter displays the **Overwrite Existing Data** dialog box.

### 4. Confirm that all your selections are correct and then click **Start**.

Teamcenter installs the specified classification hierarchies and data into your database.

## Import part family templates when you use custom item IDs

By default, the Manufacturing Resource Library installer creates items that are associated with the part family template that have the same ID as the name of the part family template. If your company uses custom IDs, you cannot use the installer to import the part family templates. You must import them using scripts delivered with the software kit.

### 1. Locate the scripts in the following directory:

**advanced\_installations\resource\_management\MRL\PartFamilyTemplates**

- Rename the part family templates located in the **Parts** subdirectory to reflect your custom item IDs.
- Adjust the **PFT\_3\_Sample\_TC101\_assign\_pfts.bat** script to reflect the new IDs.
- Adjust the **PFT\_MRL\_import\_one\_pft.bat** script and specify the correct user, password, and group.
- Run the **PFT\_0\_MRL\_PartFamily\_Templates.bat** script to install the part family templates with the new IDs.

## Install localization data for MRL

Use this procedure to populate the Teamcenter database with a translated version of the Manufacturing Resource Library.

Before you begin installation, make sure you have the latest available version of the Manufacturing Resource Library. If a later version is available, download and extract the latest version and browse to the **advanced\_installations** directory in the location of the extracted files.

For information about later versions, see Support Center.

1. In the **advanced\_installations** directory in the Teamcenter software kit, locate the **advanced\_installations.zip** file and unzip it to a local directory.

2. In the **resource\_management** directory, click **Setup.exe**.

Teamcenter displays the **Teamcenter Configuration Setup** dialog box.

3. Click **Next**.

Teamcenter displays a notification that you must check for the most recent version of the resource library.

4. Click **Next**.

Teamcenter displays the **Setup Type Selection** dialog box.

5. Select **Manufacturing Resources - Localization** and click **Next**.

Teamcenter displays the **Choose TC\_ROOT Directory** dialog box.

6. Select the path to your Teamcenter installation and click **Next**. The path must point to an existing installation.

Teamcenter displays the **Choose TC\_DATA Directory** dialog box.

7. Select the path to the directory containing the database-specific data files and click **Next**. The path must point to an existing **TC\_DATA** directory.

8. Enter the user name, password, and group of the valid Teamcenter administrative logon account and click **Next**.

9. Select the languages for which you want to make the Manufacturing Resource Library available, and then click **Next**.

10. Confirm your selections and click **Next**.

Teamcenter populates the database with localized classification structures for the selected languages.

## Configure NX Library using the installation wizard

When you install the manufacturing library data, you must configure the NX Library so that the new data appears in the library selection dialog boxes when you search in NX CAM for classified resources, such as tools, machines, or devices.

You must perform this configuration on the client on which NX is installed. If your environment includes a shared **MACH** directory, see *Configure NX Library*.

Note:

Before you begin installation, make sure you have the latest available version of the Manufacturing Resource Library (MRL). If a later version is available, download and extract the latest version and browse to the **advanced\_installations** directory in the location of the extracted files.

For information about later versions, see Support Center.

1. In the Teamcenter software kit **advanced\_installations** directory, locate the **advanced\_installations.zip** file and unzip it to a local directory.
2. In the **resource\_management** directory, click **Setup.exe**.

The Manufacturing Resource Library installer **Setup Type Selection** dialog box is displayed.



3. In the **Setup Type Selection** dialog box, select **Configure NX CAM** and click **Next**.

The **NX CAM Resource Base Directory** dialog box is displayed.

Enter the path to the NX CAM resource base directory on this system. This is the directory where the NX CAM specific files are stored. Press F1 for help or Esc to exit Setup.

C:\Program Files\Siemens\NX2306\MACH\resource|

- In the **NX CAM Resource Base Directory** dialog box, enter the path to the NX CAM resource base directory on this system and click **Next**.

Note:

This directory is where the NX CAM specific configuration files are stored, typically in the directory **(UGII\_BASE\_DIR)\MACH\resource**.

- In the operating system explorer window, select the CAM configuration that you want to configure for the MRL usage and click **Open**.

Note:

The default configuration file for NX CAM is **cam\_part\_planner\_mrl.dat**.

- In the **LANGUAGE Selection** dialog box, select the check box you want for the NX Library configuration dialog boxes and click **Next**.

Note:

NX Library configuration dialog boxes are displayed for tools, machines, devices and machining data.

The **Configure NX Library for Tools** dialog box is displayed.

Select the database to be used by the NX Tool Library. Press F1 for help or Esc to exit Setup.

- Use MRL tool database
- Use ASCII tool database
- No change to tool configuration

7. In the **Configure NX Library for Tools** dialog box, select where tools in the NX machine library are retrieved from and click **Next**.

Note:

In this example, **Use MRL tool database** is selected. If you select **No change to tool configuration**, Teamcenter retains the existing settings.

The **Configure NX Library for Machines** dialog box is displayed.

Select the database to be used by the NX Machine Library. Press F1 for help or Esc to exit Setup.

- Use MRL machine database
- Use ASCII machine database
- No change to machine configuration

8. In the **Configure NX Library for Machines** dialog box, select where the machines in the NX machine library are retrieved from and click **Next**.

Note:

In this example, **Use MRL machine database** is selected. If you select **No change to machine configuration**, Teamcenter retains the existing settings.

The **Configure NX Library for Devices** dialog box is displayed.

Select the database to be used by the NX Device Library. Press F1 for help or Esc to exit Setup.

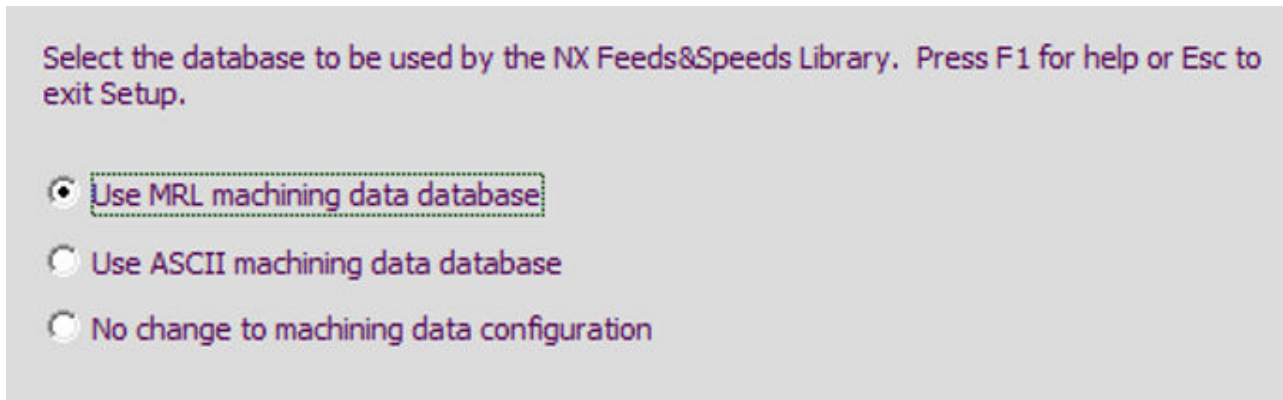
- Use MRL device database
- Use ASCII device database
- No change to device configuration

9. In this example, **Use MRL device database** is selected. If you select **No change to device configuration**, Teamcenter retains the existing settings.

Note:

If you select **No change to tool configuration**, Teamcenter retains the existing settings.

The **Configure NX Library for FeedsSpeeds** dialog box is displayed.



- In the **Configure NX Library for FeedsSpeeds** dialog box, select where the machining data in the NX machine library is retrieved from and click **Next**.

Machining data includes the following libraries:

- **Feeds\_speeds**
- **Machining\_data**
- **Tool\_machining\_data**
- **Part\_material**
- **Tool\_material**
- **Cut\_method**

Note:

In this example, **Use MRL machining data database** is selected. If you select **No change to machining data configuration**, Teamcenter retains the existing settings.

- In the **Confirmation** dialog box, confirm your selections and click **Next**.

Teamcenter performs the following actions:

- Makes adjustments to the NX CAM configuration file according to your library selections.

- Copies the definition files to the target directory.
- Copies event handler files to the target directory.
- Copies the images for the NX CAM tool dialog boxes to the target directory.

A dialog box confirms that the setup is complete.

12. When you are working in NX CAM, select the configuration file that you selected in step 5.

Note:

The default configuration file for NX CAM is **cam\_part\_planner\_mrl.dat**.

When you retrieve a tool from the Teamcenter database in NX CAM, the library selection dialog boxes now show the classification structure from the customer assembly hierarchy, including images corresponding to the new tooling data. You can search for and retrieve your customer-specific tool assembly data.

## Configure NX CMM library using the installation wizard

When you install the inspection library data, configure the NX Library so the new data appears in the library selection dialog boxes when you search in NX CMM for classified resources, such as tools, machines, or devices.

Perform this configuration on the client on which NX is installed. If your environment includes a shared **CMM\_INSPECTION** directory, see *Configure NX Library*.

Note:

Before you begin installation, make sure you have the latest available version of the Manufacturing Resource Library (MRL). If a later version is available, download and extract the latest version and browse to the **advanced\_installations** directory in the location of the extracted files.

For information about later versions, see Support Center.

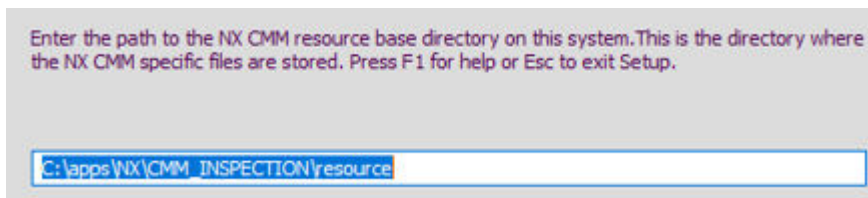
1. In the Teamcenter software kit **advanced\_installations** directory, locate the **advanced\_installations.zip** file and unzip it to a local directory.
2. In the **resource\_management** directory, click **Setup.exe**.

The Manufacturing Resource Library installer **Setup Type Selection** dialog box is displayed.



3. In the **Setup Type Selection** dialog box, select **Configure NX CMM** and click **Next**.

The **NX CMM Resource Base Directory** dialog box is displayed.



4. In the **NX CMM Resource Base Directory** dialog box, enter the path to the NX CMM resource base directory on this system and click **Next**.

Note:

This directory is where the NX CMM specific configuration files are stored, typically in the directory **(UGII\_BASE\_DIR)\CMM\_INSPECTION\resource**.

5. In the **Confirm Selections** dialog box, review your selections and click **Next** to confirm.

Teamcenter performs the following actions:

- Copies the definition files to the target directory.
- Copies event handler files to the target directory.
- Copies the images for the NX CMM tool dialog boxes to the target directory.

A message confirms the task is complete.

## Configure NX Library

your enterprise installs NX using a shared **MACH** directory, there are several steps that you must perform manually. These steps mirror the steps automatically performed by the *Configure NX Library using the installation wizard* procedure performed when you use a local **MACH** directory.

This procedure pertains to the following files:

**MACH\resource\configuration\cam\_part\_planner\_mrl.dat**

**MACH\resource\library\tool\inclass\dbc\_mrl\_tooling\_library\_tlas.tcl**

**MACH\resource\library\tool\inclass\dbc\_mrl\_tooling\_library\_tlas\_en.def**

**MACH\resource\ug\_library\dbc\_mrl\_general.tcl**

**MACH/resource/configuration/cam\_part\_planner\_mrl.dat**

**MACH/resource/library/tool/inclass/dbc\_mrl\_tooling\_library\_tlas.tcl**

**MACH/resource/library/tool/inclass/dbc\_mrl\_tooling\_library\_tlas\_en.def**

**MACH/resource/ug\_library/dbc\_mrl\_general.tcl**

1. Create a copy of your **cam\_part\_planner\_library.dat** file and rename it to **cam\_part\_planner\_mrl.dat**.
2. Change the following line in the **cam\_part\_planner\_mrl.dat** file:

Old:

```
LIBRARY_TOOL,${UGII_CAM_LIBRARY_TOOL_INCLASS_DIR}dbc_inclass_tlas.def,
${UGII_CAM_LIBRARY_TOOL_INCLASS_DIR}dbc_inclass_tlas.tcl
```

New:

```
LIBRARY_TOOL,${UGII_CAM_LIBRARY_TOOL_INCLASS_DIR}dbc_mrl_tooling_library_tlas_en.def,
${UGII_CAM_LIBRARY_TOOL_INCLASS_DIR}dbc_mrl_tooling_library_tlas.tcl
```

If you use a customized configuration file at your company, make the change in the customized file and continue to use this file to initialize NX CAM.

3. Copy the following files to your **MACH\resource\library\tool\inclass\MACH/resource/library/tool/inclass/** directory.

```
dbc_mrl_tooling_library_tlas.tcl
dbc_mrl_tooling_library_tlas_en.def
```

4. Copy `dbc_mrl_general.tcl` to `MACH\resource\ug_library\MACH\resource\ug_library`.
5. When you initialize NX CAM, use the modified `cam_part_planner_mrl.dat` file or the customized configuration file that you modified in step 2.

## Configure the graphics builder for MRL

To use the following features in Teamcenter, you must configure the NX graphics builder:

- Part family member creation
- NX auto assembly
- 3D model import for catalog data
- **Check NX CAM tool retrieve** in MRL
- Extract Holder Data

Prerequisites:

- The graphics builder must be installed on the Teamcenter server. This is the server that runs the **tcserver** process.
- **The `mrl_retrieve_camsetup_metric` or `mrl_retrieve_camsetup_inch` NX part file item must be imported using the Template parts for temporary tool retrieval option.** You can search for an item with this ID to check the existence of this file.
- The `...\MACH\resource\configuration\` directory of the NX installation that resides on the Teamcenter server where the graphics builder is installed must contain the `cam_part_planner_mrl.dat` file. This file points to the `dbc_mrl_tooling_library_tlas.tcl` event handler file required for graphics creation. The event handler file must be up-to-date. You can copy the most recent file by setting the tool database to MRL when you complete the *Configure NX Library using the installation wizard* procedure.

The following procedure installs the graphic macros on the server where the graphics builder is installed.

1. Obtain the Manufacturing Resource Library software kit.
2. Click **Setup.exe**.

Teamcenter displays the **Setup Type Selection** dialog box.

3. Select **Configure Fixture TCL Graphic Macros**.

4. Click **Next**.

Teamcenter displays the **Choose TC\_ROOT Directory** dialog box.

5. Select the path to your Teamcenter installation and click **Next**. The path must point to an existing installation.

Teamcenter displays the **Choose TC\_DATA Directory** dialog box.

6. Select the path to the directory containing the database-specific data files and click **Next**. The path must point to an existing **TC\_DATA** directory.

7. Enter the user name, password, and group of a valid Teamcenter administrative logon account.

The **plmxml\_import** utility uses this logon information to import the data into the Teamcenter database.

8. Click **Next**.

Teamcenter displays the **Choose MRL Graphic Macros Directory** dialog box containing the default path to store graphic macros.

The installation procedure modifies the **NXGraphicsBuilder** preference and copies required files into Teamcenter directories (*TC\_ROOT\bin\lx\_graph\TCL\_Create\_Graphics*).

9. Verify that this path is the correct path or modify it by clicking **Browse**.

Teamcenter displays the **Confirmation** dialog box.

10. Click **Start**.

Teamcenter copies the graphic macros to the database and modifies the **NXGraphicsBuilder** preference.

### Upgrading from an earlier MRL version

In addition to the steps outlined above, it is advised that you:

- use the MRL graphics builder to copy the latest **ics\_graphicsbuilder.tcl** file.

## Configure MRL Connect

MRL Connect allows native NX users to retrieve cutting tool assemblies from the Manufacturing Resource Library (MRL) in Teamcenter. New tools must be created in the library using Teamcenter.

### Prerequisites:

- A Teamcenter four-tier server installation must exist.
- Teamcenter client communication system (TCCS) must be installed on the machine running NX.
- Java Runtime Environment (JRE) must be installed on the machine running NX.
- The Manufacturing Resource Library must be populated on the Teamcenter server.
- The manufacturing preferences in NX CAM must point to a configuration file that has been set up for Resource Manager access, for example, the **cam\_native\_rm\_library.dat** configuration file provided with NX.

Before you begin installation, make sure you have the latest available version of the Manufacturing Resource Library. If a later version is available, download and extract the latest version and browse to the **advanced\_installations** directory in the location of the extracted files.

For information about later versions, see Support Center.

To configure MRL on the computer running NX:

1. In the **advanced\_installations** directory in the Teamcenter software kit, locate the **advanced\_installations.zip** file and unzip it to a local directory.

2. In the **resource\_management** directory, click **Setup.exe**.

Teamcenter displays the **Teamcenter Configuration Setup** dialog box.

3. Click **Next**.

Teamcenter displays a notification that you must check for the most recent version of the resource library.

4. Click **Next**.

Teamcenter displays the **Setup Type Selection** dialog box.

5. Select **Configure MRL Connect**.

6. Select **MRL Connect for NX - Configure environment on NX Client**.

This option creates the command file **start\_nx\_using\_mrl\_connect.bat** in the location *user-profile\MRLConnect*, for example:

**C:\Users\login\_username\MRLConnect\start\_nx\_using\_mrl\_connect.bat**

7. Enter the path to the NX base directory.
8. Enter the path to the directory where FMS is locally installed.
9. Enter the path to the Java Runtime Environment.
10. Enter server name and port number of the Teamcenter server.

This information is used to build the NX environment variable on the NX client.

Tip:

Ask your Teamcenter installer for the port number.

11. Check your selections and complete the installation.

MRL Connect supports only the retrieval of tools from MRL into NX CAM.

Machines, devices, feeds and speeds data, machining data, cut methods, part materials, and tool materials are *not* supported. Those modules should use ASCII, not MRL, in the configuration file.

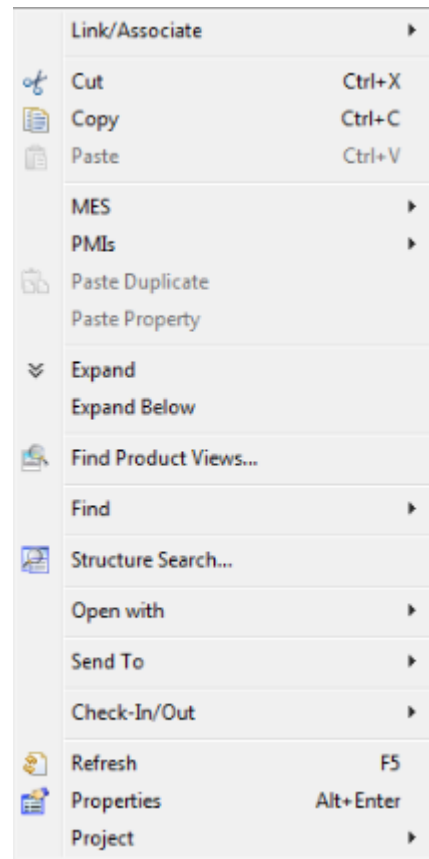
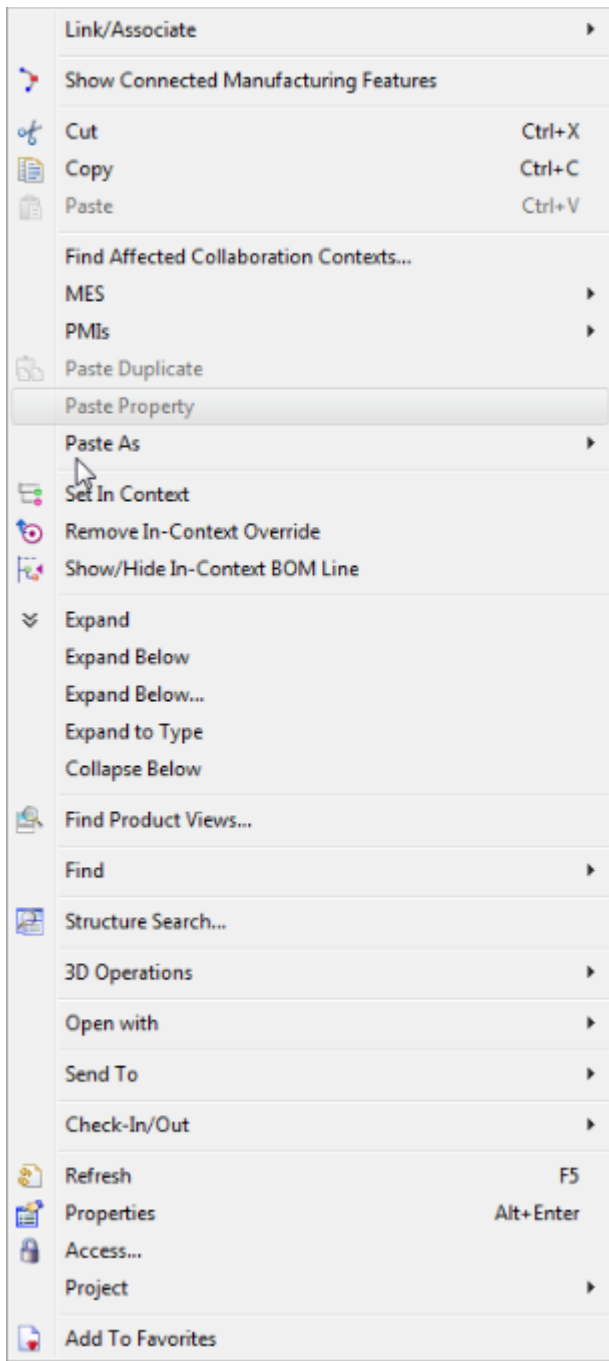
# 3. General configuration and customizing tasks

## Configure the commands in the context menu

You can customize which menu commands are displayed in the context menu (the menu that appears when you right-click an object) so that only the menu commands relevant to your tasks appear. You can specify that the commands that appear when you right-click an object in a view housed in one perspective are different than those that appear when you click the same type of object in a different perspective. This allows you to tailor the user interface to different workflows and user needs.

You must have DBA privileges to complete this task.

## Context menu — before suppression      Context menu — after suppression



1. Display the required menu command by adding the following extension to the **plugin.xml** file located in `portal\plugins\com.teamcenter.siemens.lookandfeel_10000.1.0`:

```
<extension point="org.eclipse.ui.menus">
  <menuContribution
    allPopups="false"
```

```

locationURI="menu:window?after=additions">
<command
  commandId="com.teamcenter.rac.cme.contextSuppression"
  mode="FORCE_TEXT"
  style="push">
  <visibleWhen>
    <reference
definitionId="com.teamcenter.rac.cme.mpp.inMainView">
      </reference>
    </visibleWhen>
  </command>
</menuContribution>
</extension>

```

**Note:**

To activate the command you may need to register your client by double clicking on genregxml under ..\Teamcenter Root\portal\registry. Without this the command will not be visible.

2. Choose **Window** → **Suppress Context Menus**.

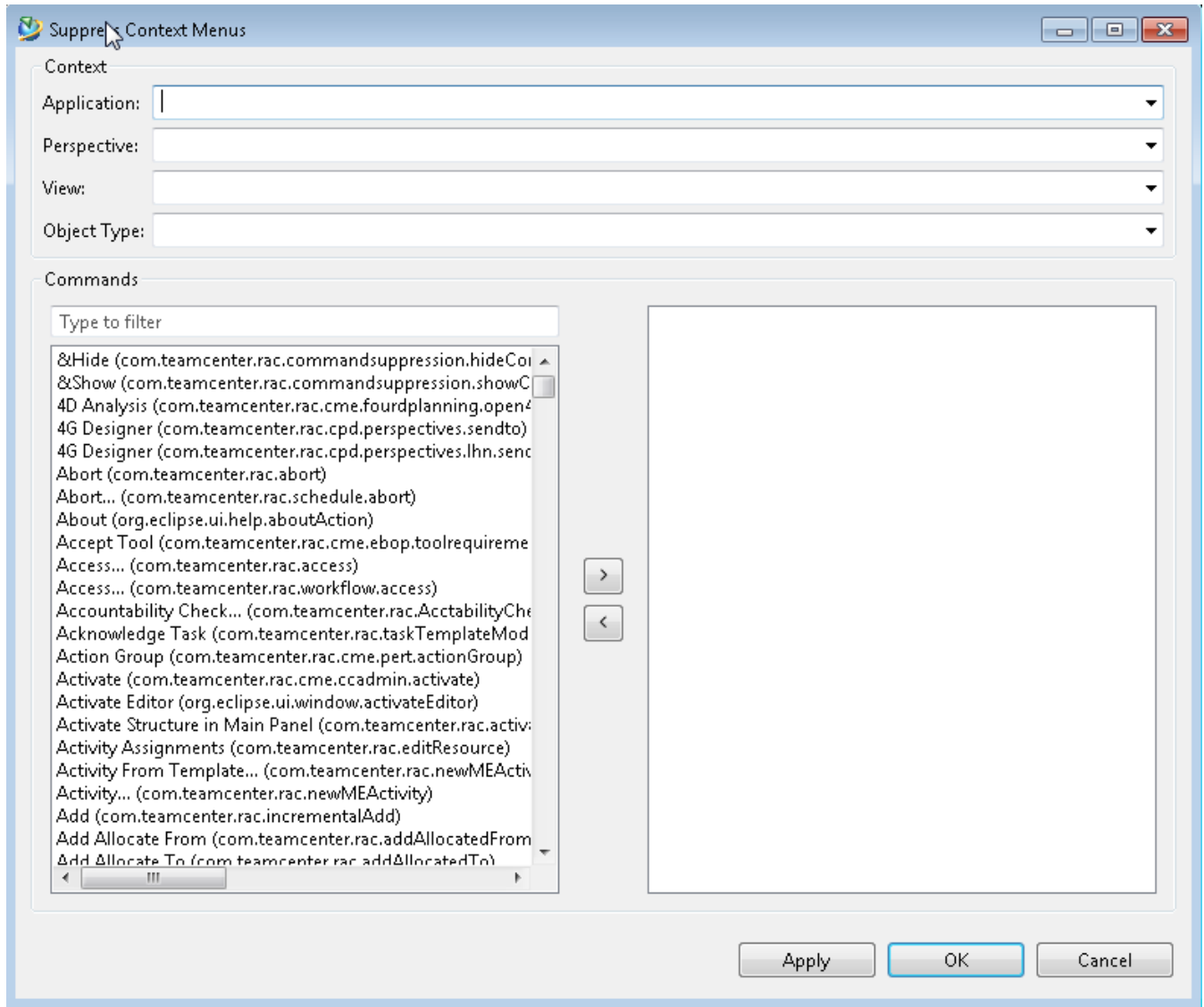
**Note:**

This command may be hidden in your installation. Your administration can display it.

Teamcenter displays the **Save** dialog box where you can create an XML file that stores the suppression data or select an existing XML file that you want to modify.

3. Type the name of a new file or select an existing file to modify and click **Save**.

Teamcenter displays the **Suppress Context Menus** dialog box. This dialog box allows you to specify the type of object for which the context menu is customized. Additionally, you select the view, perspective, and application in which this customization is visible.



4. Select the application, perspective, view, and type of selected object for which you want to limit the context menu commands.

You can select any of the **Open With** commands to suppress entries in the **Open With** context menu.

You can also suppress context menus in customized Manufacturing Process Planner perspectives.

5. Add the commands by typing their names in the **Commands** dialog box, selecting them from the commands list, and adding them to the list on the right with the arrow.
6. Click **OK**.

Teamcenter saves the results in the XML file.

7. Make the configured menu available to different groups, roles, or users as follows:
  - a. Save the XML file from the previous step to a **Context Menu Suppression** dataset.
  - b. Add the name of the dataset to the **TC\_ContextMenuSuppression** preference and publish this preference to the required group, role or user.

## Configuring the business object types in the New dialog boxes

To reduce the chance of error when Manufacturing Process Planner users select a new business object type to create, you can specify which object types are available in the **New** dialog boxes for specific users, groups, or roles. The filtering is dependent upon the type of **New** dialog box and selected object type. You perform this configuration using the following preference:

**MEFilteredChildTypes** *\_type-of-new-dialog\_type-of-selected-component*

For example:

- **MEFilteredChildTypes\_Item\_Item** for the **New Item** dialog box opened on a selected item
- **MEFilteredChildTypes\_OP\_Process** for the **New Operation** dialog box opened on a selected process
- **MEFilteredChildTypes\_Item\_None** for the **New Item** dialog box when no selection is active

You then publish the preference to the appropriate user, group, or role.

## Hide a custom perspective

As with all Teamcenter perspectives, you can hide a Manufacturing Process Planner perspective so that users cannot see it in the user interface using the **HiddenPerspectives** preference. If you create your own Manufacturing Process Planner perspective, you must perform the following steps to hide this perspective using the **HiddenPerspectives** preference. The example uses a new perspective called **MyCustomizedPerspective**.

1. Add the following information to the **org.eclipse.ui.activities** extension:

```
<activity
    description="Turn off MyCustomizedPerspective
perspective"
    id="ID-of-MyCustomizedPerspective"
    name="MyCustomizedPerspective"
</activity>
<activityPatternBinding

activityId="com.teamcenter.rac.cme.mpp.MyCustomizedPerspective"
    pattern="com\teamcenter\rac\cme\mpp/
com.teamcenter.rac.cme.mpp.
```

```
MyCustomizedPerspective">
</activityPatternBinding>
```

The ID is the value assigned in the perspective definition in the **plugin.xml** file.

The pattern value is derived from the value of the ID in the customized perspective definition in the **plugin.xml** file.

For example, if the customized perspective definition ID value is **com.rcpcompany.demo.ui.views.History**, the pattern is then **.\*/com\.rcpcompany\.demo\.ui\.views\.History**.

2. Add the perspective name (**MyCustomizedPerspective** in the example) to the **HiddenPerspectives** preference.

Teamcenter hides the custom perspective in the user interface.

## Defining which occurrence types display in paste and assign resource commands

In the **MEDisplayOccurrenceType** preference, you can define conditions to display which occurrence types appear when pasting objects in the Manufacturing Process Planner rich client. This preference is used, for example, when end users access the **Paste As**, **Paste Special**, and **Assign Resource** commands and in the **Occurrence Type** column visible in bill of materials (BOM) views.

Note:

Regardless of how you configure the preference, when pasting objects with multiple occurrence types, the system will only show occurrence types common to both the source and target. If no common occurrence types are found, the end user receives a message informing him of that fact.

## Example of how to copy and assign custom occurrence types to processes and operations

This example shows how to add copy, paste, and assign custom occurrence types created in the Business Modeler Integrated Development Environment () to processes and operations in Manufacturing Process Planner.

1. In the BMIDE, create new custom occurrence types for your project.

Project Name	Display Name	Description
DLDEV	<b>DL_MEPublication</b>	ME Publication Used for internal BOP product document.
DLDEV	<b>DL_MEReference</b>	ME Reference Used for external BOP product and supplier documents.

2. After deploying the template, add a new preference (for example, **DL\_DocumentOccTypes**), containing the new custom occurrence type values **DL\_MEPublication** and **DL\_MEReference**.
3. Update the preference **MEDisplayOccurrenceType** with the new preference values.

**MEProcess:Document:DL\_DocumentOccTypes**

**MEOP:Document:DL\_DocumentOccTypes**

4. Update the preference **MEAssignCustomizedOccurrenceType** with new values.

**MEProcess.Document:DL\_MEPublication**

**MEProcess.Document:DL\_MEReference**

**MEOP.Document:DL\_MEPublication**

**MEOP.Document:DL\_MEReference**

5. Update the preference **MERelationTypePartsOtherAssigned** with the new custom occurrence type values.

**DL\_MEPublication**

**DL\_MEReference**

In Manufacturing Process Planner, when you add a new document to a process or operation, then copy it, right-click another process or operation and then select **Paste As**→**MEReference**, the item is added with the appropriate custom occurrence type, **MEPublication** or **MEReference**.

## Configure the update of multiple product views

Teamcenter uses the Dispatcher infrastructure to update multiple product views. In addition to installing this software using Teamcenter Environment Manager, you must also install the translator specific to creating multiple product views—the **Update Mass Product Views** translator.

1. Check that **VisAutomationApp** is installed on the same machine on which Dispatcher is installed.
2. Set **FMS\_HOME** as a system environment variable with the appropriate value and restart the Dispatcher machine.
3. Install the **Update Mass Product Views** translator using the TEM installer.
4. Verify that the ACL rule for Dispatcher is set in Teamcenter.

5. Open `Dispatcher_Root_Directory\Module\conf\translator.xml` in a text editor and ensure the `isActive` attribute in the `TcUpdateProductViews` extension is set to `true`.
6. Open Windows services and restart the following two services in the following order:
  - a. **Dispatcher Scheduler**
  - b. **Dispatcher Module**
7. Run Dispatcher client manually by starting `Dispatcher_Root_Dir\DispatcherClient\bin\runDispatcherClient.bat`.

Teamcenter opens a console.

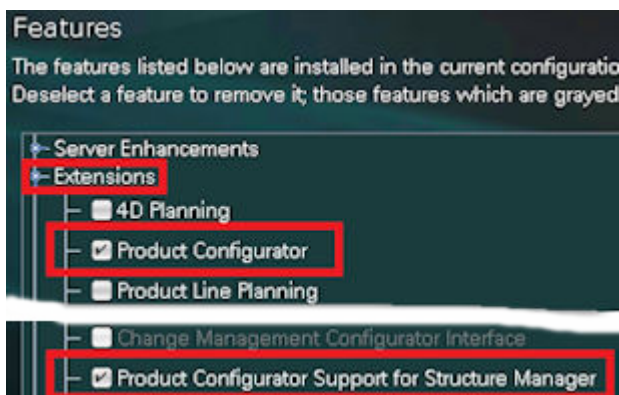
## Set up Manufacturing Process Planner for Product Configurator

Note:

If you already have Product Configurator available for Structure Manager there is no need to perform any of the following steps.

The administrator must install the following features using Teamcenter Environment Manager to connect Product Configurator to Manufacturing Process Planner and Structure Manager.

1. Run Teamcenter Environment Manager (TEM) and navigate to the **Features** list.
2. Navigate to the **Extensions** branch and select **Product Configurator** and **Product Configurator Support for Structure Manager**.



3. Complete the installation of the features in TEM.

The features you added provide access to Product Configurator.

- Restart Teamcenter and log on as administrator.

The restart is required to activate the new features.

- Choose **Edit**→**Options** to open the preferences dialog box and set the Product Configurator preference.
- Click the **Search** button and type **PSM** in the **Search On Keywords** box.

Preferences that begin with **PSM** are listed.

Name	Location
PSM_classic_variants_date_families	Site
PSM_default_configurator_context	Site
PSM_enable_PLMXML_Assembly	Site
PSM_enable_product_configurator	Site
PSM_global_option_item_ids	None

- Select **PSM\_enable\_product\_configurator** and click **Edit**.
- Type **True** in the **Value** box and click **Save**.
- Click **Close**.

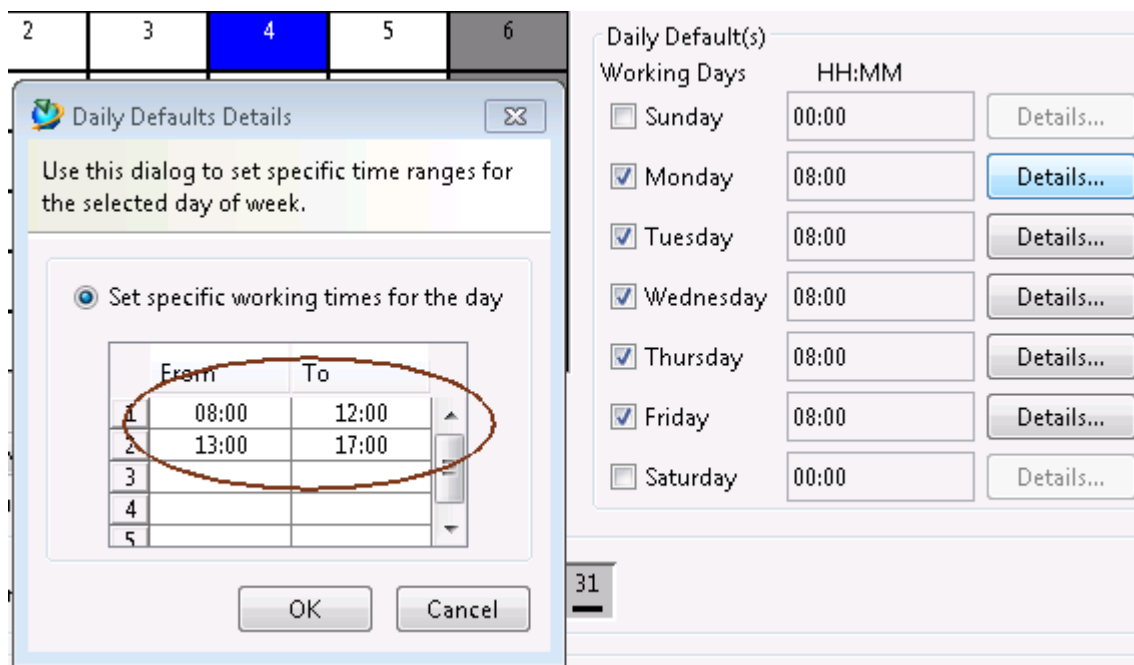
## Set up 4D planning

Adding a time component to process planning to simulate and visualize construction over an extended period of time is referred to as *4D planning*. 4D planning is used in very large structures to ascertain what parts are already assembled on a specific date in a specific part of the structure. For example, if you build a ship, you may want to begin fitting the cabins with furniture. It is necessary to know if the piping and electrical work is complete by April 30th to begin subsequent work. You can query your structure, taking the date into consideration, to see what has been assembled by April 30th.

Teamcenter uses an embedded **Schedule Manager** view to help you associate scheduling information with process planning operations. The **Manufacturing – 4D Scheduling** perspective already contains this view, in addition to structure views to assist you. This is the same view that is contained in the Schedule Manager application. Making a change in one changes the data in the other.

To configure 4D planning:

1. Install the **4D Planning** extension using Teamcenter Environment Manager (TEM).
2. Set the time zone on the computer running the rich client to **UTC (GMT)**. This is the only supported time zone.
3. In the rich client, open the Organization application and modify the **Calendar** setting.
  - a. Set the calendar time zone to **GMT (GMT +00:00)**.
  - b. Remove lunch breaks in the calendar by clicking **Details** next to the weekday entries and modifying the times.



4. Set the following preferences:

Preference	Value
MECopyIdInContextToAssignedLine	true
SiteTimeZone	Etc/UTC or Etc/GMT
ScheduleTaskAvailableAttributesWithRelations	ScheduleTask.fdp0AssociatedProcesses
QS_QSEARCH_ENABLED	true
QSEARCH_update_enabled	true
QS_BBOX_GENERATION_ENABLED	true
QS_BBOX_GENERATION_FROM_NX_ENABLED	true

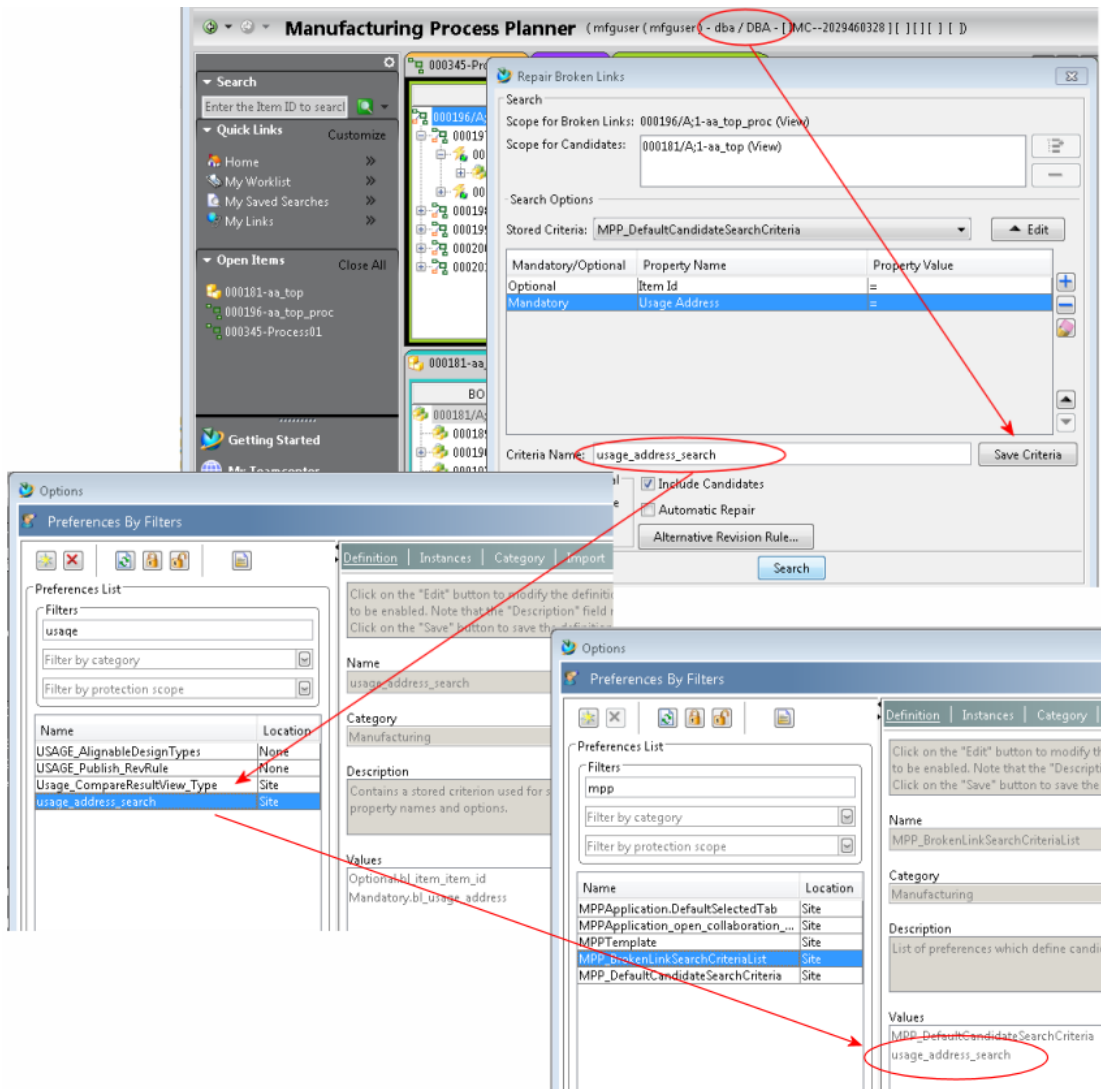
Preference	Value
QS_TRUSHAPE_GENERATION_ENABLED	true
QS_SPATIAL_ENABLED	true
QSEARCH_foreground_processing_halted	false

Teamcenter sets the following preferences for you automatically.

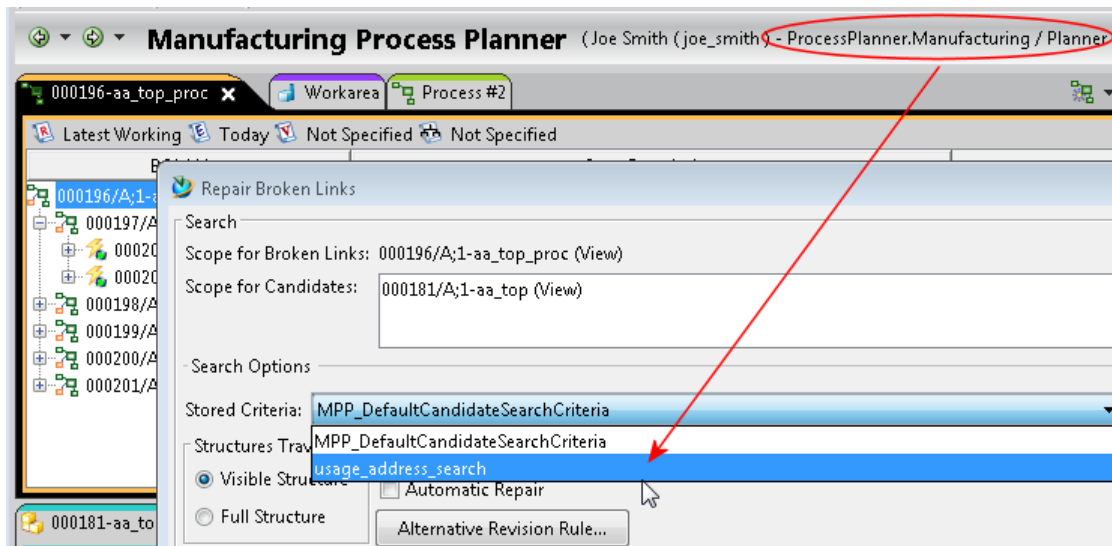
Preference	Value
ME4DResultsViewDisplayNameColumnsShownPref	Mfg0BvrProcess.bl_indented_title Mfg0BvrProcess.ScheduledStartDate Mfg0BvrProcess.ScheduledFinishDate Mfg0BvrProcess.ScheduledCalculatedDuration
ME4DResultsViewColumnsWidthPref	45, 15, 15, 15
ME4DSearchCriteriaDate	fdp0ScheduledStartDate
MEProcessIconPropertyNames	start_date
MECopyIdInContextToAssignedLine	true

## Creating new search criteria for a broken link search

The **Broken Links** dialog box allows you to select a search criteria. All available search criteria are stored in the **MPP\_BrokenLinkSearchCriteriaList** preference. Each value in this preference is, in turn, a preference that holds a list of properties contained in the search criteria. As an administrative user, you can create your own search criteria and save them under a new criteria name. Doing so creates a new preference of the same name as the search criteria that holds the list of properties you specified. Teamcenter adds the name of the new preference to the list of available search criteria in the **MPP\_DefaultCandidateSearchCriteria** preference. In this way, the new search criteria that you created is available to every user.



Nonadministrative users can see these search criteria in the **Broken Links** dialog box. They can modify them and save their changes to a user preference with the same name as the original search criteria preference. The next time they use the search criteria, they see the values stored in the user preference including their modifications. If they want to use the original search criteria that the administrative user saved, they must delete the user preference.



## Using the ME-revise-AH action handler

To better support lifecycle planning for Process Simulate users who cannot revise an item in Teamcenter, you can use the **ME-revise-AH** action handler in your Teamcenter bill of process/bill of equipment (BOP/BOE) workflow after creating a release to revise the item.

Argument	Value(s)	Description
<i>revrule</i>	<b>Latest Working</b>	Sets the criteria to manage appropriate revisions of components (based on statuses) to configure a product structure. Rules can be customized but may include values such as <b>Any Status; Working</b> or <b>Working (Curr Group); Any Status</b> . This value is required if an item or revision is specified.
<i>log</i>	<i>C:\temp\BIW_Milestone.txt</i>	Records the official record of events that occurred during the revision action.
<i>closurerule</i> (optional)	1 (Boolean=yes)	Defines the scope of the plan for traversing from a target to a related object and whether to process or ignore the object.
<i>statuslist</i>	<b>TCM;MS1;MS2</b>	Lists all the release status names, separated by semicolons, that should be considered for revision.
<i>updateprecise</i>	1 (Boolean=yes)	Forces updates on precise items. Precise items are generally used when a BOM is in its final stages and specific item revisions represent the BOM.
<i>refresh</i>	1 (Boolean=yes)	Updates or reloads your Teamcenter BOP/BOE workflow from the database after the action handler revises the item.

Example:

```
ME-revise-AH -u=myUserName -p=myPassword -g=myGroup -revrule="Latest Working"
-log=C:\temp\BIW_Milestone.txt -closuresrule=1 -statuslist=TCM -refresh
```

## Customizing the BOM line quantity

Certain sites add dimensions at the end of each part name, resulting in a long BOM line name. These dimensions at the end of the **BOM Line** column and the value in the **Quantity** column may run into each other and confuse the user.

To avoid this, you can use the **BOM Line Title** column instead of the **BOM Line** column. Add the **BOM Line Title** column, move it to the far left, and then remove the **BOM Line** column from the display.

You should then choose **Tools → Options** in Structure Manager and enter a custom title in the **BOM Line Title format for Items of type: Item** box. For example, the following entry shows only the item ID and name separated by two dashes, but no quantity value:

```
$bl_item_item_id - - $bl_item_object_name
```

Note:

As the **Options** dialog box lists only actual item types in Structure Manager, you cannot customize process or manufacturing item types directly in this window. To do this, you must first choose **Options → Preferences** and create a new preference for each applicable item type, for example, **BOMLine\_Title\_Format\_OP** or **BOMLine\_Title\_Format\_Process**. You can then define the current value of the preference name in the format shown in the example.

## Specify a property value to be a combination of other property values

You can automate the generation a property value for BOM lines to be based on the concatenation of other properties of the BOM line. The combination of these properties results in a unique string that Teamcenter assigns as a new property value.

- Define the properties that are combined to create the new property value in the **MEIdGenerationPropertySetting** preference.

These can be any single-valued BOM line property. If the property value is empty, Teamcenter uses that empty value. Two typical properties that you may combine are item ID and absolute transformation matrix.

- Define tolerances for translation and rotation in the following preferences:

## ME\_angular\_tolerance ME\_translation\_tolerance

### Note:

If you use a CAD integration with Catia or other CAD application and use BOM (Bill of Materials) to BOM or BOM to BOE (Bill of Equipment), updates sometimes cause small BOM CAD tolerance variations so that many parts in the model show as changed during advanced accountability checks. To reduce the number of outliers during such checks, increase the **ME\_angular\_tolerance** and **ME\_translation\_tolerance** values to reduce the tolerances.

Alternatively, you can:

- Run the **me\_stamp\_ids** utility to generate the new property value for an entire structure, based on a closure rule.
- Create a workflow process based on the **ME-stamp-ids-AH** action handler to automate the process from the user interface.

When using the utility or action handler, you can specify whether Teamcenter allows duplicate property values in the **MEIdGenerationAllowDuplicates** preference.

## Configure incremental changes

If you plan to use incremental changes to control the manufacturing process, you must complete the following steps:

1. By default, incremental changes are not enabled. To enable incremental changes, set the **Incremental\_Change\_Management** preference to **true**.
2. Configure the change types to use with incremental change with the Change Viewer application.
3. By default, all incremental changes (configured and unconfigured) are visible to the user. To hide unconfigured changes, set the **ShowUnconfiguredByChangeEffectivity** preference to **true**.
4. Define a release status that is attached to an incremental change when the user first creates it. For example, the following setting creates each incremental change with a **Pending** status.

```
Incremental_Change_ReleaseStatus=Pending
```

The status must already exist and have an Access Manager rule that allows write access to objects with this status.

5. Ensure an **In IC Context** rule is available in Access Manager, allowing a user write access to a released structure.

## Make manufacturing change notices mandatory for BOP structures

You can configure the database so that Manufacturing Process Planner and Part Planner end users must set a manufacturing change notice (MCN) before they can edit certain types of BOPs. If they attempt to edit a BOP in this configuration, or open a collaboration context containing such a BOP, they will receive a warning.

A Change Notice must be set before editing this structure.

Assuming that change management is installed and the site preference **MEMCN\_management\_enabled** is set to **true**, you can configure the types of structures for which the warning is displayed by adding the item type of the BOP's top line to the role-based preference **MEMCN\_change\_notice\_mandatory\_types**.

Example:

**Process**

**Mfg0MEPlantBOP**

You can also set MCN requirements by a structure's root node type and release status name.

## Configure manufacturing change notices (MCNs)

An engineering change notice (ECN) records engineering design changes to a product. Manufacturing change notices (MCNs) are essentially ECNs that are specifically intended for manufacturing rather than engineering needs. An MCN is not a COTS (commercial-off-the-shelf) object and can be any custom change notice revision that overrides the following business object constants.

Name	Value	Overri...	COTS	Template
CMHasImpactedItemCreCondition	Cm0isCMHasImpactedItemCreatable		✓	cm
CMHasProblemItemCreCondition	Cm0isCMHasProblemItemCreatable		✓	cm
CMHasSolutionItemCreCondition	Cm0isCMHasSolutionItemCreatable		✓	cm

As an administrator, you can modify the constants as shown.

Name	Value	Overridden	COTS	Template
CMHasImpactedItemCreCondi...	isTrue	✓		testmcn
CMHasProblemItemCreCondition	isTrue	✓		testmcn
CMHasSolutionItemCreCondition	isTrue	✓		testmcn

**Note:**

MCN capabilities require a special license and installation to work in Teamcenter. After installation, set the **MEMCN\_management\_enabled** preference to **true** to enable change notice revision commands in Manufacturing Process Planner.

To allow users to set statuses on an MCN, change notices must have valid release statuses defined in the **MEMCN\_release\_statuses** preference.

You can make change notices mandatory before users may edit certain types of structures. To configure such types of structures, adding the item type of the structure's top line to the role-based **MEMCN\_change\_notice\_mandatory\_types** preference.

**Example:**

**Process**

**Mfg0MEPlantBOP**

Use the **MEMCN\_release\_revision\_mode** preference to control how statuses are created on item revisions being tracked by a change notice revision. Use closure rules, such as a rule to exclude tools with a status of **TCM Released**, to exclude items from being captured by a change notice.

**Tip:**

You can customize change notice report layout and elements using preferences with the suffix **MECTRev**.

**Note:**

When a subassembly node is assigned from an EBOM to an MBOM, change notices track the assigned node and all of its children because the default value of the preference **MEMCN\_solution\_object\_collection\_rule** is **GetChangedProcessLines**. To add custom business objects to change notices, you must add them to the closure rule **GetChangedProcessLines**. If an EBOM line type is different from MBOM (for example, part versus item), you can modify the closure rule in this preference to define which lines are assigned and which are not. For example, to keep the assigned line of type **PART** from being added to the change notice, define the closure rule:

```
BOMLine:BOMLine:bl_all_child_lines:SKIP:PRIMARY.bl_item_object_type=
=
"Part"
```

## Restructure BOP product views using an upgrade utility

If you are upgrading from earlier versions of Teamcenter to the 11.6 release, you may encounter some problems with product views. You can use the **upgrade\_pv\_with\_visiblelineinfos** utility to identify and correct broken product views.

Using the utility with various mode types lets you upgrade product views in different ways. When **mode=bopwithpv**, you can generate an XML file that lists all affected product views.

Example:

```
C:\Siemens\Tc116\tcroot\tc_menu> upgrade_pv_with_visiblelineinfos -u=zz -p=zz g=myGroup
-mode=bopwithpv -outputfile=13Nov2018.xml -log=13Nov.log
```

The file lists each individual BOP with its itemID and other attributes, along with each affected product view (PV). You can compare the product views in the XML by opening the BOPs in Teamcenter and performing a product view search.

```
<BOP2 LastModifiedDate="03-Sep-2018 19:47:34" itemID="L029084" uid="CaIAAAEwJIoCfC">
  <PV1 attachedLineID="L000092_4" attachedLineName="Op10_a" pvUID="41DAAA0kJIoCfC">PV1</PV1>
  <PV2 attachedLineID="L000092_4" attachedLineName="Op10_a" pvUID="YWLAAA0kJIoCfC">PV1_zoomOut</PV2>
  <PV3 attachedLineID="L000092_4" attachedLineName="Op10_a" pvUID="oGEAAA0kJIoCfC">PV2</PV3>
</BOP2>
```

You can also extract item IDs for product views that need to be upgraded by setting **-mode=pvsneedingmigration**. The XML file can be used to upgrade all affected product views reported in a BOP by running the utility once more and setting **rootitemid** to the BOP *itemID*.

Example:

```
C:\Siemens\Tc116\tcroot\tc_menu> upgrade_pv_with_visiblelineinfos -u=zz -p=zz g=myGroup
rootitemid=L029084 -log=13Nov.log
```

Once you have run the utility, you can verify that the product views have been upgraded by running the utility again as in the first example. If fixed, the product views do not appear in the resulting XML file.

You can also batch update all BOPs in the XML file by setting **-mode=upgradepv** and specifying the XML file as the **inputfile**.

Example:

```
C:\Siemens\Tc116\tcroot\tc_menu> upgrade_pv_with_visiblelineinfos -u=zz -p=zz g=myGroup
-mode=upgradepv -inputfile=13Nov2018.xml -log=13Nov.log
```

If you have a collaboration context (CC) that contains multiple BOPs, you can also use the utility to upgrade all BOPs contained in it by extracting BOP IDs.

Example:

```
C:\Siemens\Tc116\tcroot\tc_menu> upgrade_pv_with_visiblelineinfos -u=zz -p=zz g=myGroup  
-cc=myCC -log=13Nov.log
```

To force upgrades to affected views in a BOP and thereby verify all are updated, you can set **-mode=upgradeallpvs**

Example:

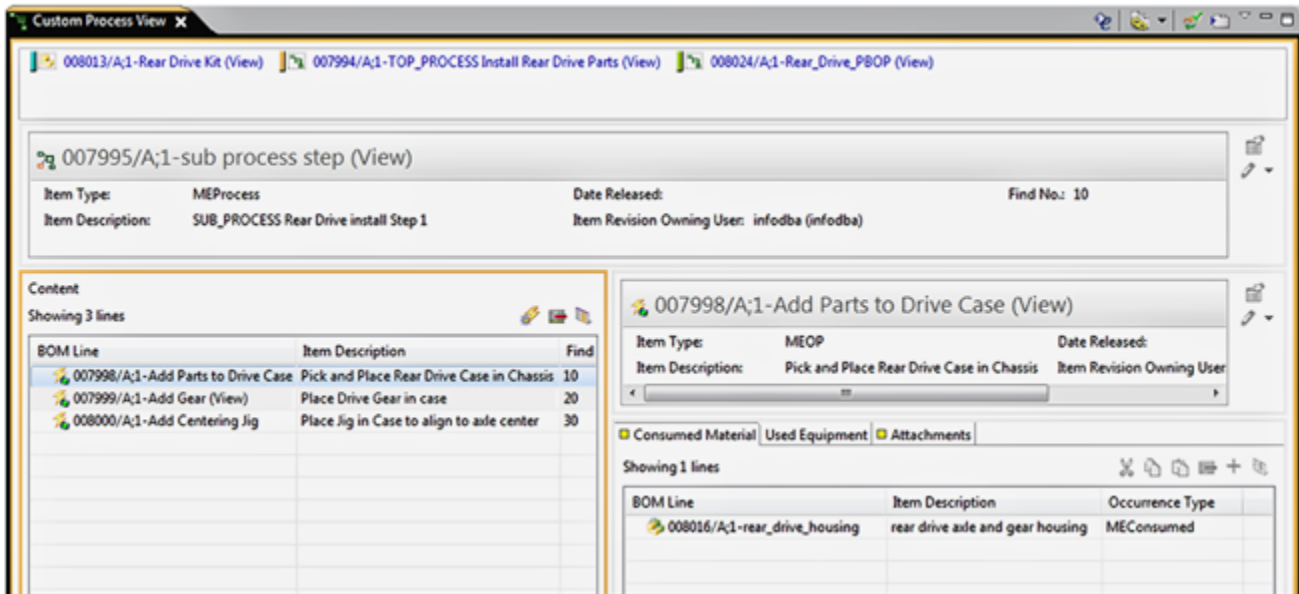
```
C:\Siemens\Tc116\tcroot\tc_menu> upgrade_pv_with_visiblelineinfos -u=zz -p=zz g=myGroup  
-mode=upgradeallpvs -log=13Nov.log
```



# 4. Setting up the Custom Process View

## Configure the Custom Process View

You can view a process plan in an alternative view called the **Custom Process View**. This view is based on configurable style sheets. Additionally, a variety of preferences define the information displayed in the view.



To configure the **Custom Process View**, you can do any of the following:

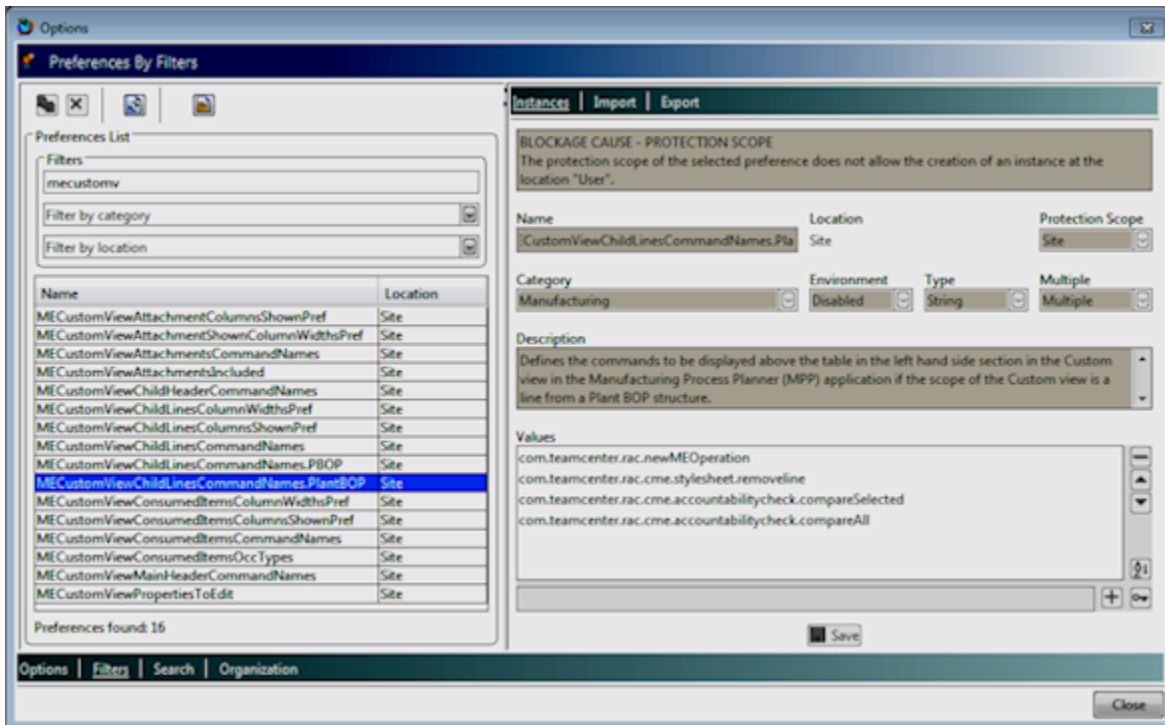
- Modify style sheets for the **Mfg0BvrProcessSummary** and **Mfg0BvrOperationSummary** datasets.
- Modify the types of tables on the right side of the view and the occurrence types that are displayed in these tables.
- Modify the commands available in the toolbars of the child lines and consumed items tables.

You can add or remove commands by setting the preferences selected in the **MECustomViewChildLinesCommandNames** and **MECustomViewConsumedItemsCommandNames** preferences.

- Set the preference **MECustomViewShowMfgFeatures** to **True** to enable the **Consumed Manufacturing Features** section of the **Custom Process View**. This section contains **Weld Point**, **Arc Weld**, and **Datum** tabs for users who want to see immediate details on manufacturing features included in the station or operation selected in the **Content** section.

## Modify the view using preferences

Use the **Preferences By Filters** dialog box to access and modify the toolbars and tabs in the **Custom Product View**.



Note:

To view all related preferences, choose **Edit**→**Options**, select **Search**, and type **mecustomv** in the top **Preference By Filters** line.

You can use the following preferences to define the appearance of the **Customer Process View**.

- Define the names of the style sheet datasets in the following preferences:

**Mfg0BvrProcess.SUMMARYRENDERING**

**Mfg0BvrOperation.SUMMARYRENDERING**

- Define the commands available for the main table on the left. These commands are shown above the table.

**MECustomViewChildLinesCommandNames**

- Define the commands available for each of the child tables on the right in this preference:

### **MECustomViewConsumedItemsCommandNames**

- Define the tables available on the right side:

### **MECustomViewConsumedItemsOccTypes**

- Define the properties shown as columns and the widths of these columns in the main table on the left:

### **MECustomViewChildLinesColumnsShownPref**

### **MECustomViewChildLinesColumnWidthsPref**

- Define the properties shown as columns and the widths of these columns in the tables on the right:

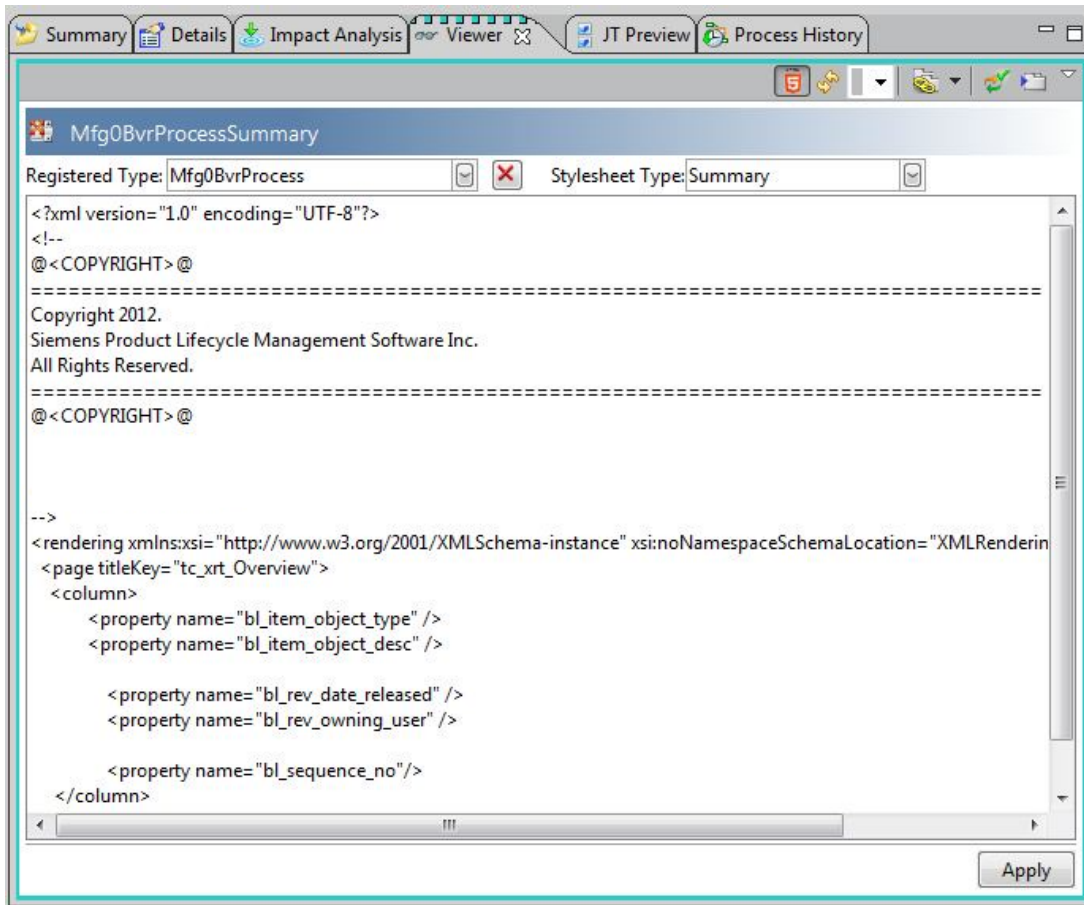
### **MECustomViewConsumedItemsColumnsShownPref**

### **MECustomViewConsumedItemsColumnWidthPref**

## Modifying style sheets

1. Open My Teamcenter.
2. In the quick search, search for the **Mfg0BvrProcessSummary** or the **Mfg0BvrOperationSummary** dataset.
3. Open the dataset in the **Viewer** view.

Teamcenter displays the XML content of the style sheet.



4. Modify the existing tags and click **Apply**.

Ensure that you have a backup copy of each of the style sheets in order to return to the default configuration if necessary.

5. Send a process from My Teamcenter to Manufacturing Process Planner. If the **Custom Process View** is already open, close it.
6. Open the **Custom Process View** view for that process line by right-clicking the process and choosing **Open With** → **Custom Process View**.

The changes you made in My Teamcenter are visible in the **Custom Process View** in the order specified in the XML.

## Specify the tables and content

The list of tables that appear on the right side of the **Custom Process View** is specified in the **MECustomViewConsumedItemsOccTypes** preference. The following table lists the values you must specify in this preference to display various tables.

The preference values have the following format:

*relation-property-type:occurrence-type*

<b>Table type</b>	<i>relation-property-type</i>	<i>occurrence-type</i>
<b>All Materials</b>	<b>Mfg0all_material</b>	<b>ME Assemble ME Assign MEConsumed ME Disassemble ME Handle MEOther METarget</b>
<b>Allocated Features</b>	<b>Mfg0allocated_features</b>	<b>ME Feature Assign ME Weld Point</b>
<b>Assigned Work Area</b>	<b>Mfg0assigned_workarea</b>	<b>MEWorkArea</b>
<b>Consumed Material</b>	<b>Mfg0consumed_material</b>	<b>MEConsumed</b>
<b>Disassembled Material</b>	<b>Mfg0disassembled_material</b>	<b>ME Disassemble</b>
<b>Handled Material</b>	<b>Mfg0handled_material</b>	<b>ME Handle</b>
<b>Other Material</b>	<b>Mfg0other_material</b>	<b>MEOther METarget</b>
<b>Used Equipment</b>	<b>Mfg0used_equipment</b>	<b>ME Compound Resource ME Plant Specific Tool MEResource METool</b>

You can add additional table types by checking the Business Modeler IDE for more occurrence types.



# 5. Configuring 4G design in manufacturing

## Configuring the comparison of design to manufacturing features

If you work with design features in 4th Generation Design (4GD), you can assign these features to the MBOM when working in Manufacturing Process Planner. The design features are stored in design container elements in 4GD. During the assignment action, Teamcenter copies the mapped property values from the design feature to the corresponding properties on the BOM line of the manufacturing feature. The mapped attribute group properties are copied to their corresponding form properties on the item revision. The forms themselves that contain mapped properties to be copied are created as part of the assign process.

To enable the comparison, you must configure the mapping of the properties between the design features and the manufacturing features.

The following constants affect the behavior of the assignment process:

- **Cpd0DFToBOMPropertyMapping** global constant

Specifies the design feature properties that are mapped to form properties on a BOM line.

- **Cpd0DFPropertiesForMBOM** global constant

Specifies the initial set of properties carried over during assignment. Your administrator can define a different set of properties for assignment based on a feature type.

- **CPD0AttachToItemRevision** business constant

Specifies whether attachments are associated to the item revision or BOM line in the context of the root level.

- **Cpd0ItemTypeForDesignFeature** business constant

Specifies what type of item is automatically created when the corresponding design feature is assigned to the MBOM.

## Mapping property and attribute groups between design elements and BOM lines

When assigning a design element into a manufacturing structure, some of the properties of the design element are copied into the new BOM line. The mapping of design element properties to BOM line properties is based on the values of the **Cpd0DEToBOMPropertyMapping** global constant in the Business Modeler IDE. This constant has the following format:

*BOM-line-property:design-element-property*

For example:

**BI\_absolute\_transform:DE\_absolute\_transform**  
**BI\_effectivity:DE\_effectivity**  
**BI\_note\_type1:DE\_custom\_attribute1**

The selection of properties that are carried forward during the assign process is specified in the **Cpd0DEPropertiesForMBOM** global constant. The value of this constant consists of a list of design element properties, for example:

**DE\_absolute\_transform**  
**DE\_effectivity**  
**DE\_custom\_attribute1**

The properties that are carried forward to the BOM line are the properties that are available for propagation after running an accountability check.

Additionally, if the design element has an associated attribute group, these attributes are copied to a designated form on the BOM line. The mapping between attribute group and form type is specified in the **Cpd0DEAttrGroupTypeToFormType** global constant. This constant has the following format:

*attribute-group:form-type*

For example:

**EngAttributes:FormEngAttributes**

Caution:

You should only map one attribute group to one form. Mapping multiple attribute groups to one form results in errors in the partial match report of the accountability check.

# 6. Administering standard text

## Standard text overview

Teamcenter enables creating work instructions using standard text as building blocks. These work instructions are associated with operations or processes in the process structure.

The standard text library is a Teamcenter structure comprised of standard text folders and standard text elements nested under the folders. Standard text is associated with standard text elements and is accessible in the Teamcenter **Attachment** pane.

The benefits of standard text include:

- Standardization and consistency of work instructions throughout the entire organization.
- Managing standard text content. Standard text enables you to write once only and reuse the same text many times.
- Easy maintenance of up-to-date work instructions when products are upgraded. Make a single change in a procedure and propagate the change through all the relevant documents.
- Improved accuracy. Work instructions are intrinsically connected with the process structure; they are a textual expression of the manufacturing process.
- Work instructions are composed by the planners or engineers who designed the product, increasing accuracy.
- Standard text is stored in the Teamcenter database, enabling you to search for text by its ID, name, or the text itself.
- Easy localization.
- Easy transfer of documents between different virtual engineering systems.

Typically, there are three user roles involved in creating work instructions:

- Administrator

Responsible for creating and maintaining standard text and work instruction templates. If required, the administrator can configure different system templates for standard text and work instructions. The administrator can change these templates or create new ones according to organizational requirements. The templates are in Microsoft Word DOCX format and are associated with a dataset object.

- Librarian

Responsible for creating standard text libraries, managing standard text folders and elements, writing standard text, and creating and managing standard text symbols. Standard text is written in a generic style so that it can be reused in varying contexts. An example of standard text is a procedure describing how to mount a wheel on a car and this can be reused in multiple projects. The librarian uses the standard text editor to write standard text elements and the standard text library interface to manage them. Additionally, the librarian may use the **Teamcenter Search** view and the **Results** window as aids.

- Planner or engineer

Uses standard text as building blocks to compose work instructions. Planners cannot edit standard text in the standard text library; however, they may copy and edit standard text and save it in a new document associated with an operation or process in the process structure. Additionally, the planner may use process structure, the textual work instructions editor, **Teamcenter Search** view, and the **Results** window as aids.

Administrators must configure Teamcenter to enable the standard text feature.

### Prerequisites to using the standard text feature

- The Teamcenter client must be installed on a Microsoft Windows platform (this limitation does not apply to the Teamcenter server).
- Requires Microsoft Word. For certified versions, see the Hardware and Software Certifications knowledge base article on Support Center.
- Requires a Teamcenter Manufacturing Documentation license.

You can use Access Manager to configure various permissions for different roles. Siemens Digital Industries Software recommends using the scenario of administrator, librarian, and planner, as described.

### Set up standard text

1. Using Teamcenter Environment Manager, install the standard text templates. By default, the same template is used by both the **Standard Text** and **Textual Work Instructions** views; however, you can set up the system to use different templates for each view.

Note:

During installation, Teamcenter imports the following work instruction templates to Teamcenter Environment Manager:

- **Default\_WIObjectTemplate.docx**

- STX\_TC9100TB\_spec\_template
- WI\_TC9100TB\_spec\_template
- TWI\_TC10100TB\_spec\_template
- TWIfor3DPDF\_TC10100TB\_spec\_template

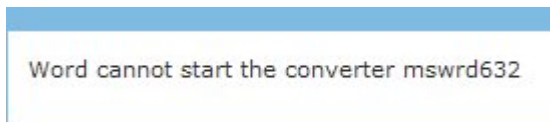
During installation, Teamcenter updates the **MEWiStandardTextTemplate** preference that defines the template to be used by the **Standard Text** view and the **MEWiWorkInstructionTemplate** preference that defines the template to be used by the **Textual Work Instructions** view. These preferences enable you to use different templates for the **Standard Text** and **Textual Work Instructions** views. Both preferences are updated to the **TWI\_TC10100TB\_spec\_template** value.

If the textual work instructions are created to be used as part of the **3DPDF Rich Textual Work Instructions** report, set the **MEWiWorkInstructionTemplate** preference to **TWIfor3DPDF\_TC10100TB\_spec\_template**.

2. If Microsoft security bulletin MS09-073 is installed on your machine, you may encounter the following error messages when loading work instructions. Microsoft has provided complete information on this issue at:

<http://support.microsoft.com/kb/973904/>.

- Converter mswrd632



Open **Regedit** and do one of the following:

- For a 32-bit machine, delete the following registry entry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Shared Tools\
Text Converters\Import\MSWord6.wpc
```

- For a 64-bit machine, delete the following registry entry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Shared Tools\
Text Converters\Import\MSWord6.wpc
```

- Windows 6.0 files

Cannot load Word for Windows 6.0 files

Open **Regedit** and do one of the following:

- For a 32-bit machine, add the following registry entry:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\
Applets\Wordpad
```

- For a 64-bit machine, add the following registry entry:

```
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\
CurrentVersion\Applets\Wordpad
```

- For both 32-bit and 64-bit machines, do the following:
  - a. Select the new entry in **Regedit**, choose **Edit→New**, and select **DWORD Value**.
  - b. Type **AllowConversion** for the name of the DWORD value, and press the Enter key.
  - c. Right-click **AllowConversion** and choose **Modify**.
  - d. In the **Value data** box, type **1** and click **OK**.
  - e. Close **Regedit**.

3. In the Business Modeler IDE:, customize **List of Values Data Collection** type.
  - a. Create an LOV business object of any basic type: **String**, **Integer**, and so on (for example, **ProjectStatusLOV: Green/Yellow/Red**).
  - b. Create a subclass of **Mes0MELOVDCDForm** (for example, **ProjectStatusDataCollection**).
  - c. For **ProjectStatusDataCollection**:
    - A. Create a property of the same type as the LOV basic type (for example, **Value**).
    - B. Connect the property to the **ProjectStatusLOV** LOV.
  - d. Deploy the changes to the database.
4. In the Business Modeler IDE, customize the **List of Values Planner Input Field** type.

- a. Create an LOV business object of any basic type: **String** (for example, **ProjectStatusLOV: Green/Yellow/Red**).
- b. Create a subclass of **Mes0MELOVPFBaseForm** (for example, **ProjectStatusPlannerInputField**).
- c. For **ProjectStatusPlannerInputField**:
  - A. Connect the property **mes0Value** to the **ProjectStatusLOV** LOV.
  - B. Connect the property **mes0DefaultValue** to the **ProjectStatusLOV** LOV.
- d. Deploy the changes to the database.

## Understanding the standard text template

By default, a single template is defined for both the standard text and work instructions editors. The template contains the following:

- Symbols section

Symbols associated with the standard text or the work instructions are displayed in a dedicated symbols section at the top of the document. When editing templates, leave this section at the top of the document (first line) or remove it if symbols should not be displayed. Do not insert the symbols section in the header or footer.

- Character styles

Use these to create the *look-and-feel* of the document. You can edit these styles, but you must retain the same style names.

- **WIDataCollectionStyle**

Applied to the placeholder string of data collection definitions. The formatting is: Font: Calibri, Size: 11 points, Color: Gray-50%, Bold.

- **WIPlannerFieldStyle**

Applied to the value of planner input fields. The formatting is: Font: Calibri, Size: 11 points, Italic.

- **WIInvalidPlannerFieldStyle**

Applied to *invalid planner input fields* identified while running the **Validate PIF** command.

The formatting is: Font: Calibri, Size: 11 points, Italic, Color: Red.

**Note:**

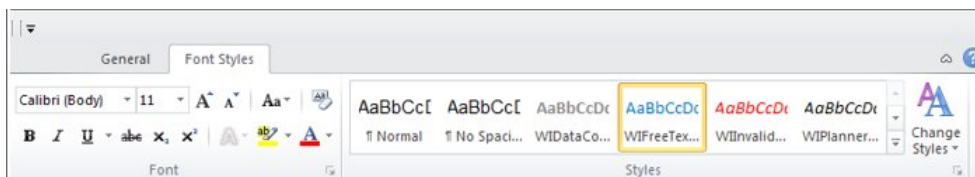
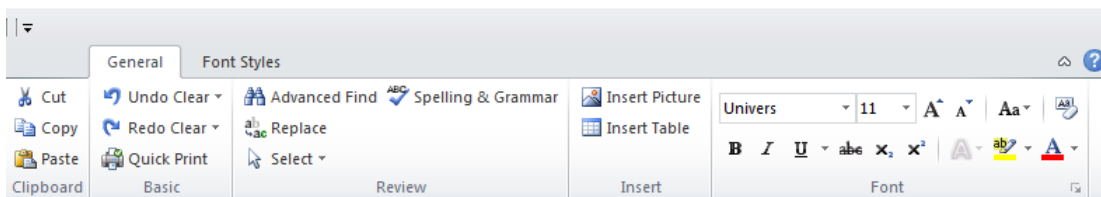
If you add new styles, ensure that **Style type** is **Character**. If you add new styles whose **Style type** is **Paragraph**, errors may occur.

- Header and footer

By default, this is empty. You can add your company logo or any other pertinent information. You can also add BOM line properties.

## Using the standard text template

When using the standard text template to create standard text, the following Microsoft Word commands are available on the **General** and **Font Style** tabs.



## Edit standard text templates

Standard text templates are stored by default in the **%Root%\install\cmtmes** directory and are also imported to Teamcenter as part of the installation. They are the responsibility of the system administrator. The administrator can format the default standard text template and create more templates by copying existing ones and editing them for different uses. Some editing functions can be performed directly on standard text templates from Teamcenter. Other editing functions require that you access the template from the **%Root%\install\cmtmes** directory and perform the following three-stage procedure on the template:

- Remove the user interface customization that limits the available Microsoft Word commands.
- Edit the template.
- Restore user interface customization.

**Note:**

When creating new templates, always use one of the document templates supplied with the Teamcenter installation as your starting point. Either copy one of these templates, rename it, and edit it; or open a template, edit it, and perform a **Save As** operation. You can clone a template in one of the following ways:

- Use a template from the installation directory:
  1. Select a template from the **%Root%\install\cmtmes** directory and copy it to an accessible location.
  2. Clear the read-only attribute of the file.
  3. Modify the file name so that you can import it later; the import fails if there is already a template with this name.
  4. Perform the following modifications:
    - a. Remove the user interface customization that limits the available Microsoft Word commands.
    - b. Edit the template.
    - c. Restore user interface customization.
  5. In My Teamcenter, choose **Tools→Import→Templates→Specification Template** to import the new template into Teamcenter.
- Clone a template within Teamcenter:
  1. In My Teamcenter, locate the template by choosing **Home→Standard Text Templates→Standard Text Spec Template**.

**Note:**

If using OOTB templates, you should only use the latest template file available – **TWI\_TC1017OOTB\_spec\_template**.

**Warning:**

Old templates for past versions of Teamcenter have known issues in newer versions.

2. In the **Summary** view, click **Save As**.

3. Modify the file name so that you can import it later, the import fails if there is already a template with this name.

4. Click **Finish**.

The specification template is created under the **Newstuff** folder. You can move it to a dedicated folder.

Note:

User interface customization ensures document standardization throughout your organization.

After importing the template, define which templates are to be used by the **Standard Text** and **Textual Work Instructions** views:

1. Edit the **MEWiStandardTextTemplate** preference so that it contains the name of the new standard text template (without the **.docx** suffix). This template becomes the default template for creating standard text.
2. Edit the **MEWiWorkInstructionTemplate** preference so that it contains the name of the new work instructions template (without the **.docx** suffix). This template becomes the default template for creating work instructions.

## Remove customization

1. Change the template file name to include the **.7z** extension. For example, rename **TWI\_TC1000TB\_spec\_template.docx** to **TWI\_TC1000TB\_spec\_template.docx.7z**.

The system recognizes the template file as an archive file.

2. Open the archive file in **7-Zip**, extract **customUI.xml**, **customUI14.xml**, and **.rels**, and store these files for later use.
3. Delete **customUI.xml** and **customUI14.xml** from the archive and close **7-Zip**.

Note:

User interface customization ensures document standardization throughout your organization. Siemens Digital Industries Software recommends retaining the default customization. However, you can edit the **customUI.xml** file to change the customization, or delete the **customUI.xml** file to remove customization.

4. Remove the **.7z** suffix from the template file name.

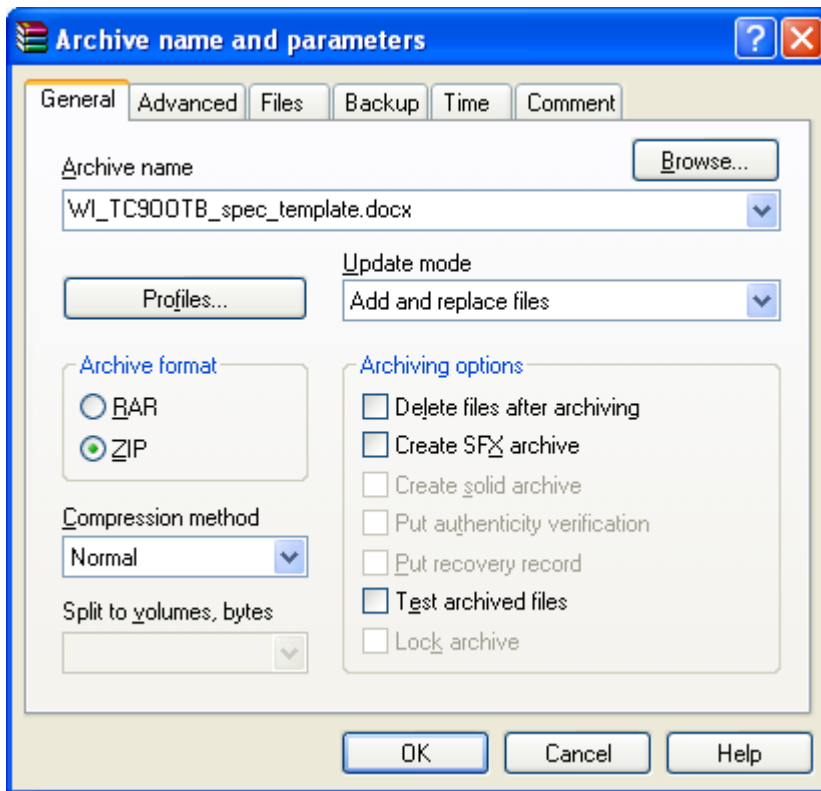
Deleting **customUI.xml** removes all customization from the template file. You can now edit it as you would any other Microsoft Word file.

## Restore customization

1. Save the modifications and close the document.
2. Change the template file name to include the **.zip** suffix. For example, rename **WI\_TC900TB\_spec\_template.docx** as **WI\_TC900TB\_spec\_template.docx.zip**.

The system recognizes the template file as an archive file.

3. Locate the **customUI.xml**, **customUI14.xml**, and **.rels** files that you saved when you removed the customization, right-click, and choose **Add to archive....**



4. Type the template ZIP file name in the **Archive name** box and click **OK**.

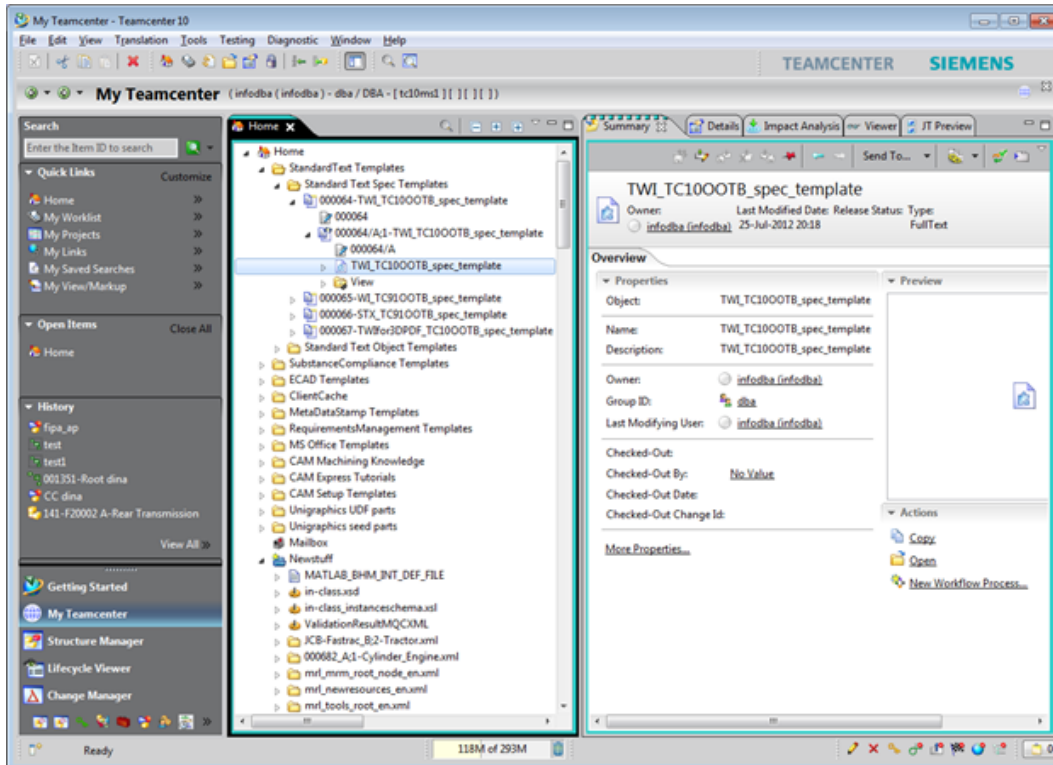
Note:

Siemens Digital Industries Software recommends restoring the **customUI.xml** and **customUI14.xml** files. This restores the default customization. However, you can edit the **customUI.xml** and **customUI14.xml** files to change the customization or not restore the **customUI.xml** and **customUI14.xml** files to remove customization.

5. Remove the **.zip** suffix from the template file name.

## Editing options

This topic details all the template editing functions. For each one, it notes whether the procedure can be performed directly from Teamcenter or whether you must access the template from the **%Root% \install\cmtmes** directory. To modify the template directly in Teamcenter, access My Teamcenter, expand under the specification template up to the **FullText** element, and double-click the **FullText** item.



## Add BOM line properties

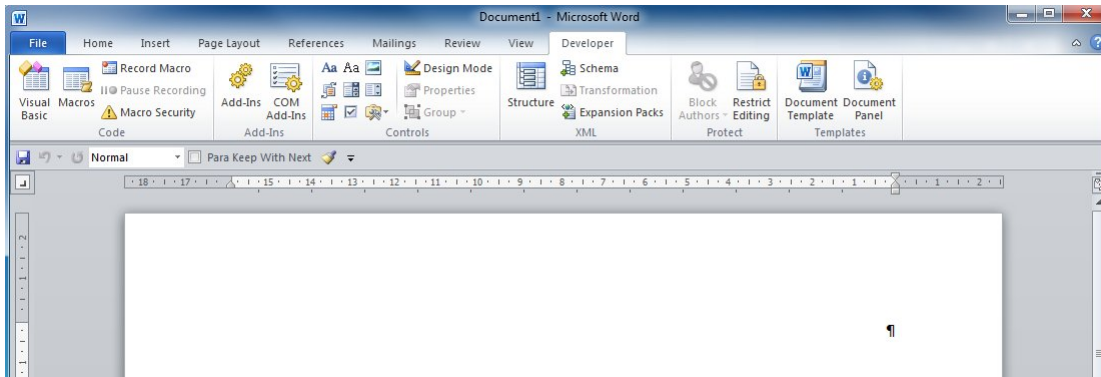
Administrators can add operation or process BOM line properties to the template header or footer. This can be useful, for example, to facilitate quick identification of different documents. This can be done directly from Teamcenter.

1. Launch Microsoft Word.

Note:

This instance of Word is independent of Teamcenter.

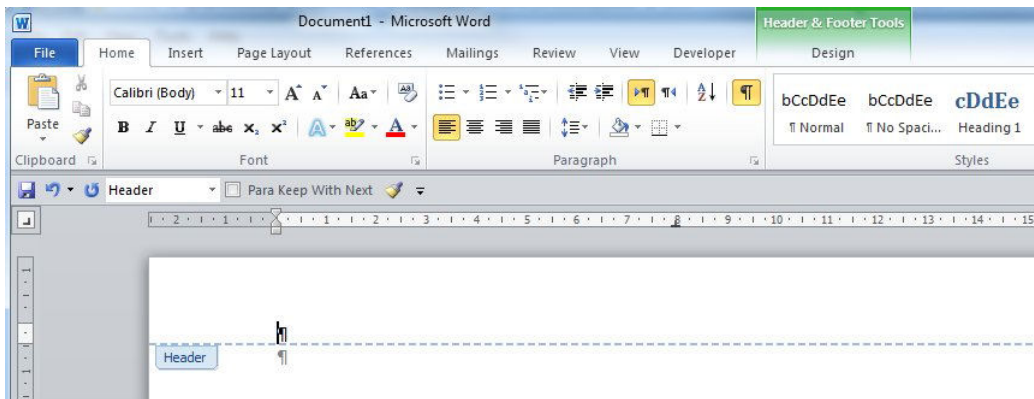
2. Open a blank Word document, and click the **Developer** tab.



If the **Developer** tab is not available, in Microsoft Word, do the following:

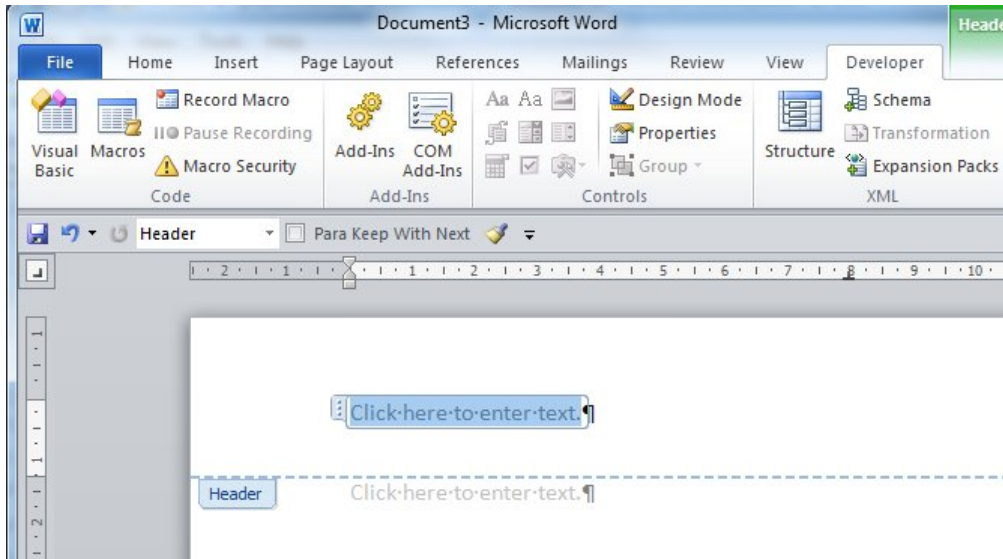
- a. Choose **File**→**Options**→**Customize Ribbon**.
  - b. From the **Customize Ribbon** list on the right, select **Main Tabs**, and select **Developer**.
  - c. Click **OK** to accept the change and exit the **Word Options** dialog box.
3. Double-click the top of the document.

The document header becomes active.




4. Click the **Developer** tab.
5. Click the **Aa Plain Text Content** control button.

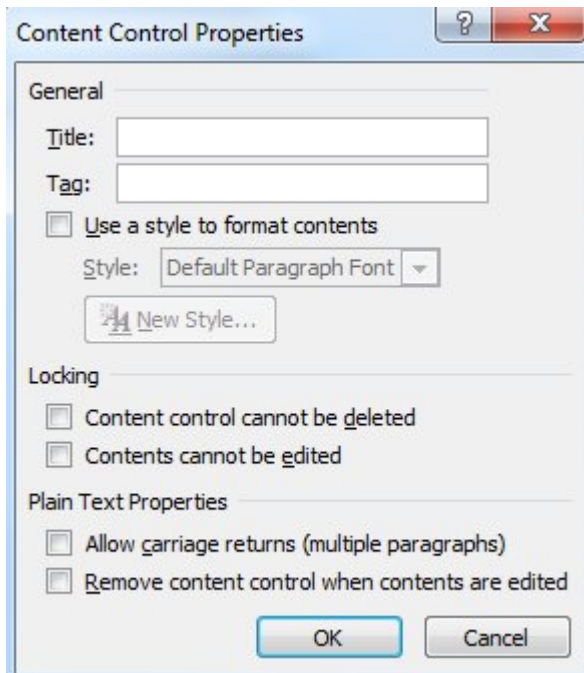
Word places an empty plain text content control in the header.



If the property you want to add requires more than a single paragraph, use a **Aa Rich Text Content** control. For example, the description property may require multiple paragraphs.

6. Select the new text control and click  **Properties**.

The **Content Control Properties** dialog box appears.

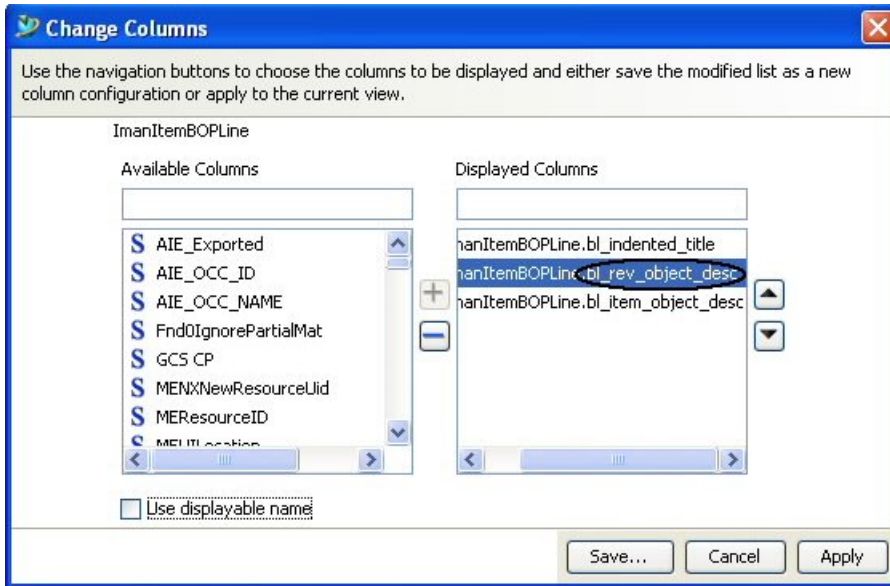


7. In the **Title** box, type a name for the property title.

The **Title** text is used for the content control's name.



8. In the **Tag** box, type the name of the BOM line property you want to insert in the template header. To retrieve the property name, proceed as follows:
  - a. Right-click the column area in Manufacturing Process Planner and choose **Insert Column**. The **Change Columns** dialog box appears.



- b. Copy the desired BOM line property name to the clipboard.
  - c. In the **Change Columns** dialog box, click **Cancel**.
9. In the **Tag** box, paste the BOM line property name.
10. In the **Tag** box, preface the BOM line property name with one of the following:
  - **LNP\_** for a line property, for example, **LNP\_bl\_rev\_object\_name** for a line object name or **LNP\_bl\_rev\_item\_revision\_id** for a line revision identification number
  - **LPP\_** for a line parent property, for example, **LPP\_bl\_rev\_object\_name** for a parent object name
11. In the **Content Control Properties** dialog box, click **OK**.

Note:  
Ignore the **Locking** and **Plain Text Properties** sections.

12. Open the target template and double-click the top to activate the header area.
13. Type the name of the property you want to insert in the template header.
14. From the Word document, copy the content control you created and paste it after the property name you typed.
15. Add as many BOM line properties as you want to the template header.

**Note:**

You can add a table to the document header and insert the BOM line properties into the table. This enables you to add multiple properties in an orderly way.

16. Save and close the template.
17. Discard the original Word document in which you created the content control.

## Add or modify template styles

This procedure can be performed directly from Teamcenter.

1. Add or modify styles as needed.
2. Close and save the template file.

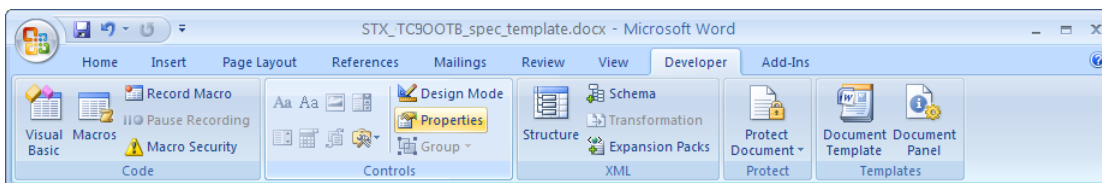
## Remove symbols section

This procedure requires you to access the template from the `%Root%\install\cmtmes` directory.

Symbols associated with the standard text or the work instructions are displayed in a dedicated symbols section at the top of the document. When editing templates, leave this section at the top of the document (first line) or remove it if symbols should not be displayed. Do not insert the symbols section in the header or footer.

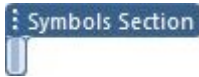
By default, the symbol section exists in the template and is protected. To prevent symbols from appearing, delete the symbol section as follows:

1. In Word, choose **Options**→**Popular** and select **Show Developer tab in the Ribbon**.



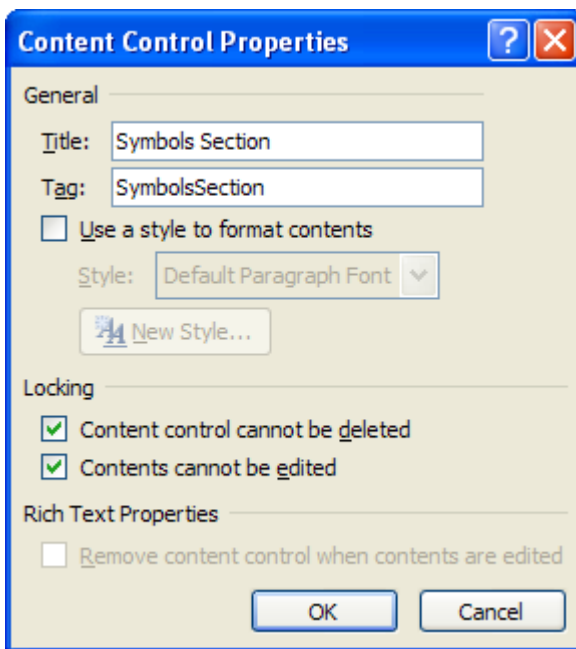
- Click the **Developer** tab.
- The symbol section is not visible, therefore, to place your cursor inside the symbol section, click anywhere in the body of the document and then press the Home key and the right arrow (→) key.

The symbol section becomes active.



- From the **Controls** section, click **Properties**.

The **Content Control Properties** dialog box appears.



- Clear the **Content control cannot be deleted** and **Contents cannot be edited** check boxes.
- Click **OK**.
- Select the symbol section and press the Delete key.

Siemens Digital Industries Software recommends deleting any empty lines in your template file.

## Add symbol section

To add a symbol section if you deleted it:

- From the **Developer** tab, choose **Controls**→**RichText** **Aa**.

Word adds a new control.

Note:

Word automatically places the new control at the top of the document, below the header.

2. Type a single space in the new control.
3. Right-click the control and choose **Properties**.

The **Content Control Properties** dialog box appears.

4. Edit the **Title** box, adding, for example, **Symbols Section**.
5. In the **Tag** box, type **SymbolsSection**.

Note:

You must type **SymbolsSection** exactly or the symbol section does not work.

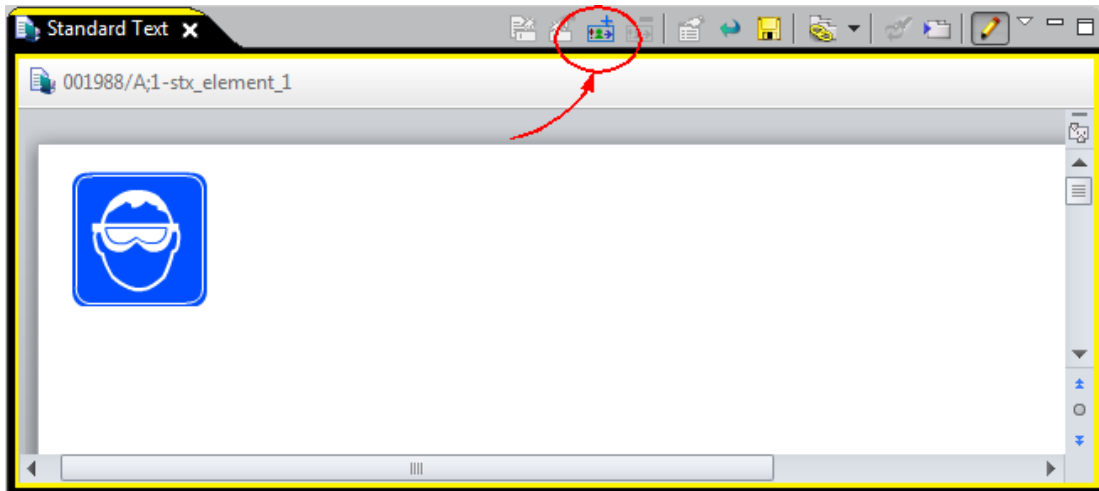
6. Select the **Content control cannot be deleted** and **Contents cannot be edited** check boxes.
7. Click **OK**.

Word creates the symbol section.

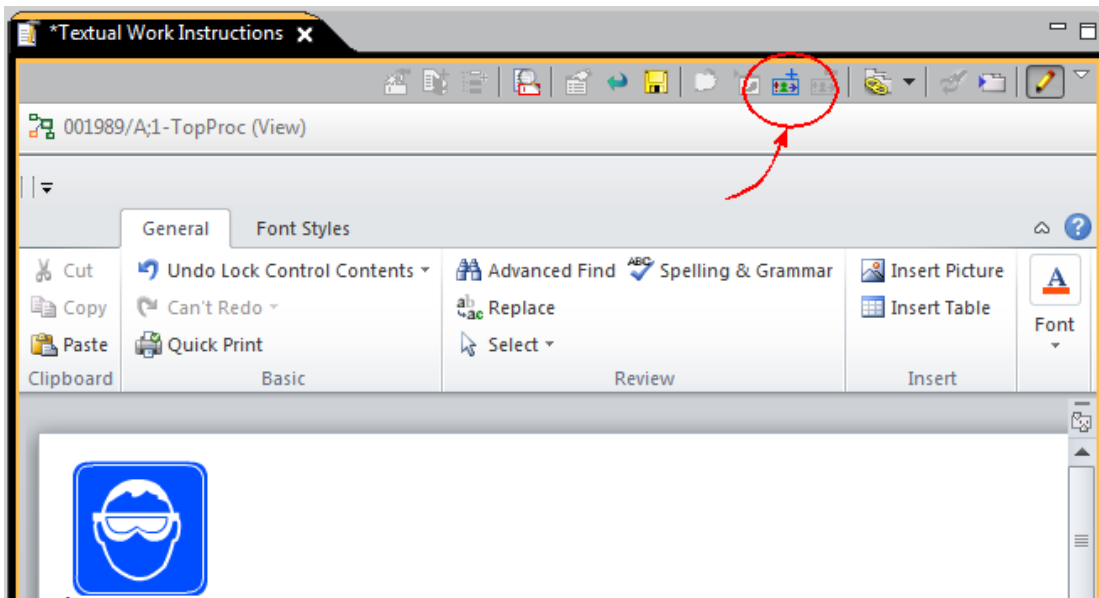
## Specifying the location of the Add Symbol button

When preparing to use the textual work instructions feature, you must determine in advance whether to enable adding symbols in standard text elements (in the **Standard Text** view by the librarian) or instead to textual work instructions (in the **Textual Work Instructions** view by the planner). This decision is applicable to all users of the feature. You cannot add symbols in both locations.

- Add a symbol to the standard text element in the **Standard Text** view. You subsequently add the standard text element to the work instruction.



- Add a symbol directly to the work instruction in the **Textual Work Instructions** view.



The position of this button is defined in the **MEWiEnableAddSymbolCommand** preference. The valid values for this preference are **STXView** or **TWIVView**.

## Importing standard text library structures from a legacy system

If you have a standard text library structure in a system other than Teamcenter, you can re-create this structure in Teamcenter using the **tcexcel\_import** utility. You define the full library structure containing folders and standard text elements in a Microsoft Excel spreadsheet and save it to a tab-delimited file. This file is then used as input for the **tcexcel\_import** utility that creates the structure and attaches documents to the standard text elements. In the spreadsheet, you specify the paths to the documents that are to be attached to the standard text elements. The textual content for each of the standard text elements can come from:

- TXT files containing only rich text.
- DOCX files containing tables, images, formatting, header and footer content.

After import, a standard text librarian can modify the library structure and an author can modify the documents and save the modifications.

The following is an example of an input file with a simple library structure. The first four columns represent the structure of the standard text library. The last column, the **StxFilePath** column, contains the path to the document that will be attached to the standard text element.

Header1	Item	Item	Item	StxFilePath
Header2	Mes0MESTXLibrary	Mes0MESTXFolder	Mes0MESTXElement	
Header3	Mes0MESTXLibrary	Mes0MESTXFolder	Mes0MESTXElement	
Title	Root	Sub1	Sub2	Sub3
	Library			
		Safety instructions		
			STX1	D:\templgoggles.docx
			STX2	D:\templboots.docx
			STX3	D:\templhelmet.docx
		Folder2		
			STX4	D:\templmagnet.docx
			STX5	D:\templtable_example.docx
			STX6	D:\templrich_text_example.docx
			STX7	D:\templhelmet.docx
			STX8	D:\templhelmet.docx

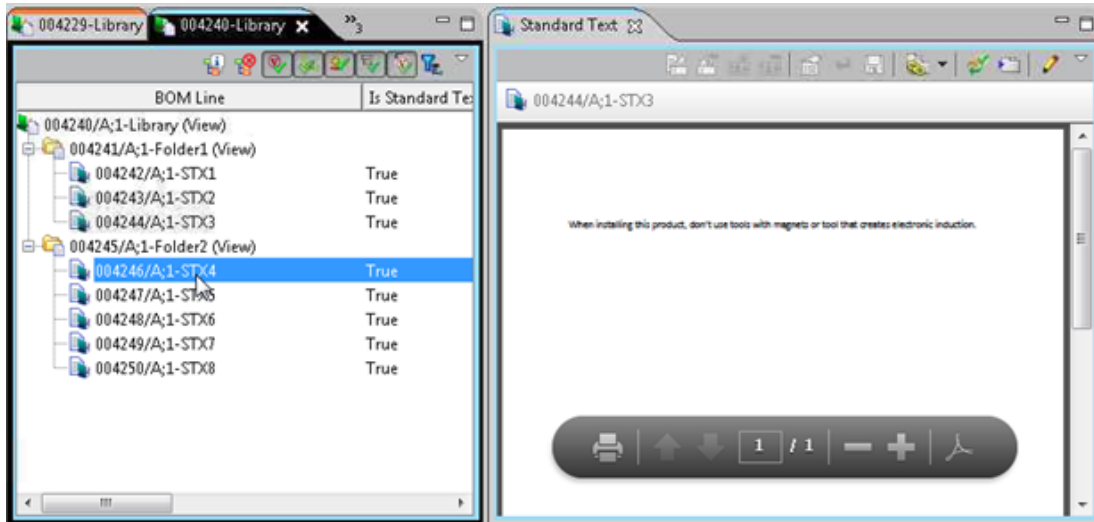
**Note:**

This example imports only DOCX files, but both DOCX and TXT files are supported.

You can import this structure into Teamcenter by saving the Microsoft Excel spreadsheet as a text file and import that file into Teamcenter using the following tcexcel\_import utility call:

```
tcexcel_import -u=username -p=password -g=dba -i=path-to-txt-file
```

The resulting structure appears as follows in Teamcenter.



Note that the standard text elements have documents attached. You can modify these documents, save them, and continue to work with them in the customary fashion.

**Note:**

You can only use the **tcexcel\_import** utility to create new library structures. You cannot use it to update structures.

## Updating standard text instances using the update\_stx\_elements utility

When an individual standard text library object changes, you can use the command line **update\_stx\_elements** utility to find and update textual work instruction preview datasets attached to process or operation revisions. Any of these preview datasets impacted by changes to a single, selected standard text object are updated when you run the utility.

### Requirements

After a standard text library object has been revised, updated and approved, it is a valid candidate for the textual work instructions (TWI) bulk update utility.

- The **update\_stx\_elements** utility requires a Microsoft Office version compatible with the Teamcenter version you have installed.
- You must run the utility from a Teamcenter server.
- To provide controlled and predictable results, it is strongly recommended that you update only a single standard library object prior to executing the bulk update. Because an individual TWI document may contain references to multiple standard library elements, all standard library elements referenced within a TWI document are regenerated within an updated TWI document.

- You should run the utility at a time when users are not authoring TWI documents, and you should prenotify users of expected changes. The TWI documents may be in a working or released state.

### Caveats

The following are not part of the **update\_stx\_elements** utility process and must be addressed separately:

- Manufacturing Execution System (MES) integration.
- Change management.

Teamcenter recommends that you set a single manufacturing change notice (MCN) on a single standard text library object. An in-depth understanding of the nature of changes made by the **update\_stx\_elements** utility is essential to successful results. The TWI preview dataset used by TWI and Electronic Work Instructions (EWI) supports one of several formats: PDF (default), MHT, docx, and rtf, based on the Teamcenter preference **MEWiPreviewFormat**. Changes made to the TWI preview dataset attached to an **Process/OP** item revision are absolute and datasets have no revision history.

- The utility does not support standard symbol library objects.
- If a standard text library object change requires additional changes to the BOP, such as changes to the number of operations or changes to consumed parts, this is not a valid use of the bulk update utility.

# 7. Administering structure cloning

## Creating structures from templates overview

You can create a product structure from a product template, a plant structure from a plant template, and process structure from a process template. You may choose to define a generic product structure that does not represent an actual manufactured product because it contains generic placeholders in place of variable parts. These placeholders are identified by absolute occurrences that are populated when an actual product structure is created from the template.

If you create a generic manufacturing process structure from the generic product structure, the occurrences in the generic product structure are consumed in the generic process structure.

In a template process, although no real parts are consumed in the operations that comprise the generic process, each consumed item has an identity. This identity of the item in the process structure is the same as it is in the product structure. Consequently, if you use this process structure as a template for the creation of a new process structure for manufacturing a product that maps exactly to the generic product structure, the product and process share the same absolute occurrences. The new process that you create from the template consumes the parts from the real product structure with the same absolute occurrences.

Similarly, if you define a mapping between the plant structure referenced by the template and the actual plant structure, generic work areas are substituted with specific absolute occurrences.

## Reusing process templates

In many manufacturing environments, process planners design standard processes that are reused several times for different products. For example, the basic structure of the manufacturing process for a car is similar for a basic model or a premium model.

You can use any process as a template for a new structure. However, if you have many of these processes, it can be more efficient to save such manufacturing processes as templates in a library, and then use the templates as the basis of new manufacturing processes for similar products. The template may include configuration rules that map the product data to process operations, and may include references to standard resources. You can classify the templates in the Classification application for easy retrieval later.

When you create a new process operation, you can browse the available templates and select one to create the process operation. After it is created, you can attach the geometry of the part to be processed.

Similarly, you can create assembly setup templates containing assembly information for a typical assembly setup. When you initialize a setup from a template, Teamcenter loads the related method, equipment, features, and manufacturing operations.

**Note:**

By default, classify immediately is disabled for a root item in Manufacturing Process Planner.

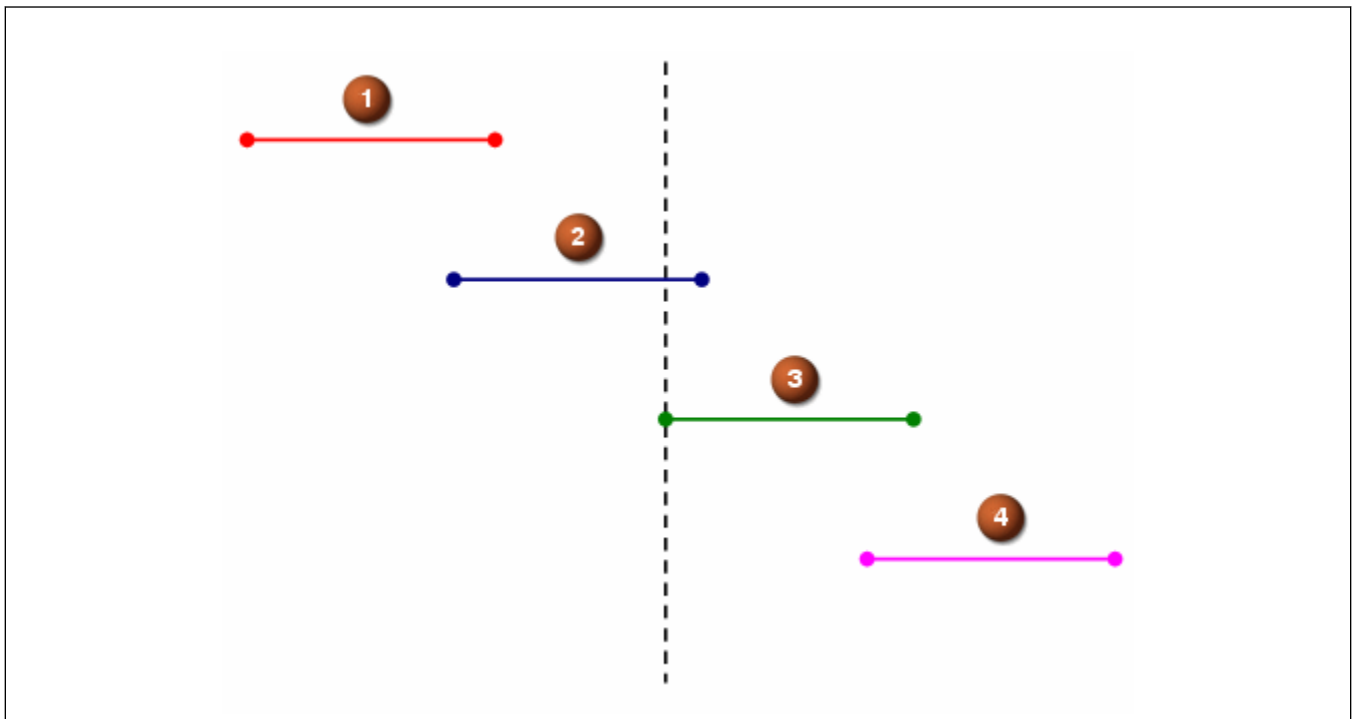
## Carrying over future incremental change when cloning

When cloning, you have the option to carry over future effective incremental changes. If you are using date effectivity, *future* is anything falling after the date set in cloning dialog box. If you are using unit effectivity, future is anything after the unit number set in the cloning dialog box. Incremental changes that are currently effective are merged with the currently set incremental change. Those that are effective in the future are carried forward to the new cloned structure or re-created under new incremental change revisions. Any incremental changes whose effectivities are in the past are dropped from the cloned incremental change.

**Note:**

- Specify the release status type that is assigned to any newly created incremental change revisions during cloning in the **Clone\_Pending\_Release\_Status** preference.
- Specify the release status type that should be considered as secure in the **Clone\_Secured\_Release\_Status** preference.

In the following figure, the vertical line represents the point of cloning.



Cloning with incremental changes

1	IC1	Incremental change 1 whose effectivity lies in the past.
2	IC2	Incremental change 2 that is secure and configured.
3	IC3	Incremental change 3 is the current IC (set at the bottom of the pane) that is configured and pending. This is the target IC.
4	IC4	Incremental change 4 is pending and unconfigured.

After cloning, Teamcenter:

- Disregards IC1.
- Clones IC2 to a new IC5 and the change elements in IC2 (that is, **add** or **remove**) that are associated with the cloned object are cloned and added to IC5. IC5 then has a **pending** status.
- IC4 is left alone and the new change elements are added to it. This is not cloned because the status is pending and the **out** effectivity does not match the currently set IC revision (IC3).

## Configuring how assemblies in context are cloned and replaced

The **Clone Assembly in Context** and **Replace Assembly with Any Item Type** commands are available in Teamcenter MBOM **Product** or **Workarea** views when you choose **Edit→In Context**. How you configure these commands affects how structures are cloned in context within the same MBOM and how items and item types are replaced within in-context assemblies.

### Clone Assembly in Context

By default, the **Clone Assembly in Context** command clones an MBOM subassembly and pastes the clone within the same MBOM. This command:

- Retains all in-context edits in the cloned subassembly.
- Lets the MBOM author copy, reference, or exclude in-context attachments from the structure being cloned, and optionally, split occurrence effectivity dates or ranges between the assembly being cloned and the cloned assembly.

Direct and in-context attachments of the new (cloned) line, cloning of children below the new line, and attachments on referenced children below the line added in the context of lines above the new line, are controlled by cloning rules. Cloning rule names are listed in the preference **MECopyAssemblyInContextTemplates**.

### Paste Assembly in Context

This command pastes a copied assembly with in-context edits elsewhere in the same MBOM, creating a new absolute occurrence of the assembly while retaining in-context edits and attachments.

Copied and referenced children are again controlled by cloning rules. Cloning rule names are listed in the preference **MEDuplicateAssemblyInContextTemplates**.

## Replace Assembly with Any Item Type

The **Replace Assembly with Any Item Type** command replaces an MBOM subassembly item with a new item of a different item type while retaining the original subassembly's in-context edits. The procedure is similar to the **File→New item** command, but retains in-context edits and also copies, references, or excludes attachments from its original structure.

The **Replace Assembly with Any Item Type** command always preserves in-context attachments of the new line. Direct attachments of the replaced line, cloning of children below the replaced line, and attachments on children below the replaced line—in the context of lines above the replaced line—are controlled by cloning rules. Cloning rule names are listed in the **MEReplaceAssemblyInContextTemplates** preference. Properties on the new line can be mapped from properties on the replaced line using the **MEReplaceAssemblyInContextAlignProperties** preference.

## Defining cloning rules

To clone structures from templates or other structures, you must define cloning rules that allow Manufacturing Process Management to determine how to map objects between the structures. These cloning rules are defined by preferences and the defined rules are available to users when creating an actual structure. The location of the cloning rule preferences for a specific user is determined by a preference.

You can define three cloning rule preferences, one each for process templates, product templates, and plant templates. You can associate multiple action rules with each preference.

When Manufacturing Process Management encounters an object that is mapped between the template and the actual structure, it determines if a substitution is necessary and takes one of four actions, according to the mapping of the object type:

- **Clone**

Copies the object referenced by the template to a new object in the structure.

- **Reference**

The structure references the same object as the template.

- **Ignore**

The new structure does not reference the object.

- **Map**

The system maps the template to a replacement structure. For example, this allows you to create a process structure from an existing process or from a process template. Optionally, this rule may be allowed by a second (default) action: **Reference** or **Ignore**.

You can also define a cloning rule to ignore the cloning of an MBOM recipe.

Example:

```
*:*:Relation.Mfg0MEAssignmentRecipeRel:Ignore
```

Note:

You cannot apply a **Reference** action to an MBOM recipe. A **Reference** action will log an error and ignore the cloning.

Manufacturing Process Management only takes the specified action if an absolute occurrence or its ID is the same in the template and the structure. If the same absolute occurrence appears more than once, only the first matched instance is mapped.

You can specify a second action after the **Map** action. The second action is the default action and is taken if the mapping fails. For example:

```
*:*:OccType.MEConsumed:Map.Ignore
*:*:OccType.MEWorkArea:Map.Reference
```

In this example, the **MEConsumed** occurrence type is mapped to the occurrence in the new structure. Manufacturing Process Management uses the absolute occurrence referenced from the template and finds the corresponding occurrence in the new product. It links the consumed items for that operation to the occurrence in the new product. When there is no mapping for the **Consumed** item, Manufacturing Process Management ignores it. When there is no mapping for the **WorkArea** item, Manufacturing Process Management creates a reference to the item from the cloned element.

Note:

By default, the **MEConsumed** occurrence type maps consumed items to the corresponding occurrences. The site administrator can change the **controllingOccsForProcessConfiguration** preference to map another occurrence type. Alternatively, this preference may be left unassigned, in which case the configuration of consumed items does not affect the configuration of operations.

Each site can define a set of cloning rules from which users may select. In addition, individual users may change the cloning action rules by editing their preference file.

Note:

These rules determine if objects are cloned, referenced or ignored. You cannot change this behavior with the Business Modeler IDE application.

The general format of the cloning rule preferences is as follows:

```

Process.Template=
Action_Rule_Name1
Action_Rule_Name2
.
.
Product.Template=
Action_Rule_Name3
Action_Rule_Name4
.
.
Plant.Template=
Action_Rule_Name5
Action_Rule_Name6

```

For each set of action rules, you must define a new variable. The following example shows a typical **Action\_Rule\_Name1** variable for the **Process.Template** variable:

```

Process.Template.Action_Rule_Name1=
*:Attribute.items_tag:Clone
*:Attribute.structure_revisions:Clone
*:Attribute.bom_view:Clone
*:Attribute.parent_item:Clone

```

The following example shows two cloning rules (**Mapping\_Consumes** and **Ignore\_Consumes**) for process templates:

```

Process.Template=
Mapping_Consumes
Ignoring_Consumes
Process.Template.Mapping_Consumes=
*:Attribute.items_tag:Clone
*:Attribute.structure_revisions:Clone
*:Attribute.bom_view:Clone
*:Attribute.parent_item:Clone
*:Attribute.variant_expression_block:Clone
*:OccType.Null:Clone
*:OccType.MEConsumed:Map.Ignore
*:OccType.MEWorkpiece:Ignore
*:OccType.*:Reference
*:Relation.TC_Rendering:Clone
*:Relation.TC_specification:Clone
*:class.MEActivity:Relation.TC_MEFolder:Clone
class.MEActivity*:Attribute.contents:Clone
class.Folder*:Attribute.contents:Reference
Process.Template.Ignoring_Consumes=
*:Attribute.items_tag:Clone

```

```

*:*:Attribute.structure_revisions:Clone
*:*:Attribute.bom_view:Clone
*:*:Attribute.parent_item:Clone
*:*:Attribute.variant_expression_block:Clone
*:*:OccType.Null:Clone
*:*:OccType.MEConsumed:Ignore
*:*:OccType.MEWorkpiece:Ignore
*:*:OccType.*:Reference
*:*:Relation.TC_Rendering:Clone
*:*:Relation.TC_specification:Clone
*:class.MEActivity:Relation.TC_MEFolder:Clone
class.MEActivity:*:Attribute.contents:Clone
class.Folder:*:Attribute.contents:Reference

```

The syntax for adding an entry to a cloning rule is as follows:

*parent-object:object:relation:action*

For example:

**\*:type.MEPSDRevision:Occtype.\*:Reference**

In this example, the system searches through any parent (\*), and when it finds a child of type **MEPSDRevision (type.MEPSDRevision)**, with any occurrence relation (**Occtype.\***), it creates a new reference to that **MEPSD** element (**Reference**).



# 8. Administering manufacturing publish

## Configuring the publish feature

Manufacturing Process Management allows you to create publishing pages that can be viewed or printed by nontechnical users with limited computer skills, such as assembly personnel. *Publishing pages* is a general term for any document used by manufacturing personnel, including work instructions, maintenance instructions, assembly instructions or any other documentation needed for your processes.

To create or view manufacturing documentation, install Microsoft Visio on each workstation where you author publishing pages. Both Visio Standard and Visio Professional are supported. You can install Visio before or after the Teamcenter installation. For certified versions, see the Hardware and Software Certifications knowledge base article on Support Center.

Publishing pages can be published to PDF or HTML so that they can be viewed independently of Teamcenter or Visio.

You create a template for each of the publishing page types needed in your organization from assets (Visio shapes) that are bound to Teamcenter data. The process planners use these templates to create publishing pages that may include text, tables, graphics, 2D or 3D snapshot (product views), and link assets. When all the publishing pages are complete, the process planner can collect them into a *technical portfolio* that may contain all the activity, operation, and process documentation for a particular assembly line. Users manipulate publishing pages and portfolios in the Manufacturing Process Planner, Part Planner, Resource Manager, Plant Designer, and Multi-Structure Manager applications. They may also add 3D annotations where appropriate. For detailed procedures, see the procedures for viewing, creating, and editing manufacturing documentation in the guides for those applications.

In addition to creating manufacturing documentation in Microsoft Visio, you can create links that point to manufacturing documentation in Microsoft Word and text files.

You can optionally publish manufacturing documentation in PDF format. If you publish a PDF of a portfolio, Teamcenter attaches the PDF dataset to the portfolio. If you create a PDF of a publishing page, Teamcenter attaches the PDF dataset to the page dataset. You must install PDF generation software to use this feature, for example, an Adobe product.

## Understanding roles

Publishing pages have content that reflects a specific purpose or role. By default, Teamcenter supports two roles, **Work Instructions** and **Product Manual**. You can customize Teamcenter to support any desired roles, for example, **Repair Manual**. The same dataset types are used for all roles. When publishing pages are attached to BOM lines or item revisions, the role is determined by the type of relation that attaches the dataset to the parent item revision. Multiple datasets can be attached to an item revision and the datasets can fulfill different roles according to the relation type. When a publishing page is attached to an activity, its role is determined by a preference setting. All publishing pages attached to an activity fulfill the same role. When a portfolio is generated from a structure, the pages

that are added to it are determined by the role that the user specifies for that portfolio. Internally, each role is typically associated with:

- A relation type used to attach a publishing page dataset to a revision.
- A transfer mode for generating a portfolio for this role.
- One or more alias files for populating assets.

A set of preferences associates the role with the relation type, the transfer mode, and the default alias files, as well as the following:

- The list of roles in the portfolio generation dialog box
- A set of useful alias files
- The user's favorite list of templates

You can create a custom roles to reflect your company's needs. There are two steps necessary to do this:

- *Add a new relation object* for the new role using the Business Modeler IDE.
- *Add a new rich client plug-in* that extends the `com.teamcenter.rac.cme.tcpublishwrapper.publishingPageViewRoles` extension point.

## Example: creating a customized publish role

The following is an example `plugin.xml` file for adding a new Teamcenter publish role named **Fnd0MyPersonalRole**. The Teamcenter publish wrapper extension point requires both a role and a view. The plug-in must define both. The view must be added to a Teamcenter menu. This example shows adding the view as an entry to the **Open With** menu command where the current **Work Instruction** menu command exists.

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
  <extension
    point="com.teamcenter.rac.vns.viewDef">
    <view alwaysUseSetRootInputObject="true"
      id="com.myCompany.viewId"
      selectionInPrimaryEffectsInvisbleSecondary="false"
      selectionInSecondaryEffectsPrimary="false">
      <viewType type="secondary"></viewType>
    </view>
  </extension>
  <extension
    point="org.eclipse.ui.views">
    <view allowMultiple="true"
      category="com.teamcenter.rac.aifrcp.manufacturingCategory"
      class="com.teamcenter.rac.cme.tcpublishwrapper.PublishingPageView">

```

```

        icon="icons/workinstructions_view_16.png"
        id="com.myCompany.viewId"
        name="%viewId.VIEW">
    </view>
</extension>
<extension
    point="com.teamcenter.rac.cme.tcpublishwrapper.publishingPageViewRoles">
    <viewRoleType
        tcPublishRole="Fnd0MyPersonalRole"
        viewId="com.myCompany.viewId">
    </viewRoleType>
</extension>
<extension
    point="com.teamcenter.rac.aifrcp.openWithViewMenuDef">
    <menuItem groupId="secondary"
        icon="icons/workinstructions_view_16.png"
        label="%viewId.VIEW"
        viewId="com.myCompany.viewId">
        <visibleWhen>
            <and>
                <count value="1"/>
                <iterate ifEmpty="false" operator="and">
                    <with variable="rac_command_suppression">
                        <not>
                            <iterate operator="or">
                                <equals value="com.teamcenter.rac.openWith"/>
                            </iterate>
                        </not>
                    </with>
                    <adapt type="com.teamcenter.rac.kernel.TCComponent">
                        <test
                            property="com.teamcenter.rac.kernel.TCComponent.typeClass"
                                value="ImanItemBOPLine,BOMLine,CfgAttachmentLine,MEActivity"/>
                    </adapt>
                </iterate>
            </and>
        </visibleWhen>
    </menuItem>
</extension>
</plugin>

```

## Set preferences and environment variables

- **IMAN\_MEWorkInstruction\_relation\_primary**

Lists the object classes that can be pasted into a work instruction using this relation.

- **AppInterface\_TCPublishingTechnicalPortfolio\_default\_relation**

Defines the default relation used when a **TCPublishingTechnicalPortfolio** dataset is pasted under an intermediate data capture (IDC) while creating technical portfolios.

- **AppInterface\_TCPublishingTechnicalPortfolio\_shown\_relations**

Defines the relations available when pasting a **TCPublishingTechnicalPortfolio** dataset under an intermediate data capture (IDC) while creating technical portfolios.

- **IMAN\_PublishingPageAssetContext\_relation\_primary**

Defines which Teamcenter objects can be used with a **IMAN\_PublishingPageAssetContext** relation.

- **IMAN\_TCPublishedPortfolio\_relation\_primary**

Defines which Teamcenter objects are used as primary objects with a **IMAN\_TCPublishedPortfolio** relation.

- **ItemRevision\_TCPublishingPage\_default\_relation**

Defines the default relation created when a publishing page is pasted within an item revision.

- *<TC Publishing Technical Portfolio type>***\_DefaultChildProperties**

Determines which properties are children of the indicated technical portfolio type. Each type requires its own preference.

- *<TC Publishing Page type>***\_DefaultChildProperties**

Determines which properties are children of the indicated publishing page type. Each type requires its own preference.

- **IMAN\_PortfolioConfiguredPage\_relation\_primary**

Defines which Teamcenter objects are used as primary objects with a **IMAN\_PortfolioConfiguredPage** relation.

- **IMAN\_PortfolioPage\_relation\_primary**

Defines which Teamcenter objects are used as primary objects with a **IMAN\_PortfolioPage** relation.

- **TechnicalPortfolioRoles**

Use this preference to define the roles of technical portfolios.

- **TechnicalPortfolioTransferMode\_IMAN\_MEWorkInstruction**

Defines the transfer mode of the specified technical portfolio. **TransferMode** objects control the related data exported into a PLM XML document.

- **TCPublishing\_Default\_Alias\_File**

The name of the installed alias file, as described later. Defines the default alias file loaded when a new publishing page is viewed.

- **TCPublishing\_Aliases.IMAN\_MEWorkInstruction**

The names of the available alias files for work instruction pages. If this preference is not set, the user can choose from any alias file in the database.

- **TCPublishing\_Activities\_Role**

Defines the role of all publishing page datasets that are attached to activities in Manufacturing Process Planner, for example, **IMAN\_MEWorkInstructions** (the default setting). The role determines the tab name shown in Manufacturing Process Planner and also how portfolios are generated. If you change the relation name defined in this preference, you must define a new closure rule and transfer mode to generate portfolios. You must also add the new role to the **TechnicalPortfolioRoles** preference.

- **TCPublishing\_Aliases.Activities**

Contains the names of the alias file datasets that appear when the user wants to load alias files for activity pages.

You should also ensure that the **Dataset\_DefaultChildProperties** preference for each user contains the following settings:

```
TC_PublishingPageAssetContext
TC_PortfolioPage
TC_PortfolioConfiguredPage
```

These settings allow users to cut and paste publishing pages and portfolios.

The following environment variables are set during installation and you should not normally change their values.

- **PUBLISH\_PDF\_TIMEOUT**

This environment variable sets a time-out period for stalled PDF conversions. Enter a number of milliseconds after which the conversion is terminated, enter **120000** for 2 minutes. If you require an infinite time-out, set this environment variable to **INFINITE**.

- **PUBLISH\_PDF\_MAX\_JOBS**

Defines the maximum number of outstanding PDF conversion jobs that are allowed to run simultaneously. This environment variable is effective only when you publish a portfolio to PDF and select multiple PDFs.

- **PUBLISH\_PDF\_INTERVAL**

This environment variable defines the number of milliseconds Teamcenter waits before issuing each subsequent PDF conversion job. Set a value that ensures there are no conflicts between PDF jobs starting simultaneously. Typically, you do not need to change the default value of **1000** (one second). This environment variable is effective only when you publish a portfolio to PDF and select multiple PDFs.

## Installing the transfer mode

When you install Teamcenter, the necessary default PLM XML transfer modes are provided. The transfer mode for publishing is **defaultTransferModes.xml**, which is located in the **TC\_DATA** data directory of the base installation (**TC\_BASE**).

If necessary, you can reinstall this transfer mode by running the following command from a Teamcenter shell:

```
%TC_BIN%\plmxml_import -u=Tc-admin-user -p=password -g=group -xml_file=
%TC_BIN%\defaultTransferModes.xml
```

Ensure you enter the complete path for the **xml\_file** argument, for example:

```
file=C:\Tc2007\tcdata\mfg\defaultTransferModes.xml
```

You would only reinstall the file if you make system-specific changes, for example, you customize the mode for your site environment.

## Defining alias files

To view Teamcenter-managed data in a publishing page, you have to create an asset, bind it, and populate it. When you bind the asset, the **Bind** dialog box contains a list of transfer modes and a number of choices, called aliases, from which you can select. The choice you make for each asset defines how Publish finds the data that is displayed in the asset when you populate it.

When you select an object in Teamcenter, and then populate an asset, a number of things happen.

1. If the selected object is a BOM line, Publish creates a structure context object in the database to persist the selection.
2. Publish creates a PLM XML file that contains information about the selected persistent object and about related objects. This additional content about the related objects is determined by the definition of the transfer mode that was assigned to the asset in the **Bind** dialog box.
3. Publish searches through the PLM XML file's content to find the exact data to expose in the asset. To do this, it retrieves the alias definition for the alias that was bound to the asset. The definition contains a list of paths. Publish attempts to follow each path from the root object in the PLM XML file until it finds some valid data. This data can consist of anything from a string, such as a form attribute value, to a file containing an image.

4. Teamcenter creates a snapshot of the data in the asset on the Visio pane.
5. Teamcenter stores the ID of the selected persistent object in the Visio file, associating it with the asset, for later retrieval during an update operation.

The transfer modes and aliases that are available in the **Bind** dialog box are all defined in the alias file. Each transfer mode functions as a recipe for what is included in the PLM XML file. The definition of each alias functions as a recipe that tells Teamcenter which parts of the exported PLM XML information to expose in an asset.

The snapshot of Teamcenter's data is visible whenever you open the page file for viewing. You can also update an asset so that it displays changes that were made in Teamcenter since the asset was first populated. When the asset is updated, new PLM XML is generated from the same selected object using the same transfer mode, and the same alias is then used to find the exact data from that PLM XML to be displayed in the asset.

A sample set of alias definitions is provided with the default installation, but you should create alias definitions and transfer modes to support other situations. To do this, you should:

- Ensure that the information appears in the PLM XML file. If it does not, create or modify a transfer mode so that the information is visible.
- Ensure that an alias definition correctly extracts this information from the PLM XML file. If not, modify existing aliases or create new aliases. The default alias definition file contains instructions and examples of aliases.

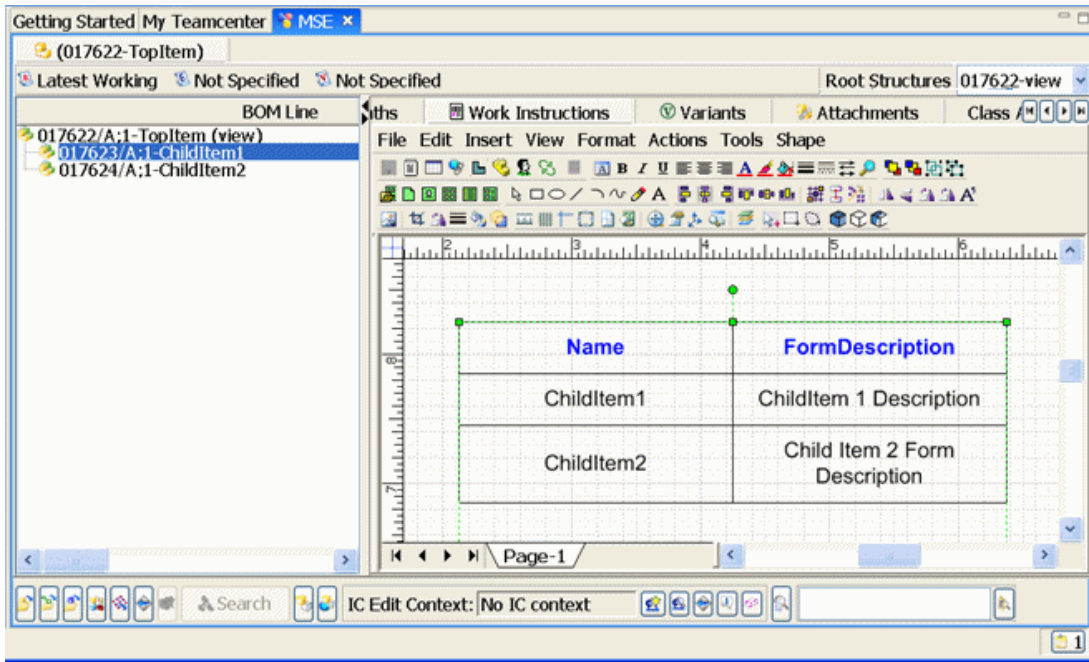
Teamcenter includes a default alias file (**TcPublishSystemAlias.xml**) that you can use as the basis for your own mapping files and its location is defined in the **TCPublishing\_Default\_Alias\_File** preference. The comments within this file contain detailed notes on the structure and usage of the alias file. Read these before beginning with customization.

Caution:

Do not edit this file directly. Create your own alias dataset and work in this file.

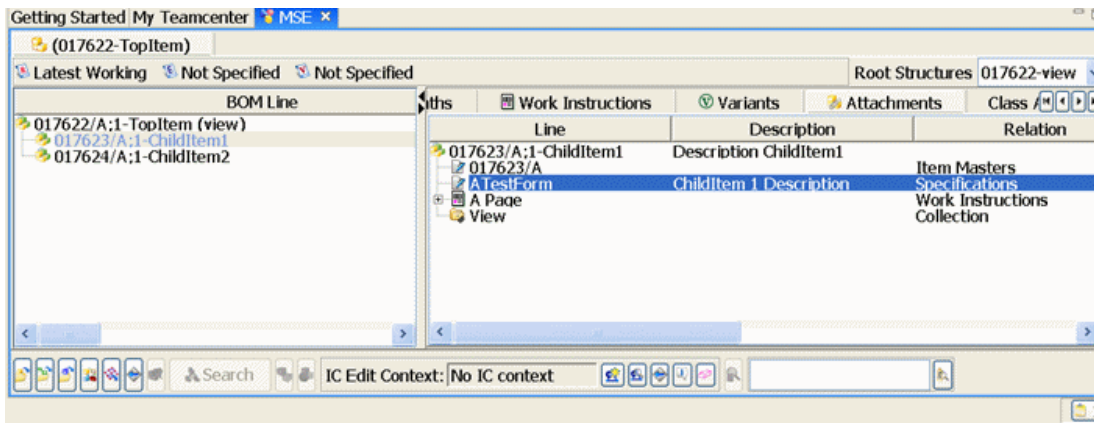
## Building a sample alias file

The following figure shows a publishing page with a single table asset that has been populated with the **Name** property of child items and the **Description** property of an attached form from each child item.



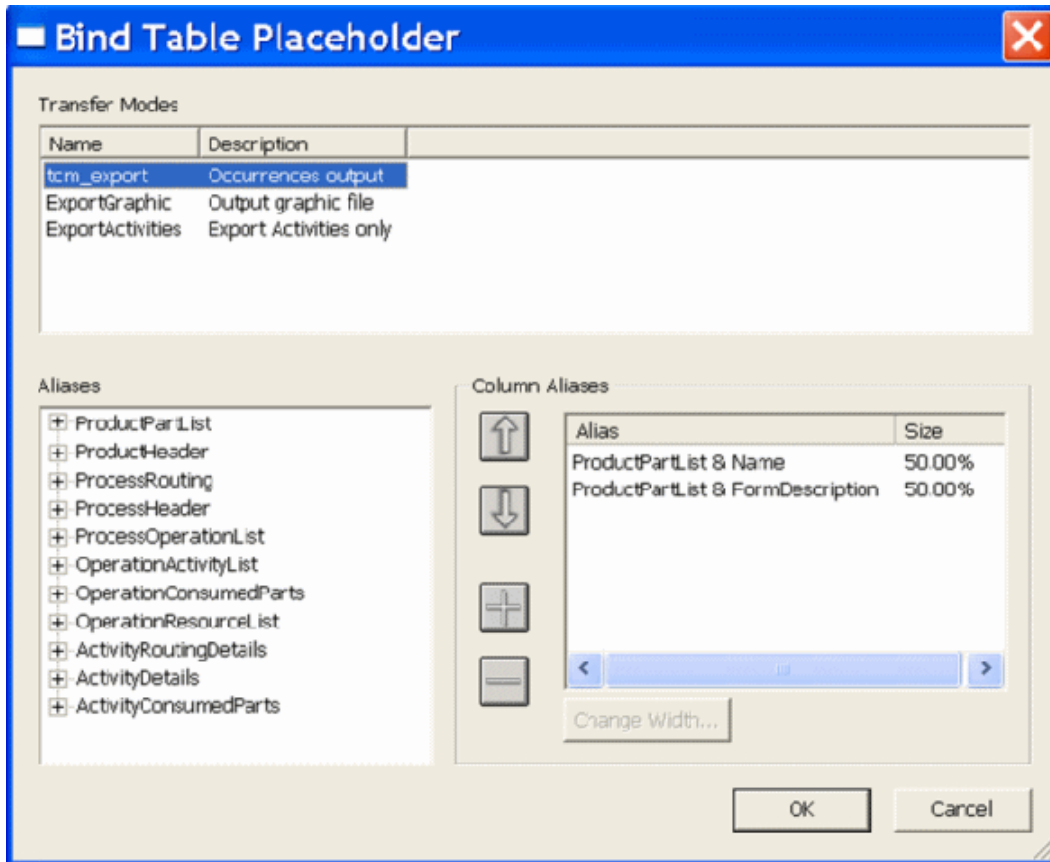
Example result – table asset populated with child item names and form descriptions

This example starts with a simple product structure containing a parent item with two direct child items. Each child item has an attached form. This scenario is shown in the following figure.



Sample product structure

To populate the table asset with data, you must assign a binding to the table asset. The binding used is **ProductPartList & Name** and **ProductPartList & FormDescription**, as shown in the following figure.

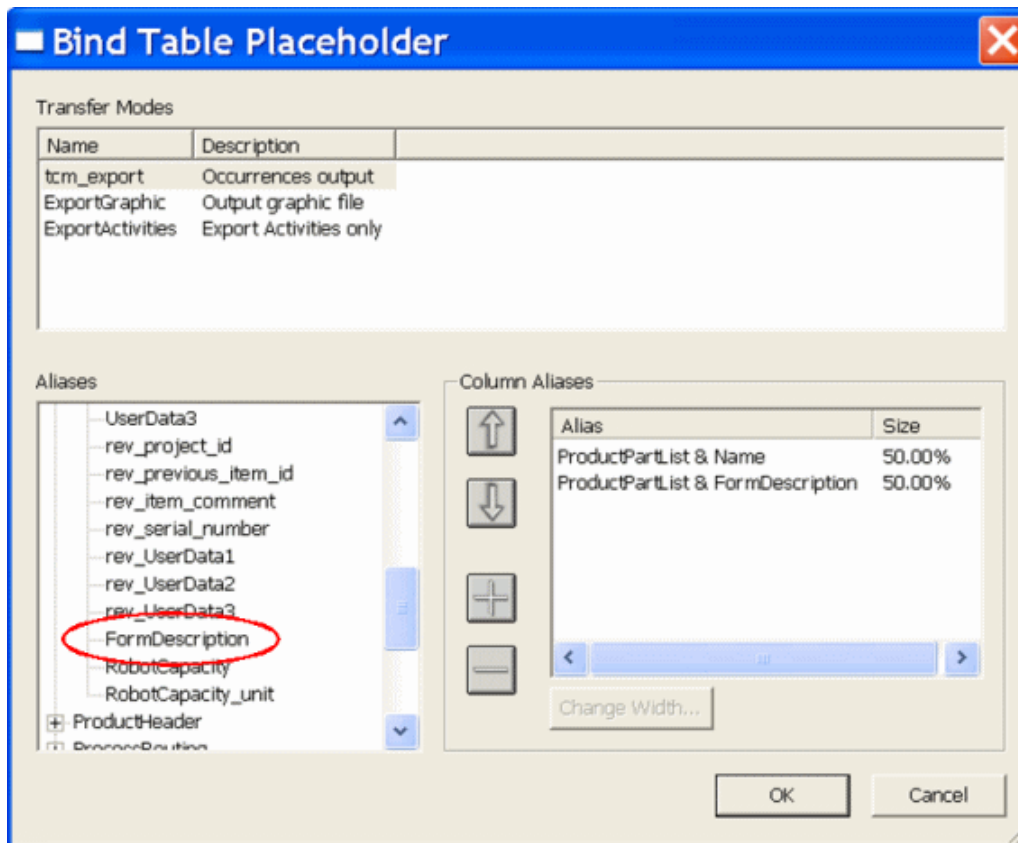


**Publish Bind dialog box**

You can find the sample alias file that was used to create this example (**ProjectTcPublishSystemAlias.xml**), as well as the PLM XML code generated by the selected transfer mode (**tcm\_export.xml**) in the **Sample** directory of your installation. This must be installed separately. If you do not have a **Sample** directory, contact your Teamcenter administrator.

In the example, the second binding involves the **Description** property of a form attached to the child item. The default alias file does not provide a definition for attaching the **Description** field of a form to a child item so you must modify this alias file.

In the following figure, notice the new **FormDescription** alias entry.



Expanded ProductPartList alias showing FormDescription entry

This new entry is created by adding a new top level alias definition into the alias file. The following code shows excerpts from the **DefaultTcPublishSystemAlias.xml** file. The line shown in bold print results in the new entry in the **Bind** dialog box shown in the figure. The statements in italics refer to other alias definitions.

```
<!--#####Product part list information addresses the first level parts list of
an
occurrence it will return only the child occurrences of the first level occurrence
selected.
```

```
#####-->
<Alias name="ProductPartList" assetType="8" tooltip="product data"
  qualifier="2" accumulate="Quantity">
  &OCC-ATTR-DEF;
</Alias>
```

```
<!-- properties that you want to see on an occurrence.
Entity definition is used to avoid lot of typing-->
```

```
<!ENTITY OCC-ATTR-DEF "
  <Alias name='ID' default='No ID' />
  <Alias name='Description' default='No description' />
  <Alias name='Name' default='no name' />
  <Alias name='Rev' default='A' />
  <Alias name='Transform' default='Identity' />
  <Alias name='RevSubType' default='ItemRevision' />
  <Alias name='LogicalID' default='Logical Identity' />
  <Alias name='SequenceNumber' default='No find#' />
```

```

    <Alias name='Quantity' default='1' />
    <Alias name='ParentName' default='No parent' />
    <Alias name='Type' default='No type' />
    &FORM-MASTER-ATTR-DEF;
    &ICO-ATTR-DEF;
  " >
<!ENTITY FORM-MASTER-ATTR-DEF "
  <Alias name='project_id' default='cannot get project_id' />
  <Alias name='previous_item_id' default='cannot get previous itemid' />
  <Alias name='item_comment' default='cannot get comment' />
  <Alias name='serial_number' default='cannot get serial_number' />
  <Alias name='UserData1' default='cannot get user_data_1' />
  <Alias name='UserData2' default='cannot get user_data_2' />
  <Alias name='UserData3' default='cannot get user_data_3' />
  <Alias name='rev_project_id' default='cannot get project_id' />
  <Alias name='rev_previous_item_id' default='cannot get previous itemid' />
  <Alias name='rev_item_comment' default='cannot get comment' />
  <Alias name='rev_serial_number' default='cannot get serial_number' />
  <Alias name='rev_UserData1' default='cannot get user_data_1' />
  <Alias name='rev_UserData2' default='cannot get user_data_2' />
  <Alias name='rev_UserData3' default='cannot get user_data_3' />
  <Alias name='FormDescription' default='No description' />
">

```

For the new alias definition to traverse the PLM XML file, another edit is necessary. The following entry shows the traversal performed by the new alias definition. This entry is added to the **<!ENTITY FORM-MASTER-ATTR ">** section of the alias file.

```

<Alias name='FormDescription'>
  <Path>*:AssociatedAttachment[@role='IMAN_specification']:@associatedAttachmentRefs</Path>
  <Path>*:Form:@attachmentRef</Path>
  <Xpath>Description</Xpath>
</Alias>

```

To determine what to add and how the traversal works, reference the PLM XML file generated by the example and found in the **Samples** directory.

In the binding, **ProductPartsList & FormDescription** was selected. Find the **Alias** and **AliasAlternate** sections for **ProductPartsList** in the alias file. The following code section shows the relevant lines from the generated PLM XML file. All PLM XML files have a **traverseRootRef** entry like this:

```
<Header id="id1" traverseRootRefs="#id3" transferContext="tcm_export"></Header>
```

**#id3** is the start node for the traversal. The **transferContext** transfer mode is used to generate this PLM XML code.

The start node is **AssociatedAttachment**.

Look through the alias file in the **ProductPartsList** section for all alias definitions that have a primary object specifying an **AssociatedAttachment** node. The following code contains the first one:

```

<!--#####Starting at a structure context as a root to a top level
item#####-->
  <AliasAlternate name="ProductPartList">
    <Path>AssociatedAttachment:AssociatedAttachment[@role='contents']:@childRefs</
Path>
    <Path>AssociatedAttachment:*:@attachmentRef</Path>
    <Path depth="1..1">*:*:@occurrenceRefs</Path>
    &OCC-ATTR;
    &FORM-MASTER-ATTR;
    &ICO-ATTR;
  </AliasAlternate>

```

The traversal tries each alias rule that has a primary object matching the start node and traverses through each path statement until it gets to the **Xpath** statements. If any **Path** statement fails during the traversal, the traversal tries the next matching alias statement. The first **Path** statement is:

```
<Path>AssociatedAttachment:AssociatedAttachment[@role='contents']:@childRefs</Path>
```

This statement causes the traversal to proceed to **AssociatedAttachment** nodes with **role=contents** via the **childRefs** attribute in the PLM XML file. The PLM XML statement **childRef** attribute is:

```
childRefs="#id4 #id72">
```

Therefore, the traversal moves to PLM XML nodes with **id4** and **id72**, if they match the **role=contents** condition. The PLM XML node containing **id4** is:

```
<AssociatedAttachment id="id4" attachmentRef="#id9" role="contents">
```

The next **Path** statement in the alias definition is:

```
<Path>AssociatedAttachment:*:@attachmentRef</Path>
```

This **Path** statement indicates that the system expected **AssociatedAttachment** nodes as results from the last statement. If it did not, this alias definition would fail at this point.

The current **Path** statement tells the system to use the **attachmentRef** and return whatever is there. The **\*** symbol is a wildcard indicating that any node is sufficient. Consequently, the traversal proceeds to the node containing **#id9**.

```

<Occurrence id="id9" instancedRef="#id43" associatedAttachmentRefs="#id45 #id46 #id49
#id67"
  occurrenceRefs="id12 id32">
  <ApplicationRef application="Teamcenter" label="SIEhslreBrSpAA/ScFh8BEQBrSpAA"></
ApplicationRef>
  <UserData id="id10" type="AttributesInContext">
  <UserValue value="" title="AO_ID"></UserValue>
  <UserValue value="" title="SequenceNumber"></UserValue>
  <UserValue value="" title="OccurrenceName"></UserValue>
  <UserValue value="" title="Quantity"></UserValue></UserData>
  <Transform id="id8">1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1</Transform></Occurrence>

```

Node 9 is an occurrence.

The next **Path** statement is:

```
<Path depth="1..1">*:*:@occurrenceRefs</Path>
```

This **Path** statement tells the system to go to the nodes referenced by the **occurrenceRefs** attribute. Again, the **\*** wildcard is used, so the traversal picks up whatever node is contained in the PLM XML. The first node ID is **#12**.

```
<Occurrence id="id12" instancedRef="#id14" associatedAttachmentRefs="#id19 #id22 #id25
#id30"
  parentRef="#id9">
```

This node is also an occurrence. The first line contains all the relevant reference information.

The next statement in the alias definition is:

```
&OCC-ATTR;
```

This is a macro statement that implies that the traversal expects to be on an occurrence node at this point and is getting ready to traverse its attributes.

At this point, the traversal again encounters alias names as shown for **OCC-ATTR**, **ID**, **Description**, **Name**, and so on.

Remember that the binding specified was **ProductPartsList & FormDescription**. The traversal started at the **ProductPartsList** node and arrived at this point. Now, it must find an alias name matching **FormDescription** to find any additional XML traversal rules to use. Going through the **OCC-ATTR** statement, the traversal cannot find any alias names that match **FormDescription** because the edit was not added to that entry. Therefore, the traversal continues on to the next entry in the original alias definition.

```
<!ENTITY OCC-ATTR "
  <Alias name='ID'>
    <Path>*:*:@instancedRef</Path>
    <Path>*:*:@masterRef</Path>
    <Xpath>@catalogueId|@productId</Xpath>
  </Alias>
  <Alias name='Description'>
    <Path>*:*:@instancedRef</Path>
    <Xpath>Description</Xpath>
  </Alias>

  <Alias name='Name'>
    <Path>*:*:@instancedRef</Path>
    <Xpath>@name</Xpath>
  </Alias>
  ... many more Occurrence attribute Alias Names
">
```

The next alias rule is:

```
&FORM-MASTER-ATTR;
```

The traversal looks in the alias file for this macro and starts traversing through the macro looking for an alias named **FormDescription**. The traversal finds the newly added alias as follows:

```
<Alias name='FormDescription'>
  <Path>*:AssociatedAttachment[@role='IMAN_specification']:@associatedAttachmentRefs</Path>
  <Path>*:Form:@attachmentRef</Path>
  <Xpath>Description</Xpath>
</Alias>
```

Remember that the last node collected was:

```
<Occurrence id="id12" instancedRef="#id14" associatedAttachmentRefs="#id19 #id22 #id25 #id30"
  parentRef="#id9">
```

Now the first **Path** statement in the current alias rule is this:

```
<Path>*:AssociatedAttachment[@role='IMAN_specification']:@associatedAttachmentRefs</Path>
```

This means the traversal is looking for **AssociatedAttachment** nodes with an **IMAN\_specification** role. This should be available through the **AssociatedAttachmentRefs** statement. In this case, the only node that matches these criteria is the one with **#id22** shown here.

```
<AssociatedAttachment id="id22" attachmentRef="#id23" role="IMAN_specification">
```

The next alias rule is:

```
<Path>*:Form:@attachmentRef</Path>
```

This means that using the **attachmentRef** attribute, the traversal expects to move to a form. **#id23** indicates the node for which the traversal needs to search.

```
<Form id="id23" name="ATestForm" accessRefs="#id15" subType="OfficeDocForm"
  subClass="OfficeDocForm">
  <Description>ChildItem 1 Description</Description>
  <ApplicationRef version="SYNhstU9BrSpAA" application="Teamcenter"
    label="SYNhstU9BrSpAA"></ApplicationRef>
  <UserData id="id24" type="FormAttributes">
  <UserValue value="" title="Title"></UserValue>
  <UserValue value="" title="Subject"></UserValue>
  <UserValue value="" title="Author"></UserValue>
  <UserValue value="" title="Manager"></UserValue>
  <UserValue value="" title="Company"></UserValue>
  <UserValue value="" title="Category"></UserValue>
  <UserValue value="" title="Keywords"></UserValue>
  <UserValue value="" title="Comments"></UserValue>
  <UserValue value="" title="Client"></UserValue>
  <UserValue value="" title="Project"></UserValue>
  <UserValue value="" title="Purpose"></UserValue></UserData></Form>
```

This shows that **#id23** is a form.

At this point, the traversal has successfully executed all the **Path** statements and ends with a final destination node. The next statement in the alias rule is:

```
<Xpath>Description</Xpath>
```

The **Xpath** statements describe what information to take from the resulting node to use in the publishing page. In the example, the property originally sought was the **Description** property of the form attached to the **ChildItem** item. This **Xpath** statement says to return the data for the **Description** entry for the current node. Looking at the PLM XML, this is:

```
<Description>ChildItem 1 Description</Description>
```

Looking at the populated publishing page in the figure entitled **Example result – table asset populated with child item names and form descriptions**, you see the same description in the **FormDescription** column for **ChildItem 1**. This accomplishes the goal of the example.

The **Path** statements are used to traverse from the start node to the final node. The **Xpath** statements are used to extract information from the final node. There are potentially several valid paths from the start node to the final node in the PLM XML. You need to determine any valid set of path statements to reach the final node.

## Specifying end objects

To specify which object is the end object that is exposed in an asset, you must mark it in a **Path** element in the alias definition. The following example shows an existing alias alternate for populating a 2D asset:

```
<AliasAlternate name="2DGraphic">
  <Path>AssociatedAttachment:AssociatedAttachment[@role='contents']:@childRefs</Path>
    <Path>AssociatedAttachment:ConfiguredOccurrenceGroup:@attachmentRef</Path>
  <Path>ConfiguredOccurrenceGroup:*:@childRefs</Path>
    <Path>Occurrence:AssociatedAttachmentRefs</Path>
      <Path>AssociatedAttachment:DataSet[@type='Image']:@attachmentRef</Path>
        <Path>DataSet:ExternalFile[@format='jpg' or @format='jpeg' or
          @format='gif' or @format='bmp']:@memberRefs</Path>
          &FILE-ATTR;
</AliasAlternate>
```

To identify the end object, use the **createRelation** attribute on the last **Path** element:

```
<Path createRelation="VisAssetPopulatedByEndobj">
```

This causes the end object UID to be sent back through Lifecycle Visualization to the Teamcenter rich client, along with the **createRelation** identifier. The rich client then creates the **VisAssetPopulatedByEndobj** relation from the publishing page dataset to the end object.

To create two different relationships to the same end object, use this syntax:

```
<Path createRelation='VisAssetPopulatedByEndobj,AnotherVisRelationship'>
```

When customizing alias definitions, you can identify the end object for any asset and Teamcenter creates the relation specified. You can define your own custom relations and have them created too by adding appropriate **createRelation** clauses or by adding additional relation names to existing **createRelation** clauses.

## Navigating through a PLM XML file

One approach to determine the path statements needed for your alias definition is to find the final node in the PLM XML file. Search the file for references to its ID. From this current node, find the node or nodes that reference it. Continue this backward traversal making sure you can get to the start node. Remember the start node is the node referenced by the **traverseRootRef** attribute. After you find a path using the backward traversal from the final node to the start node, go back and record the type of each node in the path. Use this information to build the path statements needed for an alias to your final node. Additionally, look at existing alias path definitions for examples of how to traverse from one type of node to another. A backward traversal using our example in pseudo XML would look something like this:

1. The **ChildItem 1 Description** node, the data you want to expose in the asset, is in the **Form id=id23** node.
2. The **Id23** node is referenced by **AssociatedAttachment id=id22** as an **attachmentRef** type.
3. The **Id22** node is referenced by the **Occurrence id=id12** statement as an **associatedAttachmentRef** type.
4. The **Id12** node is referenced by the **Occurrence id=id9** statement as an **occurrenceRef** type.
5. The **Id9** node is referenced by the **AssociatedAttachment id=id4** statement as an **AttachmentRef** type.
6. The **Id4** node is referenced by the **AssociatedAttachment id=id3** statement as a **childRef** type.
7. The **Id3** node is the **traverseRootRef** node from which traversal always starts.

There is another reference to **id22**, but if you follow it back you see a pseudofolder. Pseudofolders are generally not useful in a path traversal and can be ignored. After a while, it is useful to recognize what entities in the rich client relate to which nodes in the PLM XML file. Through some experimentation, these relationships become clearer. If you compare this pseudo XML code with the actual **Path** XML statements traversed in the example, you can see how they relate to each other. Therefore, when you need to update an alias file, if you can first determine a set of pseudo XML statements, it is easier to write the **Path** and **XPath** statements needed to edit the alias file.

## Strategies for working with alias files

The following are a few high-level tips and strategies about working with the alias file.

- Define a use case to work from.
  1. Determine what will be selected and what you want to see in what type of asset.
  2. When creating a template that will be used repeatedly, it is easier if the user can select a single object with which to populate all of the data in the work instruction.
  3. Find an example of the data in the database that you can select, for example, a BOM. If possible, make it unique so you can find it easily in the PLM XML file.
  4. Determine whether to use an existing alias or to add a new one to the **Bind** dialog box.
  5. Determine what alias file to use. You can develop an alias file for this user or this use case only that is simple to use and contains only one transfer mode, or you can add to an existing alias file.
- Identify a suitable transfer mode.
  1. Generate a PLM XML file by populating an asset with the transfer mode that you think will work.
  2. Examine the resulting file and make sure it has the data that you want to expose in the asset. If it does not, you must find a better transfer mode, or you must modify the transfer mode's closure rules to include your data.
  3. In some cases, it makes sense to create a custom transfer mode whose closure rules are optimized to efficiently generate XML that supports only the assets in a single template.
- Examine the PLM XML file and identify the nodes that provide a path from the root ref to the data in which you are interested.
- Define an alias alternate that follows the node path. To save time, test the alias definition using the stand-alone test utility on the generated PLM XML file. The utility is described in the alias file header.
- Decide what other use cases need to be supported by your alias definition.
  1. The data may be attached to the object in more than one way.
  2. The user may start with a different populating object.
- Create a new use case and repeat this process.

## Create a new publishing alias dataset

To create a new publishing alias from an existing alias file, do the following in My Teamcenter:

1. Use the **Search** button to find the **Installed Alias** dataset.
2. Export the **TcPublishSystemAlias.xml** file to your local disk using the **Named References** dialog box.
3. Rename the local XML file to something meaningful, such as **AliasTutorial.xml**.
4. Choose **File→New→Dataset**.

Teamcenter displays the **New Dataset** dialog box.

5. In the list of dataset types, select **TCPublishingAliases**, and then type a name and description for the new dataset.
6. Click the **Import** button, browse to the renamed alias file and click **OK**.

Teamcenter creates the new publishing alias.

7. Load the XML file into a text editor and read the notes at the top of the file.

### Note:

Each time you create an alias dataset, you must load the alias file.

You can optionally include packing of tables in the alias definition. If you choose to pack tables, lines that are identical except for quantity are collapsed into a single line. Two aliases with collapsed tables are provided as templates, **CollapsedProductPartList** and **OperationConsumedParts**.

## Update a publishing alias

To update an existing publishing alias from a modified alias file, do the following in My Teamcenter:

1. Right-click the dataset and choose **Named References**.
2. Choose **Edit→Cut** to remove the existing named reference for this alias.
3. Click the **Import** button, browse to the modified alias file and click **OK** or **Apply**. Teamcenter updates the publishing alias.

**Note:**

Each time you modify an alias dataset, you must reload the alias file.

## Load or reload the alias file

1. In My Teamcenter, create a new **TCPublishingPage** or load one of the publishing page templates.
2. Select the viewer pane and verify that the publishing page is loaded.
3. In your embedded Visio application, choose **File→Load Alias**.

Teamcenter displays the **Load Alias File** dialog box, with available aliases listed in the left-hand pane and loaded aliases listed in the right-hand pane.

To load a newly created alias dataset that is not already listed in the **Available Aliases** pane, click the **Add File by Name** button and browse to the newly created alias dataset.

4. To load a new alias file, select it in the **Available Aliases** pane and click the + button. To remove (unload) an existing alias, select it in the **Loaded Aliases** pane and click the – button.

When the aliases are listed correctly, click **OK** or **Apply** to load and unload alias files as necessary.

**Note:**

If you update an existing alias dataset by replacing its named reference, reload it by selecting the **Reload aliases** check box and clicking **OK** or **Apply**.

## Create or edit publishing pages

1. Open a configured structure in the Manufacturing Process Planner, Part Planner, Plant Designer, or Multi-Structure Manager application and select an occurrence.
2. Show the data tabs and select the tab for the required type of publishing page. You can configure Teamcenter to show each publishing context in a separate tab. For example, a user clicks the **Work Instructions** tab and the tab shows all the publishing pages associated with the occurrence by an **MEWorkInstructions** relationship. If none exist, an empty page is displayed.

**Note:**

You can create publishing pages for activities only in Manufacturing Process Planner.

3. Browse through the existing publishing pages or create a new publishing page from a defined template.

When editing an existing publishing page, the user can add, remove or edit Visio shapes and associate them with Teamcenter assets. An asset is typically 3D geometry, a 2D or 3D snapshot, a 2D image, or a link to a Microsoft Word file or a text file.

4. Save the new or changed publishing page to associate it with the selected occurrence.

## Creating manufacturing documentation in batch mode overview

You can create and update Teamcenter work instructions and portfolios in batch mode, processing multiple documents using the **Teamcenter Publish Batch** dialog box. You do not need to process individual documents manually. You can schedule these tasks to take place immediately or at a later date.

Batch processing uses the Dispatcher infrastructure for scheduling create and update features. Additionally, Dispatcher has built-in status, logging, and error reporting features. You can use the **Request Administrator Console** dialog box to obtain status information about the submitted task. Find this in the **Translation** menu of My Teamcenter after you install Dispatcher.

You must set up the batch processing in the Manufacturing Process Planner or Multi-Structure Manager applications. You can use My Teamcenter to modify the scheduling of the batch process.

## Enabling batch processing of publishing pages

Teamcenter batch processing of publishing pages uses Dispatcher infrastructure for scheduling create and update features. You must install the **PublishBatch** translator to take advantage of this feature. This enables the **PublishBatch** service.

There are several prerequisites for running the batch feature.

- The translator is installed on a Windows machine.
- VisAutomationApp, Microsoft Visio, FMS file client cache (FCC), and the Teamcenter service-oriented architecture (SOA) client are installed on the translator machine. VisAutomationApp is installed automatically when you install Lifecycle Visualization.
- The **startPublishBatch.bat** file is installed in a directory that is specified during installation of the **PublishBatch** translator. Each setting in this file must be modified as specified in the file.
- The **transmodule.properties** file in the **module/conf** directory has a **MaximumTasks** setting that defaults to **3**. This defines the maximum number of instances of publish batch tasks that can run simultaneously. Do not increase this value above **3**.
- The clients and Dispatcher Server should point to the common staging directory, and both should have read/write permissions to that directory.

## Troubleshooting the PublishBatch translator

The **startPublishBatch.bat** file is installed in a directory that is specified during installation of the **PublishBatch** translator. As described in the file, if you add **-log** to the **%TRANS\_PATH%\tcpublish\_batch** line, Teamcenter adds detailed progress and error entries to the task's log file. Teamcenter stores the log files in the directory you specify during installation.



# 9. Setting up enterprise bill of process (BOP)

## Configuring the enterprise BOP

You can use the following preferences, global constants, and cloning rules to configure the operation of the enterprise BOP for your site.

The following global constants and preferences configure the logical designator form.

Preference/Constant	Purpose
<b>Mfg0OperationLD</b>	Specifies the type of form that represents the logical designator of an operation. If empty, logical designators are not created for operations.
<b>Mfg0ProcessLD</b>	Specifies the type of form that represents the logical designator of a process. If empty, logical designators are not created for processes.
<b>Mfg0ProcessPartitionLD</b>	Specifies the type of form that represents the logical designator of a process partition. If empty, logical designators are not created for process partitions.
<b>MELogicalDesignatorDelimiter</b>	Specifies the delimiter in the display string of the logical designator.
<b>MELogicalDesignatorFormNameSuffix</b>	Specifies the suffix to the logical designator form name.

The following preferences configure cloning of generic BOPs and product BOPs.

Preference	Purpose
<b>GenericBOPTemplate</b>	Contains the default cloning rule when using a generic BOP as the template for creating a product BOP.
<b>ProductBOPTemplate</b>	Contains the default cloning rule when using a product BOP as the template for creating another product BOP.

Preference	Purpose
PlantBOPTemplate	Contains the default cloning rule when using a plant BOP as the template for creating another plant BOP.
GenericBOPAllowedTemplates	Contains a list of the cloning rules that are available when using a generic BOP as a template for creating a product BOP.
ProductBOPAllowedTemplates	Contains a list of the cloning rules that are available when using a product BOP as a template for creating another product BOP.
PlantBOPAllowedTemplates	Contains a list of the cloning rules that are available when using a plant BOP as a template for creating a plant BOP.
MEGenericBOP.MEGenericBOP.PasteDuplicateTemplate	Contains the default cloning rule used when executing the <b>Paste Duplicate</b> command from a generic BOP context to another generic BOP context.
MEGenericBOP.MEProductBOP.PasteDuplicateTemplate	Contains the default cloning rule used when executing the <b>Paste Duplicate</b> command from a generic BOP context to a product BOP context.
MEProductBOP.MEProductBOP.PasteDuplicateTemplate	Contains the default cloning rule used when executing the <b>Paste Duplicate</b> command from a product BOP context to a product BOP context.
MEProductBOP.MEGenericBOP.PasteDuplicateTemplate	Contains the default cloning rule used when executing the <b>Paste Duplicate</b> command from a product BOP context to a generic BOP context.
PasteDuplicateTemplates	Contains a list of cloning rules that are considered by the <b>Paste Duplicate</b> command.
PasteDuplicateAllowedPlantBOP Templates	Contains a list of cloning rules names to be used when pasting duplicate objects (clones) between different types of plant bills of process.
MEGenericBOP.Mfg0MEPlantBOP.PasteDuplicateTemplate	Contains the default cloning rule used when the <b>Paste Duplicate</b> command is executed between a generic BOP context and a plant BOP context.
MEProductBOP.Mfg0MEPlantBOP.PasteDuplicateTemplate	Contains the default cloning rule used when the <b>Paste Duplicate</b> command is executed between a product BOP context and a plant BOP context.
Mfg0MEPlantBOP.Mfg0MEPlantBOP.PasteDuplicateTemplate	Contains the default cloning rule used when the <b>Paste Duplicate</b> command is executed between a plant BOP context and a plant BOP context.

Preference	Purpose
<b>ProductBOPAllocationTemplate</b>	Contains the default cloning rule used when allocating from a product BOP to a plant BOP.
<b>ProductBOPAllocationCloningRule</b>	Contains the name of the cloning rule used for allocating from a product BOP to a plant BOP.

The following preferences configure the allocation mechanism between product BOP and plant BOP.

Preference	Purpose
<b>MEAllocationLevel</b>	Contains the type of process area that can have an operation or process allocation. If no type is defined, there is no limitation on the operation allocation.
<b>MEProcessIconPropertyNames</b>	Specifies whether plant BOP structures display an allocation indication symbol for processes if they were allocated from a product BOP.
<b>MEOperationIconPropertyNames</b>	Specifies whether plant BOP structures display an allocation indication symbol for operations if they were allocated from a product BOP.

The following preferences configure the expansion and display of the BOP.

Preference	Purpose
<b>typeAndRuleForProcessConfiguration</b>	BOP expand relies on this preference to evaluate whether to filter out the consumed item parent (typically an operation or process). This preference includes what types and rules to consider in order to run the evaluation (a server intensive activity). By default, this preference includes values that are OOTB types. If there is no need to filter operations or processes while expanding, you should leave the preference empty.
<b>MEBopDisplayAssyAsEndItems</b>	This preference improves BOP Display in the embedded viewer while using Manufacturing Process Planner. All consumed items in the BOP structure are considered as an enditem while being displayed in the embedded viewer. This preference enables the lightweight loading of the BOP structure where the traversal of the BOM consumed under the process will be done faster as it is being displayed in the

Preference	Purpose
<b>MEBopDisplayOccTypesList</b>	embedded viewer while using Manufacturing Process Planner.  This preference defines occurrence types used for the behavior displayed by the <b>MEBopDisplayAssyAsEndItems</b> preference. The default value is <b>MEConsumed</b> .

Two cloning actions are provided for enterprise BOP structures:

- **EBOPCreate**

Indicates the origin link relation. For example, this is used when cloning a product BOP from a generic BOP.

- **EBOPReference**

Indicates that a product BOP origin link is used as a reference to construct the generic BOP link.

For these actions, configure the following copy action rule entries to use when cloning enterprise BOP structures:

- **MapLD.Ignore**

Map according to logical designator equivalence. If unsuccessful, do not add the occurrence to the clone.

- **MAPLDAO.Reference**

(Default action) Map according to logical designator equivalence. If unsuccessful, map according to the absolute occurrence ID. If one is not found, the clone references the source.

- **MAPLDAO.Ignore**

Map according to logical designator equivalence. If unsuccessful, map according to the absolute occurrence ID. If one is not found, do not add the occurrence to the source.

## Setting up scheduled part assignment resolution

You can schedule the resolution of part assignments to run at a later date or time or at a specific recurring time. This is helpful if you are resolving large structures where the action is time-consuming or if you resolve structures on a regular basis. Teamcenter uses the Dispatcher application to run the resolution—at a time you specify.

To use this feature, your administrator must configure the asynchronous services.

The asynchronous services require the installation of Teamcenter Dispatcher.

**Note:**

Because asynchronous tasks run in a separate session, the current rich client session is not updated to reflect any changes that are made. If you choose to execute the task immediately, you do not see all of the results in the current session. Refreshing the window or reloading the structure after the task is complete updates your session to see the changes made by the asynchronous process. This is not an issue for sessions that start after asynchronous processing is complete.

## Setting up logistics planning

To set up the logistics management feature in Manufacturing Process Planner, there are the following installation prerequisites:

- Vendor management
- Enterprise bill of process

Additionally, you must select the following TEM options:

- **Logistic Process Planning**
- **Logistics for Rich Client**

Teamcenter installs or modifies the following preferences when installing this solution. Normally, you should not need to modify these preferences:

**MEPlantRevision.RENDERING**  
**MEPlantRevision.SUMMARYRENDERING**  
**MEPlantRevision\_DefaultChildProperties**  
**MEPlantRevision\_CompanyLocation\_default\_relation**  
**MEAssignCustomizedOccurrenceType**

To set up a basic logistics structure, the following is required:

- Vendor items set up in the vendor management system representing each supplier, including an exact street address for each one
- An address specified in the **Summary** view for each plant revision
- A logistic BOP structure containing a linked plant and plant BOP structure

To allow the logistics station resolver to function properly with a collapsed plant BOP, you must ensure that the in-context ID is populated on all the consumed lines of the plant BOP. Set the **MECopyIdInContextToAssignedLine** preference to **true** before you assign any objects to the plant BOP structure.

## Line balancing reports

### Customizing a line balancing report

Line balancing is a method to improve the throughput of assembly lines while reducing manpower requirements and costs. Line balancing addresses the problem of assigning operations to work stations along an assembly line in such a way that the assignment be optimized in some sense. You perform line balancing in Manufacturing Process Planner.

During the line balancing planning activities, you can create a variety of reports.

Line balancing uses the Teamcenter report framework to generate reports. The reports are created in the Microsoft Excel XML format and saved in datasets. By default, the dataset is attached to the scope of the view from which the report is launched. There is a defined default report definition and style sheet associated with each report type. These are defined in preferences.

Report	Preference for report definition ID	Preference for style sheet name
Constraints Consistency Check	MELBConstraintsConsistencyReportID	MELBConstraintsConsistencyStylesheet
Constraints Violation Check	MELBConstraintsViolationReportID	MELBConstraintsViolationStylesheet
Balancing Overview	MELBBalancingOverviewReportID	MELBBalancingOverviewStylesheet
Line Detail	MELBLineDetailReportID	MELBLineDetailStylesheet

You can create a new report definition in the Report Builder application and modify the report preference to point to that new definition. You can modify the report definition to change parameters or add style sheets.

### Customize the contents of the reports

You can customize the following in a report definition:

- Hide a section of the report.

- Hide a column of the report.
  - Add a column to show a BOMLine property.
1. Open the relevant report definition in Report Builder.
  2. In the **Parameters** section, modify the parameters using the following parameters.

Name	Meaning	Example
HIDE	Specifies a section of the default report that is not displayed.	HIDE CONSTRAINTS_DEFINITION HIDE PROCESS_AREA_INFO
HIDE_COLUMN	Specifies a column of the default report that is not displayed.	HIDE UTILIZATION HIDE CATEGORY_NVABR
SHOW_COLUMN	Specifies a BOM line property that is added as an additional column.	SHOW_COLUMN bl_resequence_no

## Line balancing report parameters

### Constraints Consistency Check Report

Constaints Consistency Check Report				
Date	13.01.2011			
Time	12:35			
Top Level Item	000001/A;1-product BOP Root			
Revision Rule	Latest Working			
Variant Rule	ProductOption= Small;			
Check Result	failed			
Check Details				
Description	Involved Objects			
The objects are in a loop	Op1, Op2, Op3			
Constraints Definition				
Object	Predecessor	Group	Simultaneous	Incompatible
Op1	Op3 Op5	Op8 Op9		
Op2	Op1			
Op3	Op2			

Enter this

To

HIDE CONSTRAINTS\_DEFINITION

Hide the **Constraints Definition** section of the report

### Constraints Violation Check Report

Constaints Violation Check Report				
Date	13.01.2011			
Time	12:35			
Product BOP Top Level Item	000001/A;1-product BOP Root			
Revision Rule	Latest Working			
Variant Rule	ProductOption= Small;			
Plant BOP Top Level Item	000002/A;1-plant BOP Root			
Revision Rule	Latest Working			
Variant Rule	ProductOption= Small;			
Check Details				
Description	Involved Objects			
The assignment causes a precedence violation	Op1, Op2, Station1, Station2			
The assignment causes a precedence violation	Op22, Op37, Station10, Station12			
Constraints Definition				
Object	Predecessor	Group	Simultaneous	Incompatible
Op1	Op5	Op8 Op9		
Op2	Op1			
Op3	Op2			
Process Area Information				
Process Area	Predecessor	Allocation		
station 1	station 2	op 2		
station 2	station 10	Op1		
station 10	station 12	op 9		
...				
station 12	station 41	Op8		

Enter this

To

HIDE CONSTRAINTS\_DEFINITION

Hide the **Constraints Definition** section of the report.

HIDE PROCESS\_AREA\_INFO

Hide the **Process Area** section of the report.

### Balancing Overview Report and Balancing Line Detail Report

Balancing Overview Report										
Date	13.01.2011									
Time	12:35									
Product BOP Top Level	000001/A;1-product BOP Root									
Scope	000030/A;1-Line 2									
Revision Rule	Latest Working									
Variant Rule	ProductOption= Small;									
Plant BOP Top Level	000002/A;1-plant BOP Root									
Revision Rule	Latest Working									
Variant Rule	ProductOption= Small;									
Summary										
Cycle Time	100									
Number of stations	10									
Time Unit	Second									
Name	Capacity	Work Content	Allocated Time	Remaining Time	Number of Workers	Utilization	Category			
							VA	NVA	NVABR	
Station 1	300	285		15	3	216	76%	69	24%	
Station 2	200	206		12	2	126	61%	80	39%	
...										
Station 10										

Balancing Line Detail Report										
Date	13.01.2011									
Time	12:35									
Product BOP Top Level	000001/A;1-product BOP Root									
Scope	000030/A;1-Line 2									
Revision Rule	Latest Working									
Variant Rule	ProductOption= Small;									
Plant BOP Top Level	000002/A;1-plant BOP Root									
Revision Rule	Latest Working									
Variant Rule	ProductOption= Small;									
Summary										
Cycle Time	100									
Number of stations	10									
Time Unit	Second									
Name	Capacity	Work Content	Allocated Time	Remaining Time	Number of Workers	Utilization	Category			
							VA	NVA	NVABR	
Station 1	300	285		15	3	216	76%	69	24%	
Process Area 1	200	187		13	2	151	81%	36	19%	
op1			98			77	79%	21	21%	
op2			89			74	83%	15	17%	
Process Area 2	100	98		2	1	65	66%	33	34%	
op3			65			65	100%			
op4			33					33	100%	
Station 2	200	206		12	2	126	61%	80	39%	
op11			97			50	52%	47	48%	
op12			33					33	100%	
op13			45			45	100%			
op14			31			31	100%			
...										
Station 10										

Enter this

To

HIDE\_COLUMN CAPACITY

Hide the **Capacity** column.

HIDE\_COLUMN WORK\_CONTENT

Hide the **Work Content** column.

HIDE\_COLUMN ALLOCATED\_TIME

Hide the **Allocated Time** column.

HIDE\_COLUMN REMAINING\_TIME

Hide the **Remaining Time** column.

HIDE\_COLUMN

Hide the **Number of Workers** column.

NUMBER\_OF\_WORKERS

HIDE\_COLUMN CATEGORY\_VA

Hide the **VA** column.

HIDE\_COLUMN CATEGORY\_NVA

Hide the **NVA** column.

HIDE\_COLUMN CATEGORY\_NVABR

Hide the **NVABR** column.SHOW\_COLUMN *BOM-line-property-name*

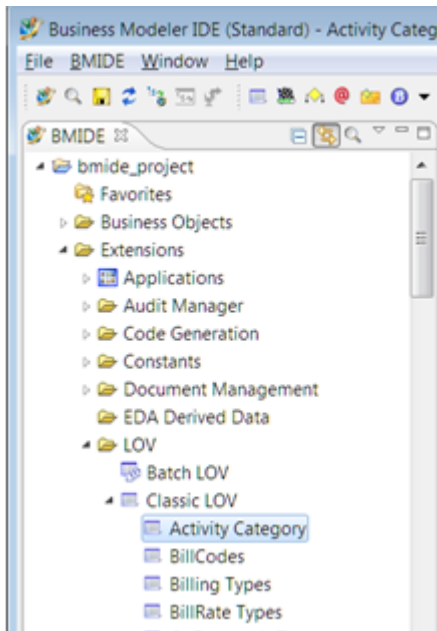
Display a column for the specified BOM line property.



# 10. Administering time data in the Time view

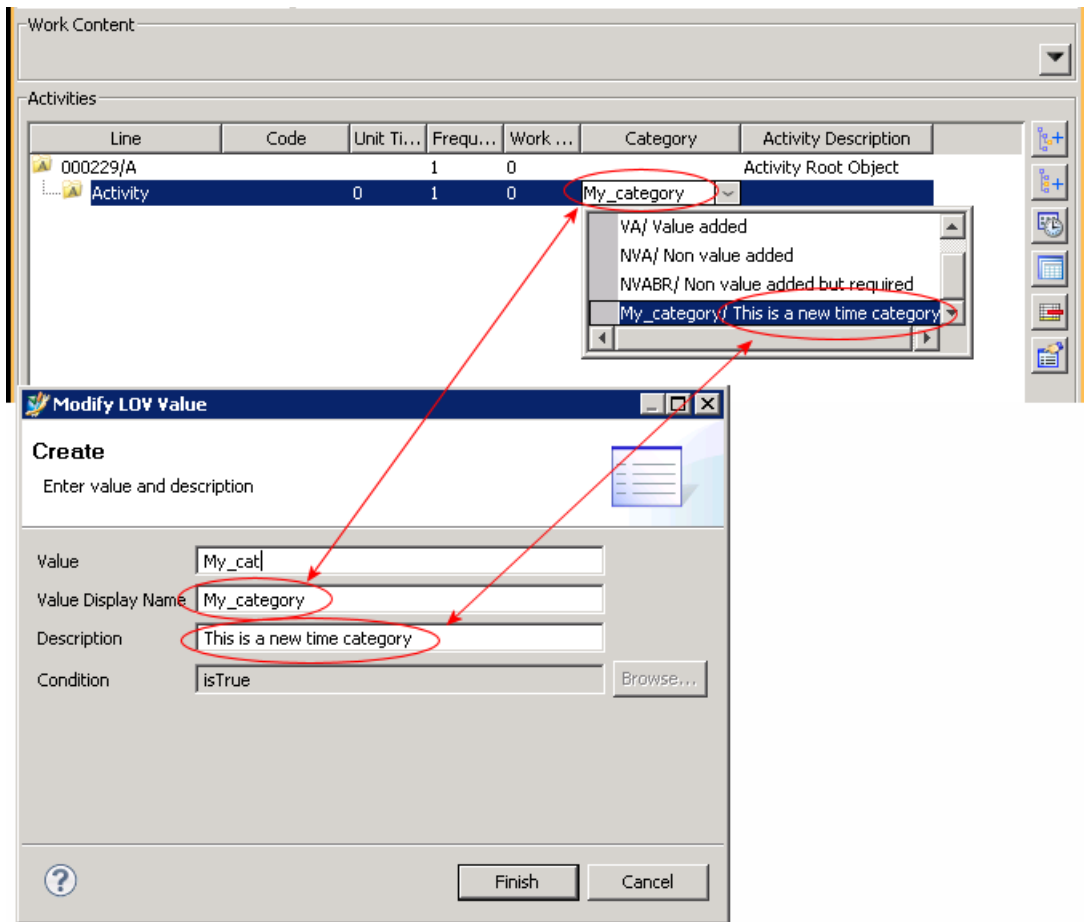
## Configure new categories in the Time view

1. In the Business Modeler IDE, open the **Advanced** perspective by choosing **Window**→**Open Perspective**→**Other**→**Advanced**.
2. Open the **Activity Category** in the **Classic LOV** folder in the **Extensions** view.

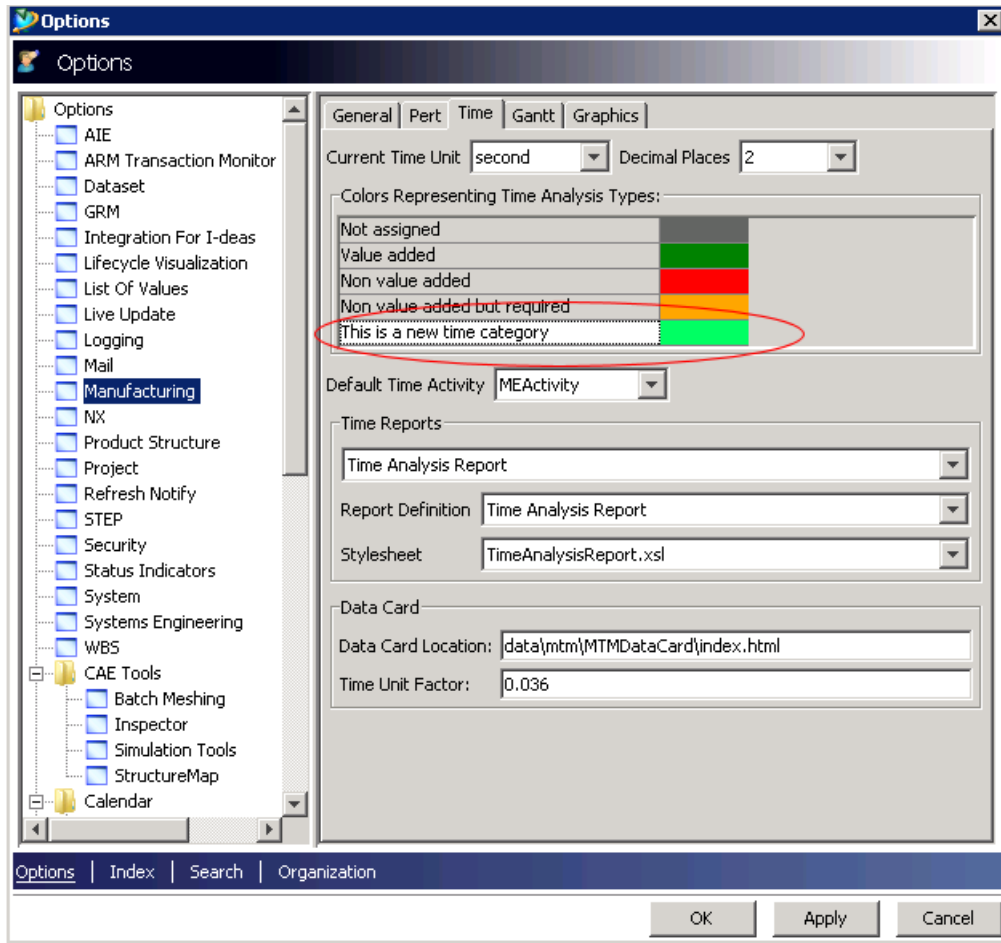


3. Add a new category, giving it a name and optional description.
4. Save and deploy the project.

Teamcenter adds a new category to the **Time** view.



- (Optional) Change the color of the category in the **Options** pane.



## MTM data card

### Configuring an MTM data card

You can add predefined activities based on information available in the Methods Time Measurement (MTM) system to operations in the **Time** view in Manufacturing Process Planner. This system sets industry-standard times required to perform standard manual tasks. The standards include a list of time elements where each element represents a specific and short step a worker can perform as part of the execution of an operation. Typically, the time elements are organized into groups according to types, distance, and difficulty. The standards are based on repeated measurements done on real production lines.

In the **Time** view, you can open a data card and select individual tasks containing a set time. Each of these tasks is added as a separate activity. You can then use these standard times to run time calculations and analyses.

The standard MTM data cards hold times in TMU (time measurement units). Teamcenter stores times in seconds. The **METimeDataCardConversionFactor** preference specifies the factor to convert from data card time units to seconds.

You can also replace the MTM data card with a customized one that includes times required for tasks that are specific to your company. If your customized data card contains times in units other than TMU, you must define the conversion rate between your data card units to seconds in the **METimeDataCardConversionFactor** preference.

Note:

Please be aware that the proper use of the time data card functionality does require thorough training in MTM methodology. Without such training the user activities may lead to erroneous results and does not represent a true MTM analysis.

For MTM training please contact your local MTM association.

## Install MTM data cards

You must install the data cards using Teamcenter Environment Manager before you can use them in Manufacturing Process Planner. The default data cards are not available until you install them.

## Create your own data card

1. Create a Microsoft Excel spreadsheet containing the data card information.

Each cell holding the time data must contain a hyperlink with the following format:

**usercode://code1|time\_as\_double|description**

Where	Is
<i>code1</i>	The time code that appears in the <b>Code</b> column in the <b>Activities</b> table.
<i>time_as_double</i>	The time required for the activity expressed as a double format.
<i>description</i>	The description of the activity. Teamcenter copies the description and uses it as the name of the activity.

Note:

You must use the pipe character (|) as a separator. To prevent illegal characters, you can use the ASCII hexadecimal representation %7C. You can use other hexadecimal symbols to replace special symbols in your URL text if you get incorrect results.

2. In Excel, choose **File**→**Save As...** to save the spreadsheet as a Web page (HTM or HTML extension).
3. Type the path to the HTM or HTML file in the **METimeDataCardPath** preference.

4. Make sure that the **METimeDataCardConversionFactor** preference is correctly set.

If you enter seconds as the unit in the Excel sheet, set this factor to **1**.

Tip:

You can also find these preferences in the **Data Card** section of the **Time** tab of the **Manufacturing** section of the preference **Options** dialog box.

5. Restart Teamcenter.

When you open a data card in the **Time** view, Teamcenter displays a tab for each sheet in the Excel file. When you click an activity cell in the data card, the activity time is displayed in the **Activities** table of the **Time** view.

## TiCon integration

### Understanding the Teamcenter to TiCon mapping

You can access time data from TiCon, an external time management system, and assign it to activities in Manufacturing Process Planner using a Web service. You can search for time elements in TiCon, create new activities, or overwrite existing activities with the timing data of time elements, and update activities assigned to time elements that were changed in TiCon. You can use the existing time management features to calculate time analysis based on the TiCon data.

By default, Teamcenter uses the following internal mapping system to map Teamcenter activity properties to TiCon time element attributes:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsi:Configuration xmlns:xsi="http://www.siemens.com/plm/Teamcenter/MSE/
TimeFieldMap">
  <FieldMapping>
    <Field TeamcenterProperty="al_activity_object_name"
TimeSystemAttribute="description"/>
    <Field TeamcenterProperty="al_activity_time_system_unit_time"
TimeSystemAttribute="tg"/>
    <Field TeamcenterProperty="al_activity_time_system_category"
TimeSystemAttribute="indicatorlid">
      <Value TeamcenterPropertyValue="NA" TimeSystemAttributeValue="1"/>
      <Value TeamcenterPropertyValue="VA"
TimeSystemAttributeValue="10000"/>
      <Value TeamcenterPropertyValue="NVA"
TimeSystemAttributeValue="10001"/>
    </Field>
  </FieldMapping>
</xsi:Configuration>
```

You can use the default mapping as a starting point to create your own mapping file. You can modify the default mapping to map your company's activity properties to the TiCon time element attributes.

## Map attributes of time elements to activity properties

1. Create a mapping file to map activity properties used by your company to TiCon time elements.
2. Create a dataset in Teamcenter and import the mapping file to it.

Siemens Digital Industries Software recommends you create a **Text** dataset as it is easiest to edit.

Note:

You must create this dataset within a folder.

3. Add the name of this dataset, its type, and reference type to the **METimeSystemFieldMappingDataset** preference.

# 11. Configuring studies

## Enable Study Manager

To enable **Study Manager** for end users, complete all steps in *Add Teamcenter Manufacturing Access Study Manager to a rich client in Easy Plan Deployment*.

Note:

You must complete this task to enable full functionality on a four-tier rich client. This task displays the **Open with Study Manager** command, which allows end users to locate and modify study data in Manufacturing Process Planner, and then send the data directly to Teamcenter Manufacturing Access Study Manager.

## Set up study synchronization

You must set up the study synchronization rules before publishing or synchronizing a study.

Note:

You must have DBA privileges to perform this task.

The synchronization rules are set up using a combination of object type and property that are to be considered.

1. Choose **Advanced** → **Study Synchronization Rules**.
2. Select the synchronization mode.
  - **Event-Based**  
Performs a partial synchronization or publish of only the selected properties.
  - **Time-Based**  
Performs a full synchronization or publish.
3. Select the direction in which the changes are synchronized.
  - Select **To Study** to set up synchronization changes from the source process structure to the study.
  - Select **From Study** to set up publish changes from the study to the source process structure.

4. Choose whether to include or exclude the properties you select in the list.
  - To exclude properties from synchronization:
    - a. Select **Exclude the following properties. Include the rest.**
    - b. Press Ctrl and select the object type and property name of each property you do not want to include.

**Tip:**

To perform a full **Time-Based** synchronization, select **Exclude the following properties** and remove all entries from the properties list.

- To include properties in synchronization:
  - a. Select **Include the following properties.**
  - b. Select the object type and property name.
  - c. Click **Add.**
  - d. Click each newly highlighted cell and open the list of entries by clicking the down arrow.
  - e. Select the desired object type and property name.

**Tip:**

To synchronize children of selected structures, select the object type and the **Mfg0sub\_elements** property.

5. Click **Save.**

Teamcenter saves the settings in study preferences. You do not need to modify these preferences manually.

**Note:**

You can set up custom synchronization rules by adding modes to the **Mfg0SimStudyRuleMode** global list of values (LOV) in the Business Modeler IDE. By default, this LOV contains only **Time-Based** and **Event-Based** entries. Your new mode is then displayed in the **Manage Synchronization Rules** dialog box, and you can create corresponding synchronization and publish rules for the custom mode.

# 12. Administering EBOM and MBOM

## Reasons for creating an MBOM

When working in the manufacturing environment, Siemens Digital Industries Software recommends that you create a manufacturing bill of materials (MBOM) from your engineering bill of materials (EBOM). There are several reasons for this:

- Generally, manufacturing engineers do not have write privileges on the EBOM and cannot modify it. For example, a design engineer may specify that a part is going to be purchased and sets a property value to **buy**. Subsequently, the manufacturing department determines that it is more economical for them to make this part. They want to change that property to **make** but do not have the privilege to do so.
- Assembling a part requires a structure that is different from the EBOM structure. For example, in the BOM, the exhaust manifold is part of the engine compartment. During assembly, however, the exhaust manifold must be added after the body is built onto the chassis as the exhaust pipe must be put through the bumper. This type of regrouping and adding of manufacturing-specific parts is most efficiently performed in a separate manufacturing BOM structure.
- Especially for final assembly, companies often have manufacturing parts that they consume in different MBOMs. These can only be created in a structure to which you have write access.
- There may be changes to an EBOM that should not immediately be reflected in the process structure. For example, you may want to continue manufacturing using the current parts to use up existing inventory. By using an MBOM, the manufacturing engineers have control over when the changes to the EBOM are reflected in the manufacturing process.
- Some companies do not author the EBOM in Teamcenter. It is imported from another system so that the EBOM in Teamcenter is a replication of the status of the EBOM in another system. Using an MBOM provides you with a buffer for these changes so that you can monitor and process them in a controlled manner.

Note:

You cannot open an MBOM in Siemens Digital Industries Software NX.

## Comparing the EBOM and MBOM

The manufacturing engineer creates a manufacturing structure (MBOM or manufacturing bill of materials) from the product structure (EBOM or engineering bill of materials). The MBOM is a different arrangement of the EBOM that can be optimized for manufacturing purposes. You must specify two structures to be the EBOM and MBOM using the **Link Structures** command. When you link an EBOM with an MBOM, their BOM view revisions are linked by an **METarget** relation.

To ensure that all required parts in the EBOM are contained in the MBOM you can run an advanced accountability check. This feature uses the in-context ID (IDIC represented by the **AbsOccID** property) to establish logical equivalence between aligned occurrences. If you want to use this feature, you cannot use the in-context ID for any other purpose. Using an accountability check, you can see if parts have been added or removed from the EBOM, discern any property changes, propagate any changes from the EBOM to the MBOM, or run the accountability check at a scheduled time.

If you use Multi-Site Collaboration and want to consume a remote EBOM in the MBOM, you must create the in-context IDs for the BOM lines to be consumed at the owning site before you export the EBOM to a remote site. Otherwise, a check does not report differences between the structures.

Caution:

- You cannot link an EBOM to an MBOM contained in a composition structure context. If you use a structure context to hold the MBOM, you cannot use the alignment check feature.
- Cutting and pasting the same line back into an MBOM creates a new occurrence without an in-context ID. This removes the equivalence between the new occurrence and the original source part in the EBOM and causes the pasted line to be displayed as missing in the other structure in an accountability check.

## EBOM-MBOM preferences

Preference	Description
<b>MEAlignedPropertiesList</b>	Defines the list of BOM line properties considered during alignment, comparison, and synchronization of aligned occurrences. By default, the <b>AbsOccID</b> , <b>Child Item</b> , <b>Notes</b> , <b>Quantity</b> , <b>Unit of Measure</b> , <b>Transform</b> , <b>Substitute List</b> , <b>Source</b> , and <b>Occurrence Type</b> properties are considered. Do not add properties to this preference unless you ensure methods exist to read from and write to them.
<b>MEAlignedPropertiestoExcludeFrom-Detection</b>	Defines a list of BOM line properties that are not compared when aligning occurrences in the EBOM and MBOM.
<b>MEAlignedPropertiestoExcludeFrom-Initialization</b>	Defines the list of BOM line properties that are not copied to an MBOM when the part is originally assigned from the EBOM to the MBOM.
<b>MEAlignedPropertiestoExcludeFromSync</b>	Defines a list of BOM line properties that are not copied to the aligned target occurrence during synchronization.
<b>MEAlignmentCheckReportLocation</b>	Defines if a persisted EBOM-MBOM alignment report is stored in a dataset in the My Teamcenter <b>Newstuff</b>

Preference	Description
	folder or if it is attached to the item revision of the top line of the target window.
<b>MEAlignmentCheckReportDefinition</b>	Contains the name of the report definition that controls the creation of the EBOM-MBOM alignment check report. A default report definition is provided but you can supply a custom report definition if necessary.
<b>MEAlignmentCheckReportStylesheet</b>	Contains the name of the Microsoft Excel style sheet on which the EBOM-MBOM alignment check report is based. A default style sheet is provided, but you can supply a custom style sheet if necessary.
<b>MEAssignAttachmentAndRelationTo-UseInCopyRule</b>	Specifies which attachment and relation types are referenced when you carry forward attachments that are associated in-context in the source structure.
<b>MEAssignAttachmentContext</b>	Specifies the context of the in-context attachments after assignment.
<b>MECreateIdInContextforAssignedchildlines</b>	<p>Assigns an IDIC to the child lines when assigning the parent from the EBOM to the MBOM.</p> <p>If you assign a line containing children from the EBOM to the MBOM, by default, only the parent line is given an IDIC. If you remove that parent line from that MBOM, the children move up a level in the hierarchy, but because they do not have an IDIC, they do not show up as a match when running an accountability check between EBOM and MBOM. You can change this behavior in the <b>MECreateIdInContextforAssignedchildlines</b> preference.</p>
<b>MEEffectivityAssignmentMode</b>	<p>Defines the assignment behavior from an Engineering BOM (EBOM) to a Manufacturing BOM (MBOM) for the BOMline property <b>bl_occ_effectivity</b>.</p> <p><b>0</b> (Default) Keep the default Teamcenter behavior.</p> <p><b>1</b> Inherit assignment behavior from parent.</p> <p><b>2</b> Apply calculated net effectivity behavior.</p>
<b>MEMBOM_Mirror_MakeRules</b>	<p>Specifies a BOM Line property, such as <b>bl_item_object_type</b>, and property value pair that serve as input for the decision as to whether to create a manufacturing-specific part.</p> <p>There are two ways to view defined BOM Line property names: 1. In the BMIDE, you can check the</p>

Preference	Description
	<p>properties defined for the object BOM Line. 2. In Structure Manager, if you right-click on the column title line and choose Insert Column(s)..., you get a list of the defined BOM Line properties.</p> <p>A common criteria may be, for example, whether you make or buy a part. Any parts designated as make in the EBOM are replaced by manufacturing-specific parts in the MBOM.</p> <div data-bbox="800 562 1450 852" style="border: 1px solid black; padding: 5px;"> <p>Note:</p> <p>This preference accepts only a string value.</p> <p>If you require a boolean or other attribute, you should write your own Business Modeler Framework extension to control a make or buy decision.</p> </div>
<b>MEMBOM_Mirror_RemoveLineWithIDIC</b>	Specifies that if a line in the EBOM is removed using the <b>me_create_mbom</b> utility, its counterpart in the MBOM is removed at update if it is already stamped with an IDIC.
<b>MEMBOM_Mirror_TypePrefixSuffix</b>	Specifies the item type for the new manufacturing-specific item and any prefix or suffix to add to the name of the part in the EBOM that is being replaced by a new part in the MBOM using the <b>me_create_mbom</b> utility.
<b>MEMBOM_Mirror_ReplaceMakeOnChange</b>	Enables the replacement of manufacturing BOM (MBOM) lines that were changed from <b>Make</b> to <b>Buy</b> in the engineering BOM (EBOM) during the MBOM update from the EBOM using the <b>me_create_mbom</b> utility.
<b>RollUp_Variant_Traverse_Level_Is_InstallationAssembly</b>	Determines the level to which rolled-up variant conditions are calculated.

## Automatically creating an MBOM from an EBOM

Teamcenter provides you with several automatic methods to create a manufacturing bill of materials (MBOM) from an EBOM (engineering bill of materials).

- **Create an MBOM from a recipe**

Create an MBOM from a template that contains a recipe for searching the EBOM for required parts. The recipe can contain a variety of search attributes such as an attribute search, spatial search, ID search, or combinations of search criteria.

This type of creation mechanism works well for *make to shelf* industries where the products are standardized such as the automotive or home appliance industries.

- **Use the `me_create_mbom` utility**

The `me_create_mbom` utility creates an MBOM based on a specific EBOM but replaces make parts or phantom levels in the EBOM with manufacturing-specific parts and directly copies over buy parts.

Your administrator can take advantage of user exits in this utility that allow you to embed your own logic into the utility that specify:

- What constitutes a make part using the `USER_create_or_ref_item` user exit (exposed in Business Modeler IDE using the `BMF_ITEM_create_or_ref_id` operation on an `Item` object).
- How to synchronize equivalent lines and align properties using the `USER_synch_item` user exit (exposed in Business Modeler IDE using the `BMF_ITEM_sync` operation on an `Item` object).
- What constitutes equivalent lines if there is no in-context ID or occurrence thread using the `USER_item_find_equivalent_line` user exit (exposed in Business Modeler IDE using the `BMF_ITEM_find_eqv_line` operation on an `Item` object).
- Which action to take on a released MBOM `Item Revision` object during update using the `USER_item_action_on_release` user exit (exposed in Business Modeler IDE using the `BMF_ITEM_action_on_release` operation on an `Item` object).

For more information, see *Integration Toolkit Function Reference*.

Your administrator can create a workflow to run this utility using the `ME-create-mirror-AH` and the `ME-update-mirror-mbom-A` action handlers.

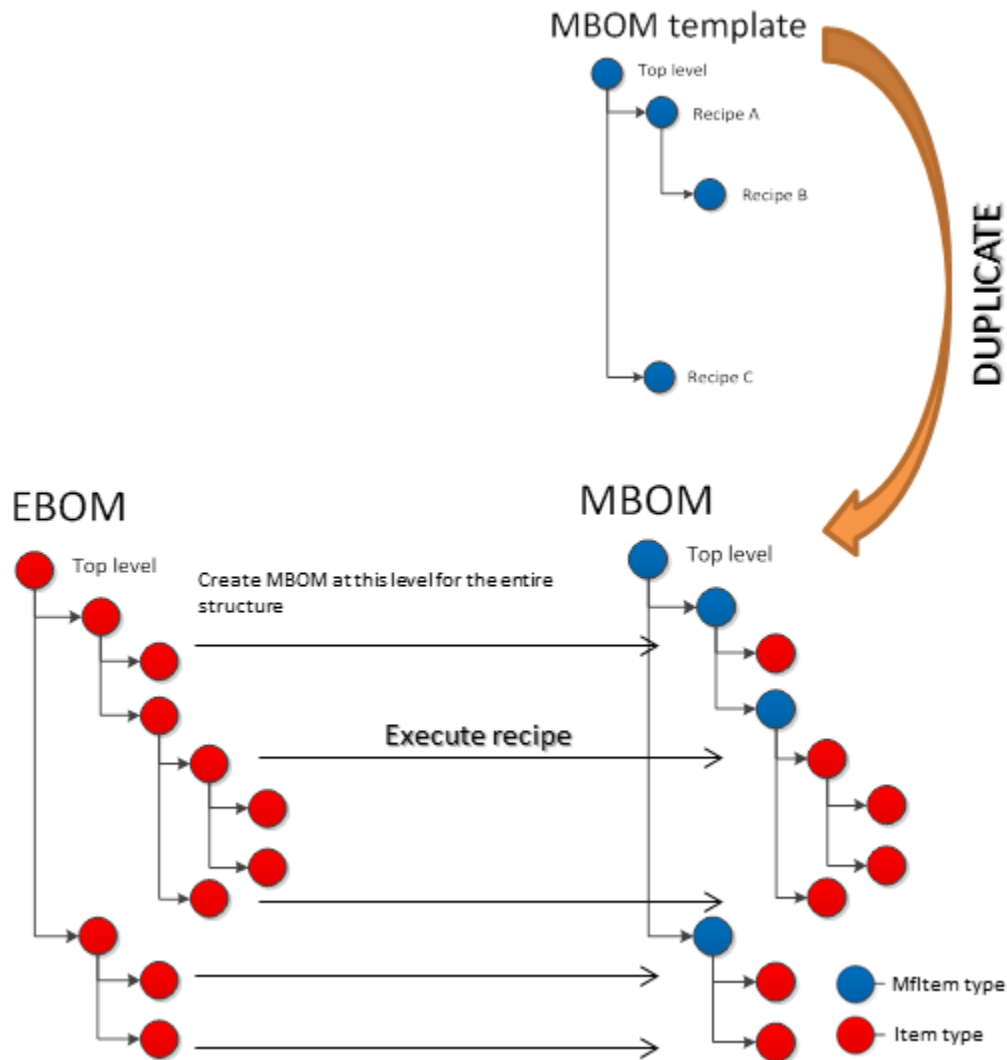
This type of MBOM creation method works well for *configure to order* customers such as heavy machinery.

Note:

Neither of the methods to create MBOMs supports part replacement through propagation.

## Create an MBOM from a recipe

You can automatically create an MBOM based on a recipe. A recipe consists of search criteria saved on each node of an MBOM template that define the parts to assign to that specific node.




Beginning with an MBOM template that contains phantom levels, you define a recipe for each part that is to be manufactured. When you run the search, Teamcenter searches in the specified EBOM for parts that fulfill the criteria and assigns them to the MBOM.

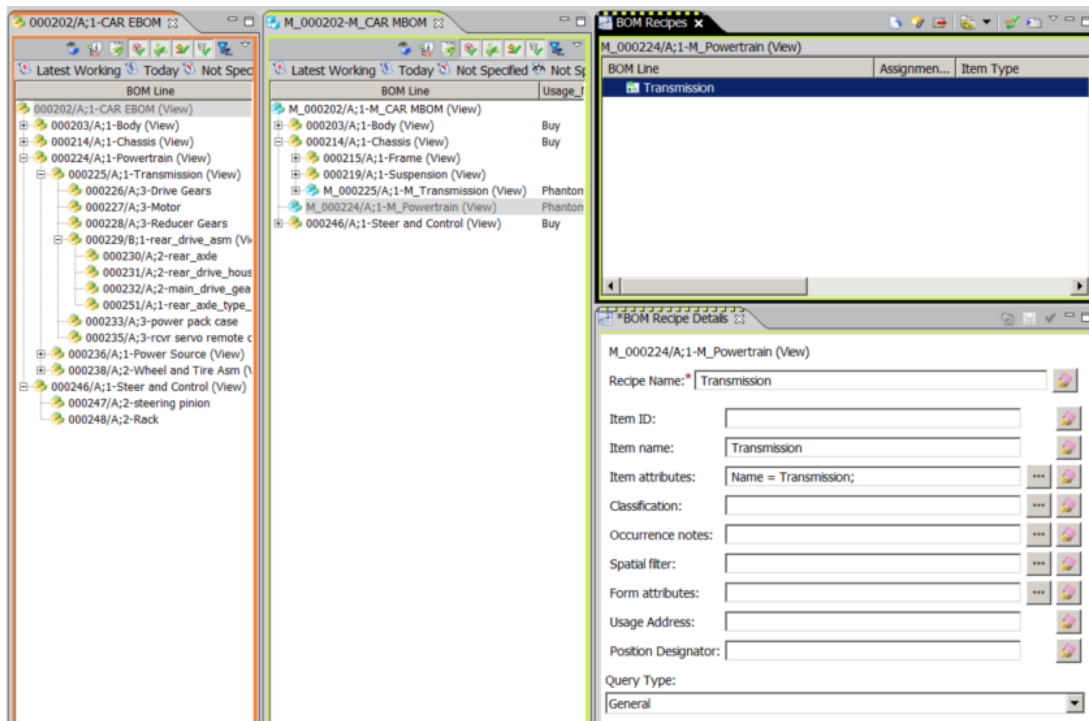
Unless a cloning rule is set up to ignore MBOM recipes, when you clone an MBOM that contains recipes, Teamcenter clones the recipes as well.

1. Open the MBOM template on which you want to save recipes.

This MBOM must already be linked to an EBOM structure.

2. Right-click a node on which you want to save a search criteria and choose **Open With**→**BOM Recipes**.
3. In the **BOM Recipes** view, click **Create/Update Recipe** .
4. In the **BOM Recipe Details** view, enter a name for the search and the search criteria that describe the parts that you want to add to the MBOM at the selected node.

What you enter as search criteria is company-specific but may include usage address, position designator, or a function-based numbering scheme making up the ID.



You can add multiple criteria to one search box by separating them with a semicolon.


5. Click **Validate BOM Recipe** .

Teamcenter runs the search on the EBOM and lists the parts that fulfill the search criteria in the **BOM Recipe Validation Results** view. Use this step to verify that you have entered the correct criteria.

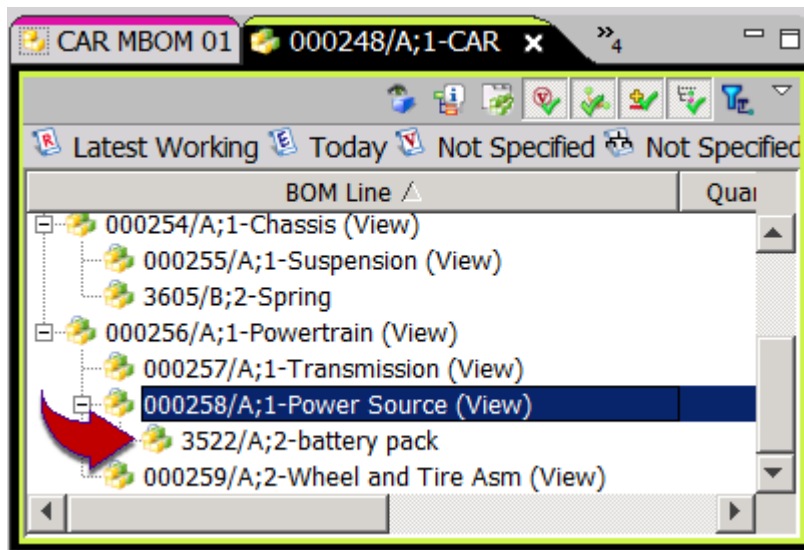
6. In the **BOM Recipe Details** view, click **Create New Recipe** .

Teamcenter lists the new recipe in the **BOM Recipes** view. You can save multiple recipes on an MBOM node.

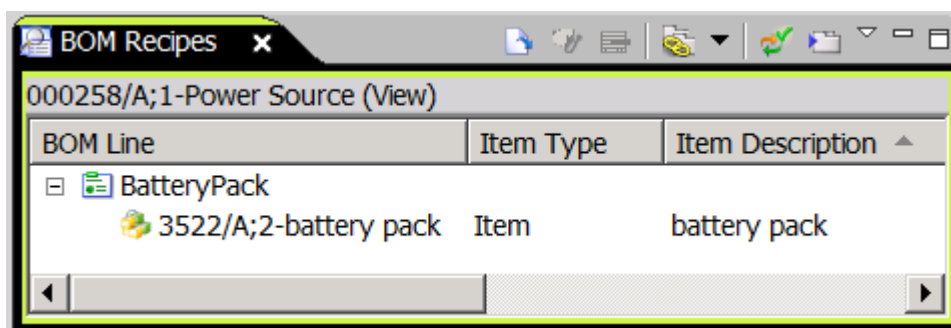
7. Do one of the following:

- To resolve one node of the MBOM template, select the recipe in the **BOM Recipes** view and click **Resolve BOM Recipe** .
  - To resolve multiple nodes of the MBOM template, select a node in the MBOM template and choose **Tools**→**Resolve BOM Recipe**.
8. In the **Resolve BOM Recipe** dialog box, select whether to remove the previously assigned parts and click **OK**. If you open the dialog box from the **Tools** menu, you can choose to resolve all nodes underneath the selected node by selecting **Resolve recipe recursively**.

Teamcenter adds the parts from the EBOM that fulfill the search criteria to the MBOM.



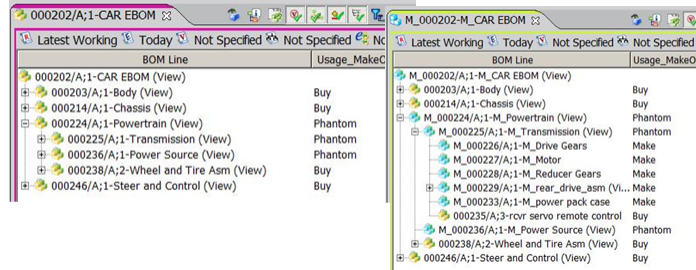
Additionally, the resolved parts are listed under the recipe in the **BOM Recipes** view.



You cannot directly copy a BOM recipe. You can, however, select a recipe in the **BOM Recipes** view, and then select a different node in the MBOM. The search criteria are still visible in the **BOM Recipe Details** view where you can save them as a new recipe on a different MBOM node.

## Creating an MBOM using the me\_create\_mbom utility

The `me_create_mbom` utility creates an MBOM based on a specific EBOM, but replaces *make* parts or phantom levels in the EBOM with manufacturing-specific parts and directly copies over *buy* parts.



Except in specific cases (see preferences below), do not use this utility to update the MBOM. To update an MBOM, use an accountability check or an advanced accountability check.

You can create a manufacturing bill of materials (MBOM) from an engineering bill of materials (EBOM) automatically using the `me_create_mbom` utility. This utility creates a new MBOM root node, and then either:

- Copies over engineering parts that are designated as **Buy** or **Phantom**.
- Replaces the parts designated as **Make** with new manufacturing parts containing the same in-context ID (IDIC) as the engineering part for future comparison.

Example:

```
me_create_mbom -u=kj -p=kj -g=Engineering
-key="item_id=000159,object_type=Item" -revrule="Latest Working"
-mrevrule="Latest Working" -log=D:\workdir\create3.csv -usemfk
```

Caution:

Log off any interactive session that uses the same structures as the structures you update with the utility before you use the utility to avoid failures due to locking of objects in that user session.

The utility is affected by the values of these preferences:

- **MEMBOM\_Mirror\_MakeRules**

This preference specifies a BOM Line property, such as **bl\_item\_object\_type**, and property value pair that serve as input for the decision as to whether to create a manufacturing-specific part.

There are two ways to view defined BOM Line property names:

1. In the BMIDE, you can check the properties defined for the object BOM Line.
2. In Structure Manager, if you right-click on the column title line and choose Insert Column(s)..., you get a list of the defined BOM Line properties.

Note:

This preference accepts only a string value.

If you require a boolean or other attribute, you should write your own Business Modeler Framework extension to control a make or buy decision.

### Preference syntax

KEY:<BOMLinePropertyInternalName>,VALUE:<PropertyValue>,ACTION:<ActionValue>

- <BOMLinePropertyInternalName> is the property on which the value criterion will apply (e.g. "GCS CP").
- <PropertyValue> is the value used by the criterion.
- <ActionValue> is the action to undertake on the BOM node.

Valid values are **PartLink** (links a Design node to Part node) and **PartAssign** (links and assigns a Design node Under a Part node).

Note:

ACTION is only meant for Design/Part like structures. If more than one value for a given property can determine the action, the operator OR (|) can be used as follows: "KEY:<BOMLinePropertyInternalName>,VALUE:<PropertyValue1>,ACTION:<ActionValue1>|VALUE:<PropertyValue2>,ACTION:< ActionValue2>" (e.g. "KEY:GCS CP,VALUE:Make|VALUE:MakePart,ACTION:PartAssign|VALUE:MakeLink,ACTION:PartLink"). If there is no preference value, the item is shared between Engineering BOM (EBOM) and Manufacturing BOM (MBOM) structures.

A common criteria may be, for example, whether you make or buy a part. Any parts designated as **Make** or **Phantom** (for organizational purposes) in the EBOM are replaced by manufacturing-specific parts in the MBOM. In this case, the value of this preference may look like this:

KEY:Usage\_MakeOrPurchase ,VALUE:Make | VALUE: Phantom

You can use a custom property in this preference. If you do not assign any value, the utility carries over the parts from the EBOM directly.

**Note:**

You should write your own Business Modeler Framework extension to control a make or buy decision.

- **MEMBOM\_Mirror\_TypePrefixSuffix**

This preference specifies the item type for the new manufacturing-specific item and any prefix or suffix to add to the item ID of the part in the EBOM that is being replaced by a new part in the MBOM. If you use custom manufacturing business object types, you can add this type to the value of this preference.

The default value of `-usemfk` is false. If true, and only the object type is present in **MEMBOM\_Mirror\_TypePrefixSuffix**, use that type and the ebom item id for MFK value. If you set the command line option `-usemfk` to `true` and specify *only* the item type, you can enter multifield keys.

If your company uses naming rules for new items, you must ensure that the prefix that you assign in this preference corresponds to the naming rules for the manufacturing item. Alternatively, you can remove all naming rules for manufacturing items. A conflict with item naming rules leads to this error message:

```
ERROR on line:250 creating mbom
74006: Valid pattern is aannnnnn. creating mbom
74007: Supplied item_id value "M_qp130410" is invalid. creating mbom
7007: Invalid Tag - the requested object does not exist creating mbom
```

- **MEMBOM\_Mirror\_ReplaceMakeOnChange**

This preference specifies the behavior when, after creation of an MBOM, you change a make/buy property value from **Make** to **Buy**. At update, by default, Teamcenter simply appends the buy part to the existing structure without removing the make part already in the MBOM. If you set this preference to **TRUE**, at update, Teamcenter replaces the make part in the MBOM with the buy part from the EBOM, avoiding duplication of parts.

- **MEMBOM\_Mirror\_RemoveLineWithIDIC**

This preference is applicable to updating an MBOM with the utility. It specifies that if the in-context ID (IDIC) of a line in the MBOM is not found in the EBOM, or if a line in the MBOM has no IDIC, it is not removed at update if it is already stamped with an IDIC.

- **MEMBOM\_Mirror\_IDICOnChildren**

This preference determines whether the IDIC is retained on a make part in the MBOM when updating it using the utility after changing a make part to a buy part in the EBOM. When updating, if the **MEMBOM\_Mirror\_ReplaceMakeOnChange** preference is set to **FALSE**, both the original make part

and the buy part exist in the MBOM after updating. This causes the accountability check to display multiple matches. Removing the IDIC on the make part ensures that there is only one match in an accountability check. If the **MEMBOM\_Mirror\_ReplaceMakeOnChange** preference is set to **TRUE**, the make line is replaced with the buy line from the EBOM so this preference is irrelevant.

- **MEMBOM\_Mirror\_IgnoreRules**

This preference uses the keyword **ACTION** to skip an EBOM node or to traverse below an EBOM node while ignoring that EBOM node. The children of the traversed EBOM node become the children of the corresponding MBOM parent.

- **MEMBOM\_Mirror\_AlignProperties**

This preference controls which properties and attachments are copied from the EBOM to the MBOM by enabling the alignment of properties on **Make** items and revisions. The **Make** nodes are of different types than the original EBOM nodes.

The following examples of running the utility demonstrate its use and the differences in output caused by changing the utility arguments. The initial values of the preferences are as follows.

```
MEMBOM_MakeRules=KEY:Usage_Product , VALUE:Make | VALUE:Phantom
```

```
MEMBOM_RemoveLineWithIDIC=FALSE
```

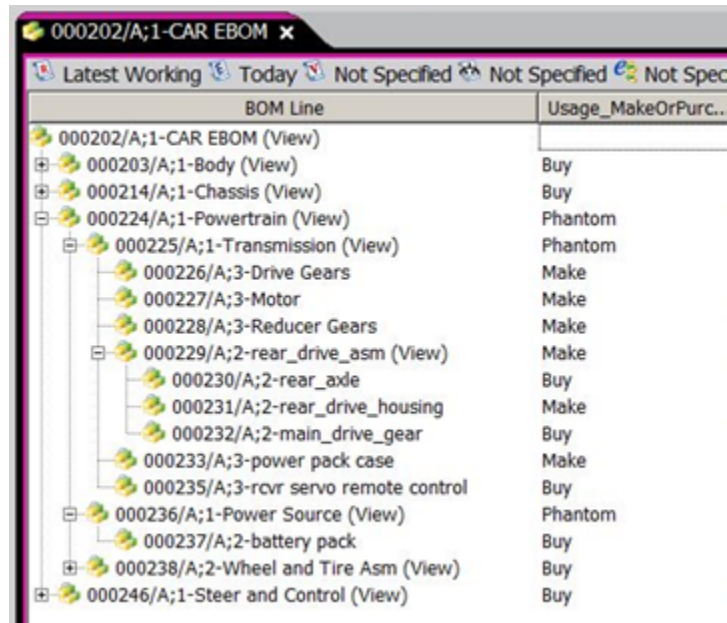
```
MEMBOM_TypePrefixSuffix=Company , ,
```

Note:

No prefix is used in this course due to the naming rule conflict.

## Creating an MBOM

For the following EBOM:

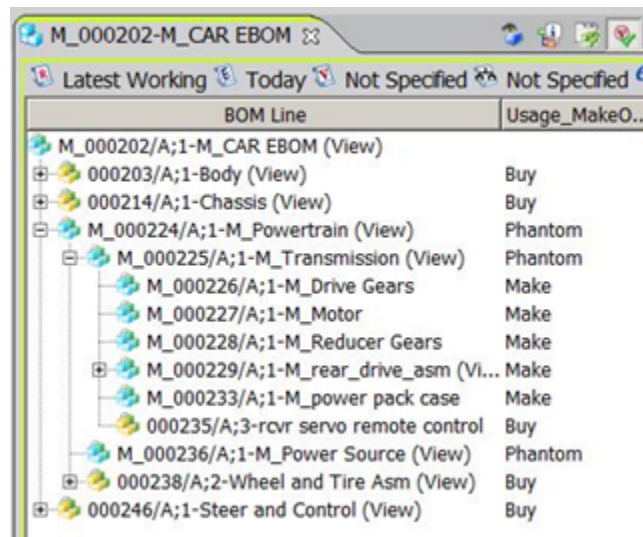


BOM Line	Usage_MakeOrPurc..
000202/A;1-CAR EBOM (View)	
000203/A;1-Body (View)	Buy
000214/A;1-Chassis (View)	Buy
000224/A;1-Powertrain (View)	Phantom
000225/A;1-Transmission (View)	Phantom
000226/A;3-Drive Gears	Make
000227/A;3-Motor	Make
000228/A;3-Reducer Gears	Make
000229/A;2-rear_drive_asm (View)	Make
000230/A;2-rear_axle	Buy
000231/A;2-rear_drive_housing	Make
000232/A;2-main_drive_gear	Buy
000233/A;3-power pack case	Make
000235/A;3-rcvr servo remote control	Buy
000236/A;1-Power Source (View)	Phantom
000237/A;2-battery pack	Buy
000238/A;2-Wheel and Tire Asm (View)	Buy
000246/A;1-Steer and Control (View)	Buy

Run the utility:

```
me_create_mbom -u=user-name -p=password -g=group -ebomroot=000202
-revrule="Latest Working" -mrevrule="Latest Working"
```

Produces the following MBOM:



BOM Line	Usage_MakeO...
M_000202/A;1-M_CAR EBOM (View)	
000203/A;1-Body (View)	Buy
000214/A;1-Chassis (View)	Buy
M_000224/A;1-M_Powertrain (View)	Phantom
M_000225/A;1-M_Transmission (View)	Phantom
M_000226/A;1-M_Drive Gears	Make
M_000227/A;1-M_Motor	Make
M_000228/A;1-M_Reducer Gears	Make
M_000229/A;1-M_rear_drive_asm (Vi...	Make
M_000233/A;1-M_power pack case	Make
000235/A;3-rcvr servo remote control	Buy
M_000236/A;1-M_Power Source (View)	Phantom
000238/A;2-Wheel and Tire Asm (View)	Buy
000246/A;1-Steer and Control (View)	Buy

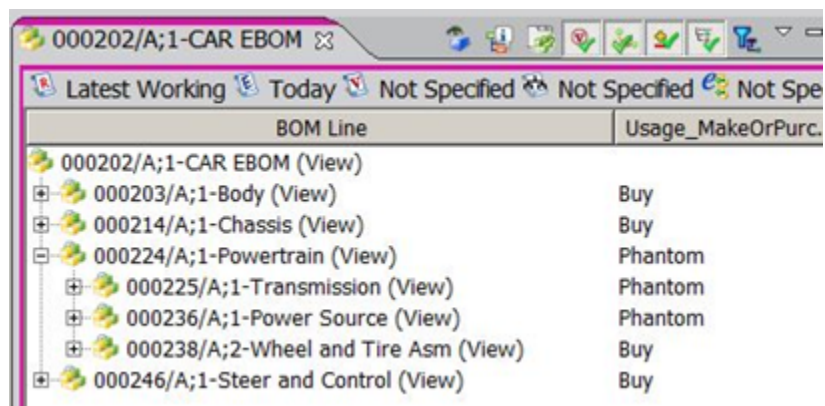
Note the following:

- The top-level node is replaced with a manufacturing-specific part.
- The make and phantom parts are replaced with new manufacturing-specific parts.

- The buy parts of the EBOM are directly copied over to the MBOM.
- An IDIC is generated for all parts except for those below a buy part.
- The in-context ID (IDIC) of the new parts is the same as the original part in the EBOM. This is true if the IDIC is already present in the EBOM part.
- When you right-click the top node of the EBOM or MBOM and choose **Link/Associate**→**Link Structures**, you can see that the utility has linked the two structures.

### Creating the MBOM to a specific level

For the following EBOM:

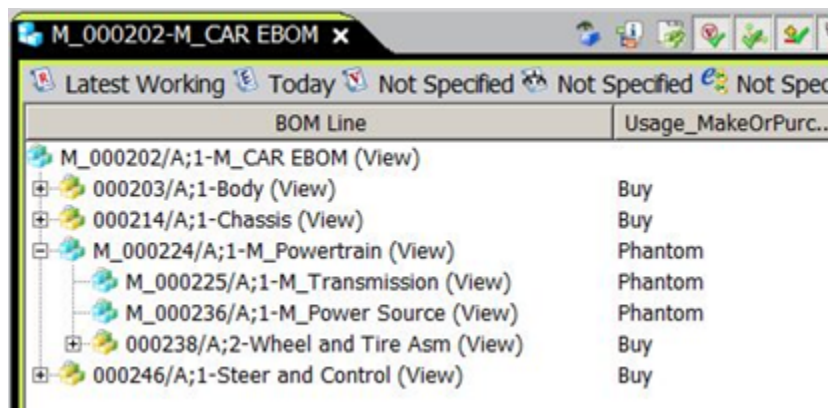


BOM Line	Usage_MakeOrPurc..
000202/A;1-CAR EBOM (View)	
000203/A;1-Body (View)	Buy
000214/A;1-Chassis (View)	Buy
000224/A;1-Powertrain (View)	Phantom
000225/A;1-Transmission (View)	Phantom
000236/A;1-Power Source (View)	Phantom
000238/A;2-Wheel and Tire Asm (View)	Buy
000246/A;1-Steer and Control (View)	Buy

Run the utility:

```
me_create_mbom -u=user-name -p=password -g=group -ebomroot=000202
-revrule="Latest Working" -mrevrule="Latest Working" -depth=2
```

Produces the following MBOM:



BOM Line	Usage_MakeOrPurc..
M_000202/A;1-M_CAR EBOM (View)	
000203/A;1-Body (View)	Buy
000214/A;1-Chassis (View)	Buy
M_000224/A;1-M_Powertrain (View)	Phantom
M_000225/A;1-M_Transmission (View)	Phantom
M_000236/A;1-M_Power Source (View)	Phantom
000238/A;2-Wheel and Tire Asm (View)	Buy
000246/A;1-Steer and Control (View)	Buy

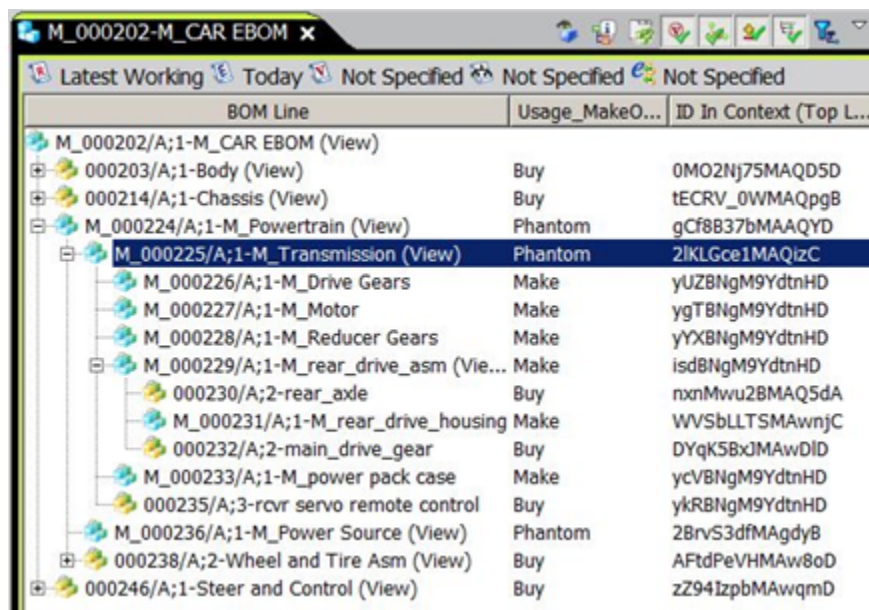
Note that the new MBOM contains only two levels of structure. The remaining levels in the EBOM are ignored.

### Updating the MBOM from a specific line

You now update the MBOM above, but only for the M\_000225 line and down, requiring the following utility command:

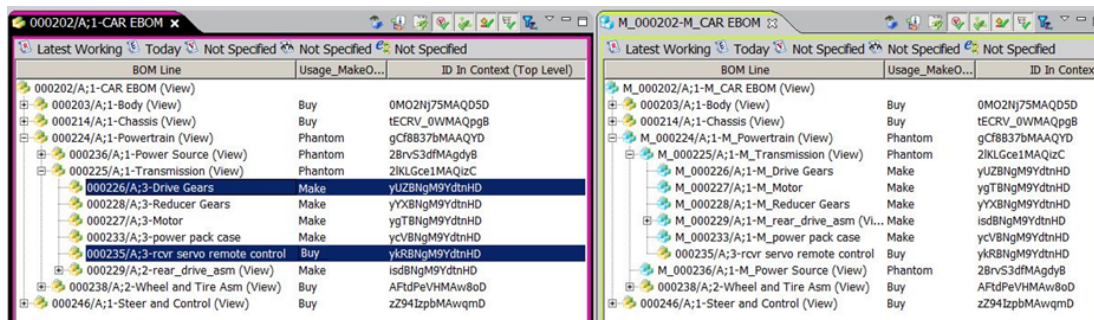
```
me_create_mbom -u=your-user-name -p=your-password -g=your-group
-ebomroot=000202 -scopeid=000225 -revrule="Latest Working"
-mbomroot=M_000202 -mrevrule="Latest Working"
```

The utility creates the following MBOM:



### Changing Make to Buy and Buy to Make in the EBOM

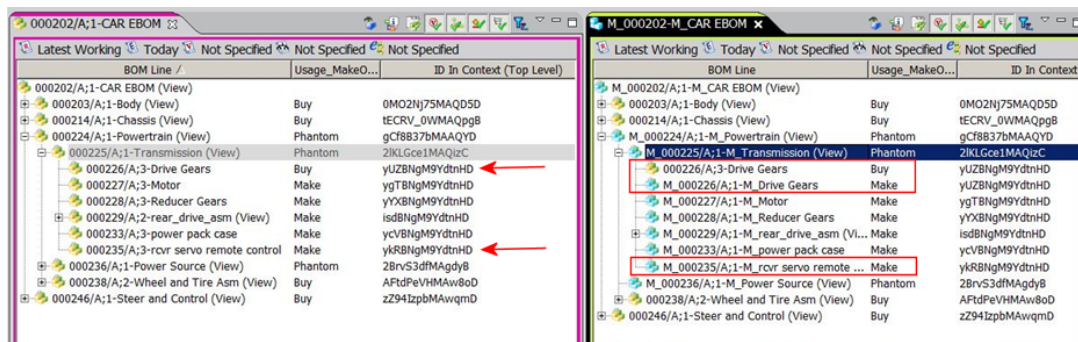
For the following EBOM and MBOM:



You change the **Make** value to **Buy** for the drive gears and change the **Buy** value to **Make** for the rcvr remote control in the EBOM and run the utility as follows:

```
me_create_mbom -u=user-name -p=password -g=group -ebomroot=000202
-scopeid=000225 -revrule="Latest Working" -mbomroot=M_000202
-mrevrule="Latest Working" -update
```

The results are as follows:



When the rcvr remote control part is changed from a **Buy** to a **Make**, the utility replaces the buy part in the MBOM during update. Buy to make behavior is not affected by preference settings. For the make to buy change (the drive gears), the behavior depends on two preferences:

- The **MEMBOM\_Mirror\_ReplaceMakeOnChange** preference is set to **FALSE**. The make part is not removed from the MBOM. The buy part is added to the structure.

The IDIC values on the make part are determined by the **MEMBOM\_Mirror\_IDICOnChildren** preference.

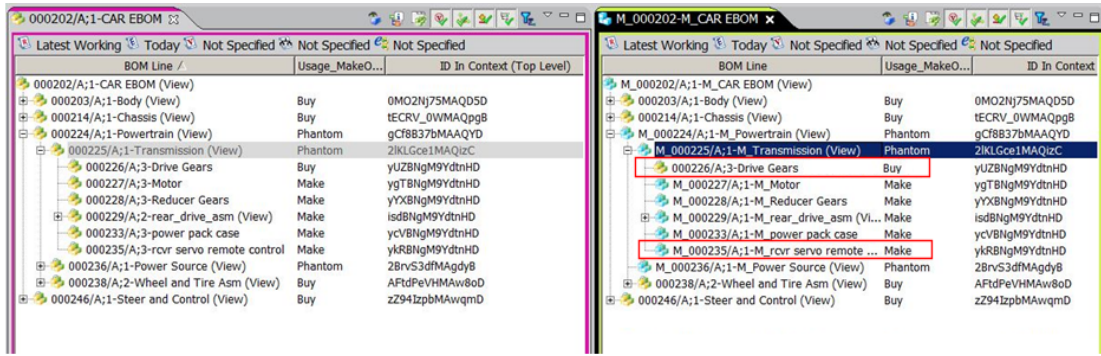
- If **MEMBOM\_Mirror\_IDICOnChildren** is set to **FALSE** (or does not exist), the IDIC on the make part in the MBOM is retained. Performing an accountability check between the EBOM and MBOM results in multiple matches as the MBOM contains both a make and a buy part with the same IDIC.
- If **MEMBOM\_Mirror\_IDICOnChildren** is set to **TRUE**, the IDIC on the make part in the MBOM is deleted and an accountability check displays the make part as missing in source.

#### Caution:

You must exercise caution when removing an IDIC on any part. This can cause problems in other areas, for example, if the part is consumed in an process structure, you lose the link to that line.

- The **MEMBOM\_Mirror\_ReplaceMakeOnChange** preference is set to **TRUE**.

The buy part (drive gears) replaces the make part. The result of the MBOM after update is as follows:

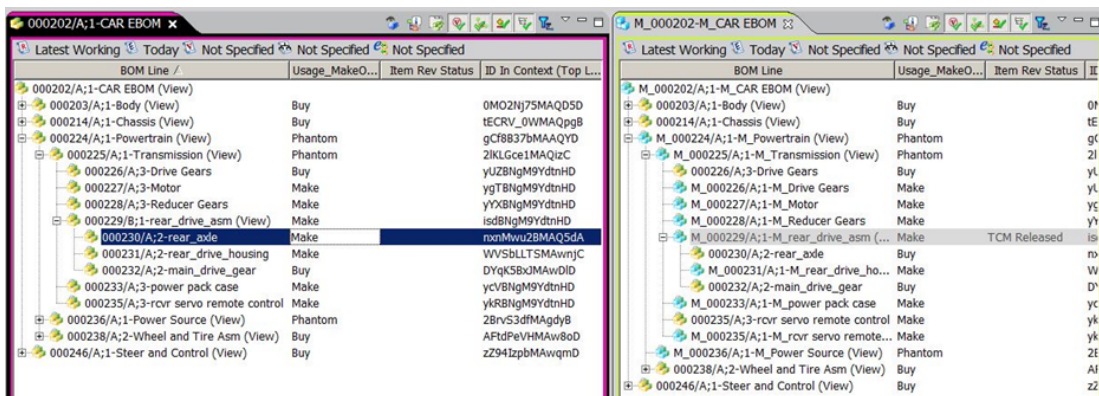


In summary, the behavior of the preferences is as follows:

<b>MEMBOM_Mirror_ReplaceMakeOnChange</b>	<b>FALSE</b>	The buy part is added to MBOM and the previous make part is kept in the structure but its IDIC is removed.
<b>MEMBOM_Mirror_IDICOnChildren</b>	<b>TRUE</b>	
<b>MEMBOM_Mirror_ReplaceMakeOnChange</b>	<b>FALSE</b>	The buy part is added to MBOM and the previous make part is kept in the structure but its IDIC is <i>not</i> removed.
<b>MEMBOM_Mirror_IDICOnChildren</b>	<b>FALSE</b>	
<b>MEMBOM_Mirror_ReplaceMakeOnChange</b>	<b>TRUE</b>	The buy part is added to MBOM and the previous make part is <i>removed</i> in the structure, irrespective of the value of <b>MEMBOM_Mirror_IDICOnChildren</b> .
<b>MEMBOM_Mirror_IDICOnChildren</b>	<b>TRUE/FALSE</b>	

### Updating a released node in the MBOM

In the following EBOM and MBOM, **M\_000229/A;1-M\_rear\_drive\_asm** is released and then the make/buy property on its child, **000230/A;2-rear\_axle**, is changed from **Buy** to **Make**.

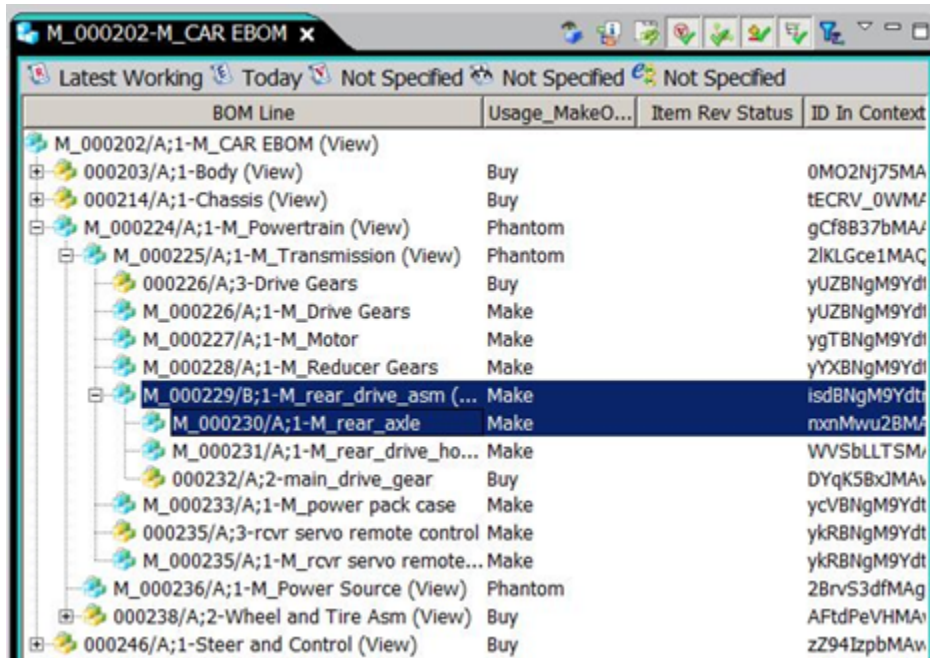


You revise the MBOM part so you have write access and run the update utility.

You run the utility with the **-actiononrelease** argument set to **2**.

```
me_create_mbom -u=your-user-name -p=your-password -g=your-group
-ebomroot=000202 -scopeid=000225 -revrule="Latest Working"
-mbomroot=M_000202 -mrevrule="Latest Working" -actiononrelease=2 -update
```

The results are as follows:



BOM Line	Usage_MakeO...	Item Rev Status	ID In Context
M_000202/A;1-M_CAR EBOM (View)			
000203/A;1-Body (View)	Buy		0MO2Nj75MA
000214/A;1-Chassis (View)	Buy		tECRV_0WMA/
M_000224/A;1-M_Powertrain (View)	Phantom		gCf8B37bMA/
M_000225/A;1-M_Transmission (View)	Phantom		2IKLGce1MAÇ
000226/A;3-Drive Gears	Buy		yUZBNgM9Ydt
M_000226/A;1-M_Drive Gears	Make		yUZBNgM9Ydt
M_000227/A;1-M_Motor	Make		ygTBNgM9Ydt
M_000228/A;1-M_Reducer Gears	Make		yYXBNgM9Ydt
M_000229/B;1-M_rear_drive_asm (...)	Make		isdBNgM9Ydt
M_000230/A;1-M_rear_axde	Make		rxnMvwu2BMA/
M_000231/A;1-M_rear_drive_ho...	Make		WVSbLLTSM/
000232/A;2-main_drive_gear	Buy		DYqK58xJMAv
M_000233/A;1-M_power pack case	Make		ycVBNgM9Ydt
000235/A;3-rcvr servo remote control	Make		ykRBNgM9Ydt
M_000235/A;1-M_rcvr servo remote...	Make		ykRBNgM9Ydt
M_000236/A;1-M_Power Source (View)	Phantom		2BrvS3dfMAg
000238/A;2-Wheel and Tire Asm (View)	Buy		AftdPeVHMAi
000246/A;1-Steer and Control (View)	Buy		zZ94IzpbMAw

The utility revises the part and replaces the old revision with the new one.

## Changing properties in the EBOM

To detect property changes in the EBOM, run an accountability check between the EBOM and MBOM. You can propagate any property differences using the accountability check mechanism.

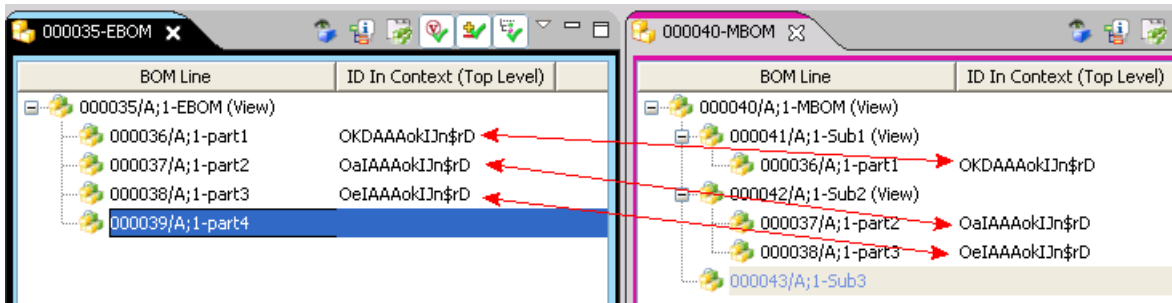
## Attachments

Attachments are always synchronized as the attachment is created as a reference on the new make part. Any changes to the attachments on the EBOM side are automatically reflected in the MBOM. If you add a new attachment to the EBOM, you must run the update to see it in the MBOM part.

## Reconciling EBOM and MBOM in a Multi-Site environment

The determination of equivalence between the engineering bill of materials (EBOM) and the manufacturing bill of materials (MBOM) is based on the in-context ID (IDIC, also referred to as the absolute occurrence ID). When you paste a part from the EBOM to the MBOM, Teamcenter generates an identical, unique absolute occurrence ID for the part on both the EBOM and the MBOM. This

mechanism is used when aligning structures to determine whether parts are missing in the source or target structures.



The ID is generated in the context of the top-level item. Even if you do not have write access to the top level of the EBOM, Teamcenter allows this ID to be generated and saved with the top-level item. If you are using Multi-Site and the EBOM is at the remote site, however, this mechanism does not work as Teamcenter must be able to check out the top level to generate the ID. A similar problem arises if you want to assign from an EBOM that exists on a remote site to a local process.

To enable a remote site to use the EBOM-MBOM mechanism in these situations, you can generate in-context IDs on the EBOM that are available when you paste parts to the MBOM. Use the **bom\_expand** utility to generate these IDs on the EBOM. For the EBOM shown in the figure, enter this command:

```
bom_expand -u=user-id -p=password -g=group -item=000035 -create_absocc_ids
```

Generate these IDs on a regular basis, for example, nightly, so that they are up-to-date when users assign parts from the EBOM to MBOM.

#### Caution:

Attaching an in-context ID to a line in a structure creates several new objects in the database. If you work with large amounts of data, this may have a detrimental affect on performance or scalability. You should only pregenerate these IDs if you use EBOM and MBOM in a Multi-Site environment.

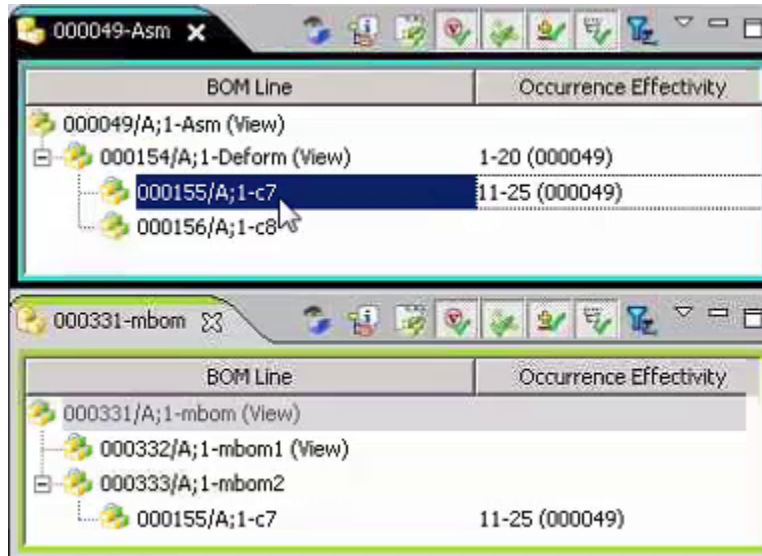
## EBOM to MBOM effectivity options

When end users are assigning BOM lines from an EBOM to a linked MBOM, you can set how the *unit-based* effectivity of the source EBOM is handled in the target MBOM after the assignments are complete. This option helps to avoid errors resulting from user attempts to manually enter the occurrence effectivities themselves.

The **MEEffectivityAssignmentMode** preference defines the assignment behavior from an engineering BOM (EBOM) to a manufacturing BOM (MBOM) for the BOM line property **bl\_occ\_effectivity**. The default preference value of **0** ensures that when an EBOM line is assigned to the MBOM, the assigned line inherits only the occurrence effectivity defined on the EBOM line, if such effectivity exists.

Example:

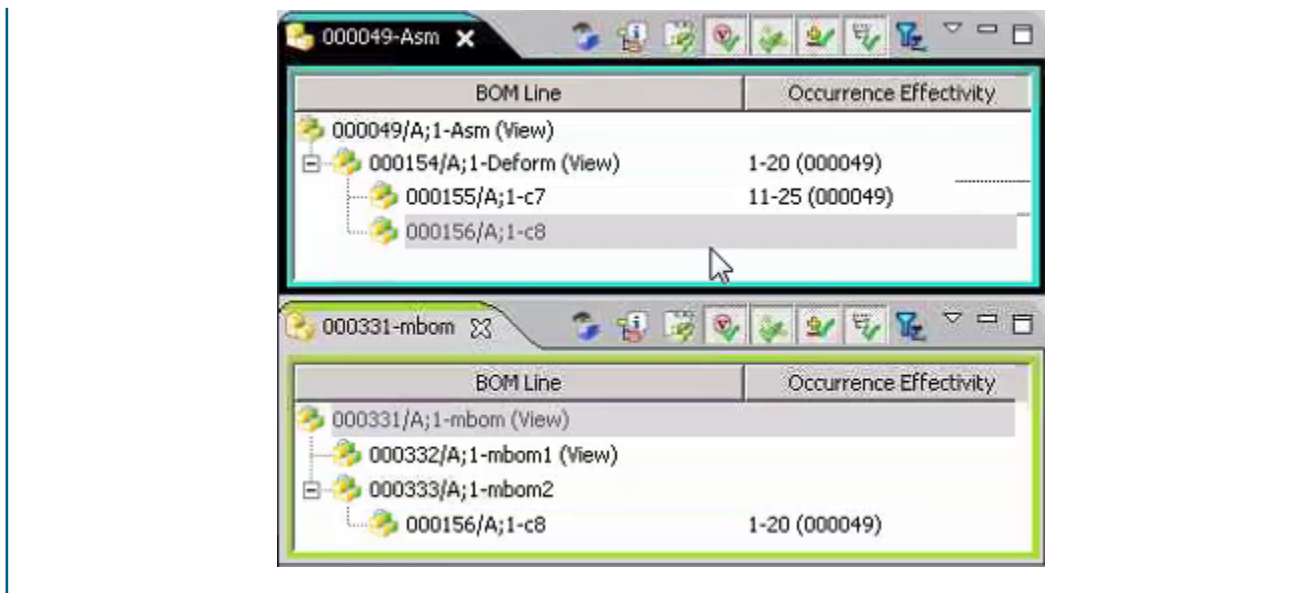
Assigning EBOM line **000155/A;1-c7** to the linked MBOM inherits its effectivity from that line, not from any line with a different effectivity higher in the structure.



Set the value of **MEEffectivityAssignmentMode** to **1** to make an assigned line with no effectivity inherit the effectivity of its parent. You must also add the **bl\_occ\_effectivity** value to the **MEAlignedPropertiesList** preference and ensure that no preference beginning with the text **MEAlignedPropertiesToExclude...** has the value **bl\_occ\_effectivity**.

Example:

Assigning EBOM line **000156/A;1-c8** to a linked MBOM results in the assigned line inheriting its effectivity from its parent line because it has no effectivity. If you assign EBOM line **000155/A;1-c7** to the MBOM, it retains its own effectivity of 11-25.

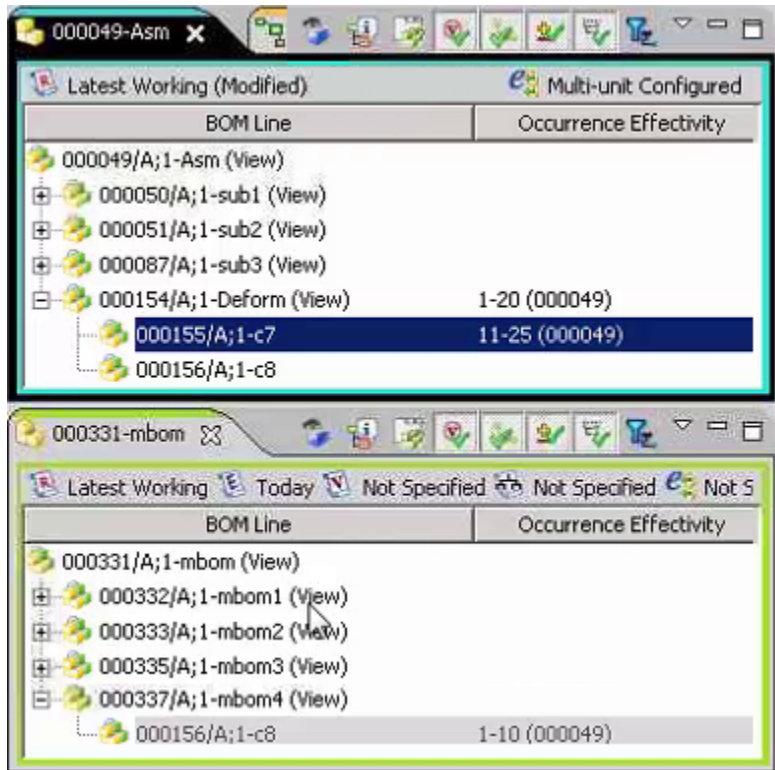


*Net effectivity* calculates the intersection of effectivities set on the path from the EBOM line to be assigned up the parent chain. Starting from the selected EBOM line, and including the EBOM line itself, net effectivity traverses up the parent chain until it finds the first parent with an explicitly defined occurrence effectivity. The occurrence effectivity defined on that parent becomes the MBOM line's net effectivity. If an effectivity is defined in the BOM window's revision rule, the computation intersects the parent's occurrence effectivity with that defined in the revision rule. The result of the intersection is the final net effectivity.

To apply calculated net effectivity behavior, set the value of **MEEffectivityAssignmentMode** to 2 and add the value **bl\_occ\_effectivity** to the **MEAlignedPropertiesList** preference. You must also ensure that no preference beginning with the text **MEAlignedPropertiesToExclude...** has the value **bl\_occ\_effectivity**, and turn on net effectivity calculations using the Business Modeler IDE.

#### Example:

The top line of the EBOM shown is associated with a multi-unit configured effectivity group with values of 1–10 (not visible). Line **000156/A;1-c8** has no effectivity, but its parent, **000154/A;1-Deform** has a value of 1–20. Net effectivity calculates the available range shared by the multi-unit configured effectivity group value (1–10) and the **000154/A;1-Deform** value (1–20), which is 1–10.



## Delegating, overriding, or copying line properties from MBOM to BOP

End users can override some delegated manufacturing bill of materials (MBOM) line properties in a bill of process (BOP). Delegated properties reference the MBOM and appear in the BOP on assigned (consumed) lines if the MBOM is loaded. To keep the BOP synchronous with the engineering bill of materials (EBOM), administrators can mark a subset of these properties so that MBOM values are delegatable using the **MEDelegatedOccurrenceProperties** preference.

By default, this preference delegates the following properties: **bl\_occ\_effectivity** (occurrence effectivity), **bl\_quantity** (quantity), **bl\_plmxml\_abs\_xform** (absolute transformation matrix), **bl\_occ\_name** (occurrence name), and **bl\_all\_notes** (all note types). If a property that was copied over in previous versions is not defined in the **MEDelegatedOccurrenceProperties** preference, that property is copied over.

Note:

- End users can edit the relative transform matrix in the BOP to align the absolute transformation matrix.
- By default, quantity is copied to the BOP. If end users delete a quantity value in the BOP, the quantity value is delegated from the MBOM.

Additional properties that can be delegated include **bl\_formula** (variant formula), **bl\_abs\_occ\_all\_ids** (ID in context, all levels), and **bl\_variant\_condition** (variant condition).

If the **MEDelegatedOccurrenceProperties** preference is absent, no properties are delegated. If your organization prefers to copy rather than delegate line properties from the MBOM to the BOP, the property names should be removed from the **MEDelegatedOccurrenceProperties** preference and added to the **MEAssignProperties** preference. Properties in **MEDelegatedOccurrenceProperties** take precedence over properties in **MEAssignProperties**.



# 13. Administering CAD BOM and part BOM alignments

## Part BOM requirements

To create a part BOM from a design BOM:

- The **MEDesignToBom\_Enabled** preference must be set to **true** (default = **false**).
- Part structures that are created in a part BOM using heavy alignment have parts with BOM view revisions (BVRs) that are not actually assemblies. Change any existing customizations to treat parts with BVRs as leaf lines if they are to be marked as end items.
- The design BOM items and structures must either be created in a CAD system or empty design items created by a manufacturing engineer as a work order for the designer.
- The design BOM must consist of design object types.
- The part BOM must consist of part object types.
- In the part BOM, add a precise clause to the top of the revision rules.

Example:

The **group** feature can be used to create the following clause: **Has Item Type (Design) {Precise}**.

To calculate rolled-up variant conditions from the design BOM and add the logically equivalent variant condition to the aligned part line, add the **bl\_condition\_tag** property to the **MEDesignToBomAlignmentProperties** preference.

To avoid errors during heavy alignments, if you are using the **TAllowedChildTypes\_<Part>** preference, you must modify it to allow design items. Similarly, if you use the **TAllowedParentTypes\_<Design>** preference, modify it to allow part items.

## Set up CAD BOM smart search criteria

While the system defines some search criteria to locate unlinked or broken alignments between a part BOM and CAD BOM, you can set the **MEDesignToBomFindDesignsSearchCriteriaProperties** preference to help end users search for CAD objects based on additional attribute criteria and then create heavy or light alignments based on search results.

For broken links, the system looks for the following in order.

1. Matching ID in context (IDIC).
2. Matching item ID.
3. Matching Absolute Transformation Matrix (ATM).
4. The **MEDesignToBomFindDesignsSearchCriteriaProperties** preference properties.

Example:

**AIE\_OCC\_ID** and **AIEAIE\_OCC\_NAME**.

If any of these do not match, the CAD object is excluded from the end user's search results. If the part object was never linked to the CAD object, the system looks for the following in order.

1. The **MEDesignToBomFindDesignsSearchCriteriaProperties** preference settings.
2. A matching ATM value.

Note:

If no previous alignment exists and there is a change to the CAD part (for example, the ATM has changed), the search may not locate the object.

## CAD and part BOM action handlers

Two action handlers are provided to make CAD bill of materials (BOM) design to part BOM alignment easier. Both parse the inputs and call USER ITKs. These ITKs can be customized.

### ME-update-part-alignment-on-design-change-AH

This action handler updates part revisions in a part BOM related to an input CAD BOM design revision.

- The target folder requires a new design revision.
- The reference folder can optionally contain a previous design revision. If a reference folder is not provided, the handler tries to find a previously linked design revision.
- Action handler mandatory argument: **-forceupdate**
- Allowed values:
  - 0 Error if part is not writable.
  - 1 Force update if part is not writable.
  - 2 Revise part and update if part is not writable.

```

/**
    @returns
    <ul>
    <li>#ITK_ok on success.
    <li>#POM_invalid_tag if either one of @p new_design_revision
and @p
        old_design_revision are invalid.
    <li>Possibly other errors.
    </ul>
*/
extern ME_API int USER_update_part_rev_from_design_rev(
    tag_t new_design_revision, /**<(I) The new design revision that
is to replace
    the old design revision.*/
    tag_t old_design_revision, /**<(I) The old design revision that
is to be replaced
    with the new design revision.*/
    unsigned int actionIfPartRevIsReleased /**<(I) Perform action.
Valid values are:
    <ul>
    <li>#ME_ERROR_IF_PART_RELEASED to return an error if the
part revision to
        be aligned is not modifiable.
    <br/>This is also the default value if an invalid
action is specified.
    <li>#ME_UPDATE_RELEASED_PART_REVISION to overwrite released
part revision on align.
    <li>#ME_REVISION_PART_REVISION to revise the non-modifiable
part revision and align.
    </ul>*/
);

```

## ME-update-part-bom-alignment-to-mature-design-AH

This action handler recursively aligns previously aligned part lines in a part BOM to a currently configured design revision in a CAD BOM that matches maturity criterion.

- The maturity criterion can be customized using **#USER\_is\_item\_rev\_mature**.
- If the **MEDesignToBom\_check\_maturity** preference value is **false**, the alignment considers the currently configured design revision as mature. The target folder requires a collaboration context containing one design scope line and one part scope line.

```

/**
    @returns
    <ul>
    <li>#ITK_ok on success.
    <li>#POM_invalid_tag if an invalid collaboration context, part

```

```

        or design in @p cc, @p design_scope_line or @p
part_scope_line is invalid.
        <li>Possibly other errors.
    </ul>
    */
extern ME_API int USER_update_part_bom_alignment_to_mature_design(

```

## Use the CBA 2.0 to CBA 3.0 `me_migrate_design_ebom` migration utility

In Teamcenter 13.2 and above, you can use the CAD-BOM Alignment (CBA) utility to migrate CBA 2.0 design or CAD bill of material (BOM) and part BOM alignment to the new CBA 3.0 CAD part alignment model, which is publish-linked based. After you migrate from CBA 2.0 to 3.0, you can view the CAD BOM and part BOM side by side in Active Workspace to review the alignment of parts and visualize the structures.

The following migration cases are supported:

- Heavy alignment
- Light alignment
- Multiple design (CAD) to one-part alignment (ND1P)
- Deformed parts (override jts and override transforms)

CAD BOM ← CBA 2.0 → EBOM → MBOM

CAD BOM ← CBA 3.0 → EBOM → MBOM

Set the following preference values before running the `me_migrate_design_ebom` utility:

- **MEDesignToBom\_Enabled=false**
- **FND0\_ENABLE\_DESIGN\_BOM\_ENGINEERING\_BOM\_ALIGNMENT=true**

For details on the `me_migrate_design_ebom` utility, use the `-h` command.

# 14. Configuring in-context IDs (IDIC)

## Understanding equivalence and in-context ID (IDIC)

Teamcenter has several features that depend on finding parts that are considered *equivalent*. This equivalency is often dependent on the in-context ID (IDIC), also known as the *absolute occurrence ID*. The in-context ID is a unique identifier, generally alphanumeric, that determines if two parts or processes are equivalent.

You can view the in-context ID by displaying the **ID in Context Top Level** column. It is usually automatically generated, and you should exercise caution in modifying its value.

The following features depend on the in-context ID:

- Reconciling and engineering bill of process (EBOM) with a manufacturing bill of process (MBOM)
- Finding products in processes that are not yet expanded
- Running an accountability check to verify line equivalence
- Assigning parts to a cloned process structure
- Comparing structures to intermediate data captures

An in-context ID is automatically generated in the following situations:

- When you assign a bill of materials (BOM) to a BOM or a bill of process (BOP) to a BOP
- When you assign a BOM to a BOP or enterprise BOP (EBOP) structure if the **MECopyIdInContextToAssignedLine** preference is set to true

An in-context ID is not generated when you assign from an EBOP structure to an EBOP structure.

### Caution:

If a line has an in-context ID and you cut it and paste the same line back into the structure, Teamcenter creates a new occurrence without an in-context ID. Therefore, any of the above comparisons based on the in-context ID no longer work.

## Setting up the search for components in other structures

You can select one or more objects in the following areas of the user interface and search for them in a different structure.

- Lines in a view
- Consumed parts in a process
- An object in the **Navigation** pane (**Favorites, History, Open Items**)
- Objects in the **Home** folder
- Objects in the **My Worklist** folder
- Items attached to a Teamcenter mail
- Lines in the **Summary** view
- Objects in the **Graphics** view

Whether you need to expand a target structure before choosing the **Find in All Visible Views** menu command depends on the settings of the following two preferences:

**MECopyIdInContextToAssignedLine**  
**MECopyIdInContextLowerLevels**

Note:

Whether you need to expand a structure before a found component can be highlighted within it depends on the value of the **MEEExpandToSelection** preference. If this preference is set to **false**, you must manually expand the target structure to highlight lines found within that structure.

## Understanding ID in context and assigned products

When assigning a product to a process or copying processes or operations from one process structure to another, Teamcenter can assign the same ID in context (IDIC) to the structure lines in the source and target structure. Doing so provides a method for Teamcenter to find the assigned objects in the other structure, for instance, when using the **Find in All Visible Views** menu command or running an accountability check.

This mechanism is controlled by two preferences:

- **MECopyIdInContextToAssignedLine**

If set to **True**, Teamcenter copies any existing in-context IDs to the target structure when assigning a product to a process or copying a process or operation from one structure to another. If an in-context ID does not exist, Teamcenter creates one on the source and copies it to the target object. Any generated IDs are unique within the source and target structures.

If set to **False**, no in-context ID is copied.

### • MECopyIdInContextLowerLevels

If set to **True**, Teamcenter copies any existing in-context IDs of consumed items under subprocesses and suboperations to the target structure when copying a process or operation from one structure to another. If an in-context ID does not exist, Teamcenter does *not* create one.

If set to **False**, no in-context ID gets copied for the consumed items under subprocesses and suboperations, even if one is present.

Note:

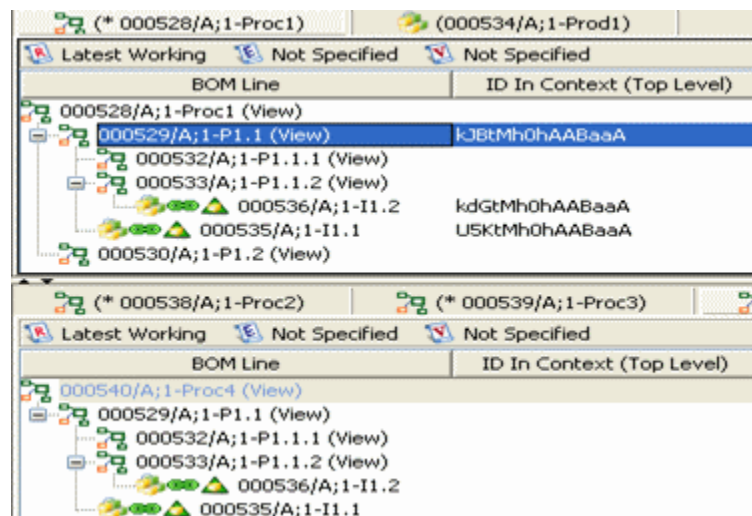
The **MECopyIdInContextLowerLevels** preference is considered only if **MECopyIdInContextToAssignedLine** is set to **True**.

In the following example, if you copy process **P1.1** from **Proc1** in the upper pane to **Proc4** in the lower pane:

- Teamcenter only generates an in-context ID for the process that is being copied. It never generates an in-context ID for subprocesses, suboperations, or their consumed items, regardless of the settings for these preferences.
- If **MECopyIdInContextToAssignedLine** = **false** and **MECopyIdInContextLowerLevels** = **false**, Teamcenter does not copy any in-context ID.

Note:

For **P1.1**, the ID is not generated at the source if it is not present (in the following example, it is already generated).



After copying, under the **Proc4** process, note that there are no in-context IDs in the **ID In Context (Top Level)** column.

- If **MECopyIdInContextToAssignedLine = true** and **MECopyIdInContextLowerLevels = false**, the in-context ID for **P1.1** and **I1.1** only are copied. The existing in-context ID for **I1.2** is not copied because **I1.2** is the consumed item of a lower level.

Note:

For **P1.1**, the in-context ID is generated at the source if it is not present and then copied. For **I1.1**, the ID is not generated if it is not present.

The screenshot displays two SAP BOM Explorer windows. The top window is for process view (\* 000528/A;1-Proc1) and the bottom window is for (\* 000538/A;1-Proc2). Both windows show a BOM structure with columns for 'BOM Line' and 'ID In Context (Top Level)'. In the top window, the ID for P1.1 is kJBtMh0hAABaaA, I1.1 is USktMh0hAABaaA, and I1.2 is kdGtMh0hAABaaA. In the bottom window, the ID for P1.1 is kJBtMh0hAABaaA and I1.1 is USktMh0hAABaaA. The I1.2 item is not present in the bottom window's BOM structure.

BOM Line	ID In Context (Top Level)
000528/A;1-Proc1 (View)	
000529/A;1-P1.1 (View)	kJBtMh0hAABaaA
000532/A;1-P1.1.1 (View)	
000533/A;1-P1.1.2 (View)	
000536/A;1-I1.2	kdGtMh0hAABaaA
000535/A;1-I1.1	USktMh0hAABaaA
000530/A;1-P1.2 (View)	

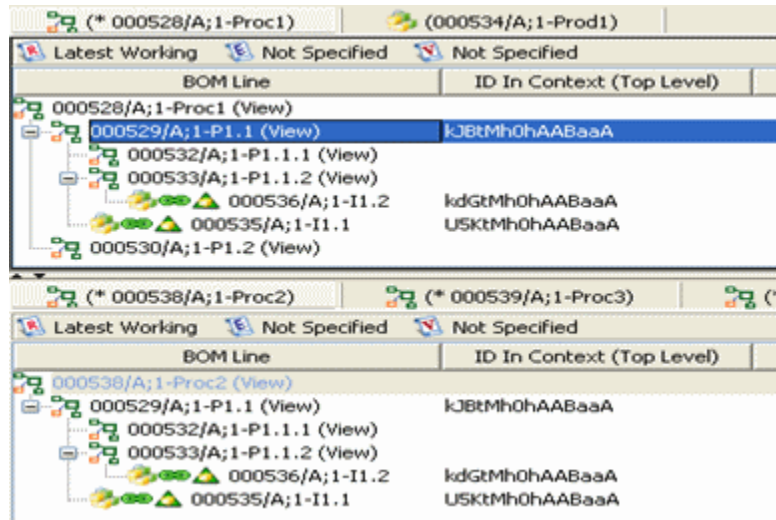
  

BOM Line	ID In Context (Top Level)
000540/A;1-Proc4 (View)	
000529/A;1-P1.1 (View)	kJBtMh0hAABaaA
000532/A;1-P1.1.1 (View)	
000533/A;1-P1.1.2 (View)	
000536/A;1-I1.2	
000535/A;1-I1.1	USktMh0hAABaaA

- If **MECopyIdInContextToAssignedLine = true** and **MECopyIdInContextLowerLevels = true**, the IDs for **P1.1**, **I1.1**, and **I1.2** are all copied.

Note:

For **P1.1** the ID is generated at the source if it is not present and then copied. For **I1.1** and **I1.2**, if the ID is not present, it is not generated, but if it is present, it is copied.



## Configuring the generation of smart IDICs

Different solutions in Teamcenter are based on the in-context ID (IDIC). It is useful to generate an IDIC on a structure to be certain that these IDs are available for subsequent actions, for example:

- You want to assign parts to a relevant process or operation where the parts in the BOM are changing often resulting in a loss of occurrence information and you want to avoid having to assign the new parts to the same process or operation manually.
- You want to clone an existing structure and automatically assign the same parts to the new cloned structure.
- You want to compare an intermediate data capture to the structure from which it was made to see what changes have occurred since you created the intermediate data capture.
- You want to compare an EBOM to an EBOM or a process to a process.
- When performing EBOM-MBOM maintenance, for example, by overwriting existing IDIC values in a BOM to ensure consistency.

You can automate the generation of IDICs for BOM lines to be based on the concatenation of other properties of the BOM line. The combination of these properties results in a unique string that Teamcenter assigns as an IDIC.

## Generate smart IDICs

- Define the properties that are combined to create the IDIC in the **MEIdGenerationPropertySetting** preference.

These can be any single-valued BOM line property. If the property value is empty, Teamcenter uses that empty value. Two typical properties that you may combine are **Usage Address** and **Position Designator**. These are used when resolving logical assignments in enterprise bill of process structures.

- Define tolerances for translation and rotation in the following preferences:
  - **ME\_angular\_tolerance**
  - **ME\_translation\_tolerance**

Note:

If you use a CAD integration with Catia or other CAD application and use a bill of materials (BOM) to BOM or BOM to bill of equipment (BOE), updates sometimes cause small BOM CAD tolerance variations so that parts in the model show as incorrectly changed during advanced accountability checks. To reduce the number of outliers during such checks, increase the **ME\_angular\_tolerance** and **ME\_translation\_tolerance** values.

- Specify whether the **MEIdGenerationPropertySetting** preference is to be used to generate in-context IDs during the assign action by setting the **MEIdGenerationOnAssignEnabled** preference to **true**.

Caution:

You may notice performance issues when setting this preference.

Alternatively, you can:

- Run the **me\_stamp\_ids** utility to generate IDICs for an entire structure, based on a closure rule.
- Create a workflow process based on the **ME-stamp-ids-AH** action handler to automate the process from the user interface.

When using the utility or action handler, you can specify whether Teamcenter allows duplicate IDICs in the **MEIdGenerationAllowDuplicates** preference.

## Configuring the IDIC mechanism when using shared assemblies

Some engineering bills of material (EBOMs) contain *shared* assemblies. The same assembly is used multiple times in the EBOM but only one is configured in and assigned to the manufacturing BOM (MBOM). If you set the **MEIdGenerationInSharedAssembly** preference to **true**, at the time of assignment, Teamcenter searches the EBOM for all other *visible* (configured in by revision rule or any of the **Show Unconfigured...** buttons) instances of the same assembly and stamps them with the same IDIC.

## Stamp missing IDIC strings

You can use the `me_fix_idic` utility to stamp ID in context (IDIC) strings from a stable manufacturing bill of materials (MBOM) structure to an engineering bill of materials (EBOM) structure in which the IDIC strings are somehow deleted. You can also use the utility to stamp IDICs from a product or plant structure to a process structure.

### Note:

- The command may take some time to complete its task.
- After the command completes you must restart Manufacturing Process Planner to see the changes.

### Warning:

IDIC values are not stamped if:

- There are multiple empty IDICs in the EBOM or assigned lines in the process structure for a given **itemid** and **propertyset** (that is, for the same **itemid** and the transform for multiple lines with no IDIC).
- An IDIC on a target line exists but is different from the source IDIC, even though the **itemid** and **propertyset** values match.



# 15. Administering the import and export of manufacturing structures

## Configuring PLM XML export

It can take hours to export large structures using PLM XML export. You can improve the performance of these jobs by setting the **PLMXML\_Pre\_Expand\_BOP\_Rule\_Name** preference to the **PreExpandProcess** closure rule. This feature works by pre-expanding the process structure by closure rule prior to PLM XML traversal. The pre-expansion performs bulk loading of BOM lines, reducing the number of calls to the Oracle server. The closure rule can be customized for maximum performance. The reduction in Oracle calls reduces the load on the rest of the system that, in turn, speeds up concurrent jobs and increases the capacity of the system to run more jobs simultaneously without sacrificing performance.

You can configure the **PreExpandProcess** closure rule to suit your needs.

## Configuring custom primary objects for use with DIPA

When you create custom primary objects such as custom plant BOPs, processes, process areas, lines, operations, and items, you must first make configuration changes using dynamic in-process assembly (DIPA) preferences and closure rules to ensure that they import correctly.

- For custom root (top level) items, the preference **MEDynamicIPARootItemTypes** needs to include all object types included in the assembly. This preference specifies the types of root items allowed in a DIPA (default = **Item**).
- For custom processes, do one of the following:
  - Change the preference **MEDynamicIPATraversalClosureRule**, to point to a new closure rule to be used for traversal of the process structure to calculate the DIPA.
  - Modify the **DynamicIPAProcessOnlyWithPA** preference to allow the custom process object types.
- To recursively generate the DIPA, modify the **MEDynamicIPARecursiveCreateProcessTypes** preference to include any new custom Bill of Process (BOP) object types.

## Configuring the import and export of dynamic in-process assemblies

To ensure that scope flows are imported and exported along with dynamic in-process assemblies (DIPAs), you must make some changes to the PLM XML transfer modes:

- To export DIPAs:

- If you use the **ConfiguredDataFilesExportDefault** transfer mode, you must add the following closure rule:

CLASS	BOMLine	CLASS	BOMLine	PROPERTY	bl_predecessor_lines	TRAVERSE_AND_PROCESS	\$bom_cb != "1" && \$depth != "0"
-------	---------	-------	---------	----------	----------------------	----------------------	-----------------------------------

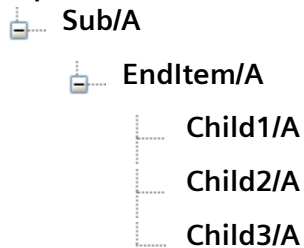
- If you use the **ConfiguredDataExportDefault** transfer mode, you do not need to make any changes.
- To import DIPAs:
  - If you use the **incremental\_import** transfer mode, you do not need to make any changes.
  - If you use any other transfer mode, including **ConfiguredDataImportDefault**, you must add the following closure rule:

CLASS	Document	CLASS	GeneralRelation	CONTENT	*	DO	
-------	----------	-------	-----------------	---------	---	----	--

## Configuring the export of end item assemblies

When using PLM XML export and the **bomwriter** utility, the export of lines under end item assemblies is controlled by closure rules. If you want to suppress the export of the lines under an end item assembly, apply conditional clauses to the closure rule clause that traverses from parent line to child lines. Ensure that there are no other overlapping clauses that export the children. The conditional clause impacts the objects that are exported. For example, in the following structure:

### Top



- If you want to export **Top** and **Sub** and suppress everything from the **EndItem** level (including **EndItem**), use the following conditional clause:

```
SECONDARY.fnd0bl_is_mono_override!= "1"
```

- If you want to export **Top**, **Sub**, and **EndItem** and suppress everything below the **EndItem** level (but not **EndItem**), use the following conditional clause:

```
PRIMARY.fnd0bl_is_mono_override!= "1"
```

The following is an example of a complete closure rule clause:

```
CLASS.BOMLine:CLASS.BOMLine:PROPERTY.bl_child_lines:PROCESS+  
TRAVERSE:PRIMARY.fnd0bl_is_mono_override != "1"
```



# 16. Configuring briefcase exchange of collaboration context

## Key preferences for configuring briefcase exchange of collaboration context

The following are important preferences for configuring a successful briefcase exchange of a collaboration context:

- **MEDefaultExportTransferOptionSet**

Specifies the list values to be made available in the **Export to Briefcase** dialog box transfer option set drop-down list. The values displayed are dependent on the values in the **MEDefaultExportTransferOptionSet** preference and the values of the transfer mode in the **TC\_gms\_export\_default\_transfermode** preference.

If **TC\_gms\_export\_default\_transfermode** has a value of **TIEExportDefaultTM**, there are two cases:

- Only the values that are specified in this preference and have a transfer mode value of **TIEExportDefaultTM** are displayed in the **Export to Briefcase** dialog box transfer option set drop-down list.
- If none of the transfer option set names specified in this preference have a transfer mode value of **TIEExportDefaultTM**, all transfer option sets with a transfer mode value of **TIEExportDefaultTM** are displayed in the **Export to Briefcase** dialog box transfer option set drop-down list.

Default values for the transfer option set drop-down list are:

- **MfgConfiguredExportDefault** — use this for briefcase configured export.
- **MfgUnconfiguredExportDefault** — use this for briefcase unconfigured export.

Note:

The first value specified in this preference defaults to the selected transfer option set in the **Export to Briefcase** dialog box.

- **MECCBriefcaseExportWithMultipleVariantRules**

Enables the briefcase export of a collaboration context containing a product structure with multiple variant rules. The default value is false.

- **MERuleForBriefcaseExport**

Defines the rule to govern creation of an appearance path node (APN) for the BOM lines during briefcase export of a collaboration context. This helps the recipient of exported data perform certain operations, such as the consumption of product to bills of process (BOPs) where if a particular object does not have ownership, Teamcenter does not allow it to create an APN there. Therefore, in such cases, it can use the created APN during export using this preference.

The default values for this preference are:

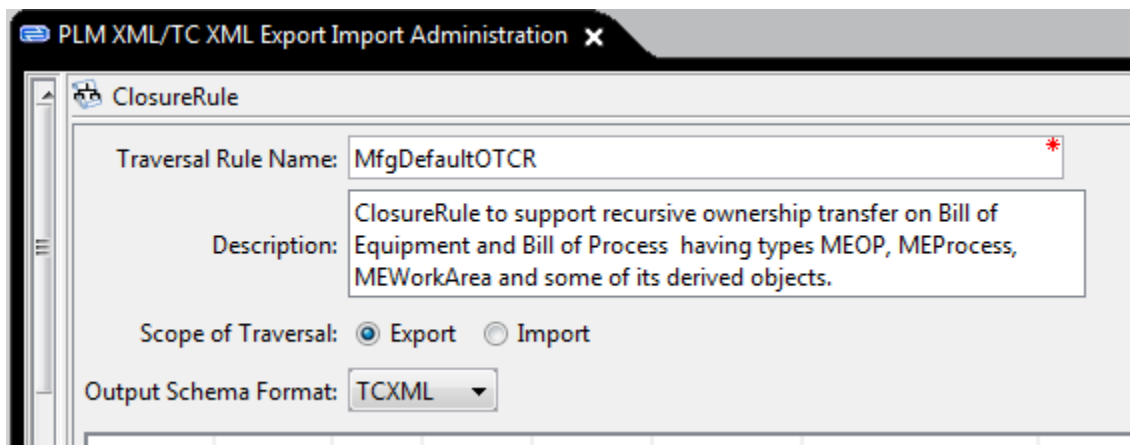
- **Type:Item; ClosureRule:; Lines:AllLines;**
- **Type:Workarea; ClosureRule:; Lines:AllLines;**
- **Type:Mfg0MEPlantBOP; ClosureRule:; Lines:AllLines;**
- **MEAutomaticOwnershipTransferClosureRules**

Specifies the closure rule to be used in recursive ownership transfer of export to briefcase functionality for collaboration context objects. The default value is **MfgDefaultOTCR**.

## Configuring recursive ownership transfer

The closure rule for defining recursive ownership transfer behavior can be specified in the **MEAutomaticOwnershipTransferClosureRules** preference. By default, it is governed by a closure rule named **MfgDefaultOTCR**. You can either modify it or create a new one.

The closure rule governing recursive ownership transfer should be based on BOM line to BOM line traversal. Do not use any persistent object for defining traversal for recursive ownership transfer. For example, the clause shown allows traversal from the parent BOM line to child BOM line when the parent is of type **MEProcessRevision** and the child is either **MEProcessRevision**, **MEOPRevision**, or **MEWorkAreaRevision**.



Primary ...	Primar...	Seco...	Secon...	Relatio...	Related Pr...	Action Type
CLASS	BOMLine	CLASS	BOMLine	PROPERTY	bl_child_lines	PROCESS+TRAVERSE

Conditional Clause

```
PRIMARY.bl_line_object_class=="MEProcessRevision" && (SECONDARY.bl_line_object_class=="MEProcessRevision" || SECONDARY.bl_line_object_class=="MEOPRevision" || SECONDARY.bl_line_object_class=="MEWorkAreaRevision")
```

In the example, if a parent line of type **MEProcessRevision** is marked for ownership transfer, any child line with types **MEProcessRevision**, **MEOPRevision**, or **MEWorkAreaRevision** are considered for ownership transfer as well when the recursive ownership check box is selected while exporting the collaboration context.

## Configuring automatic APN creation during export

This functionality is based on the **MERuleForBriefcaseExport** preference. By default, APNs are created for all structures present in a collaboration context being exported, but it can be configured to create APNs as per requirement.



# 17. Administering GD&T symbols

## Displaying GD&T symbols and rich text in forms

You can configure any form to contain geometric dimensioning and tolerance symbols or rich text in forms. It is possible to include geometric dimensioning and tolerance symbols in any box of a form but, in practice, you would restrict them to fields intended for engineering data. For example, a box titled **Work Instruction** is a good candidate for geometric dimensioning and tolerance symbols, while a box titled **Quantity** is not. To define fields that may contain geometric dimensioning and tolerance symbols, you add their names to the **GDT\_properties** preference. Once a box is added to this preference, a user can add geometric dimensioning and tolerance symbols to the box in any instance of the form.

You can activate geometric dimensioning and tolerance symbols for any type of form, including master forms for a Teamcenter object. You can apply normal Access Manager rules to any form that contains geometric dimensioning and tolerance symbols.

Users may only edit or add geometric dimensioning and tolerance symbols or rich text in a box after clicking the edit button next to the relevant box. They may then enter symbols, control characters or font formatting from the buttons provided or the keyboard. When editing or additions are complete, the user saves changes and leaves edit mode. This feature is available in any application that allows viewing of forms, including Manufacturing Process Planner, Part Planner, and Multi-Structure Manager. Detailed usage information is given in the guides for the applications that support this feature.

You can print and view reports that are generated from forms that contain geometric dimensioning and tolerance symbols or rich text. You can generate forms in two formats:

- HTML

Teamcenter generates a PLM XML file from the process structure, which is parsed, interpreted with XSL style sheets, and converted to HTML files.

- PDF

Teamcenter generates a PLM XML file from the process structure, which is parsed, interpreted with XSL style sheets, and converted to Formatting Object (FO) files. The information must then be converted to PDF format by a third-party formatting tool.

## Configure GD&T symbols and rich text in forms

Activate GD&T symbols and rich text in forms by implementing XML style sheets.

**Note:**

This feature is only supported on Microsoft Windows systems.

Ensure you have a supported browser installed on your workstation if you want to view or print HTML reports of information containing GD&T symbols.

1. Prepare the names of GD&T and rich text properties as follows:
  - a. In My Teamcenter, choose **Edit→Option→Search** and find the **GDT\_properties** preference. If the preference does not exist, create it at the **SITE** level or above.
  - b. For this preference, enter the names of the GD&T and rich text form fields you want Teamcenter to recognize in the **Values** box, for example, **work\_instruction** and **tolerance**.
  - c. Similarly, create a new preference at the **SITE** level or above called **GDT\_formtypes**, and enter a list of form type names as the value of this preference. Teamcenter recognizes these form types and, if a form of one of these types has a GDT box (that is, the name of the box is defined in the **GDT\_properties** preference), it renders this field as a GD&T/rich text box.

You have now defined names of GD&T/rich text attributes (*fields*) and form types. By default, Teamcenter renders GD&T and rich text fields as specified by the **GDT\_properties** and **GDT\_formtypes** preference entries. You can also use the XML style sheet mechanism to override these settings, as described in step 4.

2. Create the form types that can display geometric dimensioning and tolerance symbols or rich text, as follows:
  - a. Start the Business Modeler IDE and create new classes that have attributes listed in the previous step under **POM\_object**. For example, you might create a new class called **DemoGdtForm**. Define the attributes for GD&T and rich text as **String** type with the maximum length possible.
  - b. In the Business Modeler IDE, create new form types that use the class you created as the underlying POM class. Similarly, you can also create item master forms or item revision master forms to support GD&T and rich text in those master forms.
3. Copy the **gdtex.ttf** font file from the rich client installation directory to the **FONTS** directory of your Microsoft Windows system.

**Note:**

You must install this font file on every workstation where HTML are generated, viewed or printed.

4. Define an XML style sheet for the form type that contains GD&T or rich text fields. Enter **gdtpanel** as the rendering hint of the intended GD&T and rich text fields. If you do not give a rendering hint,

Teamcenter uses the default renderer (it checks if a box contains GD&T or rich text by looking at the preference settings).

The following XML example shows a style sheet for a form. In this example, **comments** and **tolerance** are rendered GD&T or rich text, while **work\_instruction** is rendered as a text area. These rendering hints override any conflicting preference settings.

```
<rendering>
  <page title="General" titleKey="General" format="TwoColumn">
    >firstcolumn>
      <property name="Quantity" modifiable="false"/>
      <property name="comments" renderingHint="gdtpanel" modifiable="true"/>
      <separator/>
      <property name="tolerance" renderingHint="gdtpanel" modifiable="false" />
    </firstcolumn>
    <secondcolumn>
      <property name="work_instruction" renderingHint="textarea"
modifiable="false"/>
    </secondcolumn>
  </page>
  <page title="Reservation" titleKey="Reservation">
  </page>
  <page title="Project" titleKey="Project">
  </page>
  <page title="All" titleKey="All">
  </page>
</rendering>
```

5. Verify you can now display and edit geometric dimensioning and tolerance symbols or rich text on the appropriate forms and can export the data into reports.

**Note:**

If Teamcenter indicates it cannot find the **corojdk11.dll** library, locate this file on the Teamcenter installation CD, and copy it into your system path.

If the display of GD&T and rich text becomes unstable when you type quickly, increase the value of the **EDITOR\_UPDATE\_LATENCY** variable in the **com\teamcenter\racl\form\gdt\gdt.properties** file from the default value of 50 and restart Teamcenter. This instability occurs on slower machines.



# 18. Setting up rich client reports

## Creating reports

You can create reports in the rich client and attach them to appropriate nodes in your process structures.

Note:

3D PDF report generation is not supported in Manufacturing Process Planner.

## Troubleshooting reports in the rich client

If the embedded viewer does not work correctly, an Applet `CMEReportJTViewerApplet` not initiated error message may be displayed in the browser status line. If so, open the Java console in the system tray and check for Java exception codes that may indicate the cause of the problem, as follows:

- **NoClassDefFoundError**

Java cannot access the embedded viewer JAR files. This error is identified by an error message similar to the following:

```
java.lang.NoClassDefFoundError:  
com/ugs/plmvis/components/<AnyClassName>
```

Possible solutions:

- Check the Java console and verify the correct classpath is set, as described previously. Ensure the **SingleEmbeddedViewer.jar** file is included in the classpath.

To view the classpath, open the Java console and type `s` to dump all system properties. Note the displayed value of the **javaplugin.vm.options** property and check that it contains the classpath string you entered when you configured the Java console.

- If the classpath is not correct, there may be more than one Java control panel; you may not have modified the correct one. Choose **Start→Settings→Control Panel** and select the button that matches the version number in the Java console window. Modify the classpath for this Java control panel.
- Verify there are no typographical errors or unnecessary spaces in the classpath definition. For example, ensure there is an `=` character between **-classpath** and the path name. Also, if you have multiple classpath entries, ensure they are separated by `;` characters.

- If none of the solutions work, copy the **SingleEmdeddedViewer.jar** file manually from *PV\_Base\_Dir\Program* to *JRE\_HOME\lib\ext* folder. Close all Internet Explorer browser instances, restart the browser and try again.

- **NoClassDefFoundError: com/eai/visweb/components/...**

You are trying to run reports from a previous version of Teamcenter. You should convert your reports to use the current embedded viewer.

- **Could not load library jawt.dll**

Java does not have permission to execute the **jawt.dll** file. Verify that you have updated the Java security policy file correctly.

- **VerifyError**

The JRE version is incompatible with the HTML page. This error is identified by an error message similar to the following:

```
java.lang.VerifyError: (class:CMEReportJTViewerApplet,
method: destroy signature: ()V)
Incompatible object argument for function call
    at java.lang.Class.getDeclaredConstructors0(Native Method)
```

Possible solutions:

- Ensure you have installed the supported JRE version, as specified in the installation guide for your operating system, and update it if necessary.
- Verify you have installed the correct version of the **CMEReportJTViewer.jar** file and update it if necessary.
- **Viewer window has no image**

The viewer window shows a green background and the icons are visible, but there is no image in the window.

To resolve this problem, try right-clicking in the JT window and choosing **All On** or **Fit All** from the menu.

- **EXCEPTION\_ACCESS\_VIOLATION**

The version of the embedded viewer you have installed is not compatible with the **CMEReportJTViewer.jar** file in the reports directory. This error is identified by an error message similar to the following:

```

An unexpected exception has been detected
Unexpected Signal : EXCEPTION_ACCESS_VIOLATION
(0xc0000005) occurred at PC=0x6FD24D6
Function=Java_com_teamcenter_visweb_components_PMI3D_nativeHideEntity+0
x1D9
Library=C:\WINDOWS\system32\VisWeb.dll
Current Java thread:
at
com.teamcenter.visweb.components.PrimaryViewer.nativeDestroyViewerDone
(Native Method)
    at
com.teamcenter.visweb.components.Viewer3D.shutdown(Viewer3D.java:718)
    - locked <0x108c3938> (a java.lang.Object)
at
com.teamcenter.visweb.components.Module.removeNotify(Module.java:285)

```

Possible solutions:

- Ensure you have the correct version of the embedded viewer. You must also update the **PATH** environment variable and modify the classpath for the new version.
- Ensure you have the latest version of the **CMEReportJTViewer.jar** file in the reports directory.
- **java.security.AccessControlException**

If you see a blank HTML page with the following error in the Java console window, the Java policy file is not edited correctly. Check the file and edit.

```

java.security.AccessControlException: access denied
    (java.lang.RuntimePermission
getenv.JRE_HOME)
    at java.security.AccessControlContext.checkPermission(Unknown
Source)
    at java.security.AccessController.checkPermission(Unknown Source)
    at java.lang.SecurityManager.checkPermission(Unknown Source)
    at java.lang.System.getenv(Unknown Source)
    at
com.ugs.plmvis.components.VisView.loadLibraries(VisView.java:2665)
    at com.ugs.plmvis.components.VisView.init(VisView.java:2580)
    at com.ugs.plmvis.components.VisView.<init>(VisView.java:241)
    at com.ugs.plmvis.components.VisView.getInstance(VisView.java:205)

```

- **To help protect your security, IE has restricted...** message

If you see a **To help protect your security, Internet Explorer has restricted this file from showing active content that could access your computer. Click here for options...** message in a yellow box instead of the viewer, you have not allowed Internet Explorer to run active content. Repeat the browser configuration procedure.

- **Java Applet Support Required...** message

If you see a **Java Applet Support Required (If you are seeing this message, you need to turn on applet support in your browser)** message in a box instead of the viewer, you have not configured applet tag support. Repeat the browser configuration procedure.

- Nothing is displayed in Firefox

If only a single box is displayed in Firefox, you have not enabled Java. Choose **Tools→Options** in Firefox, click the **Content** button, and ensure the **Enable Java** check box is selected.

## Configure reports in the rich client

If you use the Report Generator application to create reports and view JT files, Teamcenter runs a Java applet that allows you to manipulate the view (rotate, pan, and zoom), gather measurement data, and mark up the image. For correct operation of this functionality, you must configure your Web browser and related settings, as follows:

Note:

The following instructions assume you are configuring a Microsoft Windows system. Similar configuration steps are necessary if you are using another operating system.

Some of the steps may vary depending on which operating system, browser, or JRE version you use.

- If you are using the Microsoft Internet Explorer browser, modify the Internet properties, as follows:
  1. In Windows, choose **Start→Control\_Panel→Internet Options**  
Windows displays the **Internet Properties** dialog box.
  2. Click the **Advanced** tab.
  3. Select the **Allow active content to run in files on My Computer** check box in the **Security** group of entries and the **Java console enabled** check box in the **Microsoft VM** group of entries.
  4. Click **OK** or **Apply**, and then close and reopen the browser to complete any changes.

Note:

If you are using the Firefox browser, this procedure is not required. However, if you encounter problems viewing reports with Firefox, choose **Tools→Options** and verify the **Enable Java** check box on the **Content** pane is selected.

- Install the embedded viewer and run its setup program, if you did not already do so when you ran Teamcenter Environment Manager (TEM). During installation, avoid spaces in installation directory names, as these may cause incorrect operation in certain circumstances.
- Ensure that the JRE version you have installed is at least version 1.6.0\_20 or later.
- Click the **Advanced** tab of the Java control panel and ensure the **Place Java icon in system tray** check box in the **Miscellaneous** group is selected. This ensures the Java console is always present in the Windows system tray, allowing easy access to it for troubleshooting.

When troubleshooting, you can also select the **Enable tracing** and **Enable logging** check boxes in the **Debugging** group. This displays more detailed information in the Java console window that may help you debug problems.

Optionally, you can click the **Show Console** button in the **Java Console** group to open the Java console for debugging purposes whenever the applet is active.

- Click the **Browser** tab and ensure that the check box for your browser is selected. If you have more than one browser on your system, check each applicable check box.

If you changed the Java run-time parameters in the previous step, clear your current browser, click **Apply**, then recheck the current browser and click **Apply** again. This loads the new environment variable setting into the browser.

- Modify the Java security settings to allow the browser to open the DLL files for the embedded viewer and JT files in one of the following ways:
  - Edit the **java.policy** file in your **JRE\_HOME\lib\security** directory with a text editor, *not* a word processor. Add the following entry to this file:

```
grant {
    permission java.security.AllPermission;
};
```

- Use the **policytool.exe** application in your **JRE\_HOME\bin** directory to grant the same permissions. See the Sun documentation for information on using this tool.

If you have more than one version of the JRE on your system, or if you also have the Java Software Development Kit (SDK) installed, there may be more than one copy of the **java.policy** file. If possible, identify the version that is used by the browser. Identify the policy deployment directory by opening the Java console, typing **s** to dump all system properties, and noting the displayed value of the **application.home** property. Place your modified **java.policy** file in the directory indicated by this property; if the directory does not exist, create it. Alternatively, if you are using Internet Explorer, you can modify the file pointed to by the **deployment.system.security.policy** value.

**Note:**

Close all browsers before testing changes.

## View or print rich client reports with a Web browser

1. Select the process or operation on which you want a report and open the **Report** view.
2. Click the button next to the **Type** box and select a report from the list of available types.
3. Click the **Generate Reports** button to create the report. Optionally, you can click the **Save** button to save the report you generated.
4. Open the **Attachments** view. The generated report is shown as an attachment to the process or operation item revision. You can double-click the report to open and view it.

## Creating reports in the rich client

Manufacturing Process Management provides a framework on which you can create reports that are relevant to your manufacturing processes and the Report Generator application that produces the predefined reports.

For example, you may want to create a report that details the usage of various machined components in assembly operations. From the definition of the product structure, Teamcenter extracts the following information:

- The name or identification of each part consumed in the manufacturing process. Thumbnail pictures can also be included if required.
- Detailed information about each part, such as the dimensions, quantity, and **Make/Buy** status.
- A graphical representation of the manufacturing structures and assemblies. Each part is linked to others to show the hierarchy in the assembly structure.

A summary page may define the total number of components, welds, and the comparative number of bought and manufactured parts.

# 19. Configuring the accountability check

## Setting up the accountability check


Using the accountability check, you can compare two structures in various ways. You can, for example, ensure all lines in the EBOM structure are assigned to the MBOM structure, or you can compare the product structure and the process structure to ensure that all occurrences of product components and features are used in the process structure as consumed items. You can also compare a set of properties that you specify to see if they match. When running an accountability check, you can compare the entire assembly or only a subassembly to the selected process.

When running the accountability check, it is helpful to have the selection between the **Accountability Check Results** view and the structure view synchronized—selecting a line in the results view selects the same line in the process view. You can control selection synchronization using the **MExpandToSelection** preference. Set it to **True** if you want a selection on the results pane to select the equivalent line in the BOM. This can be set in the **Options** dialog box in the **Manufacturing** pane.

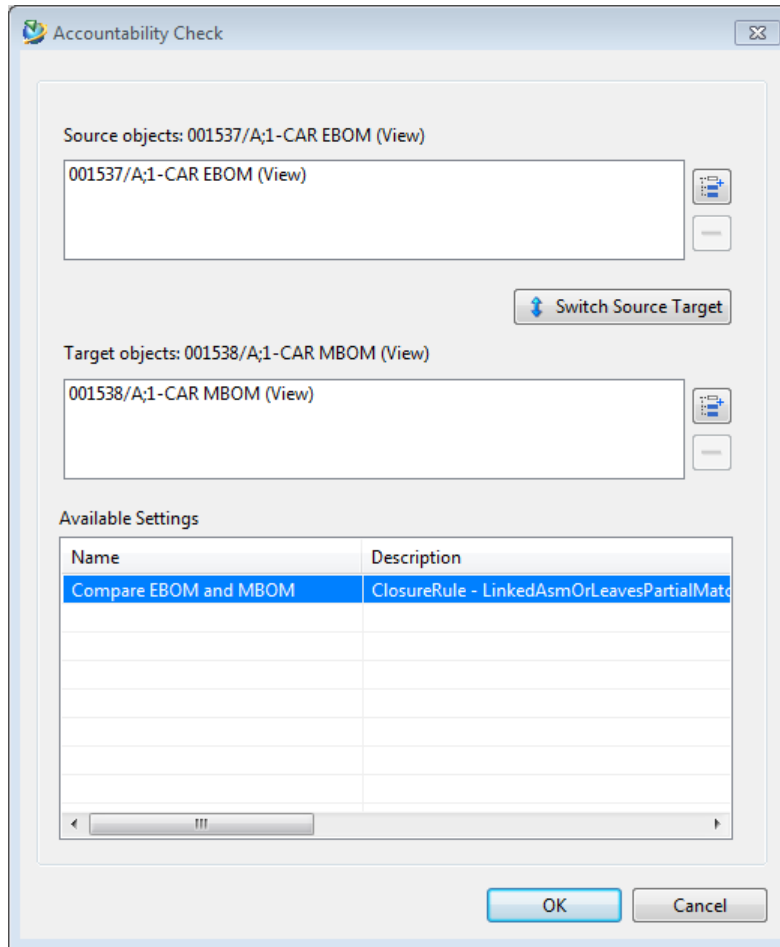
## Publish accountability check settings

When you save an accountability check as an available setting, Teamcenter stores the setting in a dataset. It also automatically creates a preference that references this dataset. The name of this preference depends on the types of structures that you used as the scope of the accountability check for which you saved the setting. For example, if you opened an engineering BOM and a manufacturing BOM, selected accountability check options for these two structures, and saved them as an available setting, the resulting setting is stored in the **MEAccountabilityCheckSettings.BOM.BOM** preference. If you used a product BOP and a plant BOP to create the setting, the setting is saved in the **MEAccountabilityCheckSettings.PBOP.PlantBOP** preference. An administrator can create a new instance of each of these preferences and assign it to a person, role, or group. The setting is then available to those members for use when running a simple accountability check.

You must be logged on as an administrative user to perform this task.

1. Search for the **MEAccountabilityCheckSettings<structureType>.<structureType>** preference that holds the setting you want to publish.
2. Choose **Edit**→**Options** and click the **Organization** tab.
3. Select the user, role, or group for whom the setting should be available.
4. Create a new instance of the preference by clicking .
5. Save the new instance of the preference.

When that user, role, or group signs in, the setting is available for them when they want to run an accountability check on structures of the same type.



The mechanism used to manage these preferences is common to all Teamcenter preferences.

## Configuring the inclusion rules

You can filter the source and target structures before you compare them to limit the number of comparisons required in an accountability check, therefore limiting the scope of the search. The filter mechanism is based on closure rules that traverse the structures and take only those objects specified in the closure rules into consideration for the check.

You may need to create new or modify existing closure rules for your own business use case. The closure rules you create for the accountability check are referred to as *inclusion rules* and are not available for importing and exporting. You can create and modify closure rules in the PLM XML/TC XML Export Import Administration application.

**Note:**

Expanding a structure based on closure rules also plays a role in which objects are taken into consideration in the accountability check. If you expand using an appropriate closure rule, you may not have to set an inclusion rule when running an accountability check.

All the existing closure rules created for the accountability check begin with **Accountability** or **AC**. When creating new closure rules, note the following:

- The scope of traversal is **Export**.
- The output schema format is **TCXML**.
- If you want a line in the structure to be considered in the accountability check, set the **Action Type** entry to **PROCESS**.

If the **Show Unconfigured** view options are set to show the unconfigured structure and you want to only consider the configured lines for the accountability check (using closure rules mode), adding the following clause to the existing list of closure rule clauses causes the accountability check to skip the unconfigured lines.

Primary Object Class Type	CLASS
Primary Object	BOMLine
Secondary Object Class Type	CLASS
Secondary Object	BOMLine
Relation Type	PROPERTY
Related Property Or Object	bl_child_lines
Action Type	SKIP
Conditional Clause	SECONDARY.bl_is_occ_configured=="false"  SECONDARY.bl_variant_state=="

If you create your own inclusion rules for the accountability check, you must add them to the following preferences to have them displayed in the **Accountability Check** dialog box.

- **MEAccountabilityCheckSourceStructureExpansionFilterRules**

List of closure rules that are displayed in the **Accountability Check** dialog box.

- **MEAccountabilityCheckTargetStructureExpansionFilterRules**

List of closure rules that are displayed in the **Accountability Check** dialog box.


You can also set the maximum number of lines to be returned by the accountability check using the preference **MEAccountabilityCheckMaximumAllowedLines**. If the actual number exceeds the maximum, a printable report is generated and attached to the accountability check target structure.

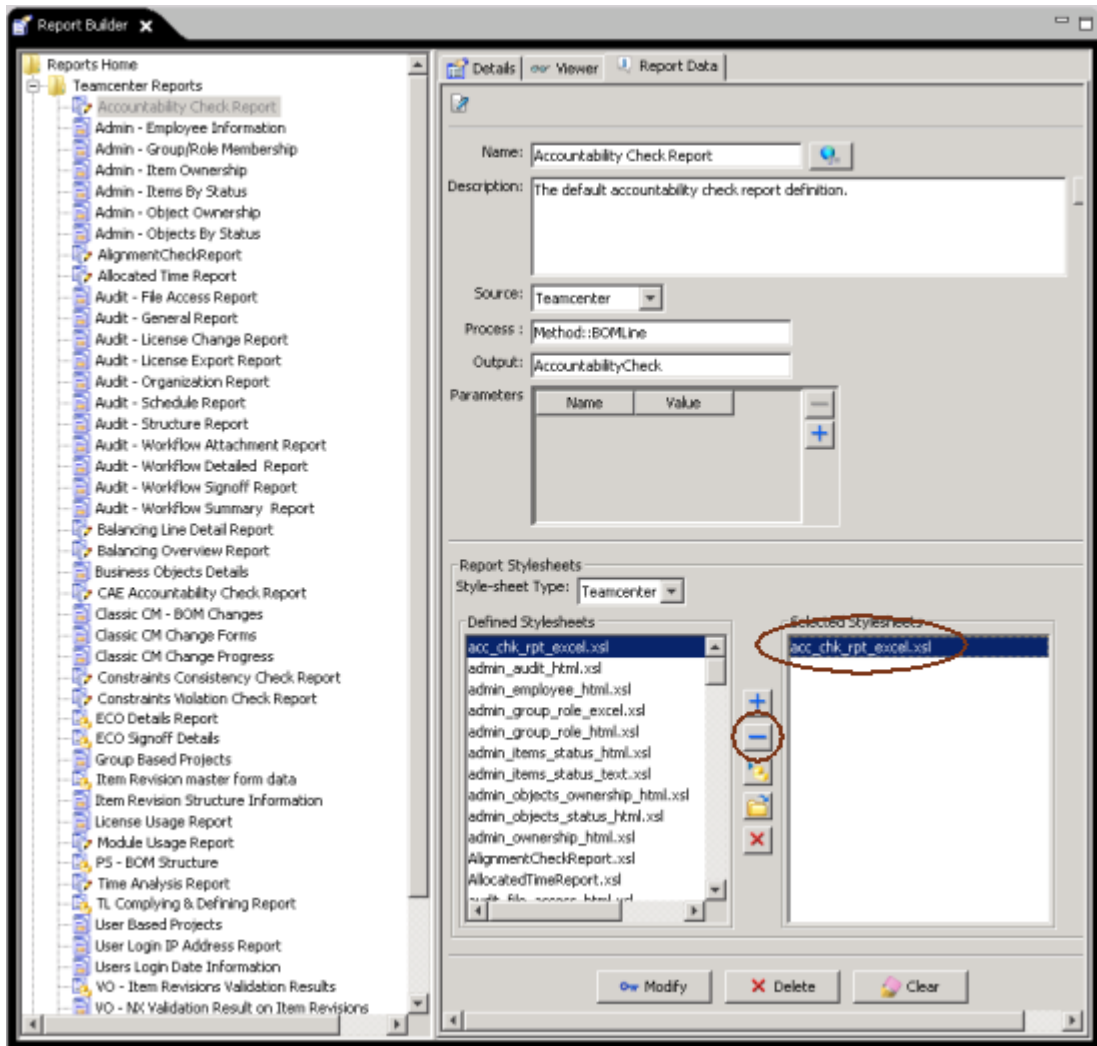
## Customize accountability check reports using a temporary XML file


When Teamcenter creates the accountability check report including report generation, it does the following:

- Creates a temporary XML file.
- Applies the stylesheet to create the Excel report.
- Deletes the XML file on your local directory.

If you want to customize the report using the temporary XML file before Teamcenter deletes it, do the following.

1. Log in to Teamcenter with administrative privileges.
2. Open the Report Builder application and in the **Reports Home** pane, expand the **Teamcenter Reports** node.
3. From the **Teamcenter Reports** list, select **Accountability Check Report**.
4. In the right panel, click the **Report Data** tab.
5. From the **Selected Stylesheets** section, select the default stylesheet and click .




6. Click **Modify**.
7. Run an accountability check including report generation.
8. Find the XML file in the *your-TMP-folder\User\2TierTransientVolume* folder.
9. Use this XML file to help you customize the stylesheet and then give it a unique name (for example, *my\_acc\_chk\_rpt\_excel.xml*).
10. Click  and in the **Import Stylesheet** dialog box, from the **Dataset Type** list, select **MSExcel**.
11. Browse to the custom stylesheet you want to use and click **OK**.
12. From the **Defined Stylesheets** list, select the custom stylesheet and click **+** to add it to the **Selected Stylesheets** list.

13. Set the value of the preference **AccountabilityCheckReportStylesheetName** to the name of your custom stylesheet.
14. Click **Modify**, log out of Teamcenter, and then log in again.

When you now generate an accountability Check Excel report, Teamcenter will use the customized stylesheet. The above steps need to be executed every time you change the stylesheet.

## Add or remove accountability check report stylesheets

To customize the accountability check report that you generate as an Excel file when executing the accountability check command, you must first create a new custom stylesheet. Do the following to add or remove stylesheets for use with your accountability check reports.

1. Log on to Teamcenter with administrative privileges.
2. Open the Report Builder application and in the **Reports Home** pane, expand the **Teamcenter Reports** node.
3. From the **Teamcenter Reports** list, select **Accountability Check Report**.
4. In the right panel, click the **Report Data** tab.
5. Click  and in the **Import Stylesheet** dialog box, from the **Dataset Type** list, select **MSEcel**.
6. Browse to the custom stylesheet you want to use and click **OK**.
7. From the **Defined Stylesheets** list, select the custom stylesheet and click **+** to add it to the **Selected Stylesheets** list.
8. Set the value of the preference **AccountabilityCheckReportStylesheetName** to the name of your custom stylesheet.
9. Click **Modify**, log out of Teamcenter, and then log in again.

When you now generate an accountability Check Excel report, Teamcenter will use the customized stylesheet. The above steps need to be executed every time you change the stylesheet.

### Note:

If you created a custom stylesheet prior to upgrading to Teamcenter 11.4 or greater, by default, the Excel report uses a new stylesheet called **acc\_chk\_rpt\_excel.xsl**. Your customized stylesheet must have a different, unique name (for example, **my\_acc\_chk\_rpt\_excel.xsl**). When the value of the preference **MEAccountabilityCheckReportCombineLines** is **true** (default), the generated xml

file will not support legacy stylesheets. To generate legacy data xml files, change the preference value to **false**.


## Configure the accountability check to use net effectivity

1. Configure unit effectivity to be valid for multiple effectivities.
2. Set the **MEAccountabilityCheckEnableNetEffectivity** preference.

This controls whether the accountability check mechanism checks for net effectivity.

3. Implement a user exit and register a method for it in the Business Modeler IDE.

## Display an icon to indicate revision mismatches on BOP lines

In some cases, bill of materials (BOM) items that are consumed in a bill of process (BOP) do not immediately allow users to recognize revision mismatches. Set the value of the preference **MECompositionMismatchPropertyName** to **fnd0CompositionRevMismatch** to display an icon  in BOPs to indicate line mismatches.

By default, the preference value is empty. The preference is at the site location but each user can create his own copy and modify the value since the protection scope is set to user. The user must restart Teamcenter after setting its value.

## Customizing the net effectivity user exit

Teamcenter provides you with customized code to help you implement net effectivity. The method returns the net effectivity that results when you intersect the effectivity defined on an occurrence (or if there is none, the effectivity of its parent) with the effectivity configured in the revision rule.

You must configure this user exit in the Business Modeler IDE application. When doing so, note that the extension point is **Fnd0BOMLINE\_ask\_effectivity\_by\_parent** and the method is **BMF\_BOMLINE\_compute\_net\_effectivity**.

## Accountability check batch mode

### Running an accountability check in batch mode

Teamcenter offers you two separate methods for running an accountability check in batch mode.

- Run an accountability check in the rich client using the **Accountability Check→Schedule Accountability Check** and **Accountability Check→Schedule Structure Propagation** commands.

Use this method if:

- You want an easy way to schedule an accountability check at a specific time.

- You do not have administrative privileges.
- Run an accountability check using a command line utility, **acc\_check\_client**.

Use this method if:

- You want to output to both Microsoft Excel and occurrence groups simultaneously.
- You want to output occurrence groups to both the source and target structures. For example, you can get **Missing in Source** results that you cannot get with the rich client commands or with a regular accountability check.
- You have administrative privileges.

## Running an accountability check or propagation at a specified time

You can schedule an accountability check or structure propagation to run at a later date or time or at a specific recurring time. This is helpful if you are comparing large structures where the action is time-consuming or if you require comparison results on a regular basis. Teamcenter uses the Dispatcher application to run the comparison and propagation at a time you specify.

To use this feature, your administrator must configure the asynchronous services.

The asynchronous services require the installation of Teamcenter Dispatcher.

### Note:

Because asynchronous tasks run in a separate session, the current rich client session is not updated to reflect any changes that are made. If you choose to execute the task immediately, you do not see all of the results in the current session. Refreshing the window or reloading the structure after the task is complete updates your session to see the changes made by the asynchronous process. This is not an issue for sessions that start after asynchronous processing is complete.

## Running an accountability check in batch mode using the **acc\_check\_client** utility

Running an accountability check of the product can take several hours. You can use the **acc\_check\_client** utility to run the accountability check if you:

- Want to output to both Microsoft Excel and occurrence groups simultaneously.
- Want to output occurrence groups to both the source and target structures. For example, you can get **Missing in Source** results that you cannot get with the rich client commands or with a regular accountability check.

- Have administrative privileges.

You can run this utility offline, after hours, using the operating system scheduling to provide up-to-date reports for Teamcenter users the following day. This utility logs onto Teamcenter, performs an accountability check, stores the results in occurrence groups and/or an Excel report and logs off. You can run this utility from a Teamcenter command window or from a batch file that is developed to perform a specific check on two specific structures.

Running the **acc\_check\_client** utility creates the same output that is obtained by running the accountability check in rich client. Teamcenter stores results in one or both of the following:

- Results are stored in occurrence groups that are created on the source and (optionally) the target structures. These occurrence groups appear as tabs when the top item or revision is opened in Manufacturing Process Planner. If structure contexts are used as input arguments, the occurrence groups appear as tabs when the structure contexts are opened in Manufacturing Process Planner.
- Results are stored in an Excel report. The report appears in a dataset that is attached to the target structure's top line item revision. You can find the dataset by opening the structure, selecting the top line, and viewing the **Attachments** tab. You can view the Excel content by opening the dataset.

## Run the **acc\_check\_client** utility from a script

Using the **acc\_check\_client** utility, you can run checks on a nightly basis and use them to update existing occurrence groups and/or excel reports. To run checks in a two-tier environment:

1. Create a script that simplifies the input of parameters. In this script:
  - a. Set the Teamcenter environment, including the *TC\_ROOT* directory.
  - b. Send standard output to a log file.
  - c. For a two-tier environment: start the server on the specific port with ID.
  - d. Start the utility executable and specify that port and ID or a four-tier connection.
  - e. Test the script by running it.
2. For nightly runs, schedule a job to run the script so that the results are updated nightly.

## Using dynamic equivalence to locate matching EBOM and MBOM lines

In some cases, end users may want to locate matching BOM lines in a linked engineering bill of materials (EBOM) and manufacturing bill of materials (MBOM) that have slightly different properties and therefore are not reported as equivalent when they run an advanced accountability check. For example, the lines

may have different ID in context values, but the manufacturing engineer wants to find them to ensure that they have the same material quantity, and later address the difference in the ID in context values.

Using the **MEAccountabilityCheckDynamicIDICProperties** preference, you can define a list of internal BOM line properties to be used as dynamic equivalence criteria when searching for equivalent lines between EBOM and MBOM. For example, to compare absolute transform values, you can add the line **bl\_plmxml\_abs\_xform** to the preference and the lines are reported as equivalent when you run an advanced accountability check. In the **Advanced Accountability Check** dialog box, on the **Equivalence** tab, end users can then select the **Dynamic Equivalence**  check box to enable a dynamic equivalence check.

Note:

Selecting the **Dynamic Equivalence**  check box automatically selects the **Multiple Match**  checkbox in the **Display Options** group on the **Reporting** tab of the **Advanced Accountability Check** dialog box. When **Dynamic Equivalence** is deselected, check boxes in the **Display Options** group return to their previous selections.

If no values are defined in **MEAccountabilityCheckDynamicIDICProperties** preference, then item ID is the default attribute for comparison.

## Modify accountability check closure rules to exclude dynamic in-process assemblies

When running accountability checks between structures that contain dynamic in-process assemblies (DIPAs), the default traversal rules used to determine the scope of the accountability check exclude the DIPA structures from the check. If you run accountability checks using customized closure rules, you must modify these to exclude the DIPA structures.

- Add a line in the transfer mode that skips the DIPA structures, for example:

```
CLASS.BOMLine:CLASS.AppGroupBOPLine:PROPERTY.bl_all_child_lines:SKIP:
SECONDARY.bl_line_object_class=="Mfg0MEDynamicIPA"
```

## Set up the accountability check to compare to intermediate data captures

You can create an intermediate data capture (IDC) of a structure that captures the current state of a configured structure in a PLM XML file. If you then continue to work on the structure, you may want to compare it at a later point to the IDC to see what changes have taken place. To do this, the original structure must contain in-context IDs *before* you create the IDC.

- If you want to compare the entire structure, create the in-context IDs using the **bom\_expand** utility.

- If you want to compare only partial structures, when writing the PLM XML transfer mode for IDC creation, include a line in the transfer mode that creates in-context IDs on the desired lines in the structure. For example:

```
"CLASS.BOMLine : CLASS.AbsOccGenerator : PROPERTY.bl_absocc_generator :  
PROCESS:
```

## Using the two-tier accountability check utility

Use the **accountability\_check** utility to run the rough equivalent of an advanced accountability check and generate Microsoft Excel and occurrence group reports in a two-tier Teamcenter environment. The utility lets you check line equivalence between several types of structures, including EBOM/MBOM, CAD BOM/design EBOM, MBOM/BOP, and product BOP/plant BOP.

Similar to the **acc\_check\_client** utility that is intended for the service-oriented architecture (SOA) client, the **accountability\_check** utility allows you to run accountability checks on sandbox machines without having to set up a four-tier infrastructure that requires you to maintain processes such as pool managers on each machine. This capability lets you run many accountability checks simultaneously and to manage the loads yourself. The utility also outputs occurrence groups to both the source and target structures, while the rich client advanced accountability check can only output occurrence groups to the source.