TEAMCENTER

Microsoft Office Integration With Teamcenter

Teamcenter 2412

SIEMENS

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**

Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

Installing and configuring Extensions for Office

Setting up the synchronization of Microsoft Office attributes with Teamcenter

Building input files for importing Excel data into Client for Office

Setting up templates to export data from Teamcenter to Microsoft Office

1. Prevent issues due to lack of interoperability of Teamcenter or Microsoft Office versions

Before you install Office Integration, ensure that you have considered the following:

1. Interoperability of Teamcenter rich client and web client versions
2. Features that are supported in different versions of Teamcenter rich client and web client
3. Microsoft Office integrations supported in different versions of Teamcenter rich client and web client

Determine the basic interoperability of Teamcenter rich client and web client versions

1. On Support Center, click **Products→Teamcenter→Downloads**.
2. Click the **Hardware and Software Certifications** tile.
3. On the **Hardware and Software Certifications** page, under **Teamcenter Certifications and Information** section, click the **Internal Interoperability - Teamcenter Compatibility Matrix (*)** link.
4. On the **Support White Papers Certifications** page, you can download the Teamcenter interoperability matrix.

Identify the features supported in different versions of Teamcenter rich client and web client

1. On Support Center, click **Products→Teamcenter→Documentation**.
2. In the **Restrict content to version** list, select the web client version.
3. Click the **Active Workspace Minimum Support Matrix** tile.

   The **AW_Minimum_Version_Support.xlsx** file is downloaded.
4. In the **AW_Minimum_Version_Support.xlsx** file, use filters to check the compatible versions of the web client and the rich client, and the supported features.

Identify the Microsoft Office integrations supported in different versions of Teamcenter rich client and web client

1. On Support Center, click **Products→Teamcenter→Documentation**.

2. In the **Restrict content to version** list, select the Teamcenter version.

3. Click the **Hardware and Software Certifications** tile.

4. On the **Hardware and Software Certifications** page, under **Teamcenter Certifications and Information** section, click the **Integrations Matrix (*)** link.

5. In the *Teamcenter Integrations Availability Matrix* file, click the **Microsoft** tab to check the compatible versions of web client and rich client, and the Microsoft Office versions supported.

2. Overview

Why integrate Teamcenter with Microsoft Office

For a more streamlined experience when working with Microsoft Office files, you can choose to work with one of the Teamcenter Microsoft Office integrations.

You may use any of these Teamcenter interfaces to Microsoft Office. Choose the interface that best suits your use of Teamcenter and Microsoft Office.

- Teamcenter Client for Microsoft Office: **A desktop version that must be installed on the client machine**
- Teamcenter Office Online: **A browser-based version of Microsoft Office named Office Online**
- Teamcenter Extensions for Microsoft Office: An integration with functionality specifically for working with Requirements Manager and Structure Manager in Teamcenter.

Each contains specialized functionality for the different types of authoring and editing you may want to do in Microsoft Office files.

- Teamcenter Client for Microsoft Office

  Teamcenter Client for Microsoft Office allows you to manage Teamcenter workspace objects in real time. Client for Office gives you access to Teamcenter objects directly through the following Microsoft Office applications:

  - Microsoft Word
  - Microsoft Excel
  - Microsoft PowerPoint
  - Microsoft Project
  - Microsoft Outlook

  From Outlook, you can save email content or attachments or both and register them in the Teamcenter database. A custom **Teamcenter** ribbon is added to these Microsoft Office applications.

  Client for Office enables you to import objects from Microsoft Excel into Teamcenter.

  Client for Office can be installed with the Teamcenter four-tier rich client or separately as an add-in to Microsoft Office. Client for Office is independent of the Teamcenter rich client. You can work in

Client for Office while a Teamcenter client is running, or you can work solely through Client for Office without running a Teamcenter client.

- Teamcenter Office Online

  Teamcenter Office Online allows you to author, co-author, and edit Teamcenter files in real time using the web client, with no need for a local install. Teamcenter Office Online works with the following Microsoft Office applications:

  - Microsoft Word
  - Microsoft Excel
  - Microsoft PowerPoint

  For more information about working with Teamcenter Office Online, see the web client documentation.

- Teamcenter Extensions for Microsoft Office

  Teamcenter Extensions for Microsoft Office provides easy access to Teamcenter through Microsoft Office Live features in Microsoft Excel. Extensions for Office allows you to edit properties of workspace objects and apply those changes to the Teamcenter database.

  Extensions for Office is available for the two-tier and four-tier Teamcenter rich client.

Setting up Teamcenter Client for Microsoft Office

To set up Teamcenter Client for Microsoft Office:

1. **Install Client for Office**
2. **Configure Client for Office**
3. Optional configurations:

   - **Configure Microsoft Office export templates to export data from Teamcenter to Microsoft Office**

     > Note:
     >
     > This is mostly used by requirements and structures.

   - **Set up attribute exchange between Office files and Teamcenter datasets**

Setting up Teamcenter Extensions for Microsoft Office

To set up Teamcenter Extensions for Microsoft Office:

1. **Install Teamcenter Extensions for Microsoft Office**
2. **Configure Teamcenter Extensions for Microsoft Office**
3. Optional configurations:
   - **Configure Microsoft Excel export templates to export data from Teamcenter to Microsoft Office**

     Note:

     This is mostly used by requirements and structures.

   - **Set up attribute exchange between Office files and Teamcenter datasets**

3. Installing and configuring Client for Office

Installing Teamcenter Client for Microsoft Office

Before you install Client for Office

Choose an installer

Client for Office can be installed alone or as part of a Teamcenter four-tier rich client configuration. Siemens Digital Industries Software provides the following ways to install Client for Office.

Standalone installation wizard	Installs Client for Office on a single host through a step-by-step interface. This wizard does not install the Teamcenter rich client.
Teamcenter Environment Manager (TEM)	Installs Client for Office alone or as part of a four-tier rich client configuration. TEM does *not* install the Teamcenter plugin for Microsoft Project. If you need the plugin for Microsoft Project, you must use the standalone installation wizard.
Deployment Center	Installs Client for Office alone or as part of a four-tier rich client configuration.

Install required software

Teamcenter Client for Microsoft Office requires the following software on your client host:

Install Microsoft Office

Install a supported Microsoft Office version. For certified versions, see the Hardware and Software Certifications knowledge base article on Support Center.

> Note:
>
> You cannot share a session between Client for Office and Teamcenter if the character sets used by both are different. Also, when you install Microsoft Office in a non–English language to work with Client for Office, ensure that you install the language pack of that language, and not that of the English language.

Client for Office installs Teamcenter plugins for Microsoft Word, Excel, PowerPoint, Project, and Outlook. (The plugin for Microsoft Project is installable only through the standalone installation wizard.)

Install Microsoft Windows Components

If you install Client for Office using TEM, install the required Microsoft Windows components *before* you install Client for Office. Installing these components requires administrative privileges.

- Microsoft.NET Framework.
- Visual Studio Runtime for Microsoft Office.
- Microsoft Office Language Pack for English for Windows. This is required if you use a non-English version of Windows because .NET add-ins require English support.

For certified versions of these libraries, see the Hardware and Software Certifications knowledge base article on Support Center.

Gather required information

You may need to provide the following information during installation of Client for Office. Obtain these values from your Teamcenter administrator if you do not have them:

Teamcenter server information	• Host name • Port • Teamcenter application name
FMS server cache (FSC) information	• Host name • Port • Protocol
Security Services information (Required only if you use Security Services with Client for Office.)	• Application ID • Application name • Server host name • Server port

Install Client for Office using the stand-alone installation wizard

Launch the installer

1. In the Teamcenter 2412 software kit, browse to the **additional_applications\OfficeClient** directory. Double-click the **setup.exe** program icon to launch the Client for Office installation wizard.

2. Proceed to the **Prerequisites required** dialog box. If the required versions of the Microsoft .NET Framework and Microsoft VSTO libraries shown in this dialog are not present, you must install them before you continue.

   You can also find the required versions of these libraries in the Hardware and Software Certifications knowledge base article on Support Center.

3. Proceed to the **Setup Type** dialog box. Specify whether to install Client for Office for one user or all users on the client host.

   The option to install for all users is available only to administrative users.

4. In the **Choose Destination Location** dialog box, enter the location in which to install Client for Office.

5. In the **Select Features** dialog box, select the features you want to include in your installation.

   The list of available plugins depends on the Microsoft Office applications installed on your host (Word, Excel, PowerPoint, Project, and Outlook).

   If you use Requirements Manager and want to manage requirements in Microsoft Office, select the **Word Applications→Requirements Management Integration** feature.

   If Teamcenter client communication system (TCCS) is not already present on your system, the **Teamcenter client communication system (TCCS)** feature is selected by default. Client for Office requires this feature.

6. In the **JRE Path** dialog box, enter a path to a supported Java Runtime Environment (JRE), or accept the JRE path shown.

   For certified versions of the JRE for Teamcenter, see the Hardware and Software Certifications knowledge base article on Support Center.

FCC settings

1. In the **Teamcenter FCC Parent settings** dialog box, type information about the FMS server caches (FSCs) your host connects to. Click **Add** to add an FSC.

Value	Description
Protocol	Specifies the communication protocol of the parent FSC. The default value is **HTTP**.
Host	Specifies the host name of the parent FSC.
Port	Specifies the number of the port used by the parent FSC.
Path	Specifies the path to the FSC on the parent FSC host.

   The FCC can connect to multiple FSCs. To add an additional FSC, click **Add** and type the values for the FSC. To remove an FSC from the list, select the row in the table and click **Remove**.

   If you use multiple FSCs, specify a connection priority for each in the **Priority** column.

2. In **Teamcenter FCC Parent settings**, click **Add** and then enter the supplied environment information.

For this parameter	Enter this value
Protocol	**https**
Host	Teamcenter URL
Port	FMS port
Path	**tc/fms**.

3. Proceed to the **Advanced Configuration** dialog box.

   If you want to specify settings for forward or reverse proxies, or other TCCS settings, select the **Advanced Configuration** check box. Otherwise, proceed to **Server connections**.

Advanced configuration

If you selected the **Advanced Configuration** check box in the **Advanced Configuration** dialog box, enter advanced configuration settings. Otherwise, skip this section and proceed to **Server connections**.

1. In the **Forward Proxy Settings** dialog box, choose a setting for the forward proxy:

Value	Description
Use web browser settings	Specifies that you want to use proxy settings from your web browser.
No proxy	Specifies that you do not want to use a forward proxy.
Detect settings from network	Specifies that you want to use proxy settings from your local network.
Use a proxy auto-configuration file	Specifies that you want to obtain settings from a proxy autoconfiguration (PAC) file.
Manually configure proxy settings	Specifies that you want to enter proxy settings manually.

2. If you chose **Manually configure proxy settings** or **Use a proxy auto-configuration file**, enter the required settings for the chosen option:

   - **Manually configure proxy settings**

     In the **Teamcenter TCCS Proxy Settings** dialog box, enter the following values:

Value	Description
All Proxy	Specifies you want to use the same host and port for all protocols. Selecting this checkbox enables the **All Proxy Host** and **All Proxy Port** boxes.
All Proxy Host	Specifies a name of a valid proxy to use for all protocols. In the accompanying **All Proxy Port** box, type the port used by the proxy host.

Value	Description
Http Proxy Host	Specifies the host of a forward proxy server for the HTTP protocol. In the accompanying **Http Proxy Port** box, type the port used by the proxy host.
Https Proxy Host	Specifies the host of a forward proxy server for the HTTPS protocol. In the accompanying **Https Proxy Port** box, type the port used by the proxy host.
Exceptions	Specifies a semicolon-delimited list of host names and IP addresses to exempt. This box is optional. This list can be used to send requests for local endpoints directly to the destination server without going through a forward proxy that is used for endpoints outside the company intranet. For example, this could allow direct access to a Teamcenter web tier hosted within the company while going through a forward proxy to access a Teamcenter web tier hosted by a business partner.

- **Use a proxy auto-configuration file**

  In the **Forward Proxy URL Setting** dialog box, enter the URL to the PAC file from which you want to obtain proxy settings in the **Proxy URL box**.

  If your network uses IPv6 (128-bit) addresses, use the hostname in URIs and do not use the literal addresses, so the domain name system (DNS) can determine which IP address should be used.

  You can configure TCCS environments further after installation.

3. In the **TCCS Environments Information** dialog box, type information about defined TCCS environments. Click **Add** to add a row to the table, and then type the required values.

Value	Description
Name	Specifies the name of the TCCS environment.
URL	Specifies the URL to the TCCS environment.
Filter Text	Specifies a string identifier for the TCCS environment. When installing a rich client, you can optionally provide a **Client Tag Filter** value to filter the list of environments displayed in the rich client to those environments that match the filter value.
SSO URL	Specifies the URL to the Security Services application you use with TCCS.
SSO App ID	Specifies the ID of the Security Services application you use with TCCS.

Note:

If your network uses IPv6 (128-bit) addresses, use the hostname in URIs and do not use the literal addresses, so the domain name system (DNS) can determine which IP address should be used.

You can configure TCCS environments further after installation.

4. In the **Reverse Proxy** dialog box, specify whether to enable TCCS reverse proxy support. Select the **Enable Reverse Proxy** check box if any URL accessed by TCCS is a reverse proxy server that requires a logon.

   If you select **No**, skip to step **6**.

5. Proceed to the **TCCS Reverse Proxy Settings** dialog box.

   Teamcenter uses reverse proxy settings to detect a logon web page from a reverse proxy server through which Teamcenter services are accessed.

   The criteria table lists the reverse proxy criteria currently defined. Each row of the table is a criteria XML element defined in the specified format. By default, the table is blank and no criteria are defined. A criterion string is of the following form:

   *Header_Name1***,** *Header_Value1***,** *Header_Name2***,** *Header_Value2***,...:***Form_Action*

   Each criterion must contain at least one header name/header value pair or at least a single form action.

   To add a criterion to the table, perform the following steps:

   a. Click **Add**.

   b. Type the header names and values for criterion you want to add. In the **Form Action** box, specify a form action.

   c. Click **OK** to add the criterion or **Cancel** to abandon your changes.

Note:

- If you must connect to a Teamcenter environment through a reverse proxy server (such as WebSEAL or SiteMinder), you may need to configure reverse proxy settings for TCCS.
- If you use SiteMinder, you must configure TCCS to detect form-based challenges originating from the reverse proxy by selecting the **Check Headers** check box.

This setting also applies to other reverse proxy servers that do not send specific header information in the 200 form-based challenge.

- Criteria definitions are written to the **reverseproxy_cfg.xml** file.

6. Proceed to the **Kerberos authentication support** dialog box.

   If you use Kerberos, enter Kerberos authentication settings.

Value	Description
Support Kerberos authentication	Specifies you want to use Kerberos authentication for Teamcenter logon.
Always prompt for user ID	Specifies you want to always prompt for a Kerberos user name. If you want to enable zero sign-on functionality on Windows hosts, clear this check box. *Zero sign-on* allows Windows users to launch a Teamcenter client without being prompted to log on to Teamcenter. Note: Zero sign-on functionality requires you configure Security Services in applet-free mode in the **Security Services** panel. For more information about configuring Security Services, see *Security Services Configuration*.

   Kerberos is a network authentication protocol that uses a system of *tickets* to allow nodes communicating over nonsecure networks to securely verify identities of each side. Using a client-server model, it provides mutual authentication: the user and the server verify each other's identities.

7. Proceed to the **Secure Socket Layer (SSL) Settings** dialog box. Enter smart card authentication settings.

Value		Description
Use Internet Explorer Certificate Store (Recommended)		Specifies you want to use certificates stored in Microsoft Internet Explorer. This option is available only on Windows hosts.
Disable SSL		Specifies you want to disable SSL authentication.
Configure Certificate Store Manually		Specifies you want to manually configure the certificate store for Teamcenter.
	Use trust store	Specifies you want to use a truststore. If you select this option, enter the path to the **File** that contains the truststore you want to use.

Value		Description
	Accept untrusted certificates	Specifies you want to accept untrusted certificates.
	Configure key store	If you want to configure a keystore, select this check box. If you choose this option, enter the path to the keystore. Also, specify the file type. The default file type is **JKS**.

8. Skip to **Complete the installation**.

Server connections

If you selected the **Advanced Configuration** check box in the **Advanced Configuration** dialog box, skip this section and proceed to **Complete the installation**. Otherwise, enter server connection settings as follows.

1. In the **Teamcenter Server Information** dialog box, click **Add**, and then enter the following values for the Teamcenter server:

   - **Connection Name**
   - **Protocol** (**HTTP** or **HTTPS**)
   - **Teamcenter Host**
   - **Port Number**
   - **Application Name**

   If you want to add additional servers, click **Add**. To remove a server from the list, select the server in the list and click **Remove**. To edit settings for a server, select the server and click **Edit**.

   Click **Up** or **Down** to assign server priority.

2. In the **Setup Type** dialog box, select whether to enable Security Services with Client for Office.

   If you select **Yes**, enter server settings for Security Services.

Value	Description
Application ID	Specifies the application ID of your Teamcenter installation as configured in the Security Services installation.
Protocol	Specifies the protocol used to access the Security Services application (**HTTP** or **HTTPS**).
SSO Server Host	Specifies the server host for the Security Services application.
Port Number	Specifies the port used by the Security Services application.
Application Name	Specifies the application name of the Security Services application.

Complete the installation

1. In the **Start Copying Files** dialog box, review your selections. Click **Back** to change your selections or click **Next** to install Client for Office.

2. When installation is complete, close the installation wizard.

3. Restart the system to enable the **FMS_HOME** environment variable required for Teamcenter client communication system.

   You can perform advanced TCCS configuration after installation is complete.

Installing or updating Client for Office using a Teamcenter patch kit

You can install Client for Office or update an existing installation using a Teamcenter patch kit, for example, Teamcenter 2412.0001. Run the **setup.exe** program from the patch kit. When the wizard displays the **Teamcenter Kit Location** dialog box, enter the path to the dependent Teamcenter software kit.

> Note:
>
> Not all Teamcenter patch kits contain an updated Client for Office installation program.

Install Client for Office using TEM

This procedure describes installing Client for Office as part of a new Teamcenter environment. If you install Client for Office into an *existing* Teamcenter environment, launch TEM in maintenance mode as you would to add features, proceed to the **Features** panel, and then proceed to step 7 here.

Launch Teamcenter Environment Manager (TEM) from the Teamcenter 2412 software kit or, if available, a Teamcenter patch kit (for example, Teamcenter 2412.0001) to install Client for Office.

1. **Install the prerequisite software** on the client host. Teamcenter Environment Manager (TEM) does not install the prerequisite software.

2. Start Teamcenter Environment Manager (TEM).

   In the Teamcenter 2412 software kit, right-click the **tem.bat** program icon and choose **Run as administrator**.

3. In the **Welcome to Teamcenter** panel, select **Teamcenter**.

   For information about any panel in TEM, click the help button  .

4. In the **Install/Upgrade Options** panel, click **Install**.

5. If you are running TEM from a Teamcenter patch kit, the **Media Locations** panel contains the **Original Media Location** box. In this box, enter the path to the dependent Teamcenter kit specified.

6. In the **Configuration** panel, type an ID and a description for the new Teamcenter configuration.

   Proceed to the **Features** panel.

7. In the **Features** panel, select the **Teamcenter Client for Microsoft Office** feature (under **Extensions→Enterprise Knowledge Foundation**).

   You may optionally include additional features in your Teamcenter configuration. If you select additional features, TEM displays panels during installation that are not described in this procedure.

   > Note:
   >
   > TEM does *not* install the Teamcenter plugin for Microsoft Project. If you need the plugin for Microsoft Project, you must use the **standalone installation wizard**. TEM installs Teamcenter plugins for Microsoft Word, Excel, PowerPoint, and Outlook.

8. In the **File Client Cache (FCC)** panel, choose whether to use a new or existing FMS client cache (FCC).

   If you want to create a new FCC using the same settings as an existing FCC on your host, select the **Merge values from an existing FMS_HOME** ☑ check box.

   If you want to specify settings for forward proxy, reverse proxy, Kerberos authentication, or other TCCS settings, click **Advanced**.

9. In the **FCC Parents** panel, enter information about the FMS server cache (FSC).

10. Proceed to the **4-tier server configurations** panel. In the **URI** column, enter the URI for the Teamcenter web tier server. In the **Connection Name** column, enter a name for the rich client connection.

    > Note:
    >
    > If your network uses IPv6 (128-bit) addresses, use the hostname in URIs and do not use the literal addresses, so the domain name system (DNS) can determine which IP address should be used.

11. In the **Office Client Requirement** panel, TEM reports whether the **required software** is detected on your system. If not, the installation cannot continue.

    If you are logged on as an administrative user on the client host, you can also choose whether to install Client for Office for the current user or for all users on the client host. This option is under **Administrator Option**.

12. In the **Office Client Configuration** panel, if you use Security Services with Client for Office, select the **Install Teamcenter Single Sign-on support for Office Client** ☑ check box, and then enter connection information for the Security Services application.

13. Proceed to the **Confirmation** panel and review your selections. Click **Start** to install Client for Office, or click **Back** to change your selections.

Install Client for Office using Deployment Center

This procedure describes installing Client for Office using Deployment Center. To install Client for Office using the Deployment Center, you must select the **Teamcenter Client for Office** component.

Components are the architectural pieces of Teamcenter, such as servers, services, and databases. Some components are automatically selected based on your selections in the **Software** and **Applications** tasks. The list of components available for installation is also determined by your selections in the **Software** and **Applications** tasks. For example, some components require a corresponding application to be selected before the component is made available. Some components are allowed only a single instance within an environment, so if a component is already installed, it may not be in the list of available components.

Configuration parameters for some components may require server names, user names, passwords, URLs, and other system information you may have previously entered for other components in your environment. When you add components, some parameters may be prepopulated with those values from other components. Some prepopulated values may not be editable. For example, in a single box environment, **Machine Name** and **OS** may not be editable.[1]

Some parameters may provide dropdown lists of values from which you can choose. For example, in a distributed environment, the **Machine Name** field for a component may provide a selection list of machine names already defined in your environment.

1. In Deployment Center, select your existing environment.

2. In the **Components** task, click **Add component to your environment** ⊕ to add the required component.

   The **Available Components** panel displays the available optional components.

3. In **Available Components**, select the **Teamcenter Client for Office** component to install. Then click **Update Selected Components** to add them to the **Selected Components** list.

   In **Selected Components**, the **COMPLETE** column displays the configuration status for each component. If all required parameters are entered for a component, its completion status is **100%**.

1 If you selected the **Single Box** environment type in the **Options** task, all Teamcenter components must reside on the same machine.

4. Click the **Teamcenter Client for Office** component in the list to display its parameters in the right panel. This panel initially displays only required parameters. You must enter values for settings that appear in required parameters view. You can toggle the view between required parameters and all parameters:

   Show all parameters

   Required parameters view displays only required parameter information. Click to expand the view to display both required and optional parameters.

   Show only required parameters

   All parameters view displays both required and optional parameter information. Click to collapse the view to required parameters.

5. In the **Client Type Settings** section, specify details such as **Machine Name** and **Setup Type**.

   To generate a special deploy script that can be deployed on multiple machines, select the **Enable Mass Client Deploy** option.

6. In the **General Settings** section, specify the installation path for the Client for Office and the **4-Tier Server Connection Name**.

7. In the **Select Features** section, some Client for Office features are selected by default. You can optionally choose additional Requirements Management Integration features as required.

8. After you enter the required parameter values, click **Save Component Settings**.

   If you don't have values for all required parameters, you can save your settings at any time and return to finish them. However, the **Deploy** task is disabled until all components in the environment show a configuration status of **100%**.

9. When all components are fully configured, go to the **Deploy** task. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

10. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see *Deployment Center — Usage*.

Install Client for Office using a silent distribution

A *silent distribution* is an XML-based configuration file you can use to install Teamcenter Client for Office *silently* (without user interaction) on another host. Silent installation suppresses most installation prompts and requires minimal user interaction. As an alternative to installing and configuring

Teamcenter Client for Office on individual hosts in your network, silent installation provides an efficient way to deploy Teamcenter Client for Office on multiple hosts in your network.

The silent installation configuration file records the selections and values you enter during installation and enables TEM to perform these steps noninteractively on other hosts. You can modify a silent configuration file to change certain Teamcenter settings before installation.

To install Client for Office using a silent installation, you must:

1. **Create a silent installation configuration file**

2. **Launch the silent installation**

Create a silent installation configuration file

1. Log on as the administrator and open the command prompt.

2. Change the directory to the Office Client directory. For example:

   Tc2412\Tc2412_wntx64\wntx64\additional_applications\OfficeClient

3. Run the command: setup.exe /r /f1"D:\localData\ResponseFiles\OfficeClient.properties"

   > Note:
   >
   > This allows you to perform the installation and creates a response file to use for silent installations.

4. Select the installation settings you need.

5. When prompted for the Teamcenter kit, point to the appropriate kit based on the operating system. For example, if you are using Teamcenter 14 on Windows 64, you would choose *Tc2412_kit\Tc2412_wntx64*.

6. Select either the TCCS or HTTP installation.

7. Complete the installation and verify that the response file is created.

   A response file named *OfficeClient.properties* is created in the *localData\ResponseFiles* folder.

Launch the silent installation

1. Log on as the administrator and open the command prompt.

2. Change the directory to the Office Client folder. For example:

Tc2412_kit\Tc2412_wntx64\wntx64\additional_applications\OfficeClient

3. Run the command:

```
setup.exe /s /f1"D:\localData\ResponseFiles\OfficeClient.properties"
```

Process of opening a Microsoft Office dataset from Teamcenter

When you open a Microsoft Office dataset from the Teamcenter rich client, Teamcenter rich client passes certain information to Client for Office. This information preserves the Teamcenter context of the dataset.

In the process of opening the dataset:

1. The rich client downloads an Office Application Integration (**.oai**) file, opens the Microsoft Office application, and passes the **.oai** file as an input.

2. The Office application starts, and the Client for Office add-in is loaded.

   Warning:

   By default, Microsoft Office allows add-ins to run without notification and does not check for digital signatures. You can change this behavior by choosing higher security settings in the Office Trust Center. Client for Office supports all add-in security settings.

   Higher security settings (for example, **Require Application Add-ins to be signed by Trusted Publisher**) can add a significant amount of time in starting the application and loading the add-in. This process can take 30 seconds or longer, depending on your settings.

3. The add-in parses the **.oai** file to obtain the Teamcenter context information about the dataset.

4. The **.oai** file closes, and the dataset is downloaded from Teamcenter and opens in the Office application.

   Normally, the **.oai** file appears only briefly before the dataset opens. Under slower network conditions, however, the **.oai** file may remain open longer.

   Caution:

   Suppose that the **.oai** file remains open and does not show the contents of the dataset file that you selected in the rich client. In such a case, ensure that the Client for Office is installed and the Teamcenter Add-in is activated in Microsoft Office. For example, suppose that you opened a Microsoft Word file from the rich client, the **.oai** file is downloaded on the computer but Microsoft Word is not opened. In such a case, ensure that the Client for Office is installed correctly and the Teamcenter Add-in is activated in Microsoft Word on your local computer.

If you have questions about this issue, consult your Teamcenter administrator.

Uninstall Microsoft Office interfaces

Choose the appropriate uninstallation method

Uninstall Teamcenter Microsoft Office interfaces using the method that reflects how you installed the interfaces.

Method	Description
Windows Control Panel	If you installed Microsoft Office interfaces using stand-alone installation wizards, uninstall them using the **Add or Remove Programs** dialog box in the Windows Control Panel.
Teamcenter Environment Manager (TEM)	If you installed Microsoft Office interfaces using TEM, uninstall them using TEM.
Silent distribution	If you installed Microsoft Offices interfaces using a **silent distribution**, uninstall them using a **silent distribution**.

Uninstall Microsoft Office interfaces using Windows Control Panel

1. Open the **Add or Remove Programs** dialog box in the Windows Control Panel.

2. Select the programs you want to remove and click **Change/Remove**:

   Teamcenter Client for Office
   Teamcenter Applications for Microsoft Office

   Note:

   Selecting this item uninstalls Teamcenter Extensions for Microsoft Office and Teamcenter Plugin for Microsoft Project.

Uninstall Microsoft Office interfaces using TEM

1. Log on to the operating system with the Teamcenter user account you created for installing and maintaining the Teamcenter installation.

2. Start Teamcenter Environment Manager (TEM):

   Start→Programs→Teamcenter→Environment Manager

3. In the **Maintenance** panel, select **Configuration Manager**.

4. If you installed Microsoft Office interfaces as part of a Teamcenter rich client configuration, you can remove the Teamcenter configuration or just remove Microsoft Office interfaces from the rich client configuration.

   - To remove the Teamcenter configuration, perform the following steps:

     a. In the **Configuration Maintenance** panel, select **Remove configuration (uninstall)**, and then click **Next**.

     b. In the **Old Configuration** panel, select the configuration you want to remove, and then click **Next**.

     c. In the **Uninstall** panel, select **Yes** to confirm that you want to uninstall the configuration. Click **Next**.

     d. In the **Uninstall Teamcenter** panel, select the **Advanced Uninstall Options** check box if you want to view additional uninstall options. Otherwise, click **Next**.

     e. In the **Confirmation** panel, click **Start** to begin the uninstallation.

   - To remove Microsoft Office interfaces from the configuration, perform the following steps:

     a. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**, then click **Next**.

     b. In the **Old Configuration** panel, select the configuration you want to modify.

     c. In the **Feature Maintenance** panel, choose **Add/Remove Features**.

     d. In the **Features** panel, clear the check boxes for the Microsoft Office interface features you want to remove:

        Extensions→Enterprise Knowledge Foundation→Teamcenter Client for Microsoft Office

        Extensions→Systems Engineering and Requirements Management→Teamcenter Extensions for Microsoft Office

     e. Proceed through the remaining panels in TEM, entering required information as needed.

     f. In the **Confirmation** panel, click **Start** to begin uninstalling features.

Uninstall Microsoft Office interfaces using a silent distribution

1. Log on as the administrator and open the command prompt.

2. Change the directory to:

   C:\Program Files (x86)\InstallShield Installation Information\{13C0720C-CA76-4E19-AF4D-DFCAC0DD2487}

3. Run the command:

   ```
   setup.exe /x /r /
   f1"D:\localData\ResponseFiles\OfficeClient_remove.properties"
   ```

   > Note:
   >
   > This creates a response file for the process to uninstall the software.

4. Run the command:

   ```
   setup.exe /x /s /
   f1"D:\localData\ResponseFiles\OfficeClient_remove.properties"
   ```

Configuring Client for Office

Enable Teamcenter Client for Office

Document authors and reviewers can create and review documents in standalone Microsoft Office applications. They can also edit or view documents by using Microsoft Office applications within the web client. To use these applications in the web client, you **set up Teamcenter Office Online**.

However, if they wish to use the standalone Office applications, you must enable Teamcenter Client for Office as follows:

1. Locate the *client_specific.properties* file in the Teamcenter installation directory.

2. In the *client_specific.properties* file, set the **useAppLauncher** value to **true**.

   If the rich client runs on a Teamcenter two-tier server, set the **preferred4TierServerUrlForMSOfficeClient** value to the URL of the Teamcenter four-tier server that connects to the same database as the two-tier server, for example, `preferred4TierServerUrlForMSOfficeClient=`*http://hostname:7001/tc*

3. Run **genregxml.bat/sh** after you update the *client_specific.properties* file.

Host the web client in Microsoft Office applications

Hosting integrates the web client with installed or cloud-based applications. To host the web client in Microsoft Office applications:

- **Install Teamcenter Client for Microsoft Office**.

  This places the Teamcenter ribbon on the Microsoft Office applications. The web client functionality can be accessed from the Teamcenter ribbon. Refer to the *Client for Microsoft Office* documentation for instructions.

- Define the following preferences to link to your web client installation:

  - Create the **ActiveWorkspaceHosting.URL** or **ActiveWorkspaceHosting.Office.URL** preferences and set the value of these preferences to the URL of the web client installation.

  - Create preferences to define whether to use the web client elements (for example, **TC_Use_ActiveWorkspace_Search**, **TC_Use_ActiveWorkspace_Create**, **OC_Use_ActiveWorkspace_Inbox**, and **OC_Use_ActiveWorkspace_Summary**.

    You must add the **OC_Use_ActiveWorkspace_Inbox**, and **OC_Use_ActiveWorkspace_Summary** preferences to the Teamcenter environment, and then override to set their values to **True**. If these preferences are not added to the Teamcenter environment or added but not overridden and set to **True**, Teamcenter uses the values of **TC_Use_ActiveWorkspace_Inbox**, and **TC_Use_ActiveWorkspace_Summary**.

Specify whether style sheets are rendered to HTML in the web client

The **UsePropertyStylesheetPlatformRenderer** preference specifies whether XML style sheets are rendered as HTML when viewing object properties in the Teamcenter Client for Microsoft Office. The only way you can display style sheets in the Teamcenter Client for Microsoft Office is in the HTML (style sheet) format.

When the **UsePropertyStylesheetPlatformRenderer** preference is set, regardless of whether the value is **true** or **false**, XML style sheets are rendered to HTML in the Teamcenter Client for Microsoft Office.

> Note:
>
> This preference is not used by the Teamcenter rich client.
>
> For more information about style sheets, see *Rich Client Customization*.

Valid Values

true	XML style sheets are rendered to HTML.
false	XML style sheets are rendered to HTML.

Default Values

true

Default Protection Scope

Site preference.

Specify whether hyperlinks point to the rich client or to the web client

As a system administrator, you can configure for which client the hyperlinks should be generated, and whether a user can change this value.

To do so:

1. Edit the config.template files for each Teamcenter add-in:

   - *TcExcelAddin.dll.config.template* (for Microsoft Excel)
   - *TcOutlookAddin.dll.config.template* (for Microsoft Outlook)
   - *TcPowerPointAddin.dll.config.template* (for Microsoft PowerPoint)
   - *TcWordAddin.dll.config.template* (for Microsoft Word)

   These files are located in the folder where the Office client is installed.

   > Note:
   >
   > If an administrator wants to pre-configure various settings prior to deployment, they would modify the *.config.template* files in the installation kit to make the desired changes to configuration settings. These *.config.template* files are then used during the installation process to programmatically generate the respective *.config* files. This way, every user does not have to make these changes to configuration settings as the installation process generates the files with all the desired configuration settings. Once the Office Client add-ins have been installed, the only way to make a change is to edit each of the *.config* files on each user's machine. If a user edits the *.config.template* files at this point it makes no difference. The *.config.template* files are only used during the installation process.
   >
   > To summarize, an administrator can either modify the *config.template* files in the software depot before deploying Client for Office, or if Client for Office is already deployed, the administrator or user must update the *.config* files on each client workstation.

2. Within each of the *config.template* files, add the following XML code in the **tcresource** section:

   ```
   <hyperlinkconfig>
          <add linkto=<client name> allowusertomodify=<value>/>
       </hyperlinkconfig>
   ```

   You can assign either of the following values to the **linkto** parameter:

- **richclient** - generates hyperlinks to the rich client.
- **thinclient** - generates hyperlinks to the web client.

You can assign either of the following values to the **allowusertomodify** parameter:

- **false** - disables the user interface so that users cannot modify the value.
- **true** - allows users to modify the value.

Specify the list of file extensions for which a warning is displayed to users

As a system administrator, you can optionally configure the **Tc_OfficeClient_HarmfulExtensionsList** preference that defines a list of potentially harmful file types. This is a site preference. For the list of extensions that the administrator specifies, the system displays a warning and asks users to confirm if they want to open the file.

Set up the automatic deletion of log files and temporary files after a specific number of days

As a system administrator, you can configure the deletion of log files and temporary files after a specific number of days. If you configure this setting, users cannot change it.

1. Edit the *config.template* files for each Teamcenter add-in:

   - *TcExcelAddin.dll.config.template* (for Microsoft Excel)
   - *TcOutlookAddin.dll.config.template* (for Microsoft Outlook)
   - *TcPowerPointAddin.dll.config.template* (for Microsoft PowerPoint)
   - *TcWordAddin.dll.config.template* (for Microsoft Word)

   These files are located in the folder where the Office client is installed.

   Note:

   If an administrator wants to pre-configure various settings prior to deployment, they would modify the *.config.template* files in the installation kit to make the desired changes to configuration settings. These *.config.template* files are then used during the installation process to programmatically generate the respective *.config* files. This way, every user does not have to make these changes to configuration settings as the installation process generates the files with all the desired configuration settings. Once the Office Client add-ins have been installed, the only way to make a change is to edit each of the *.config* files on each

user's machine. If a user edits the *.config.template* files at this point it makes no difference. The *.config.template* files are only used during the installation process.

To summarize, an administrator can either modify the *config.template* files in the software depot before deploying Client for Office, or if Client for Office is already deployed, the administrator or user must update the *.config* files on each client workstation.

2. Within each of the config.templates files, add the following XML code in the **tcresource** section:

```
<autodeletefiles>
       <add filetype="LOGFILES" daystosave="7" autodelete="true" />
       <add filetype="tempfiles" daystosave="0" autodelete="true" />
    </autodeletefiles>
```

You can assign either of the following values to the **filetype** parameter:

- **logfiles** - defines the values for the handling of log files.
- **tempfiles** - defines the values for the handling of temporary files.

You can assign either of the following values to the **autodelete** parameter:

- **false** - does not automatically delete the files.
- **true** - automatically deletes the files.

You must set the **autodelete** parameter to **true** to assign a value to the **daystosave** parameter. This value must be an integer representing the number of days (0 days implies delete immediately) to save the files.

3. You must modify the *config.template* files before running the installation process.

Note:

If the Teamcenter Client for Office was installed using the stand-alone installer (*setup.exe*), the system administrator can use the modify feature of the installer to generate the *.config* files again.

Specify the default locale and whether users can switch to another locale

As a system administrator, you can specify whether users can change the current locale or culture settings for a session. Prior to deployment, you can edit the following kitting files:

1. Edit the *config.template* files for each Teamcenter add-in:

- *TcExcelAddin.dll.config.template* (for Microsoft Excel)
- *TcOutlookAddin.dll.config.template* (for Microsoft Outlook)
- *TcPowerPointAddin.dll.config.template* (for Microsoft PowerPoint)
- *TcWordAddin.dll.config.template* (for Microsoft Word)

These files are located in the folder where the Office client is installed.

> Note:
>
> If an administrator wants to pre-configure various settings prior to deployment, they would modify the *.config.template* files in the installation kit to make the desired changes to configuration settings. These *.config.template* files are then used during the installation process to programmatically generate the respective *.config* files. This way, every user does not have to make these changes to configuration settings as the installation process generates the files with all the desired configuration settings. Once the Office Client add-ins have been installed, the only way to make a change is to edit each of the *.config* files on each user's machine. If a user edits the *.config.template* files at this point it makes no difference. The *.config.template* files are only used during the installation process.
>
> To summarize, an administrator can either modify the *config.template* files in the software depot before deploying Client for Office, or if Client for Office is already deployed, the administrator or user must update the *.config* files on each client workstation.

2. Within each *config.template* file, add the following XML code in the **tcresource** section:

```
<forcedcultureconfig forcedUICulture=<culture string>
allowUserToModify=<allow flag> />
```

- **forcedUICulture parameter**

  You can assign any supported locale or culture values to the **forcedUICulture** parameter:

  For example, **en-US** sets the locale to English (United States) and **de-DE** sets the locale to German (Germany).

  The values for **forcedUICulture** can only be set to values currently supported by Client for Office.

  For example, to let users modify the locale and to set it to US English by default, you must specify:

  ```
  <forcedcultureconfig forcedUICulture=<en-US>
  allowUserToModify=<true> />
  ```

- **allowUserToModify parameter**

You can assign a value of either **true** or **false** to the **allowUserToModify** parameter:

- **true** – enables users to change the locale or culture for the current session in the Client for Office preferences in the **Change Session Culture** area within the **Miscellaneous** tab of the **Basic Teamcenter Preferences** dialog box.

- **false** – prevents users from changing the locale or culture and disables the list of language choices in the **Change Session Culture** area within the **Miscellaneous** tab of the **Basic Teamcenter Preferences** dialog box.

> Note:
>
> If the value for **allowUserToModify** is set to **true**, any changes to the locale or culture made by the user are only valid for the current session. After closing and restarting the application, the locale or culture returns to the value specified in the **forceUICulture** parameter.

Specify the types of items users can create

As a system administrator, you can limit the data types that are displayed in the list of item types in the **Create New Item** dialog box. To do so, you must update the exclusion list or inclusion list in the application configuration file. All available object types are included in the list, except the object types that are explicitly listed in the excluded types list. An inclusion list defines the list of item types to be included in the **New Item** dialog box. By specifying item types to be included or excluded, you can limit the types of items users can create.

1. To specify the types of items users can create, edit the *config.template* files for each Teamcenter add-in:

   - *TcExcelAddin.dll.config.template* (for Microsoft Excel)

   - *TcOutlookAddin.dll.config.template* (for Microsoft Outlook)

   - *TcPowerPointAddin.dll.config.template* (for Microsoft PowerPoint)

   - *TcWordAddin.dll.config.template* (for Microsoft Word)

   These files are located in the folder where the Office client is installed.

   > Note:
   >
   > If an administrator wants to pre-configure various settings prior to deployment, they would modify the *.config.template* files in the installation kit to make the desired changes to configuration settings. These *.config.template* files are then used during the installation process to programmatically generate the respective *.config* files. This way, every user

does not have to make these changes to configuration settings as the installation process generates the files with all the desired configuration settings. Once the Office Client add-ins have been installed, the only way to make a change is to edit each of the *.config* files on each user's machine. If a user edits the *.config.template* files at this point it makes no difference. The *.config.template* files are only used during the installation process.

To summarize, an administrator can either modify the *config.template* files in the software depot before deploying Client for Office, or if Client for Office is already deployed, the administrator or user must update the *.config* files on each client workstation.

2. To specify the types of items to be excluded, within each *config.template* file, add the following XML code in the **tcresource** section:

```
<newitemexcludedtypes excludeSecondaryTypes="false">
     <add excludedtype="<type name>"
</newitemexcludedtypes>
```

You can add any number of excluded types inside the **newitemexcludedtypes** tag. You can assign any of the supported object types to the **excludedtype** parameter.

The **ExcludeSecondaryTypes** parameter is optional and has a default value of **false**.

- When **true**, object types and all corresponding secondary (derived) types are excluded from the **Create New Item** dialog box.
- When **false**, only the object types that are listed are excluded from the **Create New Item** dialog box. Secondary (derived) types are displayed in the **Create New Item** dialog box.

For example, to prevent users from creating weldpoint and CAE items, you can specify the exclusion list as follows:

```
<newitemexcludedtypes>
      <add excludedtype="WeldPoint" />
      <add excludedtype="CAEItem" />
</newitemexcludedtypes>
```

3. To specify included types, within each *config.template* file, add the following XML code in the **tcresource** section:

```
<newitemincludedtypes>
      <add includedtype="<type name>"
 </newitemincludedtypes>
```

You can add any number of excluded types inside the **newitemincludedtypes** tag. You can assign any supported object types to the **includedTypes** parameter.

For example, to enable users to create parts and items, you can specify the inclusion list as follows:

```
<newitemincludedtypes>
      <add includedtype="Part" />
      <add includedtype="Item" />
</newitemincludedtypes>
```

> Caution:
>
> The **includedTypes** list has a higher precedence. If both lists are included in the *config.template* file, only the **newitemincludedtypes** list is used.

4. To specify the default object type to create, add the following to the *config.template* file:

   ```
   <newitemdefaulttype defaulttype="Item"/>
   ```

   This automatically selects the object type specified as the default in the **Object Type** list.

   Users can still select another type.

Opting in to use the Teamcenter Model Event Manager Shared Session

As a system administrator, you can opt in to use the Teamcenter Model Event Manager shared session. The Teamcenter Model Event Manager manages event synchronization across SOA clients sharing the same Teamcenter server instance. Events that happen in one SOA client are broadcast out to all other connected SOA clients such that they can respond to the change.

If you configure this setting, users cannot change it.

Procedure

1. Edit the *config.template* files for each Teamcenter add-in:

   - *TcExcelAddin.dll.config.template* (for Microsoft Excel)
   - *TcOutlookAddin.dll.config.template* (for Microsoft Outlook)
   - *TcPowerPointAddin.dll.config.template* (for Microsoft PowerPoint)
   - *TcWordAddin.dll.config.template* (for Microsoft Word)

   These files are located in the folder where the Office client is installed.

Note:

If an administrator wants to pre-configure various settings prior to deployment, they would modify the *.config.template* files in the installation kit to make the desired changes to configuration settings. These *.config.template* files are then used during the installation process to programmatically generate the respective *.config* files. This way, every user does not have to make these changes to configuration settings as the installation process generates the files with all the desired configuration settings. Once the Office Client add-ins have been installed, the only way to make a change is to edit each of the *.config* files on each user's machine. If a user edits the *.config.template* files at this point it makes no difference. The *.config.template* files are only used during the installation process.

To summarize, an administrator can either modify the *config.template* files in the software depot before deploying Client for Office, or if Client for Office is already deployed, the administrator or user must update the *.config* files on each client workstation.

2. To specify the types of items to be excluded, within each *config.template* file, add the following XML code in the **tcresource** section:

```
<SharedSession optin="true" />
```

You can assign either of the following values to the **SharedSession optin** parameter:

- false – to opt out of using the Teamcenter Model Event Manager shared session.
- true - to opt in of using the Teamcenter Model Event Manager shared session.

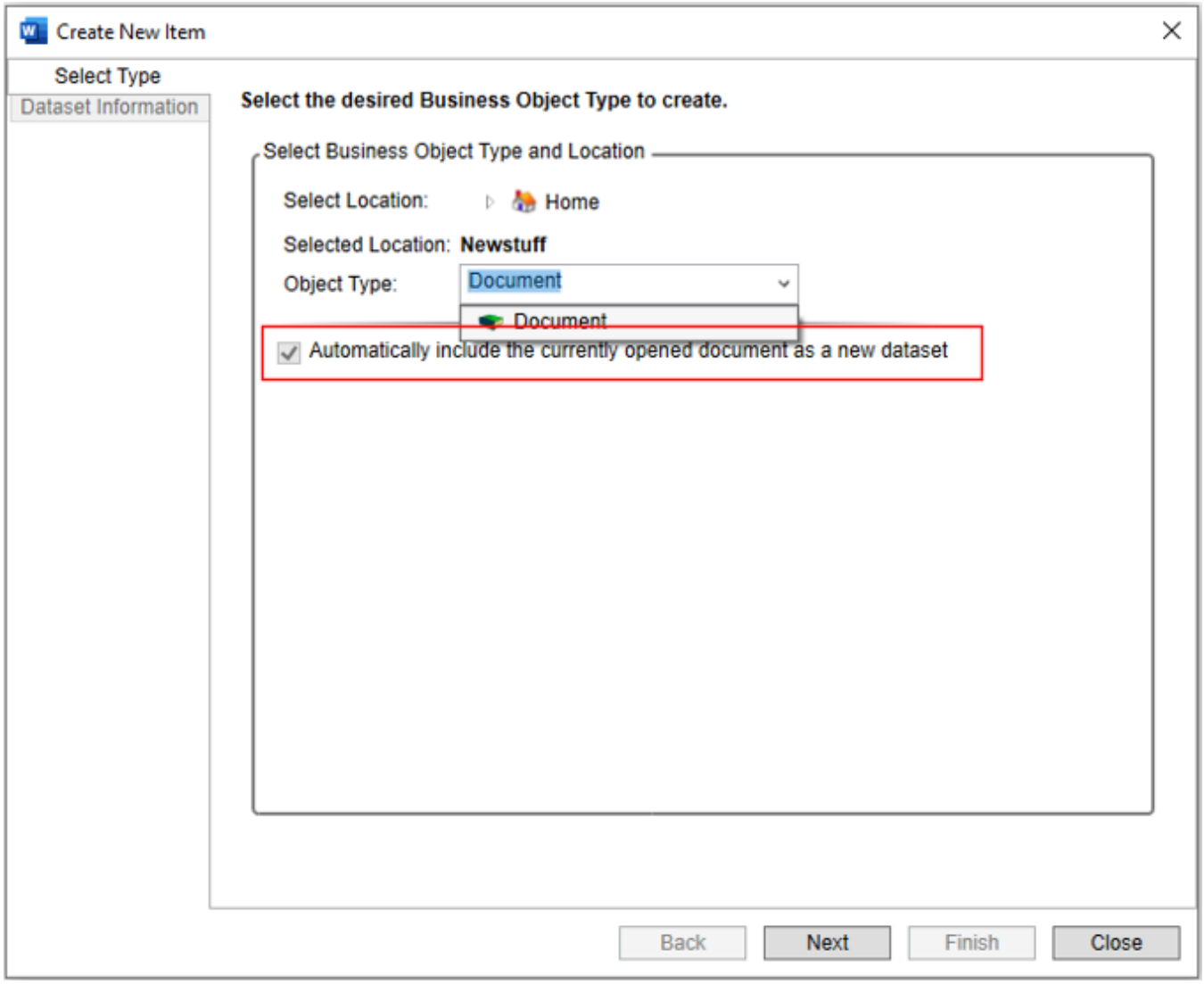
3. You must modify the **config.template** files before running the installation process.

Note:

If the Teamcenter Client for Office was installed using the stand-alone installer (**setup.exe**), the system administrator can use the modify feature of the installer to generate the **.config** files again.

Include the currently opened document as a new dataset

As an administrator, you can define if the current document should be included as a new dataset when creating a new document object. In the *app.config* file, you define if the **Automatically include the currently opened document as a new dataset** checkbox is selected by default and whether users can change the default value.



To minimize process mistakes, Siemens Digital Industries Software recommends that you do not allow users to change this selection.

```
<createdataset createdataset="true" allowUserToModify="false"/>
```

If the **createdataset** parameter is set to **true**, the checkbox is selected by default, and if it is set to **false**, the checkbox is deselected by default.

> Note:
>
> The **createdataset** option is only valid for **New→Item** menu option. The checkbox is always checked by default for the **Save As → Item with Dataset** menu option.

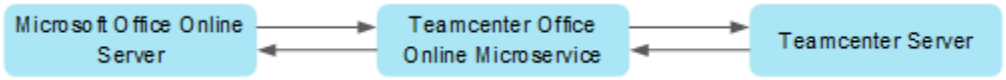
If the **allowUserToModify** parameter is set to **true**, users can select or deselect the checkbox. However, if the **allowUserToModify** parameter is set to **false**, users cannot change the state of the checkbox.

4. Setting up Microsoft Office Online within Teamcenter

About setting up Microsoft Office Online within Teamcenter

Using the Microsoft Office Online Server features, users can edit and view documents within the web client instead of using the desktop version of the Microsoft Office applications. This eliminates the need to install Microsoft Office on client machines. However, if your authors and reviewers wish to use the desktop version, you must **enable the Teamcenter Client for Microsoft Office feature**, which embeds the Teamcenter menu within Microsoft Office applications.

Components required for setting up Microsoft Office Online



To access the browser-based version of Microsoft Word, PowerPoint, and Excel within the web client, you require Microsoft Office Online Server. The web client connects to both the Microsoft Office Online Server and the Teamcenter server through the Teamcenter Office Online Server.

When a user selects a file to view it in the web client viewer:

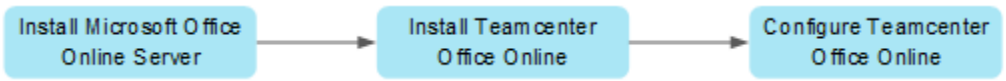
1. The web client connects to the Teamcenter Office Online microservice to get the Office Online Server launch URL.

2. The Teamcenter Office Online microservice in turn connects to the Microsoft Office Online Server to download the discovery XML used for generating the launch URL.

3. The Microsoft Office Online Server connects to the Teamcenter Office Online microservice to obtain the file information and content to be displayed in the web client viewer.

4. The Teamcenter Office Online microservice connects to the Teamcenter server to fetch the file information and content required by the Microsoft Office Online Server.

Process flow for setting up Microsoft Office Online



Install Microsoft Office Online Server

To access the browser-based version of Microsoft Word, PowerPoint, and Excel, you require Microsoft Office Online Server. To install this, perform the steps specified in *Deploy Office Online Server* to prepare the server that will run Office Online Server and to deploy the Office Online Server farm.

While preparing the server, if your site requirement is to view web-based Office files in multiple languages, install the language packs for Office Online Server.

There are several ways to deploy the Office Online Server farm. Depending on your deployment strategy, follow the installation steps specified in *Deploy Office Online Server farm*.

Install Teamcenter Office Online through Deployment Center

Add the Teamcenter Office Online application to your existing Teamcenter environment.

Prerequisites

- Microservice framework is installed.
- Microsoft Office Online Server is installed on a separate Windows machine.

Procedure

1. Log on to Deployment Center and select the environment to which you want to add Teamcenter Office Online.
2. Go to the **Applications** tab. Click **Add or Remove Selected Applications**.
3. In the **Available Applications** panel, select the **Teamcenter Office Online** application, and then click **Update Selected Applications**.
4. Go to the **Components** tab.
5. In the **Selected Components** list, perform the following tasks:
   - Click the percent link available for the **Corporate Server** component, and then enter the values for its configuration parameters in the **Teamcenter Office Online Settings** section.

Field	Action
Use Sponsored Authentication	If you want to use the sponsored authentication, click **True** and then enter the credentials of the Teamcenter *Sponsored Authentication* user. If you do not want to use the sponsored authentication, click **False**.

Field	Action
	Note: For a sponsor, do not use a user account that is typically created for administrative purposes, such as infodba, dba, administrator, or dcproxy. Instead, you must always create a new **Sponsor User ID** and **Sponsor User Password** that is specific for usage with sponsored authentication for Teamcenter Office Online. Moreover, for security reasons, Siemens Digital Industries Software does not recommend the usage of IDs that can be easily guessed such as administrator, infodba, dba, or dcproxy.

- Click the percent link available for the **Microservice Node** component, and then enter the values for its configuration parameters in the **Teamcenter Office Online Settings** section.

Field	Action
Protocol	Select **HTTP** or **HTTPS** considering your security settings. If you select **HTTPS**, the certifcate path and credentials information is provided by the **HTTPS Config** component page. Note: On Docker deployments only the gateway and service dispatcher components can use https. The Teamcenter Office Online microservice will only use http regardless of what is selected here.
Microsoft Office Online Server Discovery URL	Enter the reference URL of the deployed Microsoft Office Online Server. A sample URL is `http://office_online_servername:portnumber/hosting/discovery`.

- Click the percent link available for the **Microservice Node** component, and then enter the reference URL of the deployed Microsoft Office Online Server for **Microsoft Office Online Server Discovery URL**. A sample URL is `http://host_office_online_server/hosting/discovery`.

After you finish entering values for each component, click **Save Component Settings**.

6. In the **Selected Components** list, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter required parameters, and save component settings until all components in the environment show a configuration status of **100%**.

   When all components are fully configured, the **Deploy** tab is enabled.

7. Go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts you will use to update affected machines.

   When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

8. Locate deployment scripts, copy each script to its target machine, and then run each script on its target machine.

   For more information about running deployment scripts, see *Deployment Center — Usage*.

Install Teamcenter Office Online through TEM

Add the Teamcenter Office Online application to your existing Teamcenter environment.

Prerequisites

- Microservice framework is installed.
- Microsoft Office Online Server is installed on a separate Windows machine.

Procedure

1. Launch TEM.
2. In the **Features** panel, select the following options that are appropriate for the current deployment:
   - **Teamcenter extensions**

     Extensions→Teamcenter Office Online
   - **Active Workspace server extensions**

     Active Workspace→Server Extensions→Office Online Viewer
   - **Microservices**

     Microservices→Teamcenter Office Online Microservice
   - **Active Workspace client**

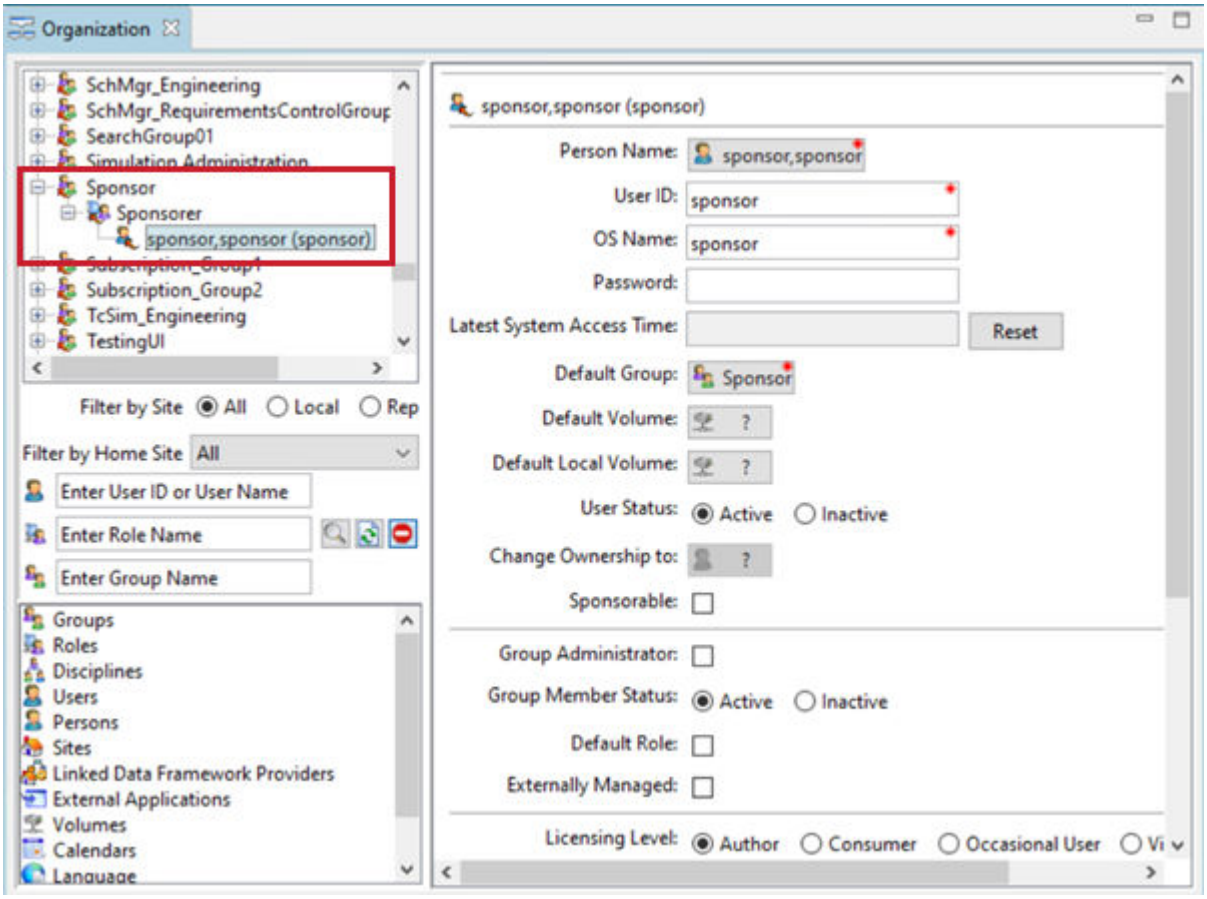
Active Workspace→Client→Office Online Viewer Client

3. In the **Teamcenter Office Online Microservice** panel, enter the following information:

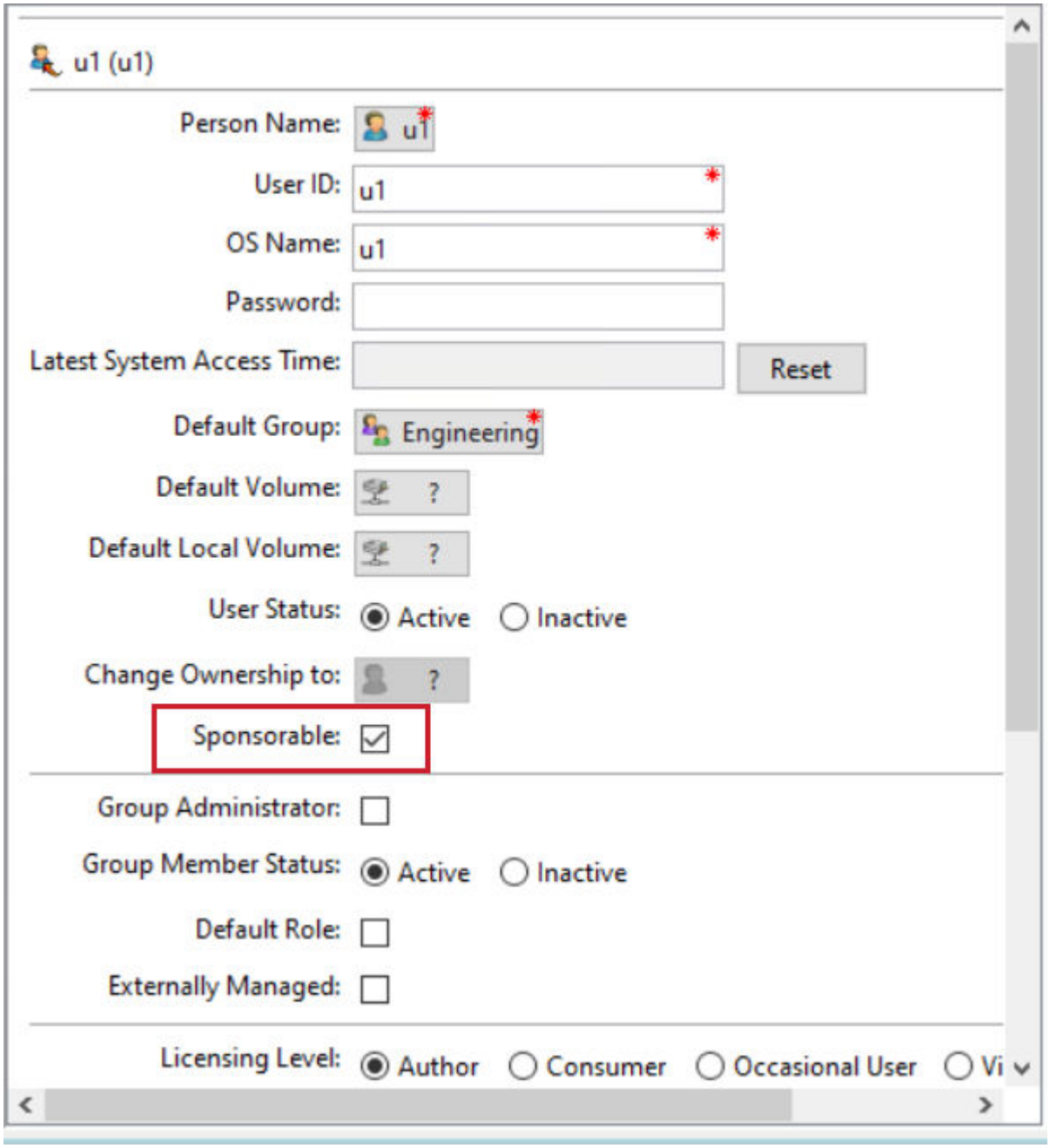
Field	Action
Microsoft Office Online Server Discovery URL	Enter the reference URL of the deployed Microsoft Office Online Server. A sample URL is `http://host_office_online_server/hosting/discovery`.
Use Sponsored Authentication	If you want to use the sponsored authentication, select **Use Sponsored Authentication** from the **Sponsor User Information** section and then enter the credentials of the Teamcenter *Sponsored Authentication* user. Note: For a sponsor, do not use a user account that is typically created for administrative purposes, such as infodba, dba, administrator, or dcproxy. Instead, you must always create a new **Sponsor User ID** and **Sponsor User Password** that is specific for usage with sponsored authentication for Teamcenter Office Online. Moreover, for security reasons, Siemens Digital Industries Software does not recommend the usage of IDs that can be easily guessed such as administrator, infodba, dba, or dcproxy.
Use HTTPS Protocol	To use the HTTPS protocol, select **Use HTTPS Protocol** from the **Protocol Selection** section and then enter the certificate path and credentials. Note: On Docker deployments only the gateway and service dispatcher components can use https. The Teamcenter Office Online microservice will only use http regardless of what is selected here.
Active WorkspaceGateway URL	Enter the web client Gateway URL in the **Active Workspace Gateway URL** field. To use single sign on, select **Teamcenter SSO**.

4. Clicking **Next** until you reach the **Confirmation** panel. Click **Start** to install Teamcenter Office Online.

5. If you want to use the sponsored authentication user in Teamcenter, you must manually create this privileged user in the Teamcenter Organization application under the **Sponsor** group and **Sponsor** role.



If you want to use the sponsored authentication, you must make a user *sponsorable* by enabling the sponsorable flag for users to view and edit Office files in the web client.



Configuring Teamcenter Office Online

About configuring Teamcenter Office Online

The Teamcenter Office Online is configured automatically when you deployed it. To make any changes later, you can use Deployment Center or Teamcenter Environment Manager. Alternatively, manually change the configurations as described below:

- **Update the sponsored authentication user credentials**.

- **Update the Gateway and Service Dispatcher URLs**.

Further, if you have set up Secure Socket Layer and single sign-on in your environment, you can (optionally) perform the following:

- **Update the Secure Socket Layer (SSL) protocol**.
- **Enable single sign-on**.

> Warning:
>
> When you are configuring, ensure you:
>
> - Do not enter the custom properties linked to a cell in Excel as they are not supported.
> - Change the property of the document only from the **Properties** panel in the web client.

Update the sponsored authentication user credentials

If you selected to use the sponsored authentication and then created the user while installing Teamcenter Office Online and Teamcenter Google Online, use the following information to update its credentials.

You can update these credentials in the `TcSecurity` section of the *appsettings.Security.json* file. However, as this section is encoded by default, you must first decode it by running the following command in the command prompt window:

```
JsonSettingsConfigUtility -decode JsonSettingsConfig.json
```

Next, update the credentials:

```
"TcSecurity":{
   "Sponsorer_Name": "Sponsor1",
   "Sponsorer_Password": "default*PW!shouldBeChanged",
}
```

After changing the credentials, you must encode the `TcSecurity` section by running the following command in the command prompt window:

```
JsonSettingsConfigUtility -encode JsonSettingsConfig.json
```

Guidelines to update the secret of the running tcooweb docker swarm service

The *appsettings.Security.json* file is copied to each running container as a secret at startup. To update the secret of a running tcooweb docker swarm service, use the following guidelines:

- To get the name of the running tcooweb service:

```
docker service ls
```

- To get a list of current secrets:

```
docker secret ls
```

- Create a new secret:

```
docker secret create <new secret name> <path to and name of updated secret
file>
```

- Update the service to use the new secret:

```
docker service update --secret-rm <current secret name> --secret-
add source=<new secret name>,target=/run/secrets/appsettings.Security.json
<service name>
```

Example:

Suppose that the name of the existing secret is `tcooweb-appsettings.Security.json`, the name of the service is `myStack_tcooweb_service`, and the folder where the updated *appsettings.Security.json* file is located at is `tcooweb_service-2.0.0`. In such a case, use the following commands:

```
docker secret create tcooweb1-appsettings.Security.json
$TC_ROOT/microservices/tcooweb_service-2.0.0/
appsettings.Security.json
docker service update --secret-
rm tcooweb-appsettings.Security.json
--secret-add source=tcooweb1-appsettings.Security.json,target=/run/
secrets/appsettings.Security.json myStack_tcooweb_service

successful output:
overall progress: 1 out of 1 tasks

1/1: running
==================================================>]

verify: Service converged
```

Note:

You must update the same user information in the Teamcenter Organization application.

Update the Gateway and Service Dispatcher URLs

You can update the Gateway and Service Dispatcher URLs in the `TcConnection` section of the *appsettings.json* file. This file is located at :

- **Windows**

  TC_ROOT\microservices\tccooweb_service-version\TcOOWeb

- **Linux**

  TC_Root/microservices/tccooweb_service-version

In the `TcConnection` section, you can change the Gateway URLs (**GatewayUrls**) and the Service Dispatcher URLs (**ServiceDispatcherUrls**), as necessary. The Gateway and Service Dispatcher URLs can be a list of comma-separated or semicolon-separated URLs.

The **appsettings.json** file is copied to each running container at the startup. To update the service, please see **Update the *appsettings.json* file in a Docker swarm deployment**.

Update the Secure Socket Layer protocol

You can use https for communication from the Teamcenter Office Online microservice to other services. The Secure Socket Layer protocol could be previously configured during the installation. However, you can also manually update these settings after deployment.

Perform the following steps if you want to use https for communication from the Teamcenter Office Online microservice to other services. For more information see, *Create a Teamcenter environment using Deployment Center* topic in Teamcenter Installation Using Deployment Center help.

1. Acquire the SSL certificate. The certificate is in the **X.509** format.

2. Open the *appsettings.json* file located at:

   - **Windows**

     TC_ROOT\microservices\tccooweb_service-version\TcOOWeb

   - **Linux**

     TC_Root/microservices/tccooweb_service-version

     The *appsettings.json* file is copied to each running container at startup. To update the service, **update the *appsettings.json* file in a Docker swarn deployment**.

3. Edit the **Kestrel** section:

   a. Change **http** to **https**.

   b. Specify the certificate file name in the certificate **Path** value, or rename the certificate file name to *certificate.pfx* and retain the certificate **Path** value as is.

   Example:

```
"Kestrel":{
   "Endpoints":{
      "Https":{
         "Url": "https://*:0",
         "Certificate":{
            "Path": "certificates/certificate.pfx"
          }
       }
   },
  }
```

4. Decode the *appsettings.Security.json* file.

```
JsonSettingCongigUtility -decode JsonSettingsConfig.json
```

5. Open the *appsettings.Security.json* (located in the same folder as *appsettings.json*) and set the **CertificatePassword** of the host SSL to the certificate's password.

   Example:

```
"Kestrel":{
   "Endpoints":{
      "Https":{
         "Certificate":{
            "Password": "ceritificatepassword"
         }
       }
   }
}
```

6. Regenerate the *JsonSettingsConfig.json* file.

```
JsonSettingsConfigUtility -generateConfigFile JsonSettingsConfig.json
appsettings.Security.json TcSecurity Kestrel
```

7. Encode the *appsettings.Security.json* file.

```
JsonSettingCongigUtility -encode JsonSettingsConfig.json
```

8. Copy the *.pfx* file to:

   - **Windows**

     tcooweb_service-version\TcOOWeb\certificates

   - **Linux**

     tcooweb_service-version/certificates

   If you copy the *.pfx* file to a different location, enter the location path as the **Certificate Path** value in the *appsettings.json* file located at:

   - **Windows**

     TC_ROOT\microservices\tccooweb_service-version\TcOOWeb

   - **Linux**

     TC_Root/microservices/tccooweb_service-version

     The *appsettings.json* file is copied to each running container at the startup. Therefore, when the file is updated, restart the containers for the changes to take effect.

9. For Linux, open the *tcooweb_service.yml* file:

   a. On the host machine, copy the root CA certificate in `pem` format to */etc/pki/ca-trust/source/anchors and run update-ca-trust*.

   b. Uncomment the **-source**, **target**, and **mode** parameters in the **configs** section after **appsettings.json** .

```
configs:
   -appsettings.json
- source: tls-ca-bundle.pem
  target: /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
  mode: 0755
```

   c. Uncomment the following lines in the lower configs section:

```
configs:
  appsettings.json:
    file: /tc_deployment_path/microservices/service-directory/
appsettings.json
    name: tcooweb-appsettings.json
```

```
  jsonsettingsconfig.json:
    file: /tc_deployment_path/microservices/service_directory/
JsonSettingsConfig.json
    name: tcooweb-JsonSettingsConfig.json
# Uncomment the following lines when CA trusts need to be
updated in the container
  tls-ca-bundle.pem:
    file: /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
    name: tcooweb-tls-ca-bundle.pem
```

> Note:
>
> If Using SuSE 15 or later use file: `/var/lib/ca-certificates/ca-bundle.pem`

Enable single sign-on

If single sign on is set up at your site, then you must enable it. To do so, in the *appsettings.json* file, ensure that the connection node has SSO enabled:

```
"TcConnection": {
   "SsoEnabled": "true"
}
```

The *appsettings.json* file is located at:

- **Windows**

  TC_ROOT\microservices\tccooweb_serivce-version\TcOOWeb

- **Linux**

  TC_Root/microservices/tccooweb_serivce-version

You can **update the *appsettings.json* file in a Docket swarm deployment**.

Update the appsettings.json file in a Docker swarm deployment

To update the *appsettings.json* file in a Docker swarm deployment, use the following guidelines:

- Get the name of the running tcooweb service: `docker service ls`
- Get a list of current configs: `docker config ls`
- Create a new config: `docker config create <new config name> <path to and name of updated config file>`

- Update the service to use the new config:

```
docker service update --config-rm <current config name> --config-add
source=<new config name>,target=appsettings.json <service name>
```

Example:

Suppose that the name of the existing config is `tcooweb-appsettings.json`, the name of the service is `myStack_tcooweb_service` and the folder where the updated *appsettings.json* file is located is *$TC_ROOT/microservices/tcooweb_service-2.0.0*. In such a case, use the following commands:

```
docker config create tcooweb1-appsettings.json
$TC_ROOT/microservices/tcooweb_service-2.0.0/appsettings.json
docker service update --config-rm tcooweb-appsettings.json
--config-add source=tcooweb1-appsettings.json,
target=appsettings.json myStack_tcooweb_service

successful output:

overall progress: 1 out of 1 tasks
1/1: running
==================================================>]

verify: Service converged
```

Guidelines to configure reverse proxy

Use the following guidelines to configure reverse proxy:

- Set up your Microsoft Office Online server with an InternalUrl and an ExternalUrl where the InternalUrl is the direct link to the Microsoft Office Online server and the ExternalUrl is the URL for the reverse proxy server.

- Enable SSL on both the reverse proxy server and the Microsoft Office Online server. Use one certificate that supports Subject Alternate Names for both these servers.

- Do not make any changes to the `Tcooweb` section of the *appsettings.json* file. The Gateway URL, service dispatcher URL and Office Online Discovery URL can all refer to the internal values.

- Microsoft Office Online does not support a virtual path. Make an entry for the Microsoft Office Online server in the reverse proxy configuration files so that all traffic (not already handled by previous entries) defaults to the Microsoft Office Online server.

```
ProxyPass / https://myInternalOfficeOnlineURL.com/
ProxyPassReverse / https://myInternalOfficeOnlineURL.com/
```

- If you are using Apache for your reverse proxy server, include the following setting in the configuration file:

  ```
  ProxyPreserveHost On
  ```

  Otherwise, the embedded javascript and CSS objects that the Microsoft Office Online server send will try to use the internal URL.

  > Note:
  >
  > For Tcgoogleweb, use the internal gateway URL in the *appsettings.json* file.

Troubleshooting Teamcenter Office Online

Error while opening a document in the web client

While opening a document in the web client, a user may receive the following errors:

- **The Office Online Server could not be accessed.**

  This error occurs when the **launchInfo** call fails due to a connection failure between Teamcenter Office Online Server and Microsoft Office Online Server.

  To resolve this issue:

  - Confirm the error. To do so:
    - Open the development tools (F12) in a browser.
    - Reproduce the issue.
    - Go to the **Network** tab and locate the **launchInfo** response.
    - Verify the error in the response.
  - Verify if the Microsoft Office Online Server is up and running. To do so, use the browser to check if the Discovery XML URL format is correct. The Discovery XML URL format is `https://ooshostname:port/hosting/discovery`
  - Verify if the Teamcenter Office Online microservice and the Microsoft Office Online Server are connected. Additionally, verify that the host name is correct and that the route is allowed.

- **The requested service, Office Online, does not exists.**

  This error occurs when Teamcenter Office Online is not deployed on the corporate server.

To resolve this issue, if your deployment setup has multiple servers such as a corporate server and a Dispatcher server, you must install Teamcenter Office Online on the corporate server.

- **There was a problem and the document cannot be opened.**

  To resolve this issue:

  - Verify the response of the alive link in a browser and two links must be verified:

    ```
    <service dispatcher URL>/tcooweb/v1/wopi/alive

    <gateway URL>/micro/tcooweb/v1/wopi/alive
    ```

    > Example:
    >
    > If the service dispatcher URL is `https://myteamcenter:9090`, the alive link must be `https://myteamcenter:9090/tcooweb/v1/wopi/alive`.

  - Verify if the **Teamcenter Office Online service log files** shows any new activity from the Microsoft Office Online Server.

Unable to view documents in the web client

If users are not able to view documents in the web client, ensure that the client connection to the web client and the client connection to Office Online Server are the same. Both the client connections must use either HTTP or HTTPS. For example, if the client connection to the web client is HTTPS, the client connection to Office Online Server must also be HTTPS.

You must also ensure that Office Online Server is deployed with an external URL irrespective of whether Secure Socket Layer (SSL) is terminated at Office Online Server or at a load balancer. For more information, see **https://docs.microsoft.com/en-us/officeonlineserver/deploy-office-online-server**.

Unable to connect to Microsoft Office Online Server on unspecified domains

The connection issue can be due to how Microsoft Office Online Server was configured and related to the mis-configured domain access.

To resolve this issue, refer to the following commands:

- **Get-OfficeWebAppsHost**: **https://docs.microsoft.com/en-us/powershell/module/officewebapps/get-officewebappshost?view=officewebapps-ps**
- **New-OfficeWebAppsHost**: **https://docs.microsoft.com/en-us/powershell/module/officewebapps/new-officewebappshost?view=officewebapps-ps**

Document icon displayed in the web client instead of the file

If the web client displays a document icon instead of a file, it may be due to some missing components. To resolve this issue, verify that the following values are included in the **AWC_defaultViewerConfig.VIEWERCONFIG** preference:

- **MSWordX.Awp0TcooViewer=word,Fnd0word**
- **MSWord.Awp0TcooViewer=word**
- **MSExcelX.Awp0TcooViewer=excel**
- **MSExcel.Awp0TcooViewer=excel**
- **MSPowerPoint.Awp0TcooViewer=powerpoint**
- **MSPowerPointX.Awp0TcooViewer=powerpoint**

If any value is missing, do not add it manually. A missing value indicates that the preference is not set correctly as the **Office Online Client Viewer** feature is not installed. To resolve this, run Teamcenter Environment Manager to install the feature.

Teamcenter login failure

If you used the sponsored authentication and then created a user while installing Teamcenter Office Online, and then a pop-up message appears and contains a message that there was a Teamcenter login failure, verify that the following configurations are correct:

- The **Sponsor user is created in Teamcenter** and the sponsor user's password is set.
- The Sponsor user is created in the LDAP if you are using TcSS.
- The user who is trying to open a document is marked as **Sponsorable in Teamcenter**.
- The Sponsor user information is correct and encoded in the *appsettings.Security.json* file. This information must match the sponsor user information in Teamcenter. (see **Update the sponsored authentication user credentials**).
- The gateway URL is correct in the *appsettings.json* file.

Intermittent failures in opening a document

If a document opens successfully, but then the next document that you try to open results in an error message, the Teamcenter server might not be configured to support enough warm processes.

Perform the following steps to troubleshoot:

1. Login to the corporate server and go to *TC_ROOT/pool_manager/confs/config-name/serverPool.properties*.

2. Search the PROCESS_WARM parameter and set the value to at least **15**.

3. Save the changes and restart the Teamcenter Server Manager Pool service.

   (See System Administration→Server manager properties files→Server manager pool-specific configuration tuning→Setting the PROCESS_WARM parameter for more information on optimizing this value.)

4. If the Teamcenter environment is configured with a load balancer and multiple web-tiers, ensure the session ID cookie is added as a sticky header so that the same web tier associated with the session ID can be targeted from Teamcenter Office Online.

Teamcenter Office Online service log files

You can refer to the Teamcenter Office Online service log files to troubleshoot issues found while setting up Microsoft Office Online within Teamcenter. You can view the log files in the log aggregator , which collects logs from distributed microservices on various microservice nodes and forwards the logs to a single location.

By default, the log level is **Information**. To troubleshoot, change the log level to **Debug**. To do so, open the *appsettings.json* configuration file, and change the **MinimumLevel** value of **Serilog** from **Information** to **Debug**.

The *appsettings.json* file is located at:

- **Windows**

  TC_ROOT\microservices\tccooweb_service-version\TcOOWeb

- **Linux**

  TC_Root/microservices/tccooweb_service-version

  The *appsettings.json* file is copied to each running container at the startup. To update the service, **update the *appsettings.json* file in a Docker swarm deployment**.

5. Installing and configuring Extensions for Office

Installing Teamcenter Extensions for Microsoft Office

Before you install Extensions for Office

Choose an installer

Extensions for Office can be installed alone or as part of a Teamcenter rich client configuration (two-tier or four-tier). Siemens Digital Industries Software provides the following ways to install Extensions for Office.

Stand-alone installation wizard	Installs Extensions for Office on a single host through a step-by-step interface. This wizard does not install the Teamcenter rich client.
Teamcenter Environment Manager (TEM)	Installs Extensions for Office alone or as part of a two- or four-tier rich client configuration.

Note:

- You must have administrative privileges to install Extensions for Office, but you do not need any special permissions to use Extensions for Office after installation.
- Extensions for Office does not include the Teamcenter plugin for Microsoft Project. The plugin for Microsoft Project is available in **Teamcenter Client for Microsoft Office**.

Install required software

Teamcenter Extensions for Microsoft Office requires the following software on your client host:

Install Microsoft Office Word

Install Microsoft Office Excel (Professional, Professional Plus, or Enterprise edition), 64-bit version.

For certified versions, see the Hardware and Software Certifications knowledge base article on Support Center.

Install Microsoft Libraries

If you install Extensions for Office using TEM, install the required 64-bit Microsoft libraries *before* you install Extensions for Office. The stand-alone Extensions for Office installation wizard installs

these libraries if they are not present, but may interrupt the installation for system restarts. Installing these libraries requires administrative privileges.

Uninstall any 32-bit versions of these libraries before installing the 64-bit versions. Download these libraries from **Microsoft Corporation** and then install:

- Microsoft .NET Framework.
- Microsoft Office Primary Interop Assemblies (PIAs) for Office.
- Microsoft Office Language Pack for English for 64-bit Windows. (Required if you use a non-English version of Windows because .NET add-ins require English support.)

For certified versions of these libraries, see the Hardware and Software Certifications knowledge base article on Support Center.

Enable .NET framework programmability support for Microsoft Office applications

1. In the Windows Control Panel, open the **Add or Remove Programs** dialog box.
2. In the list of installed programs, locate the Microsoft Office application. Select the program name in the list, and then click **Change**.
3. In the maintenance mode options dialog box, select **Add or Remove Features**, and then click **Continue** to display the installation options.
4. In the **Installation Options** dialog box, expand the features tree under **Microsoft Excel**.
5. To the left of **.NET Programmability Support**, click the down arrow, and then choose **Run from my computer**.

   > Note:
   >
   > The **.NET Programmability Support** option is available if you have installed Microsoft .NET Framework.

6. Click **Continue** to apply the changes.

   The Microsoft Office installer may prompt you for location of Microsoft Office installation media to apply the changes.

Install Extensions for Office using the stand-alone installation wizard

1. Close all instances of Microsoft Excel and the Teamcenter rich client before you run the Extensions for Office installer.

2. In the Teamcenter 2412 software kit, browse to the **additional_applications\tc_ext4mso** directory and double-click the **tc_ext4mso.exe** program icon to launch the Extensions for Office installation wizard.

3. If the prerequisite Microsoft libraries are not present on the client host, the installation wizard installs them. If this installation requires a restart, the wizard resumes after restart.

4. Proceed to the **Select Features** dialog box. Select the **Teamcenter Extensions for Microsoft Office** feature.

5. In the **Choose Destination Location** dialog box, enter the location in which to install Extensions for Office.

6. In the **Start Copying Files** dialog box, review your selections. Click **Back** to change your selections or click **Next** to install Extensions for Office.

Some additional configuration in Microsoft Office is required to enable Teamcenter Extensions for Microsoft Office.

Installing or updating Extensions for Office using a Teamcenter patch kit

You can install Extensions for Office or update an existing installation using a Teamcenter patch kit, for example, Teamcenter 2412.0001. Run the **tc_ext4mso.exe** program from the patch kit. When the wizard displays the **Teamcenter Kit Location** dialog box, enter the path to the dependent Teamcenter software kit.

> Note:
>
> Not all Teamcenter patch kits contain an updated Extensions for Office installation program.

Install Extensions for Office using TEM

Launch Teamcenter Environment Manager (TEM) from the Teamcenter 2412 software kit or, if available, a Teamcenter patch kit (for example, Teamcenter 2412.0001) to install Extensions for Office.

1. **Install the prerequisite software** on the client host. TEM does not install the prerequisite software.

2. Locate the Teamcenter software kit.

   Alternatively, you can install Extensions for Office from a compact distribution.[1]

1 A *compact distribution* is a Teamcenter software installation package created by your Teamcenter administrator that contains a selected subset of Teamcenter features. This package is smaller than a full Teamcenter software kit and is more easily distributed. However, if you attempt to install a feature not included in the compact distribution, TEM prompts for the location of a full Teamcenter software kit.

3. Start TEM:

   a. Browse to the root directory of the Teamcenter software kit.

   b. Double-click the **tem.bat** program icon.

4. In the **Welcome to Teamcenter** panel, select **Teamcenter**.

   For information about any panel in TEM, click the help button  .

5. In the **Install/Upgrade Options** panel, click **Install**.

6. If you are running TEM from a Teamcenter patch kit, the **Media Locations** panel contains the **Original Media Location** box. In this box, enter the path to the dependent Teamcenter kit specified.

7. In the **Configuration** panel, enter an ID and a description for the new Teamcenter configuration.

   Proceed to the **Features** panel.

8. In the **Features** panel, select the **Teamcenter Extensions for Microsoft Office** feature (under **Extensions→Systems Engineering and Requirements Management**).

   > Note:
   >
   > You may include additional features in your Teamcenter configuration. If you select additional features, TEM displays additional panels during installation that are not described in this procedure.

9. In the **Teamcenter Extensions for Microsoft Office**, TEM reports whether all the prerequisite libraries and settings are present on your system. Review any instructions in this panel and click **Next** to continue.

10. Proceed to the **Confirmation** panel and review your selections. Click **Start** to install Extensions for Office, or click **Back** to change your selections.

Configuring Extensions for Microsoft Office (live Excel)

Add the Excel live add-in

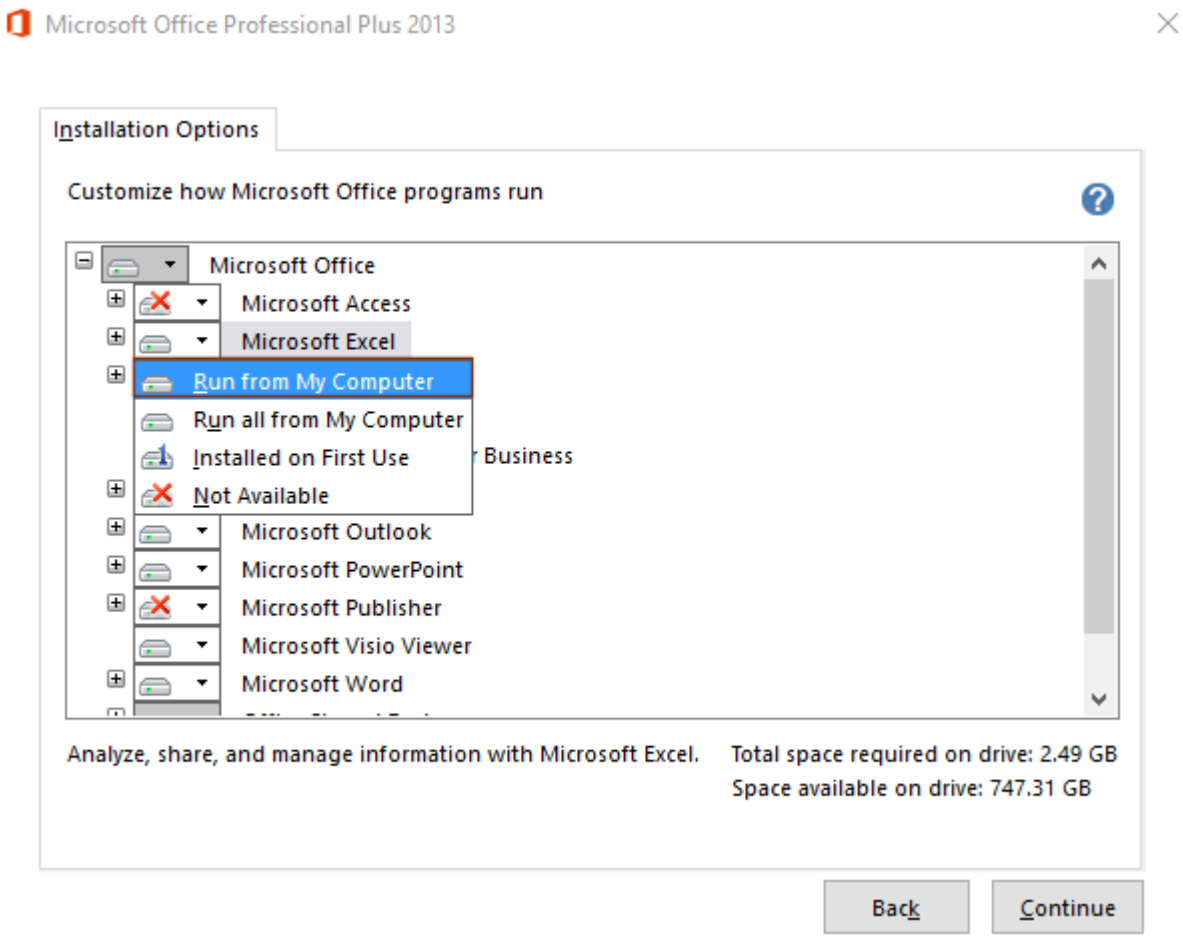
1. In the Windows Control Panel, double-click **Add/Remove Programs** to display the **Add/Remove Programs** dialog box.

2. In the **Currently installed programs** pane, select your Microsoft Office version, and then click **Change** to display the Microsoft Office Setup program.

3. Select **Add or Remove Features**, and then click **Next**.

4. Without changing the list of selected applications, select **Choose advanced customization of applications**, and then click **Next**.

5. With **Microsoft Office** expanded, expand **Microsoft Office Excel**, and then select the **Run from my computer** option for **.NET Programmability support**.



6. Click **Update**.

   If the setup program asks for the Microsoft Office installation CD-ROM or network location, insert the CD-ROM or enter the information, and then click **Next** to complete the setup.

Enable users to optionally checkout objects before editing them in live Excel

1. In Teamcenter, choose **Edit→Options** to display the **Options** dialog box.

2. In the bottom left corner of the dialog box, click **Index** to display the **Preferences By Filter** window.

3. Type **show_** in the box in the **Filters** area.

4. Select the **Show_Checkout_option** row in the **Preferences List** table.

   The definition of the preference is displayed in read-only mode in the pane on the right side of the dialog box.

5. Click the **Edit** button at the bottom of the pane.

6. In the **Value** box, type **true**.

7. Click **Save**, and then click the **Close** button in the bottom-right corner of the dialog box.

6. Setting up the synchronization of Microsoft Office attributes with Teamcenter

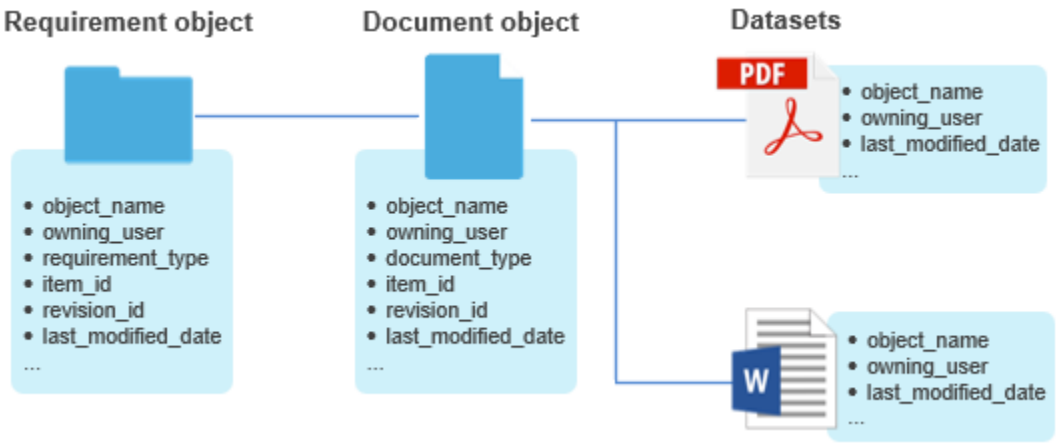
About logical objects

A *logical object* is a container that consolidates attributes (properties) from different related business objects. For example, a document (item) and its related datasets are different objects in Teamcenter. The attributes of the document can be its ID, name, or description. The different attributes of a dataset can be author, created date, or reviewed by.

The following example shows how you can synchronize attributes, consolidated by the logical object, from Teamcenter to a Microsoft Word document and to a PDF file.

Example

Consider that there is a requirement object in Teamcenter which has a reference relation to a document object. The document object has two attached files (datasets), one Word and one PDF.

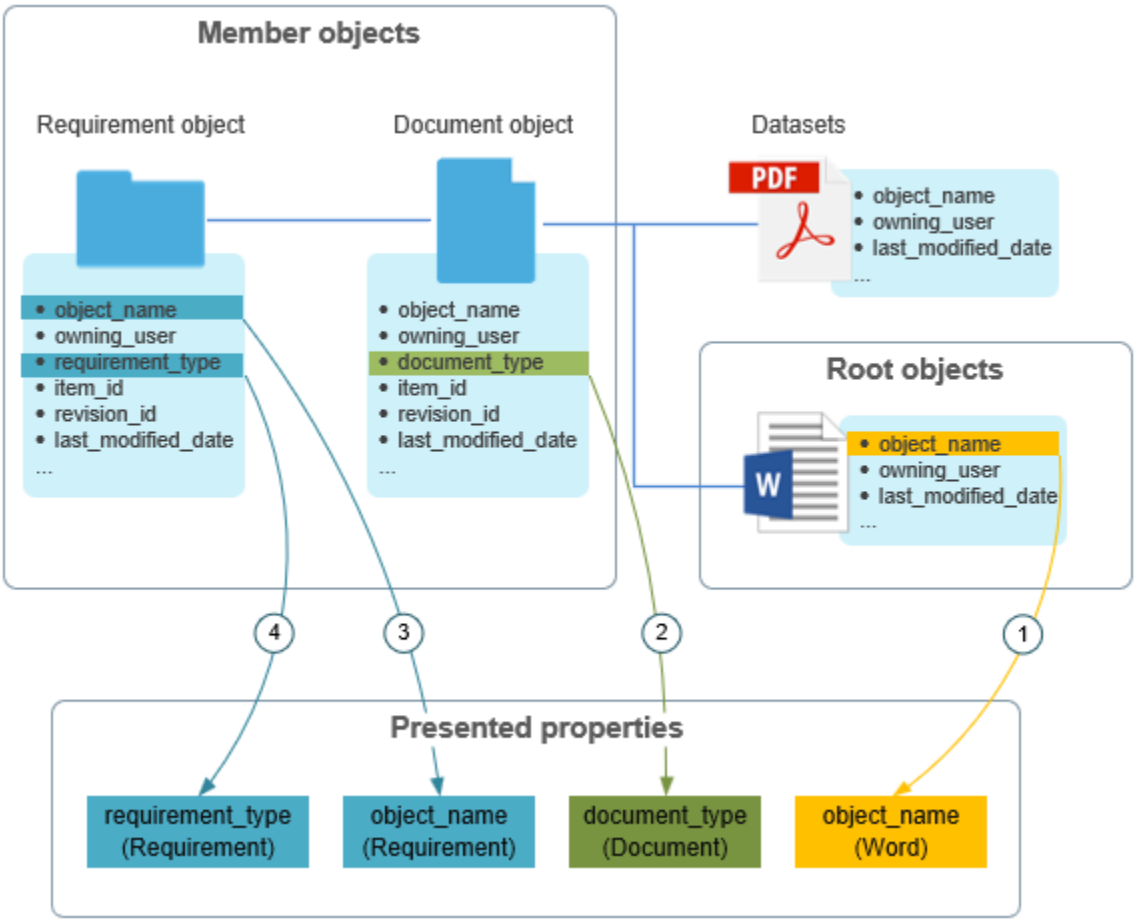


You want to consolidate four attributes from these objects:

- **object_name** and **requirement_type** from the requirement object.
- **document_type** from the document object.
- **object_name** from the Word or PDF file attached to the document.

For consolidating these attributes, you define a logical object. A logical object is composed of *root object*, *member objects*, and *presented properties*. You begin consolidating the properties from the

dataset. This makes the dataset the root object. You then traverse backward tracing its relation to the document object and then to the requirement object. This makes the document and requirement objects the member objects. While traversing from the dataset to the document object and then to the requirement object, you collect the required four attributes in the logical object. These four attributes are the presented properties.



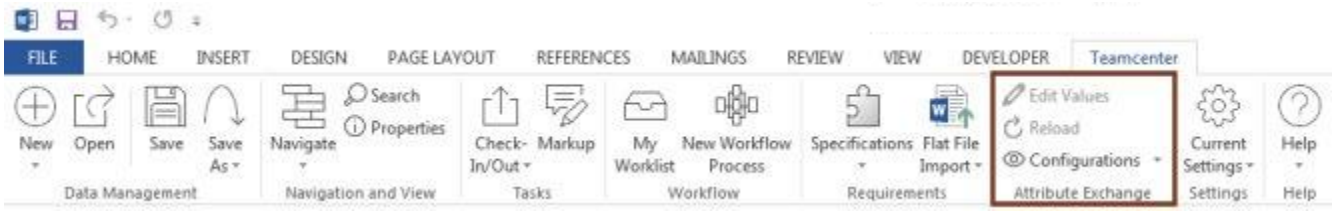
After consolidating the attributes in a logical object, you use the logical object to synchronize attributes between Teamcenter and Microsoft Office files and PDF files. When you synchronize attributes between Teamcenter and Microsoft Office files, you cannot synchronize attributes from Teamcenter to Office files and from Office files to Teamcenter at the same time.

> Note:
>
> Even if you used attribute-level security while implementing Teamcenter, the properties that are used for attribute exchange are still synchronized in the document. The attribute exchange overrides the attribute-level security.

Synchronizing attributes between Teamcenter and Microsoft Office file by using Client for Office

Exchanging attributes between Office files and Teamcenter datasets

With Client for Office attribute exchange, you can synchronize property values between Teamcenter datasets and Microsoft Office files. In Word, Excel, and PowerPoint, these features are available through the **Attribute Exchange** button group on the **Teamcenter** tab of the Office ribbon.



Note:

- Attribute exchange is not available in Outlook.
- The property values are synchronized between Teamcenter and Microsoft PowerPoint using **Attribute Exchange**. However, PowerPoint does not allow you to use the synchronized properties in the content of the slides.
- Even if the attribute-level security is used while implementing Teamcenter, the properties that are used for attribute exchange are shown in the document. The attribute exchange overrides the attribute-level security.

You can map attributes on individual datasets and on item revision definition configurations (IRDCs).

- The mappings on a single dataset do not apply to any other datasets.
- When a Teamcenter item with a related dataset is created using an IRDC template with attribute mapping, the mapping is applied to the new dataset.

  If an item and its related dataset are created based on an IRDC template, and then the **Save As** command is used to create a new dataset from the original, the attribute mappings associated with the IRDC template can be applied to the duplicate dataset.

Mapped properties are updated automatically when you do any of the following:

- Open and check out a Teamcenter dataset.
- Save changes to the same dataset.

- Save changes as a new dataset.

You can map the following Teamcenter property types:

Boolean
Date
Float
Integer
Text

> Note:
>
> The following Teamcenter properties are not supported:
>
> - System properties, such as **Date Created** and **Item ID**
> - Properties for which *property constants* are defined in the Business Modeler IDE
> - Properties with dynamic and cascading lists of values (LOVs)

Setting up Attribute Exchange for Microsoft Word and Microsoft Excel on an Item Revision

Attribute Exchange is a way to link properties in a Microsoft Office application to Teamcenter attributes. To create the mapping, open the Microsoft Office dataset in its native application and create mappings as follows:

- File to Teamcenter (unidirectional) - Mapped property values go from the Microsoft Office application to Teamcenter.
- Teamcenter to File (unidirectional) - Mapped Teamcenter attribute values go to the Microsoft Office application property values.
- Two way (bidirectional) - Mapped Teamcenter attribute values go to and come from the mapped Microsoft Office application property values depending on where they were last changed.

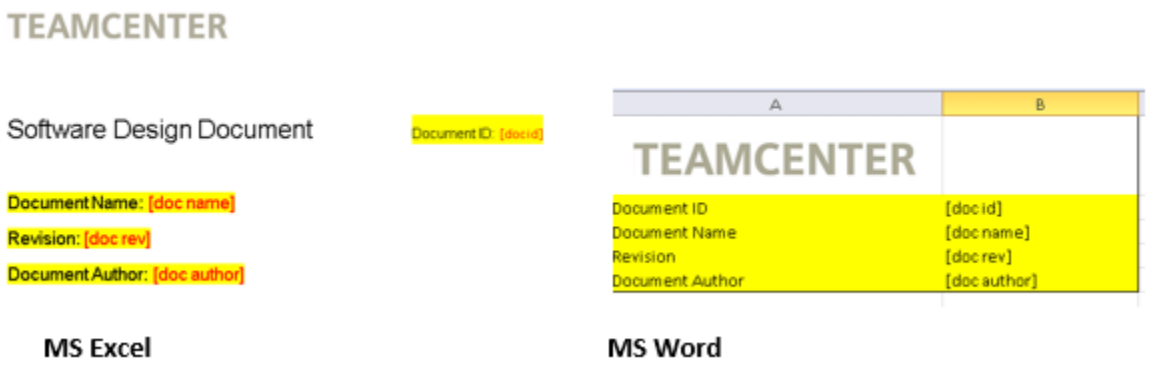
Because datasets are Teamcenter objects and can have attributes, the mapping can be directly to the Microsoft Office dataset attributes. However, this is impractical for most users as they want to pull existing Teamcenter attributes which exist on item revisions. When you set up the mapping to an item revision, the Microsoft Office dataset is usually attached to the item revision as the mapping is dependent on the relation path to the revision. Additionally, you want to be able to create a custom item and have the mapped Microsoft Office dataset attached so that you do not have to map to the dataset each time. To do this, you must use the Item Revision Definition Configuration (IRDC).

In this example, we set up attribute exchange with a Microsoft Word dataset or a Microsoft Excel dataset to an item or item revision or their child objects such as a document or document revision, or

any custom item or custom item revision. The datasets are created from an existing Word document or an Excel workbook. The task is to automatically populate the following fields with values from the document or document revision or any custom item or custom item revision. The office document properties and the Teamcenter properties they map to are as follows:

Office document properties	Mapped to Teamcenter properties
Document ID	Document's item_id (display name as **ID**)
Document Name	DocumentRevision's object_name (display name as **Name**)
Revision	DocumentRevision's item_revision_id (display name as **Revision**)
Document Author	User's owning_user (display name as **Name**)

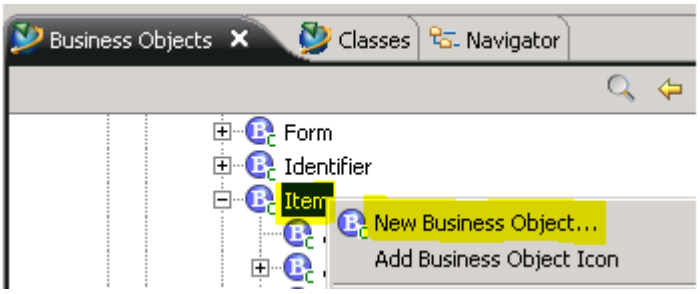
- Document ID: [doc id]
- Document Name: [doc name]
- Revision: [doc rev]
- Document Author: [doc author]



Step 1: Create the custom item or custom item revision and the properties on the revision

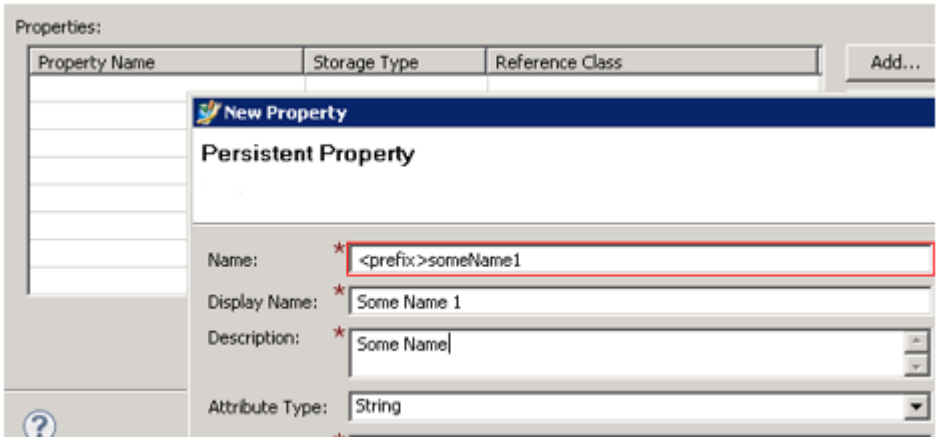
To map to custom properties on a custom item or custom item revision, you must first create the business objects. If you use COTS document or documentRevision as example above, skip the steps below.

1. Open BMIDE and search for the item.
2. Right-click the item and select **New Business Object**.



3. Set the **Name** to <*prefix*>**CustomItem**.
4. Set the **Display Name** to **CustomItem**.
5. Click **Next**.
6. Set the **Display Name** to **CustomItem Revision**.

7. Click the **Add** button in the **Properties** section.
8. Create a property named <*prefix*>**someName1**.
9. Set the **Display Name** to **Some Name 1**.
10. Click **Finish**.



11. Repeat the property creation for the following:

- Create a property named <*prefix*>**someName2**. Set its **Display Name** as **Some Name 2**.
- Create a property named <*prefix*>**someName3**. Set its **Display Name** as **Some Name 3**.

> Note:
>
> Similarly, you can create other properties as required.

12. Save and deploy the BMIDE template. See the Teamcenter guide on the deployment process.

Step 2: Create a sample of the document or document revision (or customItem or customRevision) and attach the respective dataset

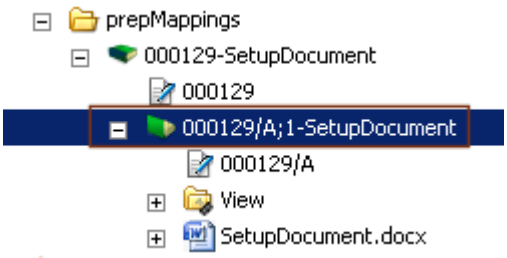
To map to properties on a document or document revision or custom item or custom item revision, let us create sample objects to attach the Microsoft Office dataset. This shows how to map the Microsoft Office properties to Teamcenter attributes.

1. In the rich client, in My Teamcenter, create a folder named **prepMappings** under the **Newstuff** folder.

2. Select the **prepMappings** folder.

3. Create a document named **SetupDocument** (or create the CustomItem). Expand the item so that the item revision is displayed.

4. Drag and drop the Microsoft Office file and onto the respective item revision to create the Microsoft Office dataset (MSWordX or MSExcelX).



The MSWordX dataset is related to the documentRevision with the **Attaches** (**TC_Attaches**) relation. (other relation can be used instead also).

If you use COTS document/documentRevision in the example above, skip the following steps.

5. If you use customItem or customItemRevision, select the customItemRevision. Right-click the revision and select **Edit Properties**. Checkout if necessary. Update the properties (Some Name 1, Some Name 2, ..., Some Name N) with some values.

6. Click **Save** and then click **Save and Check-In**.

Step 3: Open the dataset in the Teamcenter Client for Microsoft Office application and set the mappings

The mapping configuration is created in the Client for Office Ribbon on the Attribute Exchange pane. Microsoft Word uses fields to display the mapped attributes, while Microsoft Excel uses named references to display the mapped attributes.

In Microsoft Word:

1. In Microsoft Word, select the Teamcenter ribbon (installed with the Teamcenter Client for Microsoft Office).

2. Select **Current Settings →Login**.

3. Authenticate using your Teamcenter credentials.

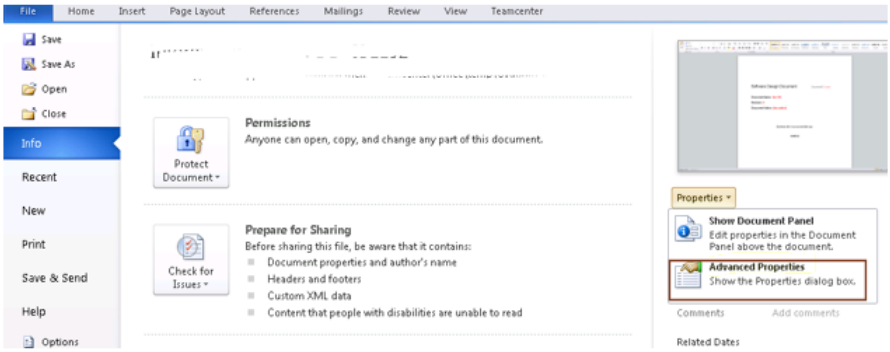
4. Click **Navigate→Browse**.

5. In the navigation pane, browse to the Microsoft Word dataset attached to the SetupDocument Revision (or create the CustomItem Revision).

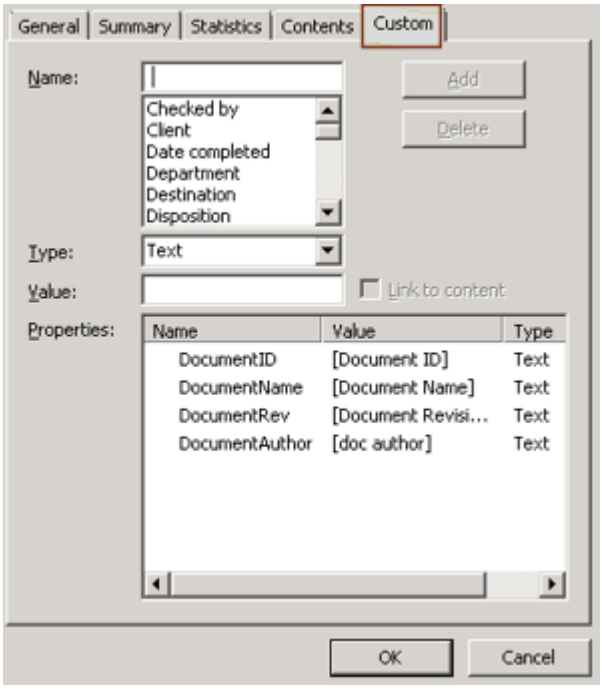
6. Right-click the dataset and select **Open** and **Check-Out File**.

7. Configure the Office Document Properties on the MSWord file.

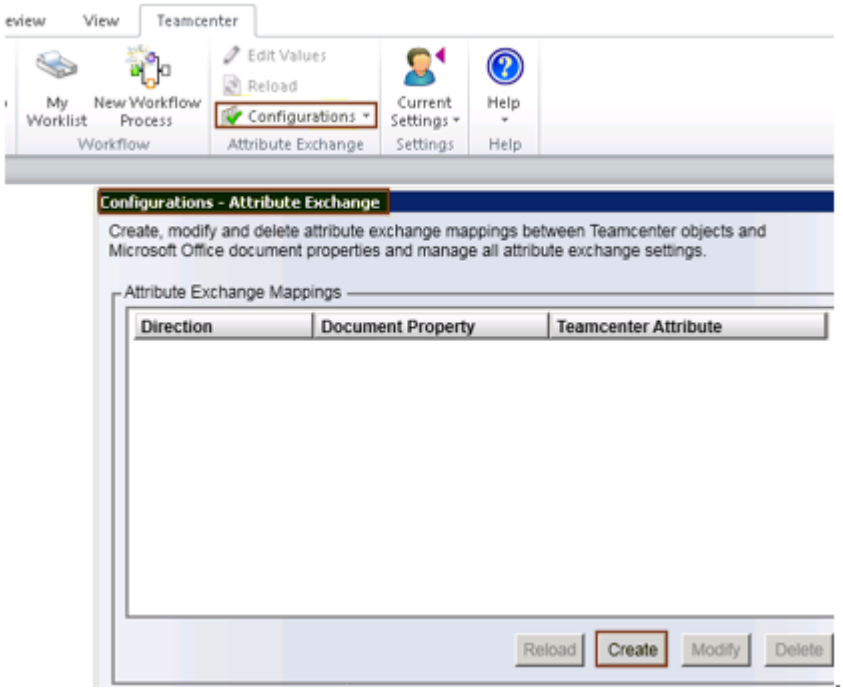
8. Choose **File→Info→Properties→Advanced Properties**.



9. In the Advanced Properties, select **Custom** tab and add the office properties such as DocumentID, DocumentName, SomeName1, and SomeName2. Click **OK**.

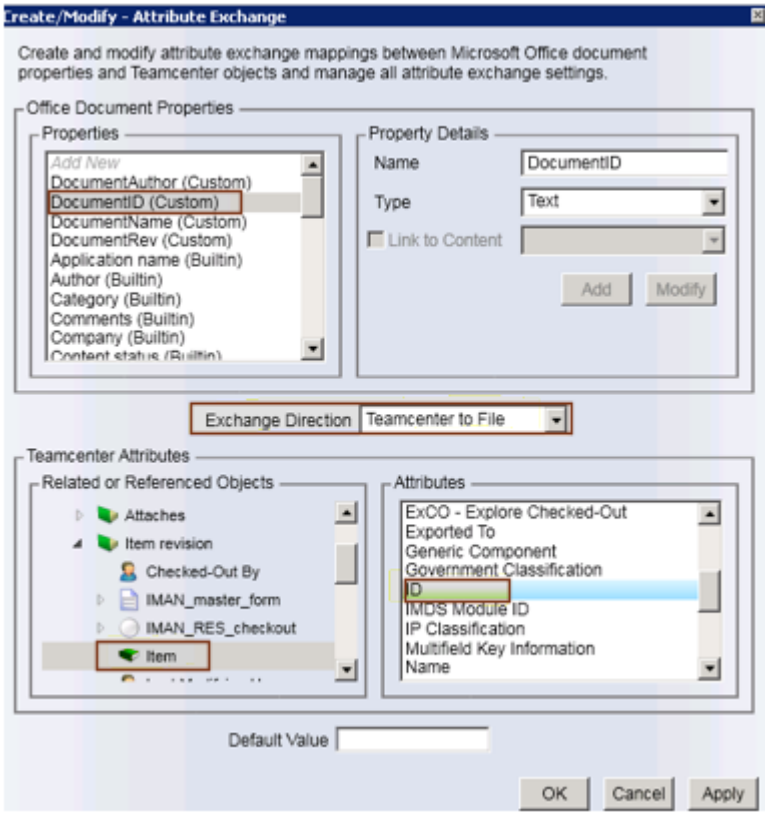


10. To configure the attribute exchange mapping, on the Teamcenter ribbon, select **Configurations→Create** in the **Attribute Exchange** pane.



11. To pull Teamcenter attributes into Microsoft Word, set the **Exchange Direction** to **Teamcenter to File**.

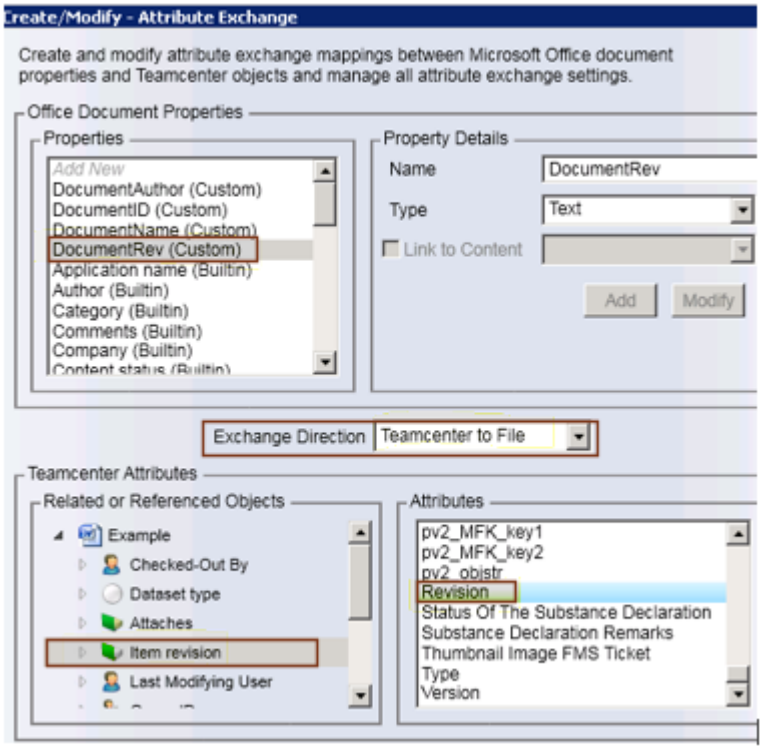
12. In the top section, configure the Microsoft Office document properties, and in the bottom section, configure the Teamcenter attributes.



13. In the Office Document Properties, select the property for the attribute exchange with Teamcenter.

14. Now to link the Teamcenter Attribute, notice that the document ID property is not in the list of Attributes initially and this is because the **Related or Referenced Objects** section has the Word Dataset selected. In example above, expand the Item revision and select the Item. Scroll through the **Attributes** list and select the attribute (for example, ID).

15. To view the Teamcenter Attributes on the SetupDocument Revision, select item revision in the **Related or Referenced Objects** section.



16. Click **Apply**. Repeat step 13 to 16 for linking other attributes.

17. After all of the mappings are created, click **OK**.

18. To verify the Teamcenter attribute mapping, select **Configurations→All Configurations** in the **Attribute Exchange** pane and click the **Close** button.

19. In the **Attribute Exchange** pane, click **Reload**.

    The attributes that are refreshed to the Microsoft Word dataset are displayed.

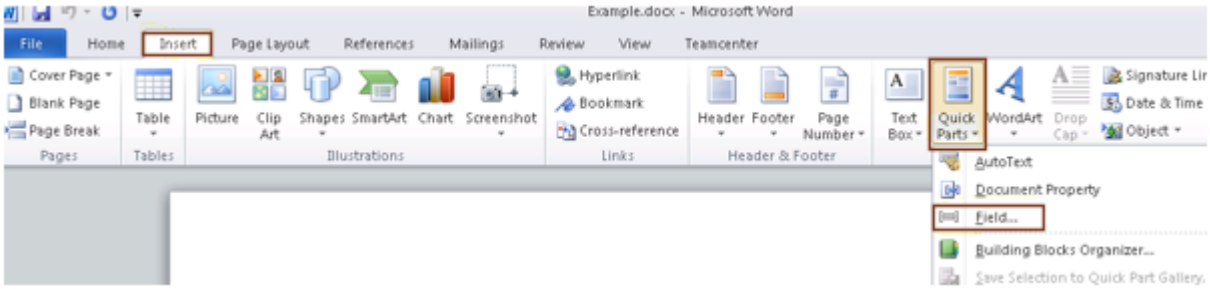
20. Click the **OK** button.

21. Verify that the Microsoft properties now have the same values by viewing the Microsoft Word Advanced Properties on the **Custom** tab. (**File** >**Info** >**Properties** > **Advanced Properties** >**Custom** tab)

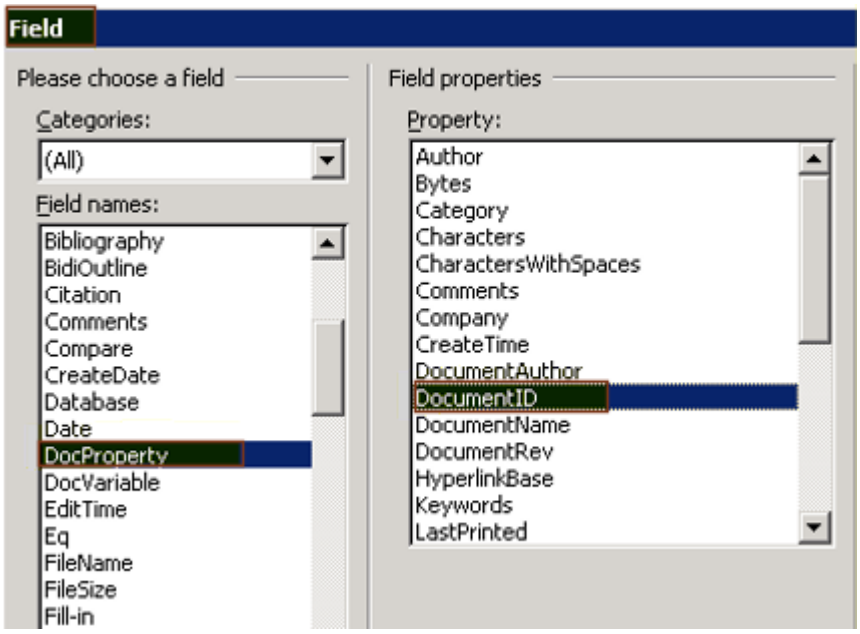
22. Click the **OK** button.

23. To insert the value of an office property in the Word document, highlight the location in the Word document where you want the Document ID attribute to be displayed for example.

24. Select the **Insert** ribbon then **Quick Parts→Field**.



25. Select the **Field Name** as **DocProperty** and **Property** as **DocumentID**.



26. Click **OK**.

    The value inserted because of the mapping is displayed.

27. Similarly repeat steps 24 to 26 to create other properties.

28. On the Teamcenter ribbon, click the **Save** button in the **Data Management** pane.

29. When presented with the **Confirm Check-In** dialog box, click **Yes**.

30. Close Microsoft Word.

In Microsoft Excel:

1. Open Microsoft Excel and select the Teamcenter ribbon (installed with the Teamcenter Client for Microsoft Office).

2. Select **Current Settings→Login**.

3. Authenticate using your Teamcenter credentials.

4. Click **Navigate→Browse**.

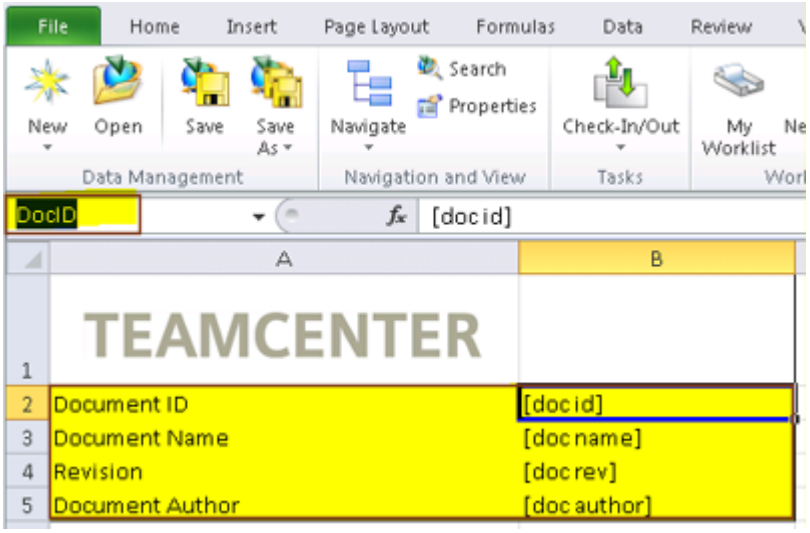
5. In the navigation pane, browse to the Microsoft Excel dataset attached to the SetupDocument Revision (or create the SetupCustomItem Revision).

6. Right-click the dataset and select **Open and Check-Out File**.

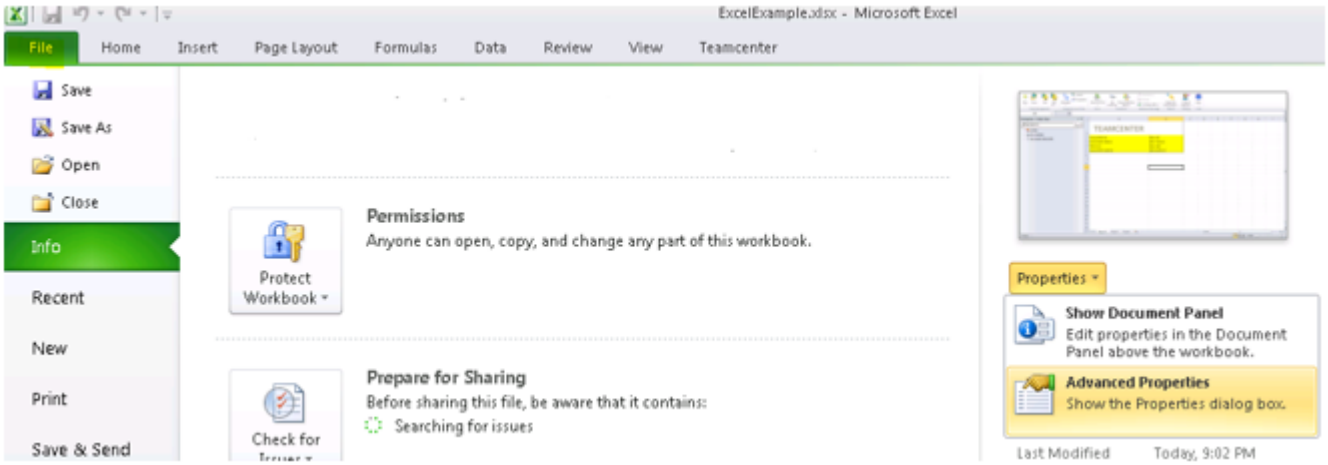
7. Configure the Office Document Properties on the MSWord file.

8. To link an office property to a cell on the spreadsheet, you must rename that cell.

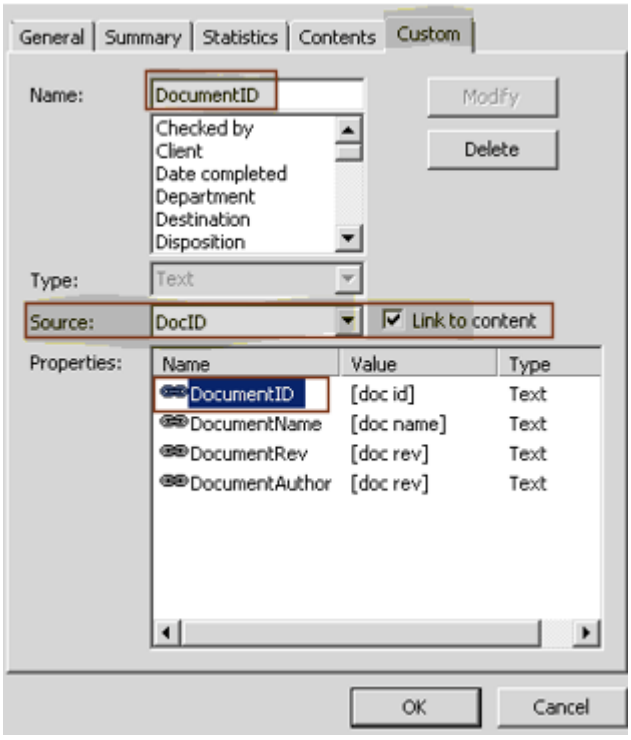
   For example, the B2 cell is renamed to DocID. Similarly, repeat the same for the other properties.



9. Select **File→Info→Properties→Advanced Properties** from the menu.

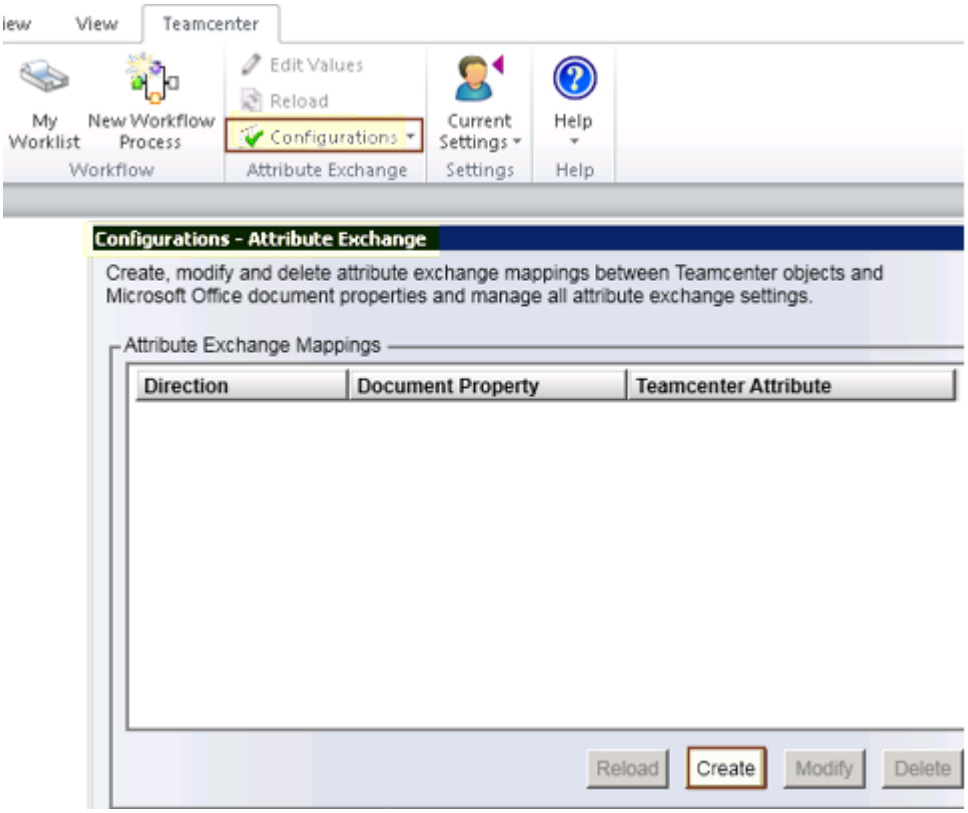


10. In the Advanced Properties, select **Custom** tab and add the office properties such as DocumentID, DocumentName, SomeName1, and SomeName2.



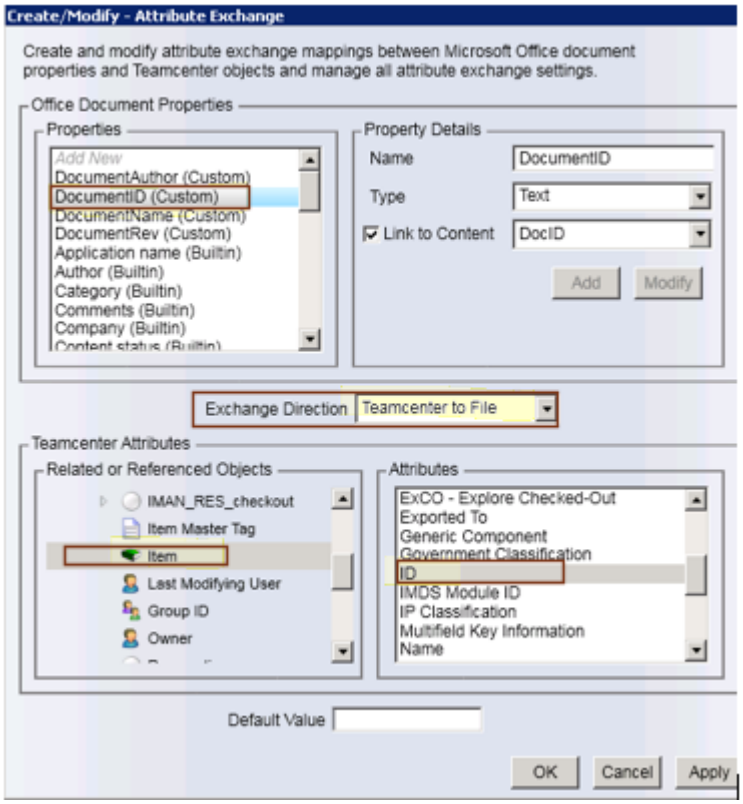
For example, add **DocumentID** in the **Name** field. Check the box **Link to content**. Scroll the **Source** field and select **DocID** (that was renamed in step 9). Repeat the same for other properties. Click **OK**.

11. To configure the attribute exchange mapping, on the Teamcenter ribbon, select **Configurations→Create** in the **Attribute Exchange** pane.



12. To pull Teamcenter Attributes into Microsoft Excel, set the **Exchange Direction** to **Teamcenter to File**.

13. The top section is to configure the Microsoft Office document properties and the bottom one is for the Teamcenter attributes.



14. In the Office Document Properties, select the property for the attribute exchange with Teamcenter.

15. Now to link the Teamcenter Attribute, notice that the document ID property is not in the list of **Attributes** initially and this is because the **Related or Referenced Objects** section has the Excel Dataset selected. In example above, expand the item revision and select the Item. Scroll through the **Attributes** list and select the attribute (for example, ID).

16. To view the Teamcenter Attributes on the SetupDocument Revision, select item revision in the **Related or Referenced Objects** section.

17. Click the **Apply** button. Repeat step 13 - 17 for linking other attributes such as DocumentName and DocumentRev.

18. To verify the Teamcenter Attribute mapping, select **Configurations→All Configurations** in the **Attribute Exchange** pane and click the **Close** button.

19. In the **Attribute Exchange** pane, click **Reload**.

    The Teamcenter attribute values are immediately updated in the Excel Workbook.

20. Click the **OK** button.

21. Verify that the Microsoft properties now have the same values by viewing the Microsoft Excel Advanced Properties on the **Custom** tab. (**File** >**Info** >**Properties** >**Advanced Properties** >**Custom** tab)

    Click **OK**.

22. On the Teamcenter ribbon, click the **Save** button in the **Data Management** pane.

23. When presented with the **Confirm Check-In** dialog, select **Yes**.

24. Close Microsoft Excel.

Step 4: Create document templates to store the mapped datasets

To create the document or custom items with the configured datasets, you need to configure IRDC in BMIDE. First, in the Teamcenter rich client you must create document templates to store the configured datasets. This creation of document templates is necessary for the IRDC configuration.

To create the document templates:

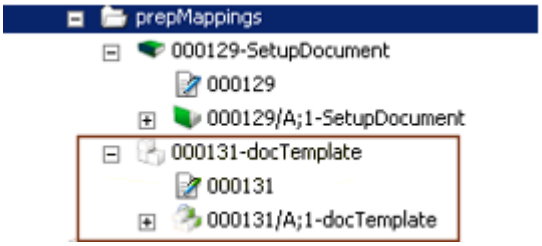
1. Select the **prepMappings** folder.

2. Select **File** →**New**→**Item** from the menu.

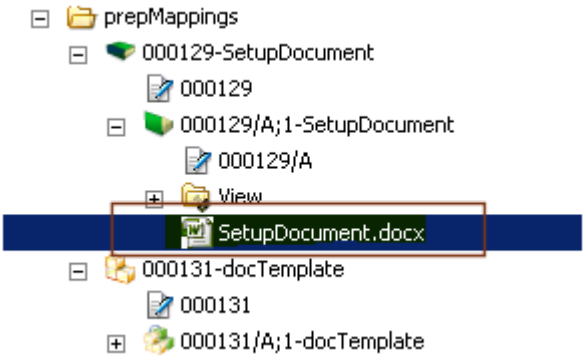
3. Select the document template and click **Next**.

4. Name the document template as **docTemplate** and click **Finish**.

5. Expand the **docTemplate** so that you can see the revision.



6. Select the MSWordX dataset attached to the SetupDocument revision and cut it by selecting **Edit**→**Cut**.



7. Select the docItemTemplate revision and select **Edit→Paste**.

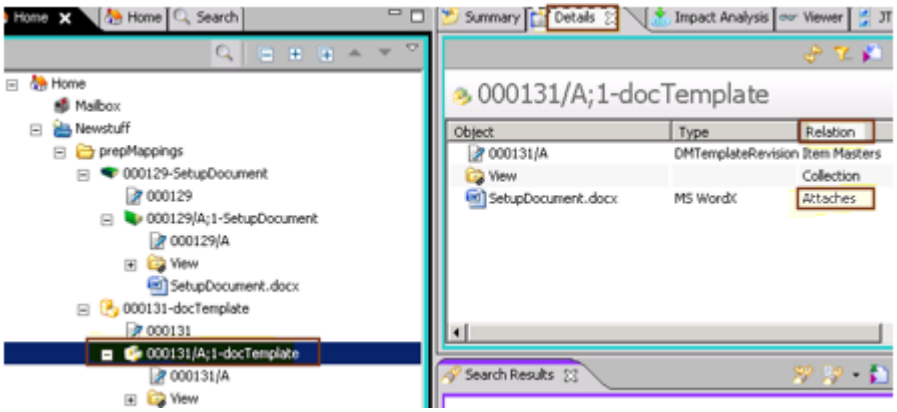
8. Select the **SetupDocument**.

9. Select **Edit→Delete** to remove it from Teamcenter.

To configure the document templates and datasets:

The document template revision and attached dataset must be in a released status. Also the dataset must be connected via a relation Attaches (TC_Attaches) in this example. Select the docTemplate revision then select the **Details** tab.

1. Select the docTemplate revision and select the **Details** tab.



2. If the Relation for the MSWordX Dataset is not Attaches, cut the MSWordX dataset and select the docTemplate Revision and select **Edit→ Paste Special**. Scroll down to select the **Attaches** relation.

3. Select docITemplate revision and the MSWordX dataset (or MSExcel dataset). You can press the CTRL key and click to select.

4. Select **File →New→Workflow Process**.

5. Select **TCM Release Process** as the **Process Template**.

6. Exit the Teamcenter rich client.

Step 5: Create IRDC in BMIDE

The final step is to create an IRDC for the custom items and map them to their respective document templates in the Teamcenter rich client. After deploying the new template they should be available for use.

1. Go to the **Extension** section in BMIDE.

2. Expand your project then expand Document Management.

To set up the IRDC for the docTemplate revision:

1. Right-click **IRDC** and select **New IRDC**.

2. Name the IRDC docTemplateIRDC (or customTemplateIRDC) for example.

3. Set the **Applies to Business Object** to **DocumentRevision**.

4. If you use your CustomRevision, then select CustomRevision.

5. Ensure that the **Condition** is set to **isTrue**.

   > Note:
   >
   > You can create a new condition for DocumentRevision (or CustomRevision) and use it.

6. Set the **Create Template** to **docTemplate**. (This selection is done via BMIDE connecting to Teamcenter and presenting choices.)

7. Click **Next**.

8. On the **IRDC Dataset Criteria** page, click **Add** in the **Source Dataset** area.

9. Set the **Source Dataset** to **MSWordX** or **MSExcelX**.

10. Set the **Item Revision Relation** to **TC_Attaches**.

11. Click **Finish**.

12. Click **Next**.

    The IRDC Dataset Naming page is displayed.

13. Click **Next**.

The IRDC Checkin page is displayed.

14. Click **Next**.

    The IRDC Rules page is displayed.

15. Set **Delete Data file** to **No**.

16. If you need to add any deep copy rules, you can add them here.

17. Click **Finish**.

18. Click **Finish**.

19. Click **Next**.

    The **Enter Markup Information** page is displayed.

20. Click **Finish**.

21. Save and deploy the BMIDE template.

Step 6: Verify

Test by creating a new document (or new customItem). If testing with your CustItem, populate the custom properties. Go to Client for Office. Browse to and open or checkout the MSWordX dataset (or MSExcelX dataset) attached to the DocumentRevision. The reload should occur automatically to display the objects.

MS Word

OR

MS Excel

Directional settings for mapping attributes

The attribute map for a dataset includes a setting for the direction of the exchange.

- **File to Teamcenter**

  The value of the Office **Comments** property is copied to the Teamcenter **Description** attribute.

- **Teamcenter to File**

  The system copies the Teamcenter **Name** attribute value, where **Name** is the owner of the Teamcenter object, to the **Author** document property. The value is copied to the document.

- **Two Way**

  When you save a document to Teamcenter, the system copies the **Title** document property value to the Teamcenter **Name** attribute, where **Name** is the object name. The value is copied to Teamcenter. When you open and check out a Teamcenter dataset, the system copies the Teamcenter **Name** attribute value, where **Name** is the object name, to the **Title** document property. The value is copied to the document.

Caution:

Properties with dynamic and cascading lists of values (LOVs) are not supported.

You can view the attribute mapping for a dataset and, if needed, modify the mapping direction.

Updating mapped properties

Several events automatically trigger attribute exchange between Teamcenter dataset objects and Microsoft Office files. The attribute or document properties exchanged depend on the configured exchange direction.

- Open and check out

  When a Teamcenter dataset is opened and checked out, the configured attribute exchange occurs. This exchange happens only on the mappings whose direction is **Teamcenter to File** and **Two Way**. A default value is used for the exchange if the following two items apply:

  - The Teamcenter dataset or its attribute does not exist on the server or the attribute value is null.
  - A default value is specified.

  Note:

  Users are informed if any errors occurred during the exchange.

- Save

  When a Teamcenter dataset is saved, the configured attribute exchange occurs. This exchange happens only on the mappings whose direction is **File to Teamcenter** and **Two Way**.

  If you delete the value for a custom Office document property that is defined for attribute exchange, the default property value is used during the save operation, if a default is specified.

  Note:

  Users are informed if any errors occur during the exchange.

- Save as

  When a Teamcenter dataset is saved using the **Save As** command, dataset creation is initiated. When the new dataset is created, if attribute exchange mappings are configured on the original dataset, they are copied to the new dataset. If the dataset is checked out, attribute exchange is performed both ways: **File to Teamcenter** and **Teamcenter to File**.

Note:

When creating a new dataset from the **Teamcenter** tab using the **Save As** command, you have the options of creating a new item where you want to save the dataset, and checking out the new item revision and the dataset.

If you check out the item revision and the related dataset file, the mappings configured for an IRDC controlled item are *not* associated with the checked out dataset. Therefore, if an item and its related dataset were originally created based on an item revision document control (IRDC) template, and if you want to retain the attribute mapping for the newly created dataset, you must:

1. Click **No** in the **Confirm Check-Out** dialog box.

2. Perform the check out operation separately from the create operation the first time you edit the object.

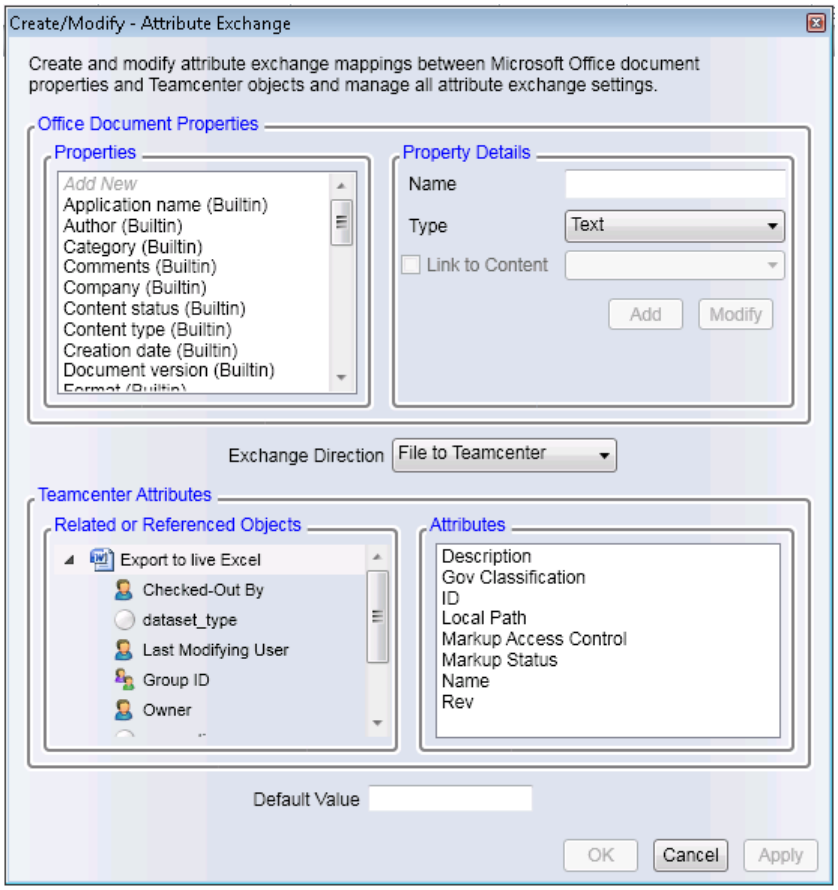
Warning:

After an attribute's value is synchronized between Teamcenter and Microsoft Office, and later it is deleted in Teamcenter, the null value cannot be synchronized to Microsoft Office, so the attribute continues to have the previous value.

Map Office document properties to Teamcenter

1. With an **MS Office** dataset selected in Teamcenter, choose **Open and Check Out**.

2. On the **Teamcenter** tab in the **Attribute Exchange** button group, click **Configurations→Create** to open the **Create/Modify - Attribute Exchange** dialog box.



3. Click **Create**.

   The **Create/Modify - Attribute Exchange** dialog box opens.

   > Note:
   >
   > You can open the **Create/Modify - Attribute Exchange** dialog box from the **Attribute Exchange** button group by clicking the list button next to **Configurations** and clicking **Create**.

4. In the **Properties** box, select an existing Office document property.

   The values for **Name** and **Type** appear in the **Property Details** pane.

   You can also click the **Link to Content** box and select the linked content from the list.

   > Note:
   >
   > When performing attribute exchange, if you set the direction of the exchange as **File to Teamcenter**, and you select a property to which you have write access, the exchange is

allowed. However, if you select a property that the system uses to control the status of the object, this can alter the status of the target object and may render the object unusable or corrupt. Therefore, when configuring attribute exchange from **File to Teamcenter**, do not select a property of the target Teamcenter object that is used to determine the processing status of the target object.

5. Click **Modify** to apply any changes you made to the property details.

   You can also create a new custom document property by adding a unique name and selecting a type.

6. Click **Add** to create a new custom document property.

7. Select the exchange direction from the list.

   - **File to Teamcenter**

     The document property value is copied to Teamcenter. The **Related or Referenced Objects** list only allows you to navigate one level to find objects attributes.

   - **Teamcenter to File**

     The Teamcenter attribute value is copied to the document. The **Related or Referenced Objects** list allows you to navigate two levels to find objects attributes.

   - **Two Way**

     The mapped property and attribute are copied both ways. The **Related or Referenced Objects** list only allows you to navigate one level to find objects attributes.

   Caution:

   Properties with dynamic and cascading lists of values (LOVs) are not supported.

8. In the **Related or Reference Objects** box, select a Teamcenter object.

   The attributes for the objects appear in the **Attributes** box. Only the attributes that match the property type display.

   Example:

   If the property type is **Date**, only Teamcenter attributes defined as date type appear.

> Note:
>
> When you select a relation and multiple objects of the same type are found by this relation, none of these objects can be selected for exchange. The valid related objects are the ones that are uniquely identified by relation and object type.

You can also enter a default value for the mapping.

9. Click **Apply** to add the attribute exchange mapping and leave the **Create/Modify - Attribute Exchange** dialog box open.

   The mapping appears as a new row in the **Configurations - Attribute Exchange** dialog box.

10. Click **OK** to add the attribute exchange mapping and close the **Create/Modify - Attribute Exchange** dialog box.

    The mapping appears as a new row in the **Configurations - Attribute Exchange** dialog box.

View mapping configuration details

1. Open and check out a Teamcenter dataset.

2. Click the **Teamcenter** tab.

3. In the **Attribute Exchange** button group, click **Configurations**.

   The system displays the **Configurations - Attribute Exchange** dialog box. Configuration mappings display IRDC template mappings and locally configured mappings.

   > Note:
   >
   > Optionally, you can open the **Configurations - Attribute Exchange** dialog box from the **Attribute Exchange** button group by clicking the list button next to **Configurations** and clicking **All Configurations**.

4. Select from the list to modify, delete, or perform exchange (reload) after necessary validations apply.

   > Caution:
   >
   > Properties with dynamic and cascading lists of values (LOVs) are not supported.

5. Click **Close** to close the **Configurations - Attribute Exchange** dialog box.

Modify an attribute map

> Note:
>
> You must be in the owning group of the user who mapped the property or a member of the configuration manager group.

1. Open and check out a Teamcenter dataset.
2. Click the **Teamcenter** tab.
3. In the **Attribute Exchange** button group, click **Configurations**.

   The system displays the **Configurations - Attribute Exchange** dialog box.
4. Select a mapping row to modify.
5. Click **Modify**.

   The system displays the **Create/Modify - Attribute Exchange** dialog box.

   > Note:
   >
   > You must be in the mapping's owning group or in the configuration manager group to modify the mapping.
6. Modify the office document properties, exchange direction, and Teamcenter attributes.

   > Caution:
   >
   > Properties with dynamic and cascading lists of values (LOVs) are not supported.
7. Click **Apply** to modify the attribute exchange mapping and leave the **Create/Modify - Attribute Exchange** dialog box open.

   The mapping is modified in the **Configurations - Attribute Exchange** dialog box.
8. Click **OK** to modify the attribute exchange mapping and close the **Create/Modify - Attribute Exchange** dialog box.

   The mapping is modified in the **Configurations - Attribute Exchange** dialog box.

Set the default locale and the manager group

1. Open and check out a Teamcenter dataset.

2. Click the **Teamcenter** tab.

3. In the **Attribute Exchange** button group, click **Configurations**.

   The system displays the **Configurations - Attribute Exchange** dialog box.

   > Note:
   >
   > Optionally, you can open the **Configurations - Attribute Exchange** dialog box from the **Attribute Exchange** button group by clicking the list button next to **Configurations** and clicking **All Configurations**.

4. In the **General Settings** pane, click the **Locale** box and select a language that is the Teamcenter default locale.

   > Note:
   >
   > This locale is only for attribute exchange. It is not the Office Client session culture. You have to be in the manager group to modify this locale.

5. In the **General Settings** pane, click the **Change** button to the right of **Configuration Manager Group**

   The system displays the **Groups - Attribute Exchange** dialog box.

   > Note:
   >
   > You must be in the configuration manager group to modify the configuration manager group.

6. Select a manager group for the attribute exchange.

   The configuration manager group is the administrator group for the mapping and the attribute exchange. This manager group defaults to the original document author's Teamcenter group. If the document is created under item revision definition configuration (IRDC) control, this is what the IRDC template is set to.

7. Type text in the **Search** box to filter the list.

8. Click **Reload** to display a complete list of Teamcenter groups.

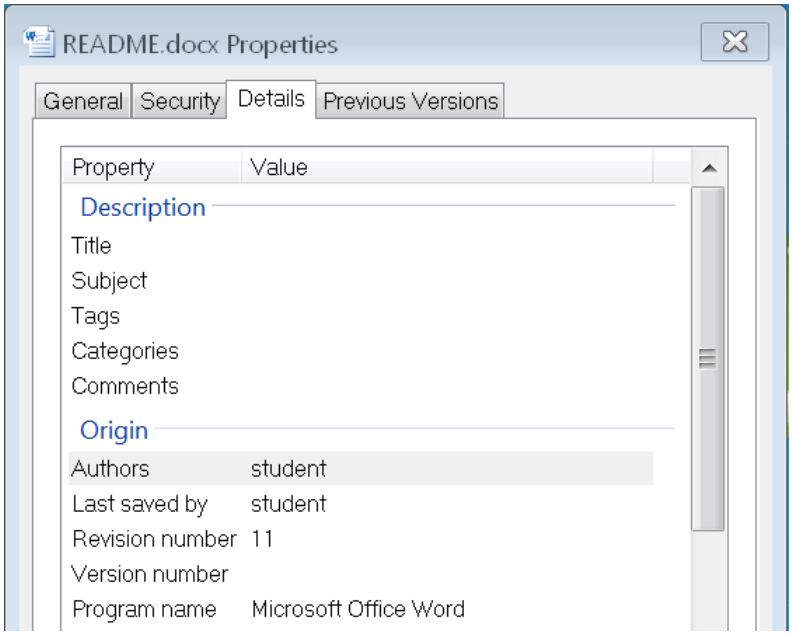
9. Click **OK** to set the configuration manager group and close the **Groups - Attribute Exchange** dialog box.

View mapped properties in Office files

1. On the Office ribbon, choose **File→Info**.

2. In the right pane, select **Properties→Advanced Properties** to open the **Properties** dialog box.

> Tip:
>
> You can also open the **Properties** dialog box by selecting **Document Properties→Advanced Properties** in the Word Document Panel.



3. Click the tabs in the dialog box to view the mapped properties by category.

Add a custom property in Word

1. In a Word document that you want to synchronize with Teamcenter, choose **File→Info**.

2. In the right pane, select **Properties→Advanced Properties** to open the **Properties** dialog box.

> Tip:
>
> You can also open the **Properties** dialog box by selecting **Document Properties→Advanced Properties** in the Word Document Panel.

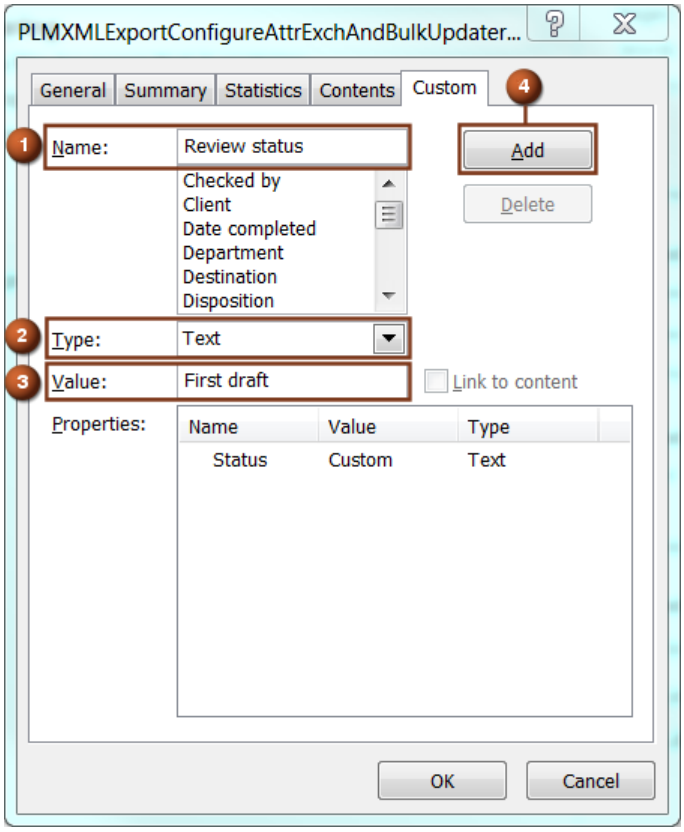
3. Click the **Custom** tab, and then do the following:

   a. Enter the property name in the **Name** box.

   b. Select the property type from the **Type** list.

   c. Enter the initial property value in the **Value** box.

d. Click **Add**.



The new property appears in the **Properties** table in the dialog box.

4. Do one of the following:

   - To add another custom property, repeat this procedure.
   - To commit your additions and close the **Properties** dialog box, click **OK**.

> Note:
>
> When using a custom attribute of the **Date** type, you cannot enter a date prior to Jan 1, 1900 as Teamcenter initializes the start date as Jan 02, 1900.

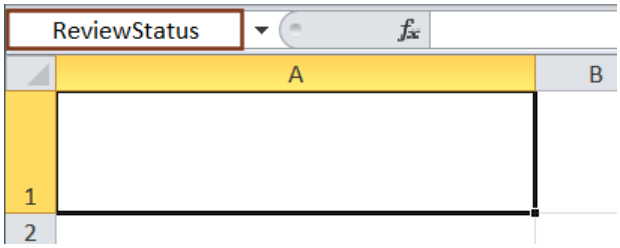
Add a custom property in Excel

1. In an Excel spreadsheet that you want to synchronize with Teamcenter, rename the cell that you want to contain the property value.

   a. Right-click the cell (for example, **A1**) and choose **Define Name** to open the **New Name** dialog box.

b. Enter the new name in the **Name** box, and then click **OK**.

The new name appears to the left of the **fx** field.



2. Choose **File→Info**.

3. In the right pane, select **Properties→Advanced Properties** to open the **Properties** dialog box.

> Tip:
>
> You can also open the **Properties** dialog box by selecting **Document Properties→Advanced Properties** in the Word Document Panel.

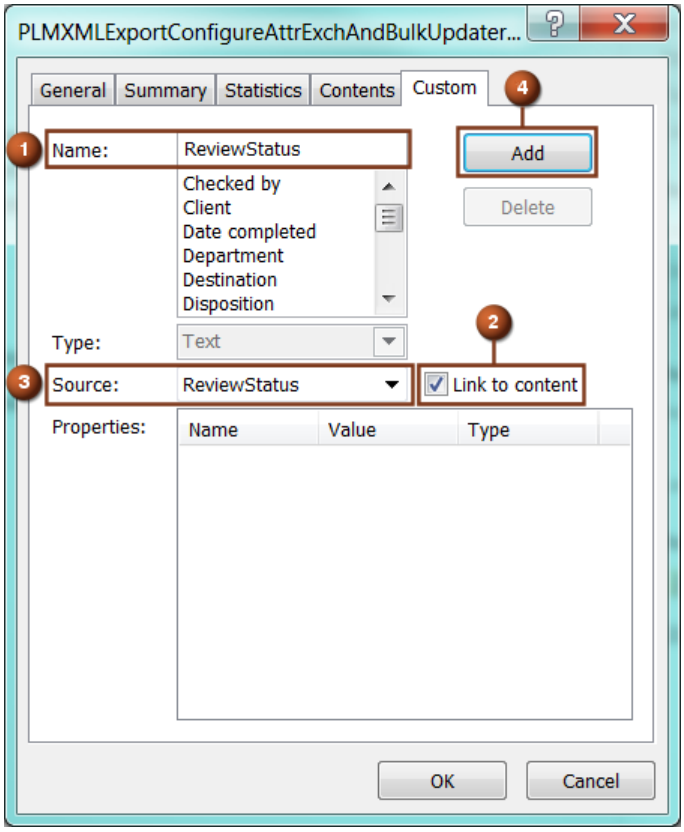
4. Click the **Custom** tab, and then do the following:

a. Enter the property name in the **Name** box.

b. Check the **Link to content** box to display the **Source** list in place of the **Value** box.

c. Select the renamed cell in the **Source** list.

d. Click **Add**.



The new property appears in the **Properties** table in the dialog box.

5. Do one of the following:

   - To add another custom property, repeat this procedure.
   - To commit your additions and close the **Properties** dialog box, click **OK**.

Note:

When using a custom attribute of the **Date** type, you cannot enter a date prior to Jan 1, 1900 as Teamcenter initializes the start date as Jan 02, 1900.

Also, if you enter a date in the months of January or February 1900 in Excel, the value is not displayed in Teamcenter correctly. This is because Microsoft Excel treats the year 1900 as leap year; therefore, the date conversion in Microsoft Excel is off by one day for the months of January and February of the year 1900.

Edit document property values

1. Open and check out a Teamcenter dataset.
2. Click the **Teamcenter** tab.

3. In the **Attribute Exchange** button group, click **Edit Values**.

   The system displays the **Edit Values - Attribute Exchange** dialog box.

   > Note:
   >
   > Only the mappings for **File to Teamcenter** and **Two Way** allow edits. The mappings for **Teamcenter to File** are not displayed.

4. Select a row in the table to edit.

5. Enter a value in the **Value** box.

   > Tip:
   >
   > You can click **Refresh** to return the original value or click **Reset to Defaults** to return the default value.

6. Click **Apply** to apply the value updates and leave the **Edit Values - Attribute Exchange** dialog box open.

7. Click **OK** to apply the value updates and close the **Edit Values - Attribute Exchange** dialog box.

Reload Teamcenter values to Office

1. Open and check out a Teamcenter dataset.

2. Click the **Teamcenter** tab.

3. In the **Attribute Exchange** button group, click **Reload**.

   The system displays the **Confirm Reload** dialog box. Any locally modified Office properties are lost with the updated values from Teamcenter.

4. Click **OK** to continue with the reload.

Delete an attribute map

> Note:
>
> You must be in the owning group of the user who mapped the property or a member of the configuration manager group.

1. Open and check out a Teamcenter dataset.

2. Click the **Teamcenter** tab.

3. In the **Attribute Exchange** button group, click **Configurations**.

   The system displays the **Configurations - Attribute Exchange** dialog box.

4. Select a mapping row to delete.

5. Click **Delete**.

   The system displays the **Delete - Warning** dialog box.

   > Note:
   >
   > You must be in the mapping's owning group or in the configuration manager group to delete the mapping.

6. Click **OK**.

   The selected mapping is removed from the list and deleted when the file is saved back to Teamcenter.

Converting legacy PropSync datasets for attribute exchange

Mapping attributes for PropSync datasets

When PropSync forms are attached to legacy datasets, that data is incompatible with Client for Office. For such datasets, attribute mapping conversion involves the following procedures.

> Note:
>
> This only works to convert to Microsoft based attribute exchange, not to logical objects.

1. Set up the **fnd0InstanceAttrExMappings** property for export.

   The **fnd0InstanceAttrExMappings** property contains the attribute mapping data for Teamcenter datasets. You use PLM XML/TC XML Export Import Administration to set up the property for export.

   In this procedure, you first create a *property set* that relates the **fnd0InstanceAttrExMappings** property to the **Dataset** primary object type. Next, you associate the new property set with the **unconfiguredDataFileExport** *transfer mode*.

2. Export the **fnd0InstanceAttrExMappings** data to PLM XML.

Using the **unconfiguredDataFileExport** transfer mode, you export the attribute mapping data to a PLM XML output file. Later in the conversion process, you use this data to replace the legacy data that you export with the **attribute_export** utility.

3. Build an XML file as input to the **attribute_export** utility.

   In the XML input file, you specify property names and values as search criteria for the datasets to be updated. You also specify a temporary value for the **fnd0InstanceAttrExMappings** property. When you run the **attribute_export** utility, you enter this file name in the **-inputfile** argument.

4. Export the legacy data with the **attribute_export** utility.

   The **attribute_export** utility queries for the datasets that meet the criteria you specify in the XML input file. For each found dataset, the utility inserts the temporary value in the **fnd0InstanceAttrExMappings** property.

   The utility outputs the data to a TC XML file. In this file, you replace each temporary value with the corresponding data from the PLM XML output file. Then, you save the changes and import the replacement data to update the datasets.

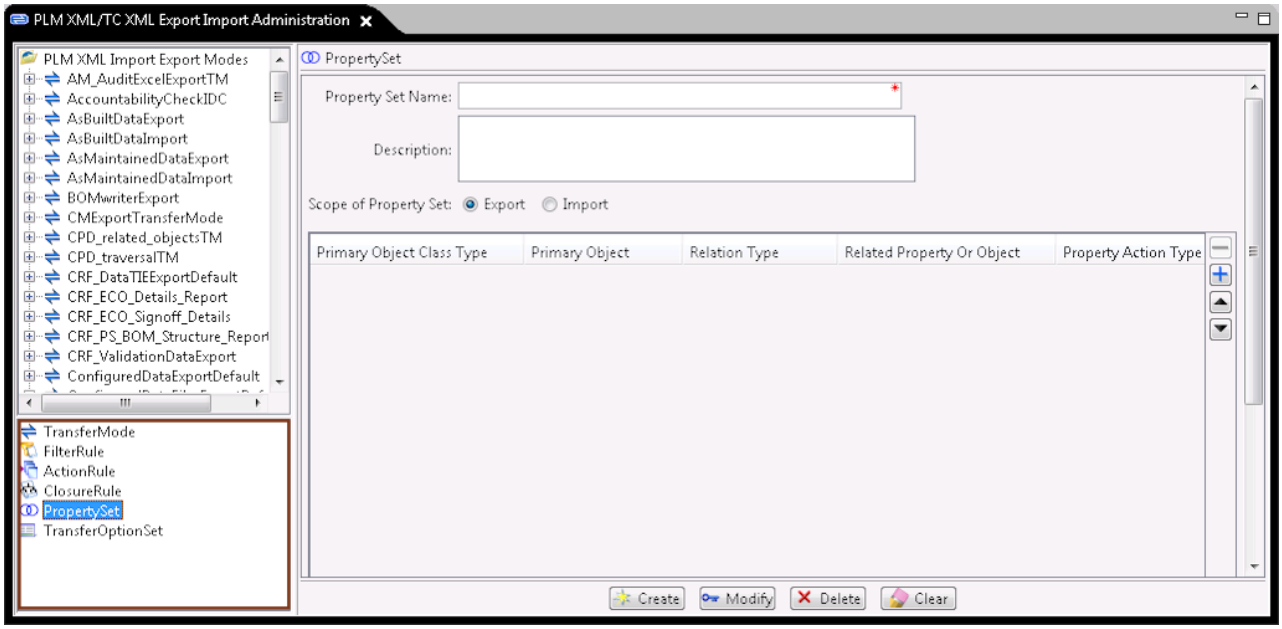
5. Import the replacement data to the legacy datasets.

   You run the **tcxml_import** utility to update the dataset properties with the replacement data from the TC XML output file. For a fast bulk update, you use the **-bulk_load** and **-bypassSiteCheck** arguments.

   > Note:
   >
   > The **-bulk_load** argument requires a Site Consolidation license key value in the **SITCONS_AUTH_KEY** environment variable. The license key is available on the Support Center:
   > **https://support.sw.siemens.com**

Set up the fnd0InstanceAttrExMappings property for export

1. In PLM XML/TC XML Export Import Administration, select **PropertySet** in the **TransferMode** tree.



2. In the **TransferMode** pane, enter the object name in the **Property Set Name** box.

   You can also enter additional information in the **Description** box.

3. Beside **Scope of Property Set**, select **Export**.

4. Add an ordered clause that specifies how to traverse the data structure.

   a. Click the **Add clause** button + to the right of the clause table.

   b. Under **Primary Object Class Type**, click the empty cell and select **CLASS**.

   c. Under **Primary Object**, double-click the empty cell and type **Dataset**.

   d. Under **Relation Type**, click the empty cell and select **ATTRIBUTE**.

   e. Under **Related Property Or Object**, double-click the empty cell and type **fnd0InstanceAttrExMappings**.

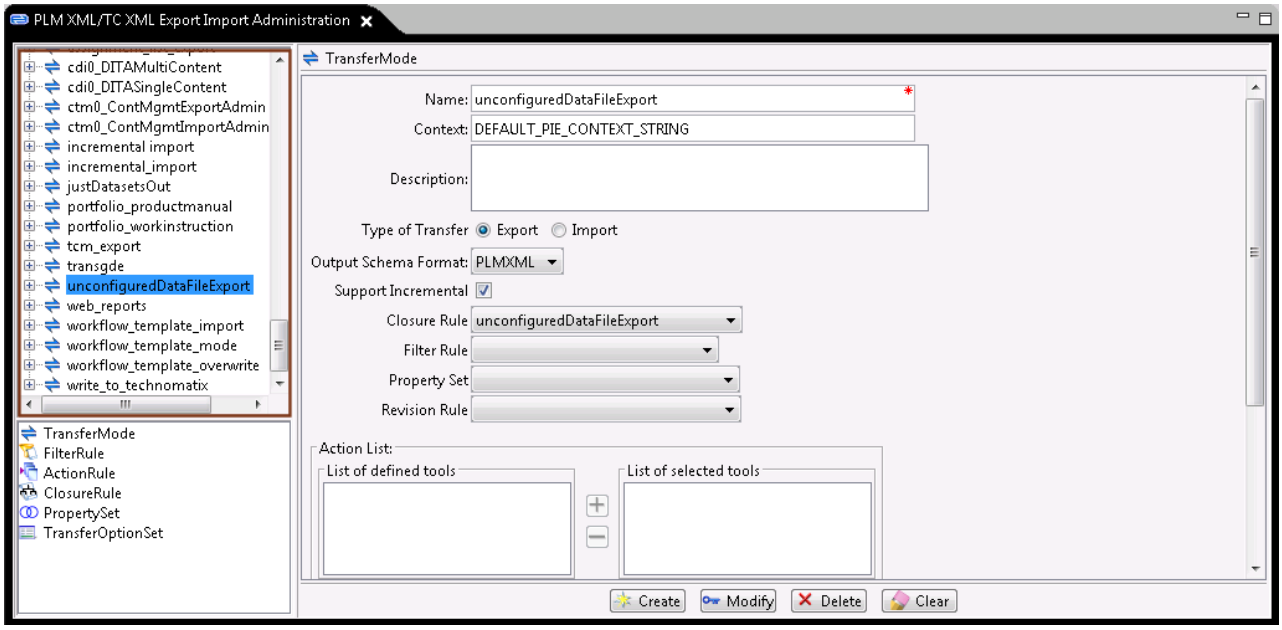
   f. Under **Property Action Type**, click the empty cell and select **DO**.

Primary Object Class Type	Primary Object	Relation Type	Related Property Or Object	Property Action Type
CLASS	Dataset	ATTRIBUTE	fnd0InstanceAttrExMappings	DO

5. At the bottom of the **TransferMode** pane, click **Create**.

The new property set is added under **PropertySet** in the bottom left pane.

6. In the object tree, select the **unconfiguredDataFileExport** transfer mode.



7. In the **TransferMode** pane, select the new property set in the **Property Set** list, and then click **Modify**.

The property set is added to the **unconfiguredDataFileExport** transfer mode in the object tree.

Export the fnd0InstanceAttrExMappings data to PLM XML

1. Select the DMTemplate item revision or its dataset.
2. Choose **Tools→Export→To PLMXML** to open the **PLMXML Export** dialog box.
3. In the **Export Directory** box, enter the full path of the directory where you want to store the XML output file.
4. In the **Export Filename** box, enter the name of the XML output file.
5. In the **Transfer Mode** list, select **unconfiguredDataFileExport**.
6. Click **OK** to initiate the export.
7. When the export is complete, open the output file and verify that it contains the **fnd0InstanceAttrExMappings** property data.

The data in an **Item** entry is a string from the string array in the **fnd0InstanceAttrExMappings** property.

```
<DataSet id="id13" name="DMTemplateForSampleSDDWordXDatasetAttrExch"
accessRefs="#id3" releaseStatusRefs="#id18" version="3" memberRefs="#id17"
type="MSWordX"> <Description>Sample SDD</Description>
<ApplicationRef version="gmdFGCjDAABaaA" application="Teamcenter"
label="gmdFGCjDAABaaA"></ApplicationRef>
<UserData id="id15"> <UserValue value=""
title="fnd0InstanceAttrExMappings">
<UserList id="id3" type="list">
<Item value="&lt;?xml version=&quot;1.0&quot;encoding=&quot;utf-8&quot;
?&gt;&lt;AttributeExchangeConfiguration xmlns:xsi=&quot;http://www.w3.org/
2001/
XMLSchema-instance&quot; xmlns:xsd=&quot;http://www.w3.org/2001/
XMLSchema&quot;version=&quot;1&quot;&gt;&lt;AdminGroup&gt;dba&lt;/
AdminGroup&gt;&lt;
Locale&gt;en_US&lt;/Locale&gt;&lt;MappingList&gt;&lt;Mapping source=&quot;
instance&quot;&gt;&lt;Direction&gt;TeamcenterToFile&lt;/Direction&gt;&lt;
OfficeProperty name=&quot;Author&quot; format=&quot;Text&quot;
propertySet=&quot;Builtin&quot;&gt;&lt;LinkToContent&gt;&lt;Linked&gt;false&
lt;
/Linked&gt;&lt;/LinkToContent&gt;&lt;/
OfficeProperty&gt;&lt;TeamcenterMapping;
bOType=&quot;MSWordX&quot;&gt;&lt;MappingPath&gt;MappingObject&lt;/
MappingPath&gt;
&lt;MappingObject bOType=&quot;DocumentRevision&quot; referencedName=&quot;
item_revision&quot; displayReferencedName=&quot;item_revision&quot;&gt;&lt;
MappingPath&gt;Leaf&lt;/MappingPath&gt;&lt;TcAttribute
name=&quot;DocumentSubject&quot; displayName=&quot;Document Subject&quot; /
&gt;&lt;
/MappingObject&gt;&lt;/
TeamcenterMapping&gt;&lt;Permissions&gt;&lt;OwningGroup&gt;
dba&lt;/OwningGroup&gt;&lt;/Permissions&gt;&lt;/Ma"></Item>
<Item value="pping&gt;&lt;Mappingsource=&quot;instance&quot;&gt;&lt;
Direction&gt;TeamcenterToFile&lt;/Direction&gt;&lt;OfficeProperty
name=&quot;Title&quot;
format=&quot;Text&quot;
propertySet=&quot;Builtin&quot;&gt;&lt;LinkToContent&gt;&lt;
Linked&gt;false&lt;/Linked&gt;&lt;/LinkToContent&gt;&lt;/
OfficeProperty&gt;&lt;
TeamcenterMappingbOType=&quot;MSWordX&quot;&gt;&lt;MappingPath&gt;MappingObj
ect&lt;
/MappingPath&gt;&lt;MappingObject bOType=&quot;DocumentRevision&quot;
referencedName=&quot;item_revision&quot;displayReferencedName=&quot;
item_revision&quot; &gt;&lt;MappingPath&gt;Leaf&lt; /MappingPath&gt;&lt;
TcAttribute name=&quot;DocumentTitle&quot; displayName=&quot;Document
Title&quot;
/&gt;&lt;/MappingObject&gt;&lt;/TeamcenterMapping&gt;&lt;Permissions&gt;&lt;
```

```
OwningGroup&gt;dba&lt;/OwningGroup&gt;&lt;/Permissions&gt;&lt;/
Mapping&gt;&lt;
/MappingList&gt;&lt;/AttributeExchangeConfiguration&gt;"></Item>
</UserList></UserValue></UserData></DataSet>
```

Build an XML file as input to the attribute_export utility

In a new XML file, enter the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<BulkUpdate>
    <UpdateSets>
        <UpdateSet>
            <type name="Dataset"/>
            <where>
                <cond_prop attrName="object_type" attrValue="MSWordX"/>
                <cond_prop attrName="fnd0InstanceAttrExMappings"
attrValue=""/>
                <cond_prop attrName="last_mod_date" attrValue="18-
Jul-2024 14:56"
                    cond_operator="GE"/>
            </where>
            <update>
                <update_prop attrName="fnd0InstanceAttrExMappings"
                    attrValue="temp_val"/>
            </update>
        </UpdateSet>
    </UpdateSets>
</BulkUpdate>
```

Replace *temp_val* with a temporary value. The **attribute_export** utility inserts this value in the **fnd0InstanceAttrExMappings** property for each exported dataset.

Export the legacy data with the attribute_export utility

1. To run the **attribute_export** utility, enter the following at the command prompt:

   attribute_export -u=*userid* **-p=***password* **-g=***group* **-inputfile=***xml-file-name* **-outdir=***path*

userid	Your Teamcenter user identifier.
password	Your Teamcenter password.
group	The Teamcenter user group to which your user identifier is assigned. You must be a member of a group that has **dba** privileges.

- For the **-inputfile** argument, replace *xml-file-name* with the name of the XML file that you built as input to the **attribute_export** utility.

- For the **-outdir** argument, replace *path* with the full path of the output file directory.

The utility generates a TC XML output file, which contains data for each exported dataset. This data includes the temporary value of the **fnd0InstanceAttrExMappings** property. For example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<TCXML xmlns="http://www.tcxml.org/Schemas/TCXMLSchema" format="low_level">
       <Dataset elemId="id1" island_id="1" puid="ADYNcOCF6hSNuC"
         lsd="2014-07-18T22:01:56Z" last_mod_date="2014-07-18T22:01:56Z"
fnd0InstanceAttrExMappings="temp_val"/>
       <Dataset elemId="id2" island_id="2" puid="AHSNcOCF6hSNuC"
         lsd="2014-07-18T22:01:56Z" last_mod_date="2014-07-18T22:01:56Z"
fnd0InstanceAttrExMappings="temp_val"/>
       <Dataset elemId="id3" island_id="3" puid="AHXNcOCF6hSNuC"
         lsd="2014-07-18T22:01:56Z" last_mod_date="2014-07-18T22:01:56Z"
fnd0InstanceAttrExMappings="temp_val"/>
...
```

2. For each **fnd0InstanceAttrExMappings** property in the TC XML output file, replace the temporary value with the corresponding data from the PLM XML file.

   In the PLM XML file, the replacement data is the **Item** value string that corresponds to the dataset. For example:

```
<Item value="&lt;?xml version=&quot;1.0&quot;encoding=&quot;
utf-8&quot;?&gt;&lt;
AttributeExchangeConfiguration xmlns:xsi=&quot;http://www.w3.org/
2001/
XMLSchema-instance&quot; xmlns:xsd=&quot;http://www.w3.org/2001/
XMLSchema&quot;
version=&quot;1&quot;&gt;&lt;AdminGroup&gt;dba&lt;/
AdminGroup&gt;&lt;Locale&gt;
en_US&lt;/Locale&gt;&lt;MappingList&gt;&lt;Mapping source=&quot;
instance&quot;&gt;&lt;Direction&gt;TeamcenterToFile&lt;/
Direction&gt;&lt;
OfficeProperty name=&quot;Author&quot; format=&quot;Text&quot;
propertySet=&quot;Builtin&quot;&gt;&lt;LinkToContent&gt;&lt;Linked&gt
;false&lt;
/Linked&gt;&lt;/LinkToContent&gt;&lt;/
OfficeProperty&gt;&lt;TeamcenterMapping;
bOType=&quot;MSWordX&quot;&gt;&lt;MappingPath&gt;
MappingObject&lt;/MappingPath&gt;
&lt;MappingObject bOType=&quot;DocumentRevision&quot;
referencedName=&quot;
item_revision&quot;
displayReferencedName=&quot;item_revision&quot;&gt;&lt;
MappingPath&gt;Leaf&lt;/MappingPath&gt;&lt;TcAttribute
```

```
name=&quot;DocumentSubject&quot;
displayName=&quot;Document Subject&quot; /&gt;&lt;
/MappingObject&gt;&lt;/TeamcenterMapping&gt;&lt;Permissions&gt;&lt;
OwningGroup&gt; dba&lt;/OwningGroup&gt;&lt;/Permissions&gt;&lt;
;/Mapping&gt;&lt;/MappingList&gt;&lt;
/AttributeExchangeConfiguration&gt;"></Item>
```

3. Save the changes.

Import the replacement data to the legacy datasets

> Note:
>
> In the **tcxml_import** utility, the **-bulk_load** argument requires a Site Consolidation license key value in the **SITCONS_AUTH_KEY** environment variable. The license key is available on the Support Center:
> **https://support.sw.siemens.com**

1. To run the **tcxml_import** utility, enter the following at the command prompt:

   tcxml_import -u=*userid* **-p=***password* **-g=***group* **-bulk_load -file=***xml-file-name* **-bypassSiteCheck**

userid	Your Teamcenter user identifier.
password	Your Teamcenter password.
group	The Teamcenter user group to which your user identifier is assigned. You must be a member of a group that has **dba** privileges.

   - The **-bulk_load** argument performs a fast bulk update of legacy data from a TC XML file.
   - For the **-file** argument, replace *xml-file-name* with the name of the TC XML file that contains the input data.
   - The **-bypassSiteCheck** argument bypasses the check that prevents the utility from updating objects at the same site and the check that prevents the update of replica objects.

   All other **tcxml_import** arguments are ignored.

2. Verify the data in Teamcenter or in Client for Office.
   - In Teamcenter, you can select a dataset and view its properties in the **Summary** view.
   - In Client for Office, you can open a dataset and view mapped properties in the **Properties** dialog box.

Synchronizing attributes between Teamcenter and Microsoft Office files by using logical objects

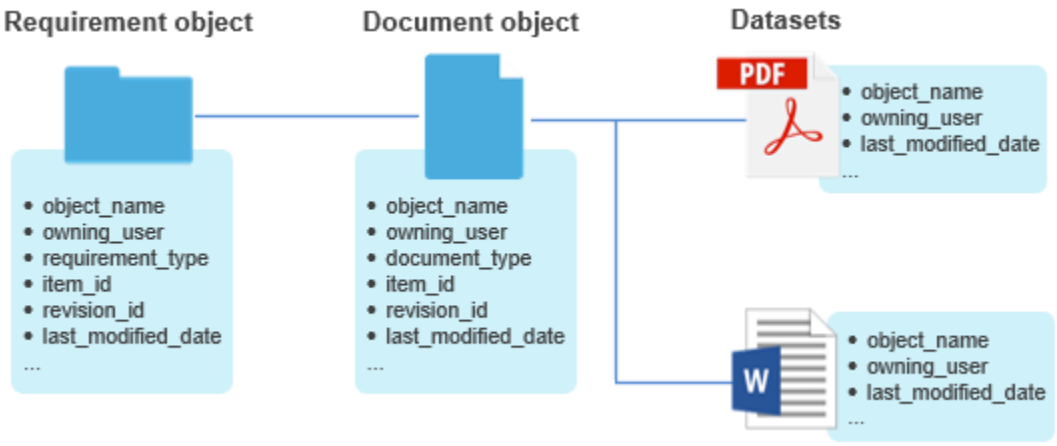
About logical objects

A *logical object* is a container that consolidates attributes (properties) from different related business objects. For example, a document (item) and its related datasets are different objects in Teamcenter. The attributes of the document can be its ID, name, or description. The different attributes of a dataset can be author, created date, or reviewed by.

The following example shows how you can synchronize attributes, consolidated by the logical object, from Teamcenter to a Microsoft Word document and to a PDF file.

Example

Consider that there is a requirement object in Teamcenter which has a reference relation to a document object. The document object has two attached files (datasets), one Word and one PDF.

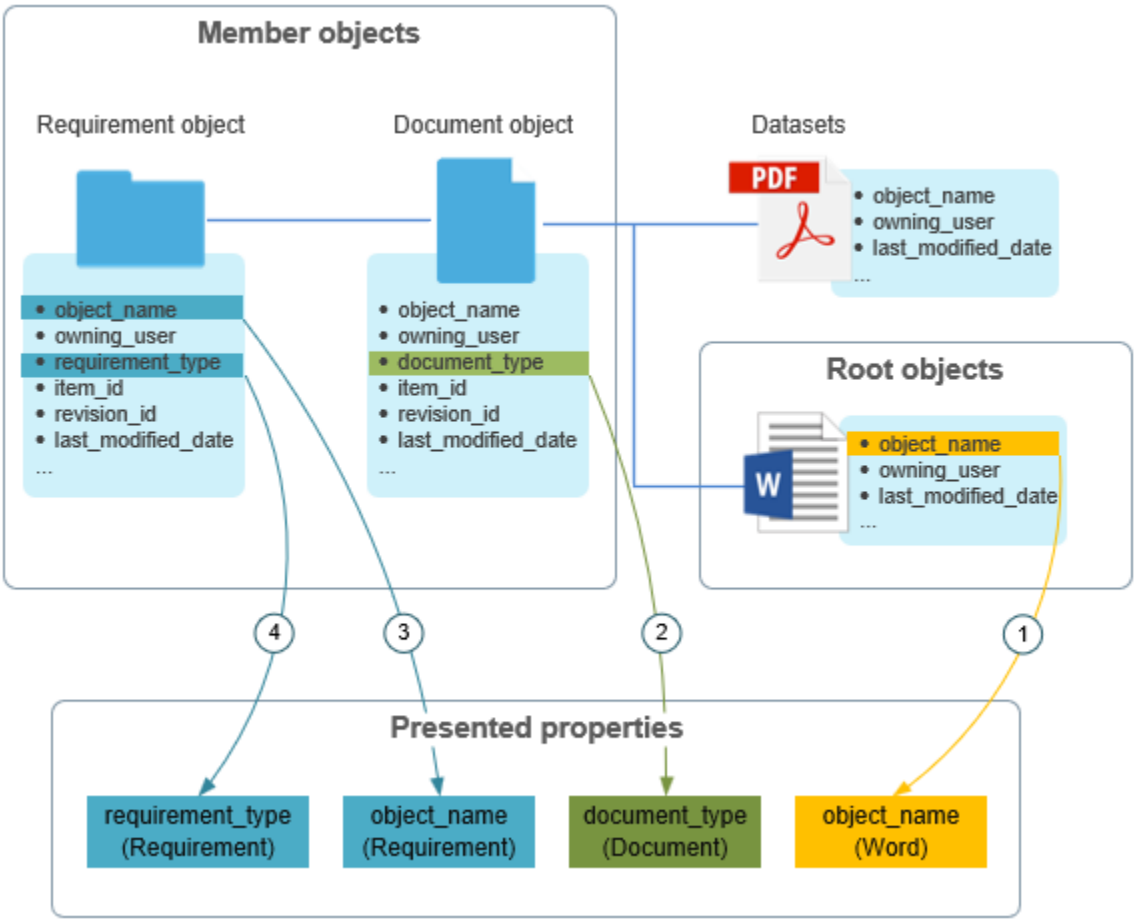


You want to consolidate four attributes from these objects:

- **object_name** and **requirement_type** from the requirement object.
- **document_type** from the document object.
- **object_name** from the Word or PDF file attached to the document.

For consolidating these attributes, you define a logical object. A logical object is composed of *root object*, *member objects*, and *presented properties*. You begin consolidating the properties from the dataset. This makes the dataset the root object. You then traverse backward tracing its relation to the document object and then to the requirement object. This makes the document and requirement objects the member objects. While traversing from the dataset to the document object and then to the

requirement object, you collect the required four attributes in the logical object. These four attributes are the presented properties.



After consolidating the attributes in a logical object, you use the logical object to synchronize attributes between Teamcenter and Microsoft Office files and PDF files. When you synchronize attributes between Teamcenter and Microsoft Office files, you cannot synchronize attributes from Teamcenter to Office files and from Office files to Teamcenter at the same time.

> Note:
>
> Even if you used attribute-level security while implementing Teamcenter, the properties that are used for attribute exchange are still synchronized in the document. The attribute exchange overrides the attribute-level security.

Task flow to synchronize attributes by using logical objects

1. **Define a logical object**.

2. **Define when to add the Teamcenter attributes in template files**.

3. To synchronize the attributes between Teamcenter and Microsoft Office files, **specify the logos, distribution statements, and workflow signoff tables** to be included in the Office files in the Document Management configuration file.

4. To synchronize the attributes between Teamcenter and Microsoft Office files, define the placement of attributes, logos, distribution statements, and workflow signoff tables in the **Word**, **Excel**, and **PowerPoint** files, if required.

   To synchronize the attributes between Teamcenter and PDF files, define the placement of attributes, logos, distribution statements, and workflow signoff tables in the PDF files, if required.

5. Include the logos, distribution statements, and workflow signoff tables in the synchronized files by setting up a **system stamp configuration**.

6. **Relate the logical object with a file (dataset)**.

When you synchronize attributes between Teamcenter and Microsoft Office files, the Teamcenter attributes along with logos, distribution statements, and workflow signoff tables are included in a file synchronously when a user creates or renders a file. They are included in a file asynchronously through Dispatcher when a user checks in, saves as, or revises a document revision. For more information about workflow signoff table, see *server-install-location\samples\document_management\readme.pdf*. To get the latest information in an Office file, a user can also send the file through an attribute exchange workflow process. For this, you must set up a workflow task for the users by using the **DOCMGT-update-docprop-logicalobject** action handler.

When you synchronize attributes between Teamcenter and PDF files, The Teamcenter attributes along with logos, distribution statements, and workflow signoff tables are included in a PDF file when a user sends the file through an attribute exchange workflow process. For this, you must set up a workflow task for the users by using the **DOCMGTAPP-apply-pdf-control** action handler.

Finally, **you may verify if you have set up the attribute exchange correctly**.

Define logical objects

You can define logical objects for attribute synchronization. Note that the logical objects do not support the runtime properties for attribute synchronization.

1. In Active Workspace, log on with DBA credentials and select **Default** as the **Workspace**.

2. Click the **LOGICAL OBJECTS** tile on the home page.

3. On the **Logical Object Configuration** page, choose **More Commands** ••• > **New** ✳ > **Define logical object** ⊕.

4. In the **Add** panel:

   a. Enter **Internal Name**, **Name**, and **Description** for the logical object.

b. In the **Root Object** list, click a root object. For example, click **Dataset**.

c. Click the **Retrieve Classification Data** checkbox if you want to retrieve the classification data.

d. In the **Parent Logical Object** list, click a parent logical object. For example, click **Logical Object**.

e. Click **Add**.

5. In the **Overview** tab, click **Add** ⊕ to add new members to the logical object.

6. In the **Add Member** panel:

a. Enter **Member ID** and **Display Name**.

b. Select **Backward** in **Segment 1**.

c. Select **Business Object**, for example, **DocumentRevision**.

d. Select **Relation or Reference**, for example, **Tc_Attaches**.

e. Click **Add Segment** ⊕ in **Segment 2**.

f. Select **Forward**.

g. Select **Relation or Reference**, for example, **items_tag**.

h. Select **Business Object**, for example, **Document**.

i. Click **Add**.

7. Click **Add** ⊕ again to add another member to the logical object to relate the business object with the dataset by using a different relation.

8. For each member, add member properties that you want to consider for document rendering. To do so:

a. In the **Presented Properties** section, click **Add** ⊕.

b. In the **Add** panel, click a property of the member in the Root Or Member ID list. For example, **fnd0Root**.

c. In the **Member Properties** list, click the required property, for example, **last_mod_date**.

d. Click **Add**.

You can add as many properties as required for each member.

9. Go to the **Exchange Configuration** tab to optionally set the attribute synchronization direction (**Teamcenter to File** or **File to Teamcenter**). If you do not specify the exchange direction, attributes are exchanged from **Teamcenter to File** by default.

10. In the **Exchange Configuration** tab, click **Refresh** to load the exchange configurations corresponding to the presented properties that you added earlier.

    Default direction for attribute exchange is **Teamcenter to File**. If **File to Teamcenter** direction for attribute exchange is required, you must set **Is Writable** property to **TRUE** in the **Overview** tab.

> Note:
>
> Attribute exchange cannot be displayed in Microsoft Office Online Excel files. This is a limitation of Microsoft Office Online.

> Tip:
>
> To delete a logical object, you must first delete all the presented properties available in the **Exchange Configuration** tab.
>
> Refresh the presented properties in the **Exchange Direction** tab after deleting presented properties in the **Overview** tab.

Define when to add the Teamcenter attributes in template files

1. In the existing BMIDE template project, click **Business Objects→ItemRevision**.

2. Double-click **ItemRevision** and in the **Main→Business Object Constants** tab, locate **Fnd0RelToDatasetForLOAttrExch** and click **Edit**.

3. In the **Business Object Constant** dialog box, enter the relations that must be considered including attributes in datasets attached to a Document Management template revision. You can specify more than one relation by using commas, for example, **TC_Attaches**, **IMAN_specification**.

4. Click **Finish**.

5. In the **Main→Business Object Constants** tab, locate **Fnd0TriggerLOAttrExch** and click **Edit**.

6. In the **Business Object Constant** dialog box, select one of the following in **Value** and click **Finish**:

   - **Disabled**

     This value indicates that the Teamcenter attributes will not be added to template files automatically. This is the default value.

- **Configured**

  This value indicates that the Teamcenter attributes will be added to template files automatically. The attributes will also be added to the datasets attached to the Document Management template revision with relations specified in the **Fnd0RelToDatasetForLOAttrExch** business constant.

- **Enabled**

  This value indicates that the Teamcenter attributes will be added to template files and to all the attached datasets automatically.

Specify the logos, distribution statements, and workflow signoff tables to be displayed in the stamped files

You specify information about logos, distribution statements, and workflow signoff tables in the Document Management configuration XML file. You may refer to the Document Management configuration schema file (**DocMgmtConfig.xsd**) located at *TC_ROOT\sample\document_management\schema* to create the configuration XML file. In this file, you can specify the logos to be used, when to use a logo, and where to place the logo in a PDF file. You can also specify similar information for distribution statements, workflow signoff tables, and Teamcenter attributes. After specifying the required information, you must **attach the configuration XML file** to a Document Management template revision for stamps.

For more information about workflow signoff table, see *server-install-location\samples\document_management\readme.pdf*.

A snippet of a list of columns in the workflow signoff table

You can define the columns to be displayed in a workflow signoff table. To do so, add the column names in the Document Management configuration XML file:

```
...
<!--workflow signoff table column names to display, can be removed or
reordered
These names listed below are internal names, not the display names-->

<dmcfg:WorkflowSignOffColumnsToDisplay>
   <dmcfg:WorkflowSignoffColumnName>processName</
dmcfg:WorkflowSignoffColumnName>
   <dmcfg:WorkflowSignoffColumnName>group</dmcfg:WorkflowSignoffColumnName>
   <dmcfg:WorkflowSignoffColumnName>role</dmcfg:WorkflowSignoffColumnName>
   <dmcfg:WorkflowSignoffColumnName>userName</
dmcfg:WorkflowSignoffColumnName>
   <dmcfg:WorkflowSignoffColumnName>userId</dmcfg:WorkflowSignoffColumnName>
   <dmcfg:WorkflowSignoffColumnName>decisionStringDisplay
   </dmcfg:WorkflowSignoffColumnName>
```

```
   <dmcfg:WorkflowSignoffColumnName>decisionDateString
   </dmcfg:WorkflowSignoffColumnName>
   <dmcfg:WorkflowSignoffColumnName>comments</
dmcfg:WorkflowSignoffColumnName>
</dmcfg:WorkflowSignOffColumnsToDisplay>
...
```

By default, all columns are displayed in the workflow signoff table. You can add, remove, or reorder the columns as required.

A snippet of the logo and distribution statement information

```
...
<dmcfg:logoList>
   <dmcfg:logoDataset datasetName="Logo" ID="1">
      <dmcfg:enableWhen>
         <dmcfg:property key="fnd0CurrentLocationCode" value="Shoreview" />
      </dmcfg:enableWhen>

      <dmcfg:MSOfficePlacement>
         <dmcfg:tag value="Logo_1" />
      </dmcfg:MSOfficePlacement>
   </dmcfg:logoDataset>
</dmcfg:logoList>

<dmcfg:distStatementList>
   <dmcfg:distStatementDataset datasetName="ExportControlledITAR" ID="1" />
      <dmcfg:enableWhen>
         <dmcfg:property key="fnd0CurrentLocationCode" value="Shoreview" />
      </dmcfg:enableWhen>

      <dmcfg:MSOfficePlacement>
         <dmcfg:tag value="Statement_1" />
      </dmcfg:MSOfficePlacement>


   <dmcfg:distStatementDataset datasetName="NonExportControlledITAR"
ID="2" />
      <dmcfg:enableWhen>
         <dmcfg:property key="fnd0CurrentLocationCode" value="Cypress" />
      </dmcfg:enableWhen>

      <dmcfg:MSOfficePlacement>
         <dmcfg:tag value="Statement_2" />
      </dmcfg:MSOfficePlacement>

   </dmcfg:distStatementDataset>
```

```
</dmcfg:distStatementList>
...
```

Attach files to a Document Management template revision for stamps

To attach the logo, distribution statement, Document Management configuration XML file, PDF command file, and MetaData Stamp file to a Document Management template revision that you created for stamps:

1. In My Teamcenter, select the required system stamp Document Management template revision.

2. Click **File→New→Dataset**.

3. In the **New Dataset** dialog box:

    a. Enter the **Name** of the dataset as the name of the logo or distribution statement specified in the Document Management configuration XML file. For example, the **datasetName** of a logo in the configuration XML file is **Logo_1**. Therefore, the name of the dataset for the logo must also be **Logo_1**.

    b. Import the logo image file, distribution statement text file, or Document Management configuration XML file.

    c. In **Relation**, select **Document Page** while attaching the logo, distribution statement, or Document Management configuration XML file.

    For PDF command and MetaData Stamp files, select **Tc_Attaches**.

    d. Click **OK**.

4. Right-click the newly created dataset and click **Properties On Relation**.

5. In the **Properties** dialog box:

    a. Click **Show Empty Properties** to view all properties.

    b. In **Page Type**, select **Logo**, **Distribution Statement**, or **Document Configuration** based on the entity you are creating the dataset for.

    c. Click **OK**.

6. Select the dataset and click **File→New→Workflow Process** to start a workflow process to release the dataset.

7. To release the dataset, in the **New Process Dialog** dialog box:

a. Select **TCM Release Process** in **Process Template**.

> Note:
>
> The default value of the release status for Document Management template (*DMTemplate*) is **TCM Released**. This value is set to *MaturityStatuses* constant in BMIDE. You can customize it by adding a status type. You can also understand what are global constants, business objects constants, and how to run data model reports to understand the *MaturityStatuses* constant.

b. In the **Attachments** tab, expand **Targets** to select the logo, distribution statement, or Document Management configuration schema file that is attached to the dataset.

c. Click **OK**.

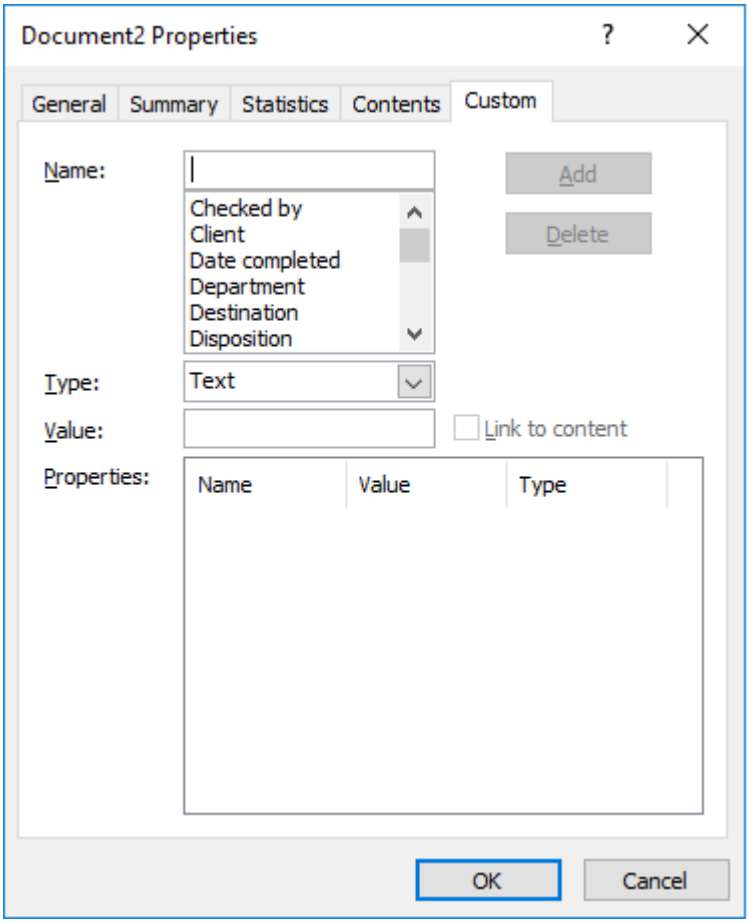
A release flag is displayed next to the released dataset.

Define the placement of attributes in Microsoft Word documents

1. Check out the required template file.
2. Open a Word file in which you want to include information.
3. You must add each property that you defined for a logical object member as a custom property. To add each property:

   a. Click **File→Info→Properties→Advanced Properties**.

   b. In the **Properties** dialog box, click **Custom**.

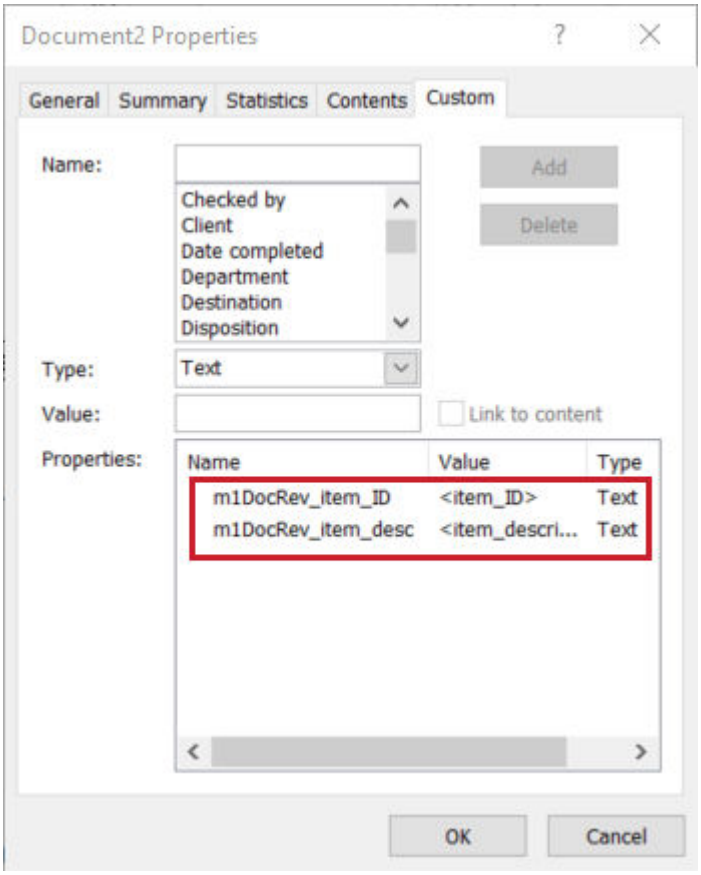


c. Enter the ID of the member property in **Name**, for example, **m1DocRev_item_ID**.

d. Select **Type**.

e. Enter the information in **Value**. As a default value is required, enter either a space or information that must be displayed in the document.

f. Click **Add**.

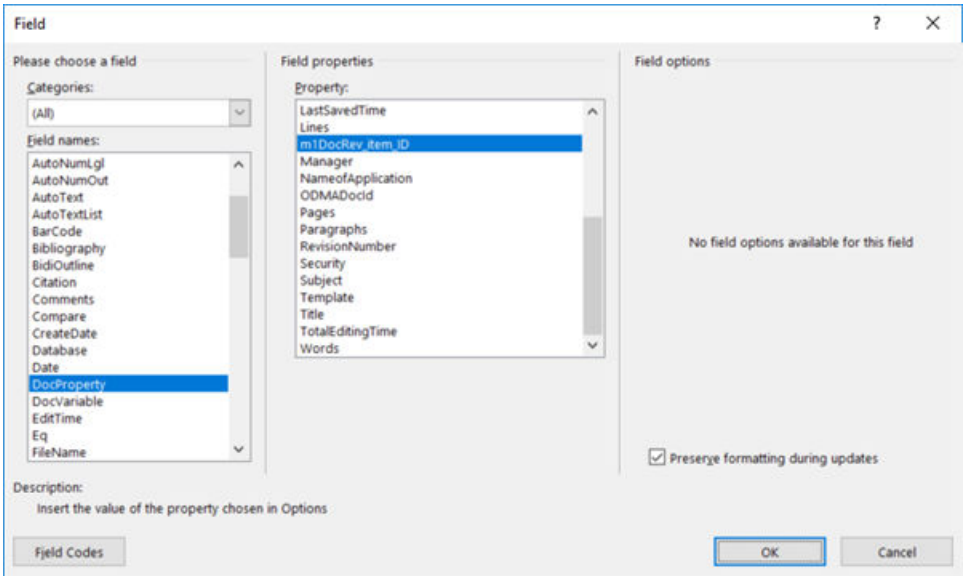


g. Add other required properties and finally click **OK**.

h. To insert the newly created custom properties:

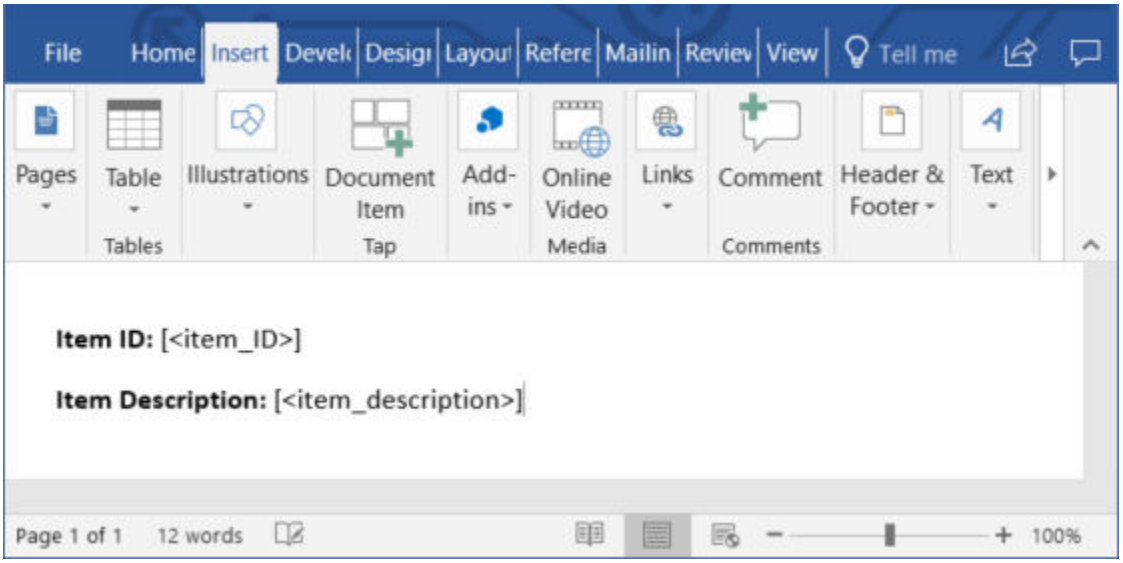
   A. In Word, place the cursor at the location where you want to enter a custom property and click **Insert→Quick Parts→Field**.

   B. In the **Field** dialog box, select **DocProperty** in **Field names** and select a custom property that you created in **Property**.



C. Click **OK**.

The value of the custom property is inserted at the selected location.



To see the value of a custom property later, right click the inserted property and click **Toggle Field Codes**.

4. To insert distributions statements:

   a. Click **FILE→Options**.

   b. In the **Word Options** dialog box, click **Customize Ribbon**.

c. In **Customize the Ribbon**, select **Main Tabs**.

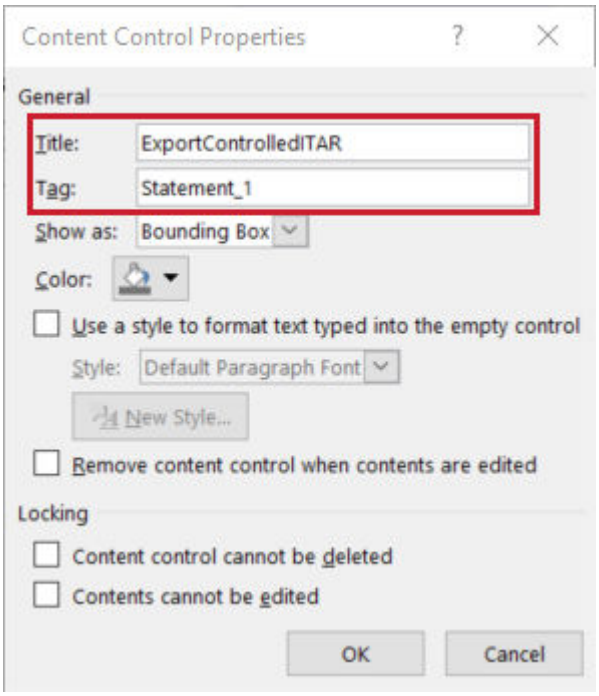
d. In **Main Tabs**, select the **Developer** check box and click **OK**.

e. Go to the **DEVELOPER** tab.

f. Place the cursor where you want to position the distribution statement and click one of the **Aa** icons to insert either rich text content or plain text content.

g. Select **Click here to enter text** and click **Properties**.

h. In the **Content Control Properties** dialog box, enter **Title** and **Tag**. The value of the tag must be same as that specified in the **MSOfficePlacement** tag value in the **Document Management configuration file**.



The value of the **datasetName** that matches the title value you specified in Word is placed as the distribution statement.

i. Click **OK**.

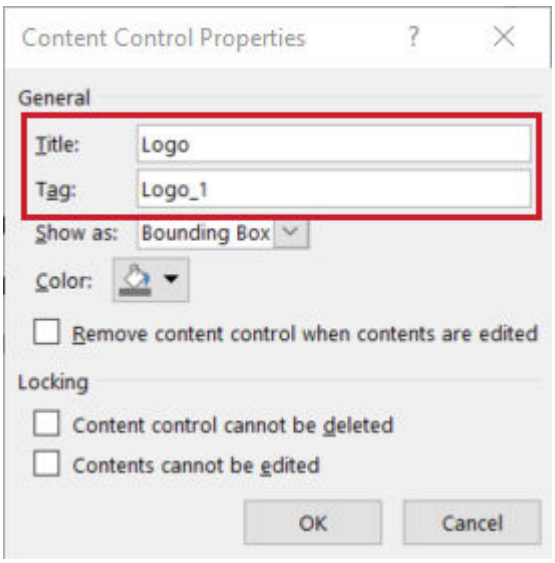
5. To insert logos:

a. Go to the **DEVELOPER** tab.

b. Click the **Picture Content Control** icon.

c. Select the image and click **Properties**.

d. In the **Content Control Properties** dialog box, enter **Title** and **Tag**. The value of the tag must be same as that specified in the **logoList** tag value in the **Document Management configuration file**.



The value of the **datasetName** that matches the title value you specified in Word is placed as the logo.

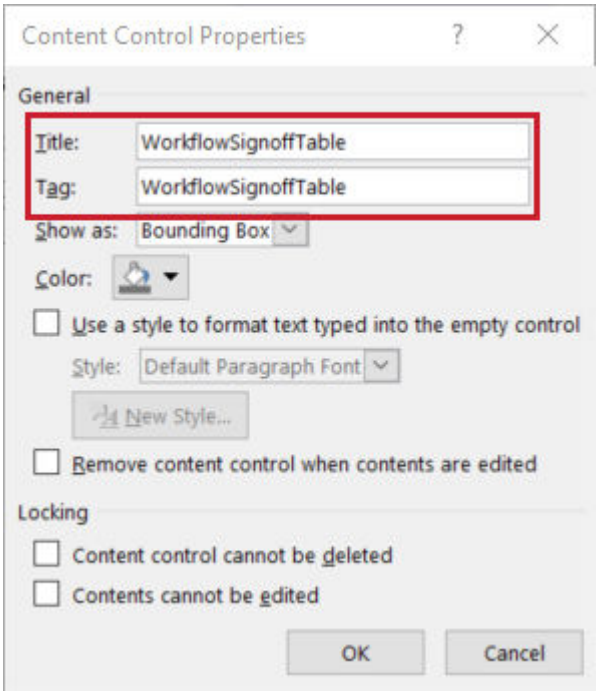
6. To insert the workflow signoff table:

a. Go to the **DEVELOPER** tab.

b. Place the cursor where you want to position the workflow signoff table and click one of the **Aa** icons to insert either rich text content or plain text content.

c. Select **Click here to enter text** and click **Properties**.

d. In the **Content Control Properties** dialog box, enter **Title** and **Tag** as **WorkflowSignoffTable**.



e. Click **OK**.

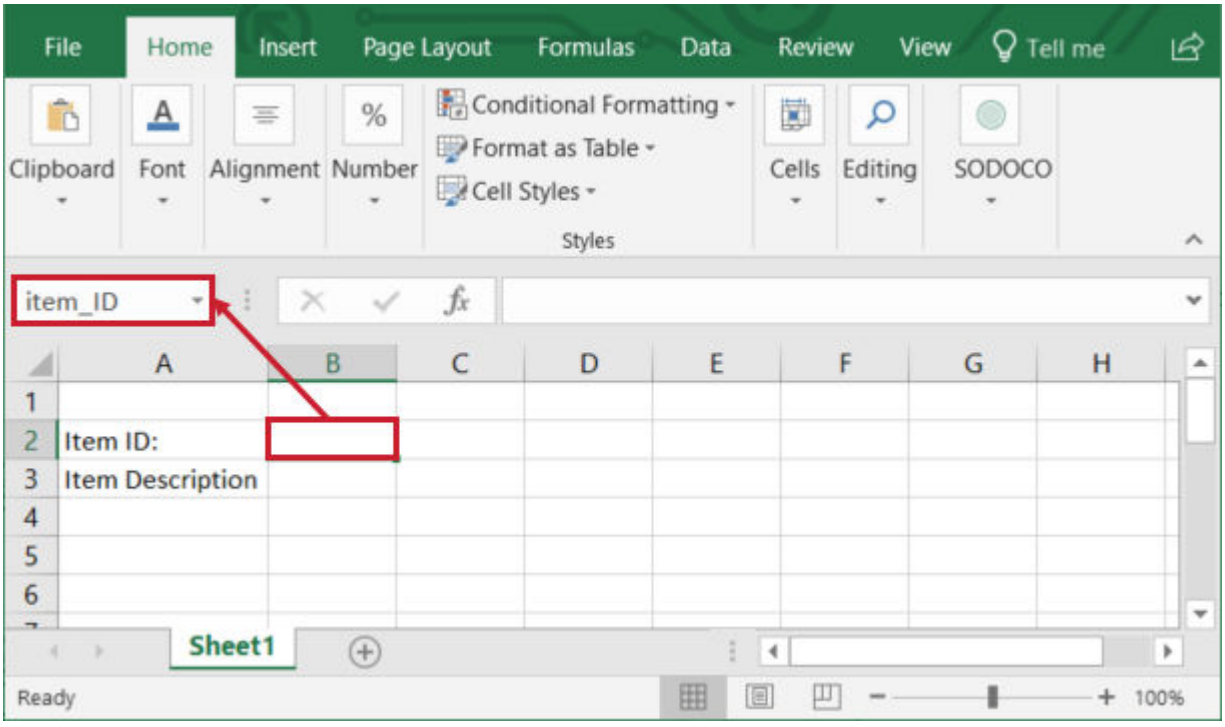
7. After creating the Word document, **attach it to the required document template revision item** and check in the template.

Define the attributes of information in Microsoft Excel files

1. Check out the required template file.

2. Open an Excel file in which you want to include attributes.

3. Place the cursor at the cell where you want to enter a custom property and change the cell name to the property value that you specified while defining the logical object, for example, **item_ID**.



4. To add properties:

   a. Click **File→Info→Properties→Advanced Properties**.

   b. In the **Properties** dialog box, click **Custom**.

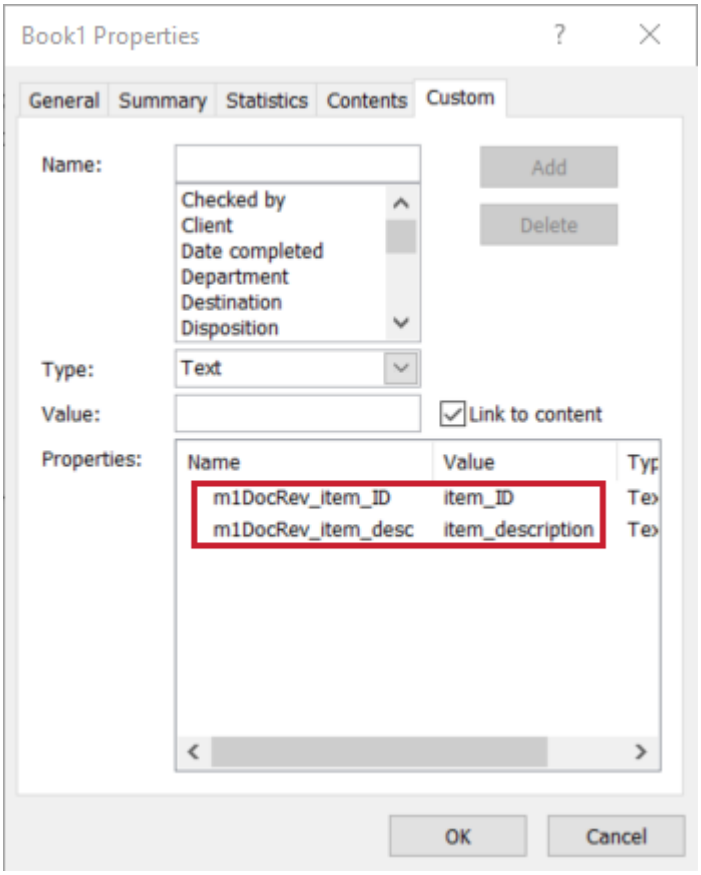
   c. Enter the ID of the member property in **Name**, for example, **item_ID**.

   d. Select **Type**.

   e. Enter the information in **Value**. As a default value is required, enter either a space or information that must be displayed in the file.

   f. Select the **Link to content** check box.

   g. Click **Add**.

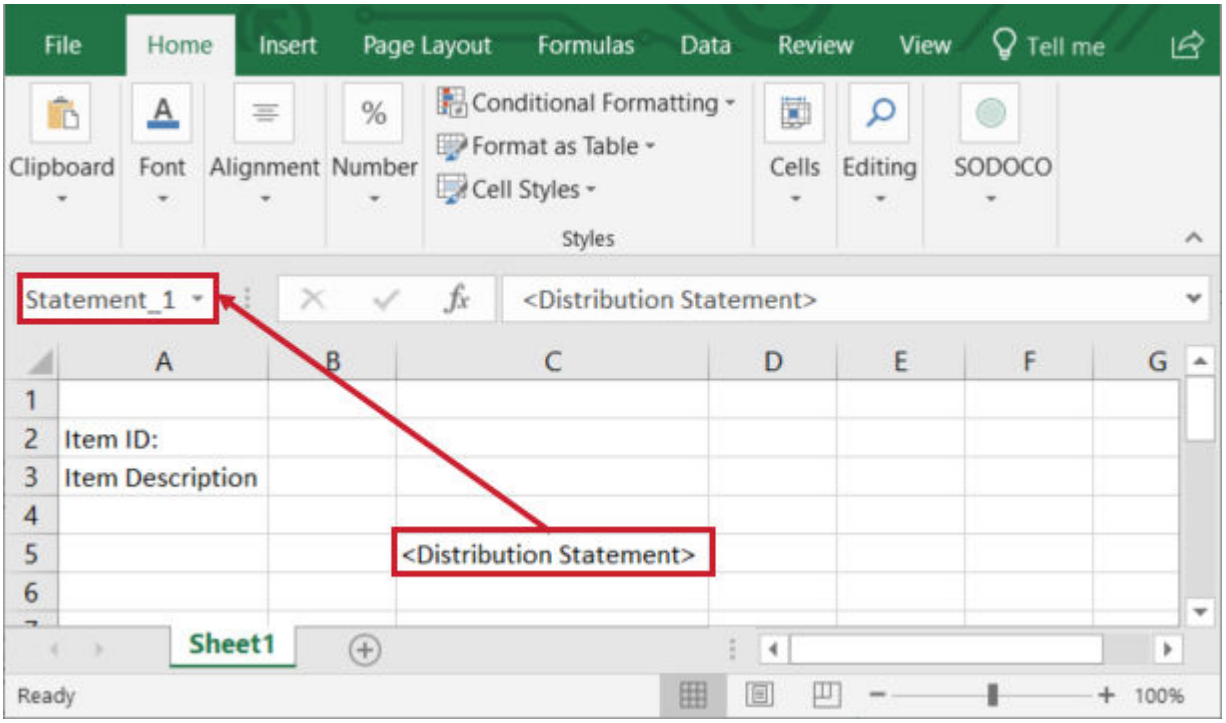


h. Add other required properties and finally click **OK**.

5. To insert distribution statements:

   a. Select a cell in which you want to insert the distribution statement.

   b. In the **Name Box**, rename the cell to the name of the distribution statement that you specified in the **MSOfficePlacement** tag value in the **Document Management configuration file**.

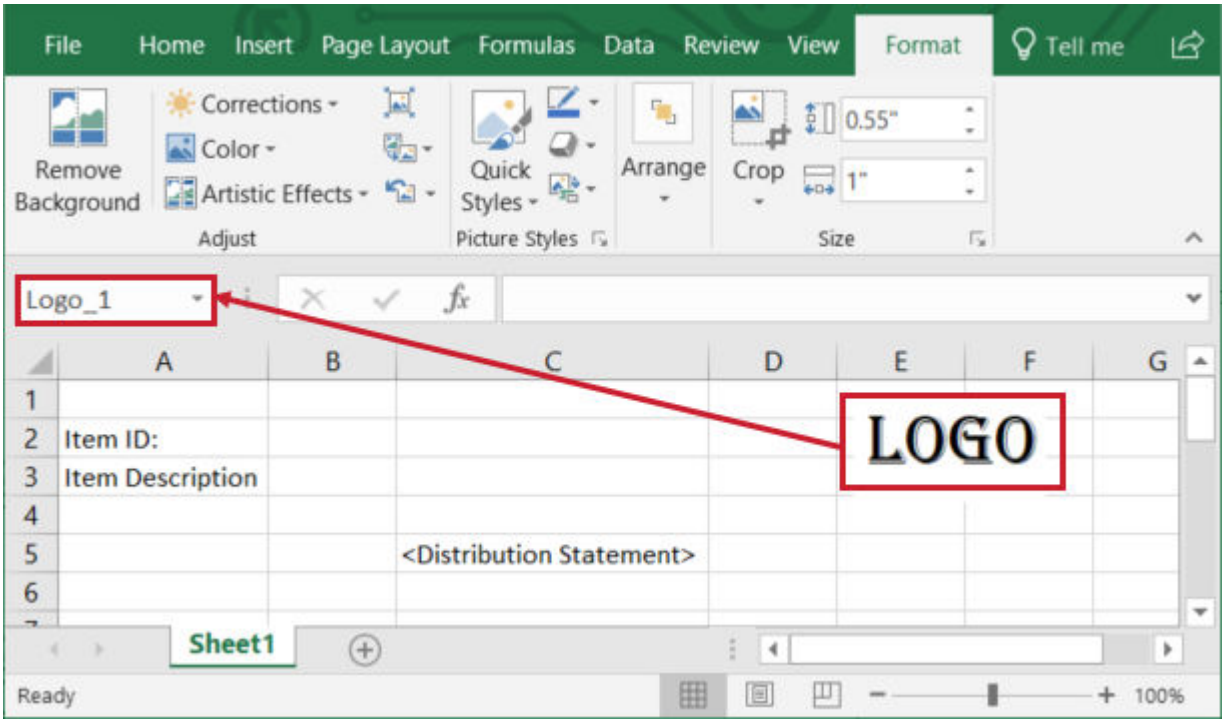


6. To insert logos:

   a. Click **Insert→Pictures**.

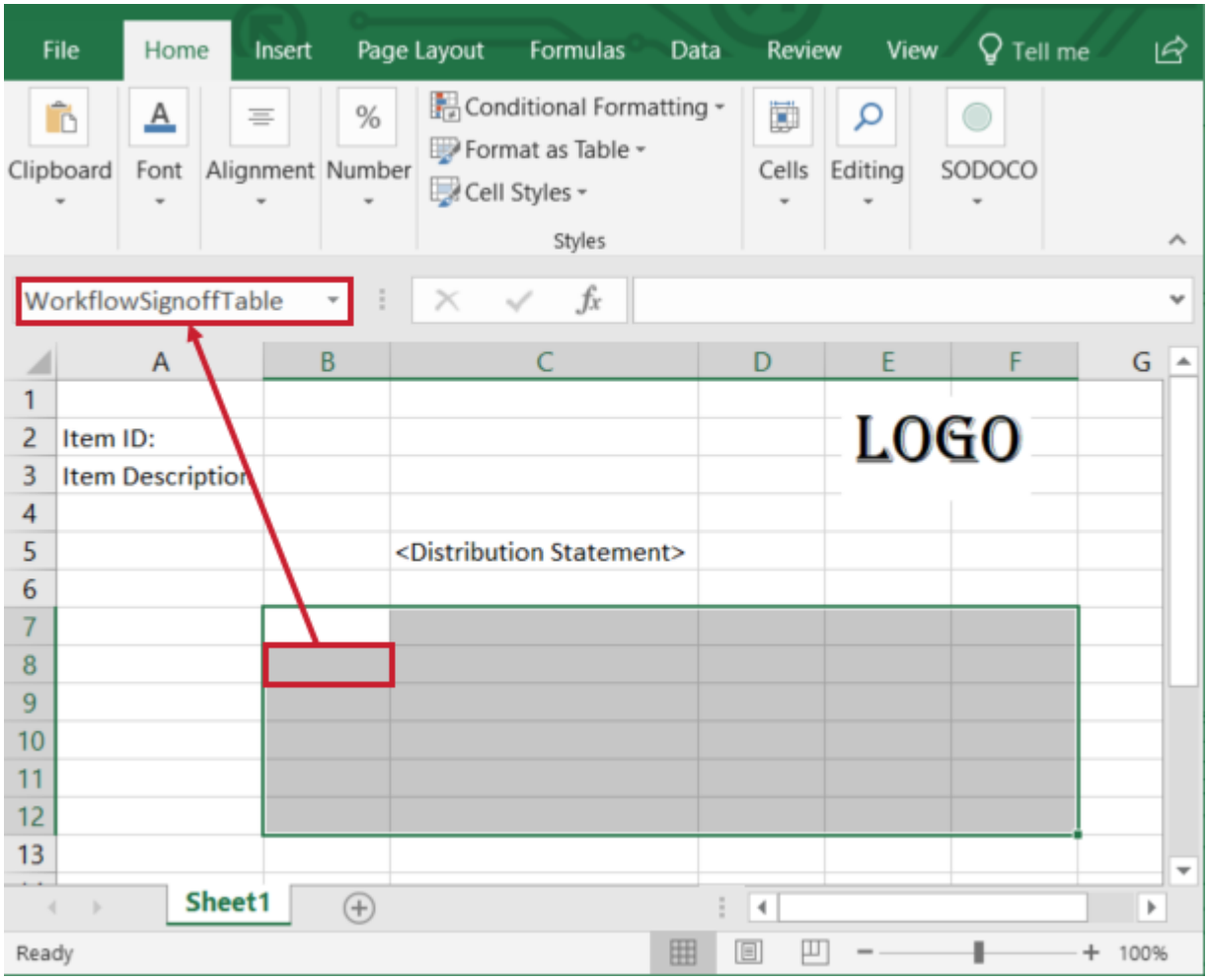
   b. Select the required logo and click **Insert**.

      Ensure that the name of the logo is the same as that specified in the **Document Management configuration file**.



7. To insert the workflow signoff table:

   - Select a cell where you want to position the workflow signoff table and change the name of the cell to **WorkflowSignoffTable**. For example, if you want the workflow signoff table to appear from cells **B7** to **F12**, you must select the cell **B8** and rename it to **WorkflowSignoffTable**. This is because the header of the workflow signoff table is placed in cells **B7** to **F12**.



8. After creating the Excel file, **attach it to the required document template revision item** and check in the template.

Define the placement of attributes in Microsoft PowerPoint files

1. Check out the required template file.

2. Open a PowerPoint file in which you want to include attributes.

3. You must add each property that you defined for a logical object member as a custom property. To add each property:

   a. Click **File→Info→Properties→Advanced Properties**.

   b. In the **Properties** dialog box, click **Custom**.

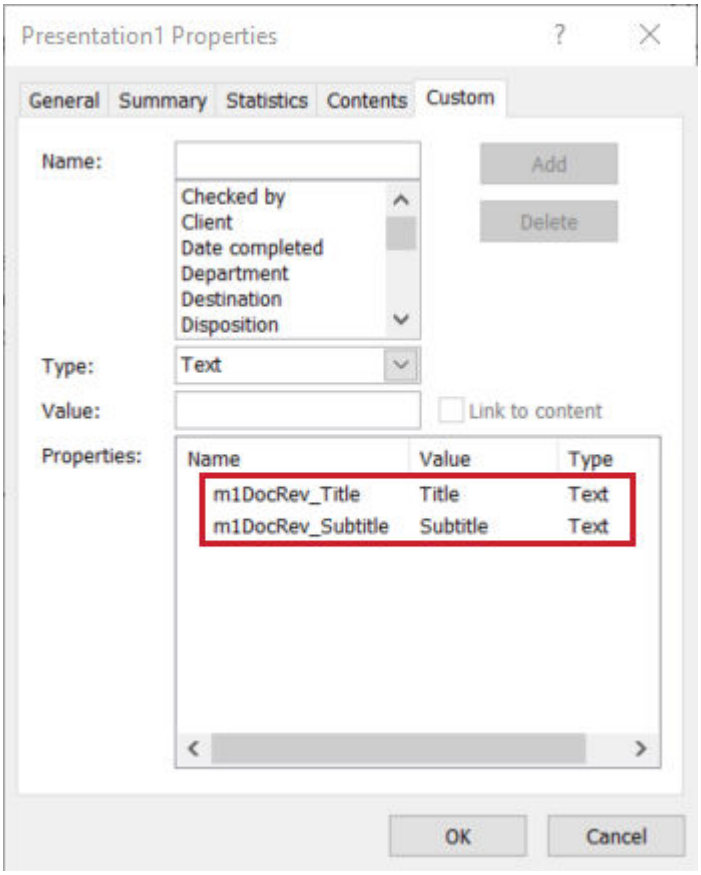
   c. Enter the ID of the member property in **Name**.

d. Select **Type**.

e. Enter the information in **Value**. As a default value is required, enter either a space or information that must be displayed in the document.

f. Click **Add**.

g. Add other required properties and finally click **OK**.



4. To insert the newly created custom properties:

a. Insert a text box for each property. You can also use the existing text boxes, if any.

b. Click **Home→Arrange→Selection Pane**.

c. In the **Selection** pane, rename each text box. The name of the text box must be same as the name of the property you added earlier.

For example, rename the text box in which you the title to appear as **m1DocRev_Title**.

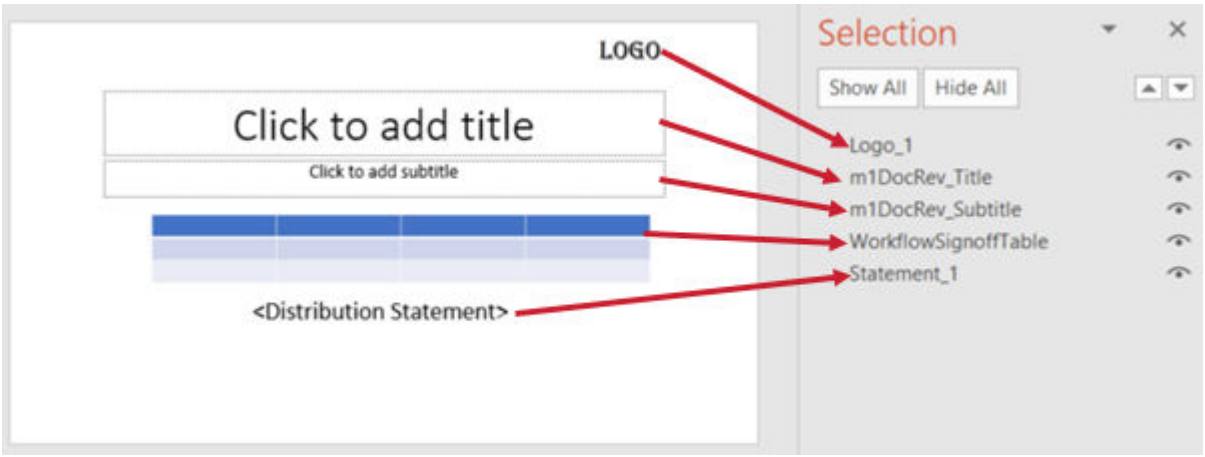
5. To place a logo, click **Insert→Pictures** and select an image.

Place the image where you wish the logo to be placed. In addition, rename the image to the value of the logo specified in the **Document Management configuration file**.

6. To place a distribution statement, insert a text box at the appropriate place and rename the text box so that its name is same as the tag value of the distribution statement in the **Document Management configuration file**.

7. To include the workflow signoff table, click **Insert→Table**.

   Place the table where you wish the workflow signoff table to be displayed and rename the table as **WorkflowSignOffTable**.



8. After creating the PowerPoint file, **attach it to the required document template revision item** and check in the template.

Set up system stamps

1. In the Business Modeler Integrated Development Environment (BMIDE), click **BMIDE** > **New Model Element**.

2. In the **Add New Model Element** dialog box, select **System Stamp Configuration** in **Wizards** and click **Next**.

3. Enter the following information for the system stamp configuration:

Field	Action
Business Object	Select **ItemRevision**.
Condition	Select the condition under which this stamp is applicable.
Applies To	Choose one of the following:

Field	Action
	• **Print**: To include the system stamps in a printed file. • **Render**: To include the system stamps in a rendered file. • **PDF_Control**: To include the system stamps in an existing PDF file when it is sent to a stamp workflow. • **PrintAndRender**: To include the system stamps in both printed and rendered files.
Include User Name?	Select this check box to include the print requester's name in the printed files.
Include Date And Time?	Select this check box to include the print date and time in the printed files.

4. Click **Next** and enter the following stamp information:

Field	Action
Properties	Specify the properties that must be included in the printed or rendered files. You can optionally specify a prefix that must precede a property, for example, **Document Name**: <**object_name**>.
User Stamp	Type the text that you want to appear on the document, such as *Internal Distribution*.
Watermark	Type the text that you want to appear as a watermark, such as *Confidential*.
MDS Template	Select the MetaData Stamp template that defines how the stamps must be applied in the printed and rendered files. To apply stamps in existing PDF files, select the PDF control command XML file. To apply stamps during attribute synchronization, select the Document Management template revision for stamps.

5. Click **Finish**.

The newly created system stamp configuration is listed under **Extensions**>**Document Management** > **System Stamp Configuration**.

Relate logical objects with datasets

You can relate one or more logical objects to a dataset. To do so:

1. If you are an administrator, in Active Workspace, open and check out the document template revision containing the required dataset.

   Note:

   To apply the **Logical Object Type Relation** property to a document revision other than the document template, ensure that the style sheet of that document revision contains the *Fnd0LogicalObjectTypeRel* property added to it. If the property is already not added to the document revision, edit the style sheet for adding the property and then apply it.

2. In the **Files** section, change the view to **Table**.

3. In the **Files** section, select the dataset and double-click in the **Logical Object Type Relation** column of the table.

4. Select the logical object that you created for the document revision in **Logical Object Type Relation**.

5. Click **Edit > Check In**.

Note:

Modifications to a *.docx*, *.pptx*, or *.xlsx* file's body text (irrespective of the values of the mapped attributes) can get lost if a logical object is attached to it, that is, if the **Logical Object Type Relation** is defined for the MS WordX, MS PowerPointX, or MS ExcelX dataset. This can happen because the logical object attribute exchange is asynchronous by design. If you update the dataset file during this time, both processes (dispatcher and user) update two different versions of the same datasets.

To work around this problem, set the **Create Derived Visualization Data** IRDC to perform an **Optional render after checkin** (so that check-in is not stopped). Before the system sends the MS WordX, MS PowerPointX, or MS ExcelX dataset to the dispatcher for rendering, it triggers the logical object attribute exchange first synchronously.

Set up a workflow to include attributes and stamps in Microsoft Office files

By default, the Teamcenter attributes, logos, distribution statements, and workflow signoff tables are included in Microsoft Word, Excel, and PowerPoint files when a user creates, checks in, revises, saves, or renders a document revision. However, a user can also send a file through an attribute exchange workflow to include the information. For this, you must set up a workflow task for the users by using the **DOCMGT-update-docprop-logicalobject** action handler.

To set up a workflow task:

1. Create a workflow process template.

   a. In Workflow Designer, click **File→New Root Template**.

b. In the **New Root Template** dialog box, enter a name in **New Root Template Name**.

c. Select **Empty Template** in **Based On Root Template** and **Process** in **Template Type**.

d. Click **OK** to create a workflow process template.

2. Add a task to the workflow process template.

a. On the toolbar, click **Edit Mode** and then click **Task**.

b. In the process flow pane, double-click where you want to place the new task.

A new task appears with the default name **New Task #**. In the **Name** box, type a name for the task.

3. Link the task to its predecessor and successor tasks.

a. Click the task node you want to specify as the predecessor task.

> Caution:
>
> Do not click the title bar of the task node as this action drags the task node to a different location.

b. Drag your cursor to the task node that you want to specify as the successor task.

A link arrow follows the cursor as you drag. When your cursor moves over a task node, the node is highlighted.

c. Release the mouse button.

A link arrow connects the predecessor and successor nodes.

4. Configure the task.

a. Right-click the task and click **Task Properties**.

b. Click the **Task Handler** pane.

c. In **Task Action**, select **Complete**.

d. In **Action Handler**, select **DOCMGT-update-docprop-logicalobject** to create a workflow task to include stamps in Microsoft Word, Excel, and PowerPoint files during attribute exchange through logical objects.

e. Click **Create**.

Understand the DOCMGT-update-docprop-logicalobject action handler

Description Updates the datasets (for example, MSWordX with a **.docx** extension or MSExcelX with a **.xlsx** extension) associated with the target item revisions with the latest attribute exchange data. Attribute exchange data can include Teamcenter properties, logos, distribution statements, and workflow sign off tables, if the target object is in a review task. Attributes are exchanged between Teamcenter and the files.

Note:

- The generic (logical object) attribute exchange currently supports Microsoft Word, Excel, and PowerPoint datasets only.
- The Microsoft Word, Excel, and PowerPoint datasets must be **related to the logical objects** for the generic attribute exchange to occur.
- Target item revisions must be valid and checked in.

  The attribute exchange process from this workflow action handler bypasses the **Fnd0TriggerLOAttrExch** business object constant configuration.
- Logos and distribution statements must be enabled based on their document configuration setting.
- System stamp must be enabled for a business object revision and logical objects must be defined for its datasets.

Caution:

Document Management does not support multiple references in a dataset. Attempts to use such dataset can produce unpredictable results.

The update is synchronous.

Syntax **DOCMGT-update-docprop-logicalobject**

Arguments **-ignore_errors**

(Optional) Specifies that the current task can be continued on to the next task even if the current task fails. If this optional argument is not specified, the current task might stop on failures (including password protected file or dataset is checked out, or object cannot be saved), and might not continue to the next task.

Placement Place on the **Start** action of a **Task**.

> Caution:
>
> Do not place this handler on the **perform** action of the **perform-signoffs** task. Otherwise, this handler runs multiple times.

Restrictions Item revisions with attached datasets such as Microsoft Word, Excel, or PowerPoint must be included as targets of the workflow process.

Export files along with their attributes to another site

You can export files attached to a document revision from one site to another site. Along with the files, their attributes also get exported. Before exporting the files,you must first **set up logical objects**. Document authors later attach files as datasets to a document revision by using **Logical Object Type Relation**. In this relation, they select the logical object that you set.

Exporting files in a multisite environment

The export of the **Fnd0LogicalObjectTypeRel** relation is not supported by the **objio** export mechanism.

1. Import the logical object that you created at the source and the target sites.

   admin_data_import -adminDataTypes=LogicalObjects -u=*username* **-p=***password* **-g=***group_name* **-inputPackage=***logical_object_path*

   For example:

   admin_data_import -adminDataTypes=LogicalObjects -u=*Tc-admin-user* **-p=***password* **-g=***group* **-inputPackage=***C:\Development\Export\LO\LogicalObjectsExport106.zip*

2. Export the document revision from the source site.

   data_share -u=*username* **-p=***password* **-f=offline_export -item_id=***document_rev_item_id* **-optionset -dir=***folder_name*

3. Import the document revision at the target site.

   data_share -u=*username* **-p=***password* **-f=offline_export -optionset -dir=***folder_name*

Exporting files in a site consolidated environment

1. Import the logical object that you created at the source and the target sites.

   admin_data_import -adminDataTypes=LogicalObjects -u=*username* **-p=***password* **-g=***group_name* **-inputPackage=***logical_object_path*

For example:

admin_data_import -adminDataTypes=LogicalObjects -u=*Tc-admin-user* **-p=***password* **-g=***group* **-inputPackage=***C:\Development\Export\LO\LogicalObjectsExport106.zip*

2. Export the document revision.

   tcxml_export -u=*username* **-p=***password* **-f=***Site_Consolidation_xml_output* **-item=***document_rev_item_id* **-low_level -force_retraverse**

Verify if attributes and stamps are included in template files automatically

To verify if you have configured generic attribute exchange correctly, you can perform one of the following:

- Start a workflow process by using the **DOCMGT-update-docprop-logicalobject** workflow action handler.

- Create, check in, revise, save as, or render a document revision with an attached MSWordX, MSExcelX, or MSPowerPointX file. Verify that attributes are automatically filled in the attached dataset. Also verify that other information such as logos, distribution statements, and workflow signoff tables are inserted into the Word document during the attribute exchange.

  Currently, on creating a document revision, the attributes are synchronized only if the attribute exchange direction is set as **Teamcenter to File**.

> Note:
>
> Logical object attribute exchange does not support **Teamcenter to File** with the display values of the properties. In stead, use the internal values of the properties. The properties include list of values (LOV).

7. Building input files for importing Excel data into Client for Office

Input to the Excel import process

To import structures or requirements into Teamcenter, you can use the Excel import process. This process enables you to import your existing data into Teamcenter.

> Caution:
>
> To define Excel import files manually, you must understand the data in each data file column and the way in which the data relates to the Teamcenter schema. Siemens Digital Industries Software recommends that control files be defined only by Teamcenter administrators.

The Excel import process requires input from two separate Excel files, a *data file* and a *control file*.

- The data file carries the actual structure data to import. The columns contain the names and values of Teamcenter object properties, relationships, and relationship properties. The import process creates or modifies the structure in Teamcenter according to this data.
- The *control file* supplies instructions that tell the import process how to build an intermediate structure in memory.

  The control file columns contain certain keywords and tags that govern data parsing and that map the data to the Teamcenter schema. Each data file column corresponds to one control file column.

> Caution:
>
> - The data file contents must conform with the processing instructions in the control file.
> - The control file contents must conform with the Teamcenter schema.

During the import procedure, you associate the data file with the appropriate control file. Teamcenter refers to the control file instructions to build an intermediate structure in memory.

Teamcenter performs the following checks against the intermediate structure:

- Determines if the control file schema mapping is valid for Teamcenter schema elements.
- Compares the data file column sequence with the control file column sequence and displays the column mapping for verification.
- Examines reference designators, find numbers, and quantities for format validation.

The data is applied in Teamcenter when the import is complete.

Note:

- Only imprecise structures can be imported from Excel.
- The import process does not configure the structure in Teamcenter, for example, by defining revision rules.
- Absolute occurrences and incremental changes are not supported.

Data file contents

The data file rows represent the elements in a structure hierarchy such as functions or logical blocks in a design model including interfaces, connections, and relationships among elements.

The data file columns can contain the following:

- Teamcenter object properties such as object types, object identifiers, object names, part numbers and part identifiers.
- Relationships of the following types:
  - A **PSOccurrence** object is used to distinguish how items occur in a structure.

    An occurrence consists of one component in an assembly including its relative position with respect to its parent assembly.
  - A Generic Relation Management (GRM) rule applies constraints on objects based on the relationship between primary and secondary objects.

    The primary and secondary business object trees in the GRM rule interface allow relation rules to be configured between primary and secondary objects at any level of the business object hierarchy.
- Relationship properties such as reference designators, budget values, and quantities.
  - A *reference designator* is an identifier appended to part properties in the structure.

    It allows unique identification when the part is used several times in the same structure. Reference designators need only be unique across a single level of the structure.
  - *Find numbers* are additional identifiers or labels for organizing items in a structure relationship.

    Teamcenter assigns a unique find number to each line in a structure. When an item is added to a structure, the new line receives the next available find number in the defined sequence.

- A *quantity* is the number of individual components associated with a line in the structure.

  If the structure relationship has an associated quantity, the number of relative occurrences associated with the structure relation equals the quantity.

- The **No Load** property indicates a part that is designed into the assembly but is not to be loaded during the manufacturing process.

  You can mark such parts in the data file by adding a column for the **No Load** property and entering **Yes** in the cells for the particular items.

Control file contents

To build an intermediate structure in memory, the Excel import process refers to a control file for instructions. The control file has two sections of processing instructions.

- *Context tags* occupy the first four rows, beginning in column **A**.

  Context tags let you assign custom identifiers to imported object types. These identifiers, the tag values, start in column **B**.

- A *rule table* starts in row **5** of the control file worksheet, beginning in column **A** or in any other column.

  The rule table columns contain certain keywords and tags that:

  - Specify which data file columns to include or exclude in the intermediate structure.
  - Mark transitions in the processing, for example, the beginning of the data for the next structure element.
  - Map included data file columns to the Teamcenter schema.

  Except for columns that mark transitions, each rule table column corresponds to one data sheet column, in sequence from left to right.

Control sheet validation

The control sheet information is checked against the Teamcenter schema during the import process.

- If invalid information is found, you must cancel the process, correct the control sheet, and restart the import.
- If the control sheet information is valid, you can continue the import.

Column mapping verification

When the control sheet is validated, a summary of the column mapping displays for your verification.

- If the mapping is incorrect, you can cancel the process, make the necessary corrections, and restart the import.
- If the mapping is correct, you can continue the import.

In the **Column Mappings** dialog box:

- Each top-level node represents an object type or a relationship type in the control sheet.
- For that object type or relationship type, each second-level node shows the following:
  - First, an Excel column heading in the data sheet.
  - In parentheses, that column's name in the data sheet.
  - To the right of the equal sign (=), one of the following:
    - The corresponding keyword tag in the control sheet.

      The dialog box shows the keyword tag itself if the tag value is blank in the control sheet.
    - The value of the corresponding keyword tag in the control sheet.

      The dialog box shows the tag value if it is entered below the keyword tag in the control sheet.

Caution:

The import process cannot continue if the dialog box does not display at least one second-level node for a top-level node.

The process can continue if control columns map to data columns other than the ones that you intend.

Data error detection

On continuation after column mapping verification, the Excel Import wizard opens. Here, the process checks the data sheet for errors.

> Note:
>
> If the data sheet contains errors, processing stops and the entire transaction is rolled back.

If there are no data errors, the import process compares the intermediate structure with Teamcenter.

- If a corresponding structure does not exist in Teamcenter, the structure is created according to the Excel data.

  > Note:
  >
  > For a product structure, only imprecise assemblies can be created.

  > Tip:
  >
  > To work with the new structure in the Teamcenter rich client:
  >
  > 1. Use the search feature to find the structure's peak element.
  > 2. In the search results, select the peak element and send it to the desired rich client application.

- If the structure exists in Teamcenter:

  - New elements and attributes in the Excel data are created in the structure.
  - Changes in the Excel data are applied to the existing structure.

    > Note:
    >
    > Only the latest working revisions are modified.

  - Existing structure elements not represented in the Excel data are removed from the structure.

    > Note:
    >
    > Elements that are removed from a structure remain in Teamcenter.

Defining Excel import data files manually

Process for defining Excel import data files

Note:

Client for Office import features support only versions 2010 and 2013 of Microsoft Office Excel.

Caution:

To define a data file manually, you must understand the data in each column and the way in which the data relates to the Teamcenter schema. Siemens Digital Industries Software recommends that data files be defined only by Teamcenter administrators.

The process for defining a data file manually includes the following steps:

1. Start with a new Excel workbook.
2. Enter the data that you want to import to Teamcenter.
3. (Optional) Group multiple identical occurrences by *packing* the instances together in the same row.

Note:

You can also edit files from external sources to further define the data file.

Caution:

The import process checks the data file for errors. If the data contains errors, processing stops and the entire transaction is rolled back.

Enter object data in the data file

Caution:

You must also define a control file that conforms with the object data configuration.

Note:

Data is applied in Teamcenter when the import process is complete.

Tip:

On the **Data** tab of the Excel ribbon, you can use the **Outline** features to show or hide levels in the structure hierarchy.

Action in Teamcenter structure	Steps in Excel
Create a new element	1. Insert a row at the desired position in the hierarchy. 2. Enter the element attributes in the appropriate cells of the new row. Note: • A corresponding item and item revision are created in Teamcenter. • The element is added to the structure as an occurrence of the new item revision.
Modify element attributes	• Enter the new values in the corresponding cells.
Remove an element	• Delete the corresponding row. Warning: • Do not delete the row if the element is the only child for a given parent in both Teamcenter and Excel. Instead, add the suffix **_void** in the cell for the child object identifier. • Do not add the **_void** object identifier suffix for an element that has a sibling at the same level. Note: Only the occurrence of the element is removed. The underlying item and item revision remain in Teamcenter.
Create a new attribute	1. Insert a column to the right of the current rightmost column. 2. Enter the attribute name in the new column's top cell. 3. Enter the attribute values in the cells below the name.

Action in Teamcenter structure	Steps in Excel
	Warning: You must insert a column for each of the default attributes. If the data file does not contain all of the default columns, import processing stops and the entire transaction is rolled back.
Remove an attribute	• Delete the corresponding column. Warning: Do not delete any of the default columns. If the data sheet does not contain all of the default columns, import processing stops and the entire transaction is rolled back.

Group multiple identical occurrences

Packing multiple occurrences in one row

A row in a data file represents a single level of a structure. When a structure contains many identical occurrences at the same level, you can save time by *packing* them in one row.

For example, assume that:

- You are entering data for a bicycle wheel assembly that contains 50 spokes.
- Each spoke is identical to the others except for its position.

In this example, it is time-consuming to enter a separate line for each spoke. Instead, you can:

- Enter one line that represents all spokes.
- Use the **Quantity** property to indicate that there are 50 spokes.

> Note:
>
> You can use the **Quantity** property to pack occurrences that do not have *reference designators*, as well as those that do.
>
> Typically, reference designators are used to label components in schematics and printed circuit board layouts. For example, the occurrence of a part can have a reference designator that relates the part to a particular location on a circuit board.

You can pack occurrences if they meet all of the following conditions:

- They have the same item revision.
- They have the same find number.
- Their other attributes are identical.
- None have variant conditions or they all have the same variant condition.

Caution:

Do not pack an occurrence for which the **No Load** attribute value is **Yes**. Teamcenter does not allow these lines to be packed with others that share the same reference designator and find number.

Note:

Special considerations are involved in packing occurrences that have reference designators.

Packing occurrences that have reference designators

Caution:

- If you pack occurrences that have reference designators, a **Quantity** property column is required for the row. The **Quantity** value must equal the number of reference designators.

  The **Quantity** property can be used also to pack occurrences without reference designators.

- Do not pack lines with blank reference designators in the data file.

  Such lines cannot be packed in the Teamcenter rich client and cannot be imported.

For packing occurrences with different reference designators, you add a column for the **Reference Designator** attribute. In the cell for the packed occurrence, you enter a value that represents the entire set of reference designator values.

For example, assume that each of the reference designator values **R1**, **R2**, and **R3** identifies the position of one occurrence. When you pack the occurrences, you can enter the range of values, **R1-R3**, in the **Reference Designator** cell for the line.

Note:

Use a hyphen (-) to separate a range of reference designators, for example, **R1-R3**.

You can also enter a combination of ranges and individual values. For example, if you pack five occurrences and the reference designator values are **R1**, **R2**, **R3**, **R7**, and **R9**, you can enter **R1-R3,R7,R9** in the **Reference Designator** cell.

> Note:
>
> Use a comma (**,**) to separate individual reference designators, for example, **R7,R9**.

Reference designators in packed lines are subject to the following conditions:

- The prefixes must be the same. For example:

  Valid: **R1-R3**

  Invalid: **R1-C1**

- The numbers must be in sequence. For example:

  Valid: **R1-R3**

  Invalid: **R3-R1**

> Note:
>
> A reference designator containing the concatenated characters of other reference designators, for example, **R1SK1**, is treated as a single reference designator for packing.

Excel import data validation

Data validation checks

Before converting the data into an intermediate structure in memory, the Excel import process checks the format of reference designators, find numbers, and quantities for validity.

> Warning:
>
> To avoid processing errors, ensure that you enter each of these data elements in its valid format.

Data element	Description
Reference designators	Controlled by the **PS_Reference_Designator_Validation** preference. • If the preference value is **true**, the import process checks reference designator values for uniqueness and the correct format.

Data element	Description
	Note: Additional considerations apply when you pack lines containing reference designators. • If the value is **false**, reference designator validation is not enabled. The value is set to **false** by default.
Find numbers	Controlled by the **PS_Find_Number_Validation** preference. • If the preference value is **true**, the import process checks find numbers for the correct format. If the preference value is **true**, find numbers must meet the following conditions: • Find numbers must not be null or zero. • Each find number must be unique within the same parent. • Find numbers in the Excel data file must conform to the find number configuration at the importing site. • If the value is **false**, the import process does not validate find numbers. The value is set to **false** by default. Note: Your Teamcenter administrator can customize the find number configuration at your site. If you have questions about find numbers, consult your Teamcenter administrator.

Reference designator format

A reference designator consists of a prefix and a number, in the following format:

PrefixNumber

Replace *Prefix* with one of the following:

- A single uppercase letter, for example, **R***Number*.

- A string of two or more uppercase letters, for example, **SK***Number*.
- A string of one or more uppercase letters and integers, for example, **R1***Number*, **SK1***Number*, or **R1SK1***Number*.

Replace *Number* with an integer, for example, *Prefix***1**.

Caution:

The reference designator must be unique for each child of a given parent.

Find number format

Note:

Your Teamcenter administrator can customize the find number configuration at your site. If you have questions about find numbers, consult your Teamcenter administrator.

If the **PS_Find_Number_Validation** preference value is **true**, find numbers must meet the following conditions:

- Find numbers must not be null or zero.
- Each find number must be unique within the same parent.
- Find numbers in the Excel data file must conform to the find number configuration at the importing site.

If the **PS_Find_Number_Validation** preference value is **false**, the import process does not validate find numbers.

The value is set to **false** by default.

Defining Excel import control files manually

Associate a manually defined data file with a control file during import

Caution:

To manually define Excel import control files, you must be familiar with the Teamcenter schema for the data objects and relations to be imported. Siemens Digital Industries Software recommends that control files be manually defined only by Teamcenter administrators.

For each import operation involving a manually defined data file, you must associate a manually defined control file when you initiate the import operation.

- A single control file can be associated with different data files as long as the data files have the same column layout.
- A separate control file must be created for each distinct column layout.
- Control files can be stored in any location to which you can browse when you initiate the import.
- From left to right in the control file, the columns correspond in sequence to the columns in the data file.

> Note:
>
> The control file column sequence does not necessarily match the data file sequence exactly. Some control file columns can mark points where transitions occur in the processing. Because the data file does not contain such transitions, each of these columns sets back the control file sequence by one.

Control file description

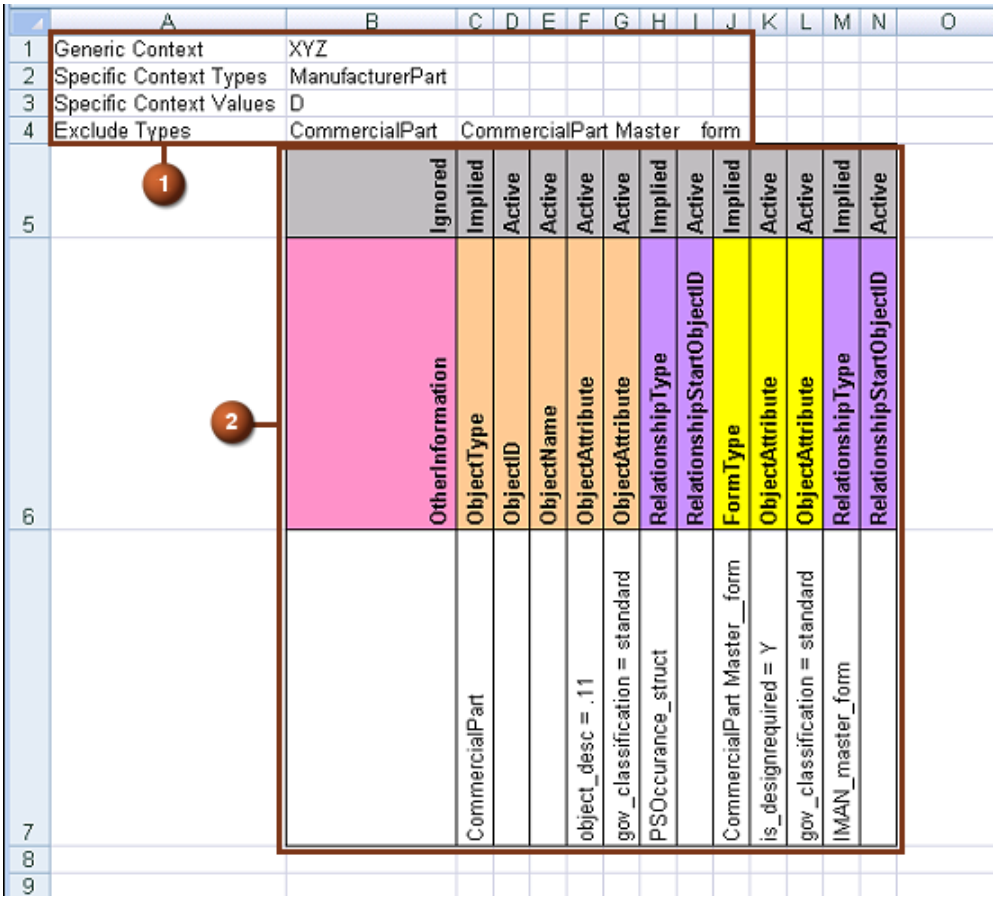
Each control file consists of two sections:

- Context tags and values

  Context tags help you track structure elements that are created and updated by the import process. For example, a tag containing an identifier for the CDM who supplied the data file can show that the imported elements relate to that CDM.

- A rule table

  A *rule table* consists of *keywords* and *keyword tags* and their values. Together, these elements tell the import process how to build the intermediate data structure.



1 Context tags and values

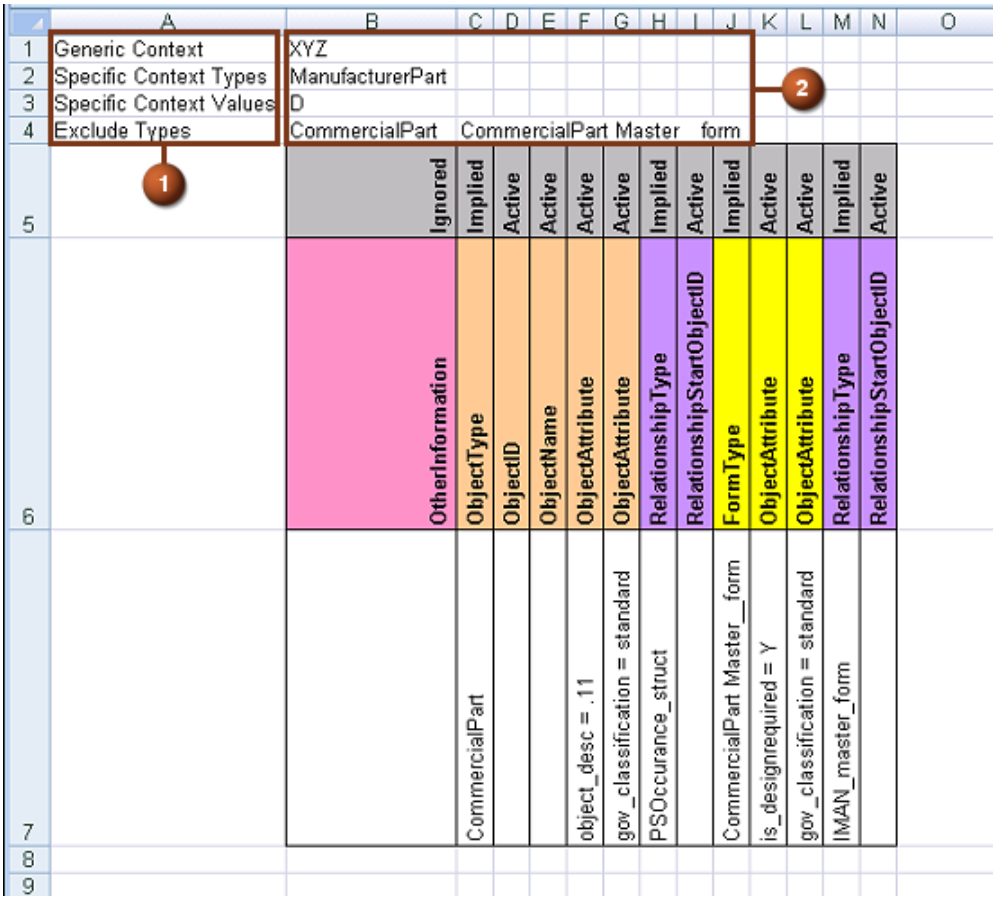
2 Rule table

Mapping context tags to object types

In the first four rows of the control file, you can map context tags to object types that you specify with the **ObjectType** keyword tag in the rule table.

> Note:
>
> Context tags and values do not relate to entries in the first four rows of the data file, for example, boilerplate material.

- You enter the tags in the leftmost column, column **A**, starting in the first row and continuing through the fourth row.
- You enter the tag values starting in column **B** of each row.



1 Context tags

2 Context tag values

Caution:

- You must enter all four context tags, in the order shown in the following table.
- Each context tag in row **1** through **4** must match the capitalization and spelling of the corresponding tag in the table.

Context tag	Value
Generic Context	Prefix that the import process assigns to the object identifiers in the data file. This value is optional.

Context tag	Value
	Note: The prefix is not assigned for object types that are named in a **Specific Context Type(s)** or **Exclude Type(s)** tag value. You must enter only one value, in column **B**. Caution: The import process does not verify whether the value is unique in Teamcenter. For objects whose data you do not want to overwrite, ensure that the prefix is not already assigned to the object identifiers.
Specific Context Type(s)	Object type for which a business rule requires an additional identifier for the object identifier in the data file, for example, a manufacturing part that also requires the vendor identifier. This value is optional. • You can enter multiple values, one in each successive cell of this row. • Each value must match the object type name in Teamcenter. Note: Directly below in the same column, the **Specific Context Value(s)** value completes the context for this object type.
Specific Context Value(s)	Excel heading of the control file column that maps to the data file column containing the additional identifier. • This value is required if the **Specific Context Type(s)** value is entered directly above. • This value must be blank if the **Specific Context Type(s)** value is blank. Caution: The import process does not verify whether the additional identifier is unique in Teamcenter. For objects whose data you do not want to overwrite, ensure that the additional identifier is not already assigned to the object identifiers.

Context tag	Value
	For example, if data file column **O** contains the additional identifier, and if control file column **W** maps to column **O**, enter **W** for this tag value. Caution: The control file column must contain the **Active** keyword in row **5**. You can enter multiple values, one for each cell that has a corresponding object type in the row above. Note: This value overrides the **Generic Context** prefix and identifier for the corresponding object type.
Exclude Type(s)	Object type for which the import process ignores the **Generic Context** prefix but assigns the object identifiers in the data file. This value is optional. • You can enter multiple values, one in each successive cell of this row. • Each value must match the object type name in Teamcenter.

Mapping a rule table

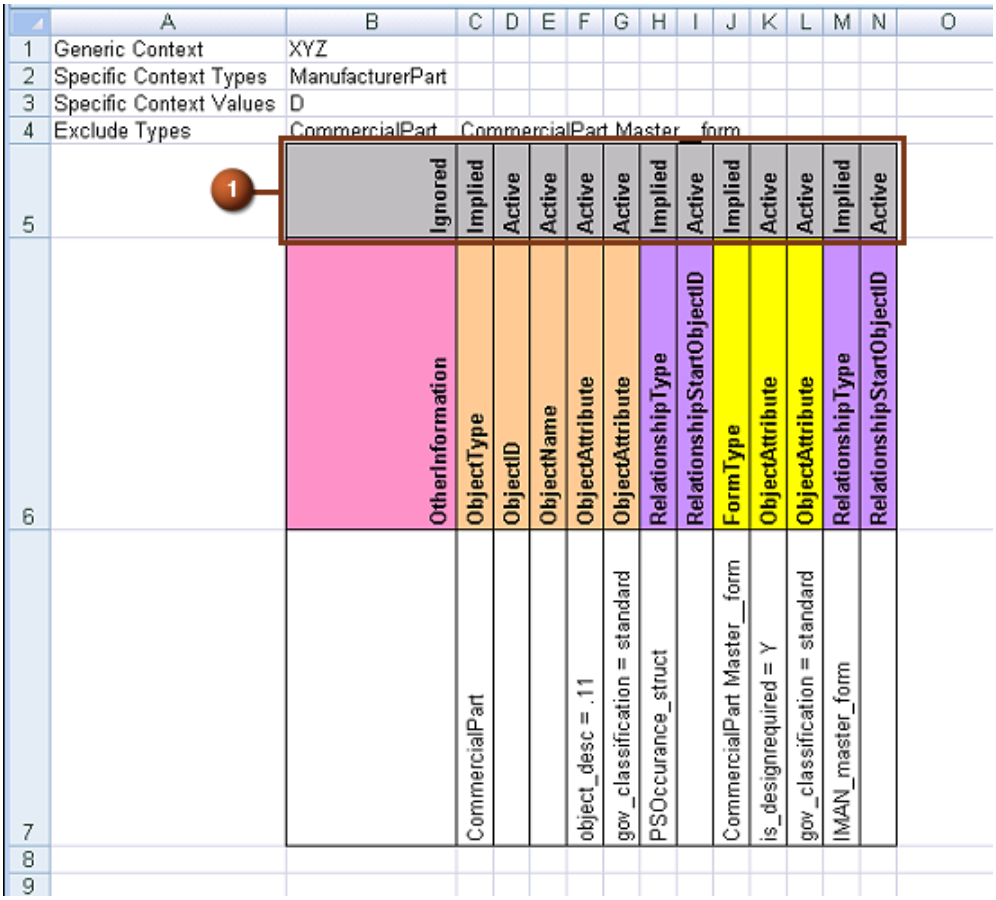
Entering keywords

Rule table keywords specify which data sheet columns are parsed into the intermediate structure.

- Enter keywords in row **5** of the control file, as the topmost elements from left to right in the rule table.
- You can start in column **A** or in any other column.

Caution:

Reserve rows **1** through **4** for context tags, which you map to object types after defining the rule table.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Generic Context	XYZ													
2	Specific Context Types	ManufacturerPart													
3	Specific Context Values	D													
4	Exclude Types	CommercialPart	CommercialPart Master__form												
5		Ignored	Implied	Active	Active	Active	Active	Implied	Active	Implied	Active	Active	Implied	Active	
6		OtherInformation	ObjectType	ObjectID	ObjectName	ObjectAttribute	ObjectAttribute	RelationshipType	RelationshipStartObjectID	FormType	ObjectAttribute	ObjectAttribute	RelationshipType	RelationshipStartObjectID	
7			CommercialPart			object_desc = .11	gov_classification = standard	PSOccurance_struct		CommercialPart Master__form	is_designrequired = Y	gov_classification = standard	IMAN_master_form		
8															
9															

1 Keywords

Caution:

Each keyword in row **5** must match the capitalization and spelling of the corresponding keyword in the following table.

Keyword	Description
Active	Data file contains a corresponding column that is parsed into the intermediate structure. For example, if a data file column contains an object name, and if the corresponding rule table column contains the **Active** keyword above the **ObjectName** keyword tag, the object name is parsed into the intermediate structure.
Ignored	Data file contains a corresponding column, but that column is not parsed into the intermediate structure.

Keyword	Description
	For example, if a data file column contains comments that you do not want to import, you can enter **Ignored** in the related rule table column to bypass the comments.
Implied	Data file does not contain a corresponding column, but this rule table column supplies processing information for building the intermediate structure. For example, if the **Implied** keyword is entered above the **ObjectType** keyword tag, the specified object type is interpreted as a transition between two different objects, marking where the next object starts in the data file.

Each rule table column containing an **Implied** keyword sets back by one the subsequent column mapping sequence.

For example, assume that:

- Data file column **A** contains comments, and rule table column **A** contains the **Ignored** keyword in row **5**.
- Data file column **B** contains object identifiers. The data file does not contain a column for the object type.
- Rule table column **B** contains the **Implied** keyword in row **5**, the **ObjectType** keyword tag in row **6**, and the specific object type in row **7**.

In this example:

- Rule table column **A** maps to data file column **A** because it does not contain the **Implied** keyword.
- With the **Implied** keyword, rule table column **B** provides information that is not contained in the data file.

  Column **B** marks a transition point in processing and does not map to a data file column.
- If the next rule table column, **C**, does not contain the **Implied** keyword, that column maps to data file column **B**.

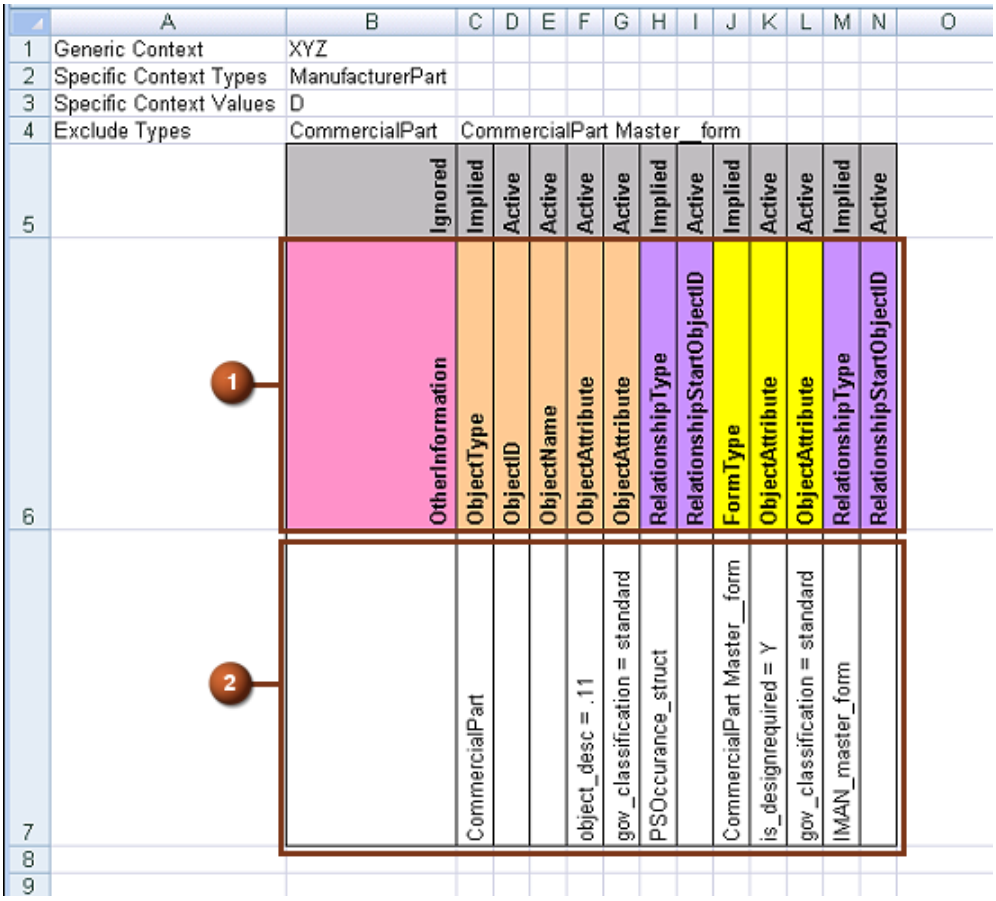
Entering keyword tags and values

Keyword tags associate data sheet columns with Teamcenter schema elements.

- In each cell below a keyword tag, the tag value names the schema element for the corresponding cell in the data sheet.

- The data sheet cell value maps to the intermediate object for that row.
- Enter keyword tags in row **6** of the control file, directly below the keywords from left to right in the rule table.
- Enter the tag values in row **7**, directly below the keyword tag.

> Caution:
>
> Some keyword tag values must be entered in the control file, whereas some values must be left blank because they are supplied by the data file.



1 Keyword tags

2 Keyword tag values

> Caution:
>
> Each keyword tag in row **6** must match the capitalization and spelling of the corresponding keyword tag in the following table.

Keyword tag	Description
ObjectAttribute	Maps the specified property to the imported object. This value is optional. You can enter any number of **ObjectAttribute** tags. **Value:** Exact name of the property in the database.
ObjectID	Maps object identifier from the corresponding data file column to the imported object. This value is required. **Value:** Unique identifier of the imported object. Can identify a new or existing object. Caution: An existing object's properties are overwritten with the values in the data file.
ObjectName	Maps object names from the corresponding data file column to the imported object. This value is required. **Value:** Blank
ObjectType	Maps the specified object type to the imported object. This value is required. **Value:** Exact name of the object type in the database. Note: To enable property editing for the **Item Revision** type and its subtypes, the name must be followed by the suffix **_rev**.
Relationship Attribute	Maps the specified relation object property from the corresponding data file column to the imported object. This value is optional. You can enter any number of **RelationshipAttribute** tags.

Keyword tag	Description
	Value: Exact name of the property in the database.
RelationshipType	Maps the specified relation object type to the imported object. This value is optional. **Value:** Exact name of the relation type in the database. Use the relation type associated with the object type specified in the **ObjectType** tag value. Note: • For the **PSOccurrence** type or a subtype, append the suffix **_struct** to the name. For example: **PSOccurrence_rev** • For master form revisions, append the suffix **_rev** to the name. For example: **IMAN_master_form_rev**
RelationshipStart ObjectID	Maps identifier of the primary object from the corresponding data file column to the imported object. This value is required if you enter a **RelationshipType** tag. **Value:** Blank
RelationshipEnd ObjectID	Maps identifier of the secondary object from the corresponding data file column to the imported object. This value is optional. **Value:** Blank
FormType	Maps the specified form type from the corresponding data file column to the imported object. This value is required if the data applies to the form for an object. **Value:** Exact name of the form type in the database.

Keyword tag	Description
	Note: The form type name must be followed by the suffix **__form** (using two underscore characters). For example: **CommercialPart Master__form**
	You can enter any number of **ObjectAttribute** tags for the form.

Import data from a manually defined file

Note:

You cannot import precise structures, only imprecise ones. The import always updates the latest revision of an item.

1. In the data file:

   a. Click the sheet tab of the data that you want to import.

   b. Click the **Teamcenter** tab on the ribbon to display the Client for Office buttons.

Tip:

To automatically fill in empty cells in object identifier property columns:

a. Click the text portion of the **Import to Teamcenter** button 

.

b. In the list, select **Assign Teamcenter IDs**.

Empty object identifier cells are populated with the next available identifiers from Teamcenter.

Note:

Folder identifier cells cannot be filled in automatically.

2. To start the import process, click **Import to Teamcenter** .

> Note:
>
> If the **Teamcenter Login** page displays, enter your Teamcenter user information, and then click **Login**. If you have questions about logging on to Teamcenter, consult your Teamcenter administrator.

3. Associate the data with the control file, using one of the following dialog boxes:

- The **Select Control File** dialog box displays if you are importing data from this file for the first time since you opened it.

  a. Select the control file and click **Open**.

     You can use the standard browsing features to navigate to the control file.

     The **Select Sheet** dialog box displays.

  b. In the **Select Excel Sheet** list, select the control file sheet that contains the processing instructions for the data.

- The **Associate Control File** dialog box displays if you previously imported data from this file since you opened it.

  a. Do one of the following:

     - Click **Continue** to use the last selected control file, shown in the path name box.
     - Click **Browse** to select another control file by using the standard browsing features.

     The **Select Sheet** dialog box displays.

  b. In the **Select Excel Sheet** list, select the control file sheet that contains the processing instructions for the data.

First, the import process performs checks on the control sheet:

- If invalid instructions are found in the control sheet, the **Control File Validation** dialog box displays.

  a. Click **Cancel** to cancel the import process.

  b. Correct the items listed in the dialog box.

  c. Restart this procedure.

- If the control sheet contains no errors, a message states that control file parsing is complete.

  Click **OK** to close the message and continue the import.

Next, the import process performs checks on the data sheet and displays the **Column Mappings** dialog box. This dialog box shows how the data sheet columns currently correspond to the control sheet columns.

- Each top-level node represents an object type or a relationship type in the control sheet.
- For that object type or relationship type, each second-level node shows the following:
  - First, an Excel column heading in the data sheet.
  - In parentheses, that column's name in the data sheet.
  - To the right of the equal sign (=), one of the following:
    - The corresponding keyword tag in the control sheet.

      The dialog box shows the keyword tag itself if the tag value is blank in the control sheet.
    - The value of the corresponding keyword tag in the control sheet.

      The dialog box shows the tag value if it is entered below the keyword tag in the control sheet.

Caution:

- The import process cannot continue if the dialog box does not display at least one second-level node for a top-level node.
- The process can continue if control columns map to data columns other than the ones that you intend.

a. Review the information in the **Column Mappings Information** dialog box to determine if the mapping is correct.

b. Do one of the following:

   - Click **Correct Data** to stop the import and correct the column mapping.

> Caution:
>
> This action cancels the import. You must restart this procedure after making the corrections.

- Click **Continue** to continue the import using this column mapping.

  The import process checks the data sheet for errors.

  - If errors are found in the data sheet, the **Validation Errors** dialog box displays.

    A. Click **Cancel** to cancel the import process.

    B. Correct the items listed in the dialog box.

    C. Restart this procedure.

  - If the data sheet contains no errors, the Excel Import wizard displays.

4. Follow the instructions on each tab in the wizard.

   On the **Step 1** tab, you can specify the following processing options.

Option	Description
Create new revision	Specifies whether to create new revisions when structure elements are updated, added, or removed. • Select the check box to create new revisions. • Clear the check box to overwrite the latest revisions. Note: The **createNewRevision** custom preference determines the check box default state. • The check box is selected if the preference value is **true**. • The check box is cleared if the preference value is **false** or if the preference is not present on the Teamcenter server.

Option	Description
	If you have questions about preferences, consult your Teamcenter administrator.
Number of validation errors allowed	Specifies the maximum number of data sheet rows that the import process validates. The default value is **50**.
Logging Method	Specifies the granularity of the log file that is generated for each import operation. • Select **Normal** to log only the basic information about exceptions apart from the process feedback from the server. • Select **Verbose** to log details for analyzing processing errors, such as failures and performance bottlenecks. You can view the log file for general information and to research errors.

While the **Step 4** tab indicates progress, the import process can display any or all of the following messages.

Message	Action
`Some objects being imported are currently checked out.`	• Click **Yes** to continue the import without overwriting the objects with data from the import file. • Click **No** to stop the import so that objects can be checked in. In the import file, a list of the checked out objects and their reserving users is inserted on a separate sheet. Note: You must restart the import if you click **No**.
`Some objects being imported were modified in Teamcenter after this sheet was exported.`	• Click **Yes** to continue the import and overwrite the objects with data from the import file. • Click **No** to stop the import and identify the objects.

Message	Action
	In the import file, the objects that were modified in Teamcenter are highlighted in red.
`The text of some requirements was created in MS Word and may have rich content.`	• Click **Yes** to continue the import without overwriting the **Body Text** property values with data from the import file. • Click **No** to stop the import and identify the requirements. In the import file, the requirements with Word content are highlighted in orange.

When the **Step 4** tab states that the import process is complete, you can do the following:

- Click **Save Report** to save the process feedback in a separate file and specify the file location.
- Use the standard Windows functions to copy the process feedback from the text pane to another application, for example, Microsoft Notepad.
- Click the **View Diagnostic Logs** link to open the log file specified by the **Logging Method** option.

5. Click **Finish** to close the wizard and apply the data in Teamcenter.

> Tip:
>
> To work with a new imported structure in the Teamcenter rich client:
>
> a. Use the search feature to find the structure's peak element.
>
> b. In the search results, select the peak element and send it to the desired rich client application.

8. Setting up templates to export data from Teamcenter to Microsoft Office

Introduction to Microsoft Office templates

Export templates define what information is exported from the Teamcenter database and how the information is formatted when exported to a Microsoft Word document or Microsoft Excel spreadsheet. Exported data can be sent to a static document or live document. Exported data can also be sent to a static Excel spreadsheet or live Excel document. Data sent to a live document can be edited and saved to the Teamcenter database in real time, in most, but not all cases.

You can configure the following types of export templates:

- **Specification export templates**

  Microsoft Word files control document format when you export a Teamcenter structure from Systems Engineering.

  > Note:
  >
  > **Closure rules** can be used to control exported content with use of **transfer mode objects**.

- **Object export templates**

  Microsoft Word files that control the content and property format, and the specific properties which are exported, when you export Teamcenter objects.

- **Excel export templates**

  Microsoft Excel files that define the property data and the output format that is used when you export objects from Teamcenter to Excel. Property data can be exported to live Excel or to a static Excel spreadsheet.

> Note:
>
> Microsoft Word is required to generate specification and object export templates and work with live Word. Microsoft Excel is required to use live Excel.

For supported versions of Microsoft Word and Excel, see the Hardware and Software Certifications knowledge base article on Support Center.

Exporting object data to Microsoft Word

When you export object data to Word, Teamcenter uses a combination of two Word templates to define which properties are exported and how the data is formatted in the output document.

Specifically, to export object data to Word, you need:

- Data

  The data exported to Microsoft Word is assigned one or more object templates and a specification template for export. For each object type in the data, you can use a different object template or you can use the same object template for multiple object types.

- Object template(s)

  An object template contains keyword tags that designate the properties that are exported for the selected objects.

  > Caution:
  >
  > Properties with dynamic or cascading lists of values (LOVs) cannot be modified with live Word.

  Style tags are applied to the **Property** keyword to specify which style definition should be used from the specification template. Using a single object template specifies that the same object properties for each object gets exported to Microsoft Word. Using multiple object templates allow you to specify the object properties that are exported for the assigned object type.

- Specification template

  A specification template controls the output document format, such as title page, headers and footers, and page layout. It also defines the Word styles that are applied to the exported data.

  The style definitions match the style tags assigned to the **Property** keyword tags.

> Note:
>
> Only properties of the object revision are supported for Word Export.

By maintaining the same style set in both the object template and the specification template, you can apply identical styles to object content and to Word output documents. Use the Word **Organizer** dialog box to manage the style set. Ensure that the style names and attributes match in both templates.

- Styles in the specification template override those with the same name but different attributes in the object template.

- Body text attributes in the output document originate from the content formatting for the individual objects.

  Styles applied to **{%Body_Text}** tags in the object template do not affect object content in the output document.

  > Note:
  >
  > Teamcenter does not set indentation for headings in the output document. To indent exported object names as in the structure hierarchy, modify the corresponding Word heading styles in the template.

> Warning:
>
> Modify Word styles only in the templates. Style modification through Word's associated XML files can corrupt the templates and their output documents.

Working with specification templates

What is a specification template?

> Note:
>
> Microsoft Office Word is required for creating and modifying specification templates.

A *specification template* is a Microsoft Office Word document that controls the content and format of the data that is exported to Word.

- Defines the overall layout and content of the Word output document.

  > Caution:
  >
  > All live Word documents use portrait orientation. Live Word does not recognize landscape orientation.

- Can be associated with specific business object types (such as **SpecElementRevision** objects, which are requirement object types) by an item revision definition configuration (IRDC).

- Can be selected manually when you export data.

- Can be viewed and edited in Word or in the Teamcenter embedded viewer.

Caution:

In output documents based on the default specification template, Word's **Normal** style can overwrite styles that are applied to object content. To work around this condition, do the following in the default specification template:

1. Add a dummy paragraph, and then apply the overwritten style to the dummy.
2. Save the changes.
3. Delete the dummy paragraph, and then save and close the specification template.

Spec Template items and **Spec Template Revision** item revisions contain datasets of the **FullText** subclass. **Full Text** datasets represent Word content by named references to XML files.

Business object types supported by specification templates

Specification templates support the following business object types:

- **Specification** item objects and subtypes, such as **RequirementSpec** objects.
- **SpecElement** item objects and subtypes, such as **Requirement** and **Paragraph** objects.
- **Full Text** datasets and subtypes, such as **Note** datasets.
- **Trace link** objects and subtypes.

Specification template layout

Specification template contents

Caution:

Landscape orientation in specification templates is ignored by the **Live integration with Word** output mode. The page layout is portrait orientation in all live Word documents.

A specification template contains the following elements of Microsoft Office Word:

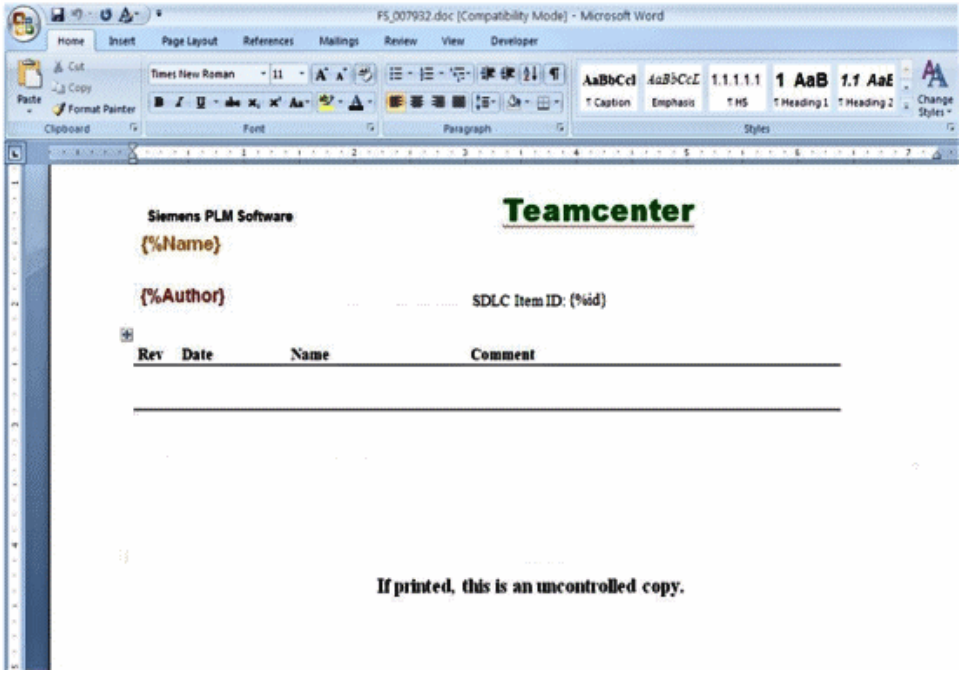
- Style and format definition
- Title page
- Table of contents
- Header and footer information

- Object export conditions
- Object export structure

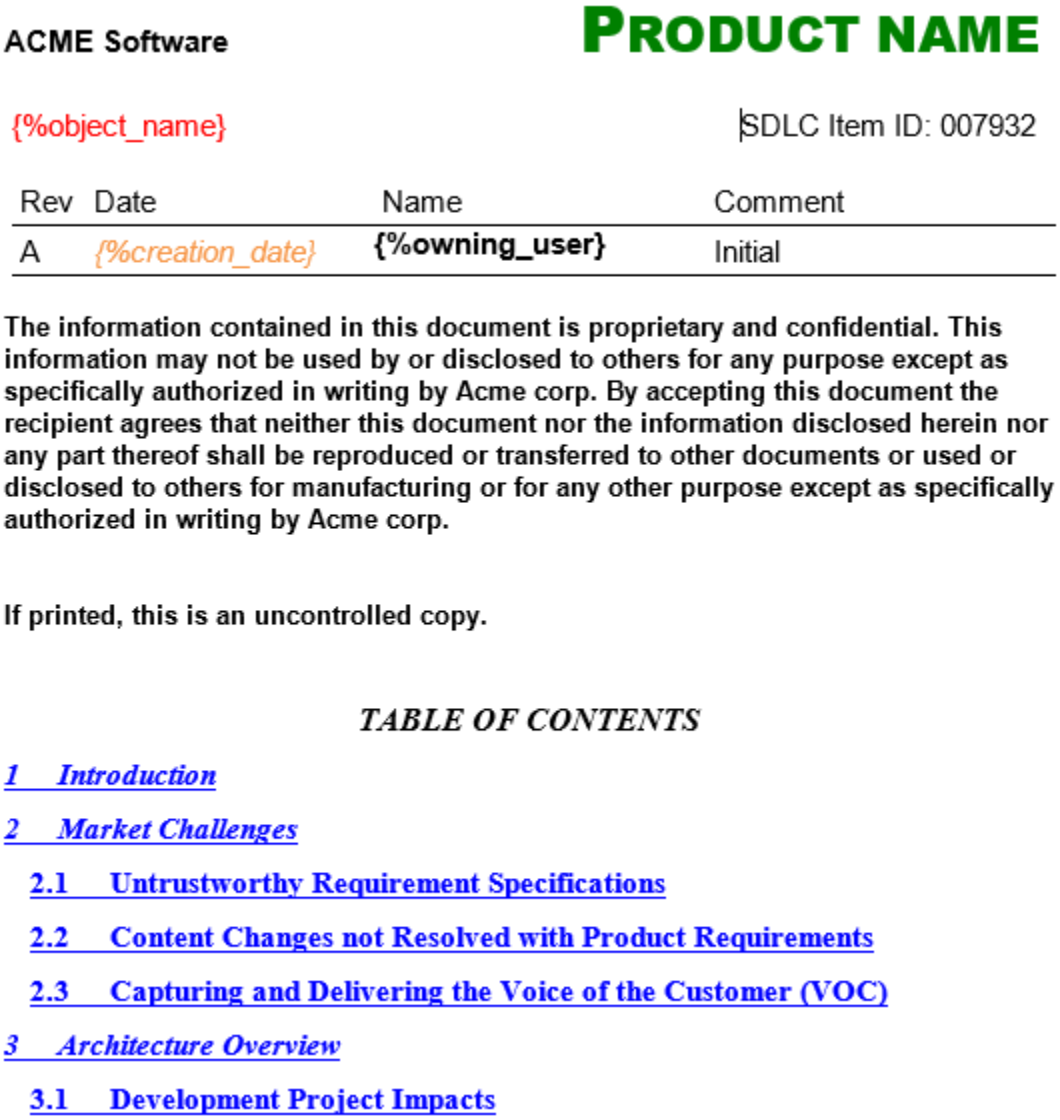
Specification template title page

The title page of a specification template can be a combination of static and dynamic text. Title pages often include:

- Document title derived from a Teamcenter property value identified by a **keyword**.
- Copyright and trademark information.
- Author, project ID, and revision history extracted from the Teamcenter database using a **keyword**.



Here is an example of a specification template.



ACME Software

PRODUCT NAME

{%object_name}

SDLC Item ID: 007932

Rev	Date	Name	Comment
A	{%creation_date}	{%owning_user}	Initial

The information contained in this document is proprietary and confidential. This information may not be used by or disclosed to others for any purpose except as specifically authorized in writing by Acme corp. By accepting this document the recipient agrees that neither this document nor the information disclosed herein nor any part thereof shall be reproduced or transferred to other documents or used or disclosed to others for manufacturing or for any other purpose except as specifically authorized in writing by Acme corp.

If printed, this is an uncontrolled copy.

TABLE OF CONTENTS

Specification template table of contents

Tables of contents are added to the specification template using the **Table of Contents** options on the **References** tab in Microsoft Word.

You can specify the number of heading levels and choose a format for the table of contents.

For more information, see *Microsoft Word Help*.

Specification template headers and footers

Headers and footers can be defined for each section of the specification template. You can define:

- First page headers and footers.

- Odd page headers and footers.
- Even page headers and footers.

Headers and footers are added to the Word document using the **Header** and **Footer** options in the **Header & Footer** group on the **Insert** tab in Microsoft Word.

You can also add footnotes and endnotes to the specification template.

For more information, see *Microsoft Word Help*.

> Note:
>
> Siemens Digital Industries Software recommends that headers and footers not be used in object templates.

Specification template styles

Specification templates use standard Microsoft Word style tools. You can define:

- Paragraph styles.
- Character styles.
- Link styles.
- Numbering styles.
- Table styles.
- Default document paragraph and character properties.
- Style inheritance.
- Style application.
- Latent styles.

> Caution:
>
> When exporting documents based on the default specification template, Word's **Normal** style can overwrite styles that are applied to object content. To work around this condition, do the following in the default specification template:

1. Add a dummy paragraph, and then apply the overwritten style to the dummy.
2. Save the changes.

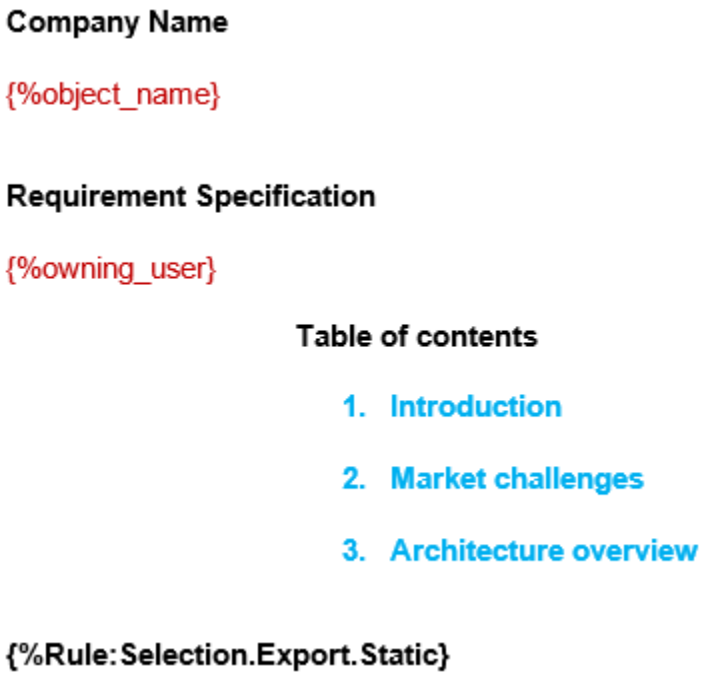
   This action writes the style to Word's **styles.xml** file.
3. Delete the dummy paragraph, and then save and close the specification template.

Specification template body contents

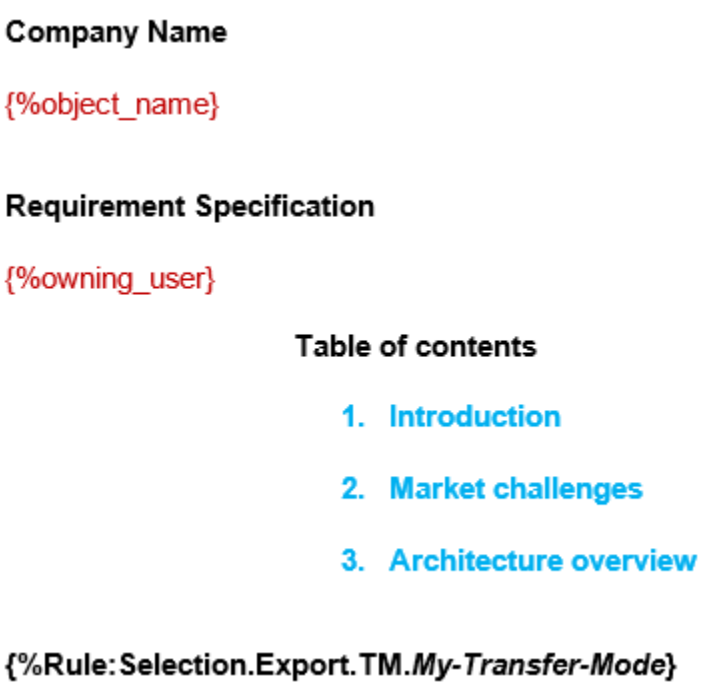
The body content of the specification template contains free-form text and information extracted from the Teamcenter database using keywords. A **keyword syntax** must be used. When following **body content rules**, the body content can reflect:

- A static structure exported from the Systems Engineering application.
- A dynamic structure generated using transfer modes and conditions in transfer modes.

Example: Static structure in a specification template



Example: Dynamic structure in a specification template



Create a specification template

1. In your **Home** view, choose **Tools→Microsoft Office Templates→Create Specification Template**.

2. Select **Spec Template** from the list of item types, and then click **Next**.

3. Type an item ID and revision ID in the **Item ID** and **Revision ID** boxes or click **Assign** to automatically assign the IDs.

4. Type a name for the specification template in the **Name** box.

   The name must be unique; duplicates are not allowed. It is recommended to use the same name for specification template revisions, as the names of the specification templates appear in the **Export to Word** dialog box.

5. (Optional) Type a description in the **Description** box.

6. Click **Finish**.

   The **Spec Template** item, **Spec Template Revision** item revision, and the associated **Full Text** dataset are created in the **SpecTemplates** folder, which is located in the **RequirementsManagement Templates** folder in your **Home** folder.

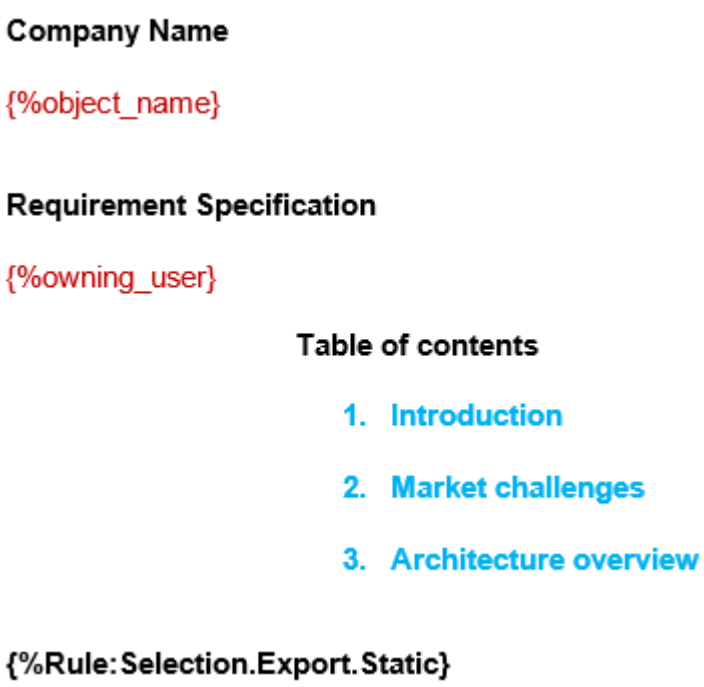
7. (Optional) To update the specification template content, double-click the **Full Text** dataset.

The template opens in Microsoft Office Word.

8. (Optional) Modify the content as required.

You can **specify a transfer mode in the specification template to collect additional objects for export**.

Example: Static structure in a specification template



Example: Dynamic structure in a specification template

Company Name

{%object_name}

Requirement Specification

{%owning_user}

Table of contents

1. Introduction
2. Market challenges
3. Architecture overview

{%Rule:Selection.Export.TM.*My-Transfer-Mode*}

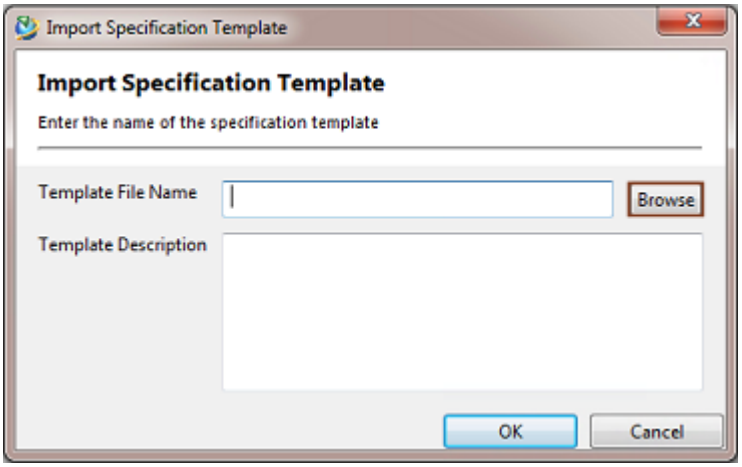
9. Save the document, and close Microsoft Office Word.

Import a specification template

> Note:
>
> When exporting a Requirement Management template to PLMXML, use the **ConfiguredRequirementDataExport** mode, and when importing a Requirement Management template from PLMXML, use the **REQ_IMPORT_TEMPLATE** mode.

1. In the **Home** view, choose **Tools→Import→Templates→Specification Template**.

2. In the **Import Specification Template** dialog box, click **Browse** and locate the Word file (**.docx**) containing the template you want to import.



3. Select the file and click **Save**.

4. (Optional) Type a description in the **Template Description** box.

5. Click **OK**.

   The template is imported in to the Teamcenter database.

6. Click **OK** in the **Import Template** dialog box.

   The **Spec Template** item, **Spec Template Revision** item revision, and the associated **Full Text** dataset are created in the **SpecTemplates** folder, which is located in the **RequirementsManagement Templates** folder in your **Home** folder.

Modify a specification template

1. In the **Home** view, select the **Full Text** dataset associated with the **Spec Template Revision** item revision out of the database.

2. Double-click the **Full Text** dataset to open it in Microsoft Office Word.

3. Edit the content of the Word document.

4. Save the file in Word.

5. Close the file and check it in to the Teamcenter database.

Create a specification template by copying an existing one

1. In the **Home** view, select the **Spec Template** item containing the template that you want to duplicate, and choose **File→Save As**.

2. Click **Assign** in the **Item Details** pane of the **Save Item As** dialog box.

   The system assigns an item ID and revision ID for the new item.

3. Type a name for the new item in the **Name** box.

   You can type an optional description in the **Description** box.

4. (Optional) Select the **Open on Create** check box to open the item after it is created.

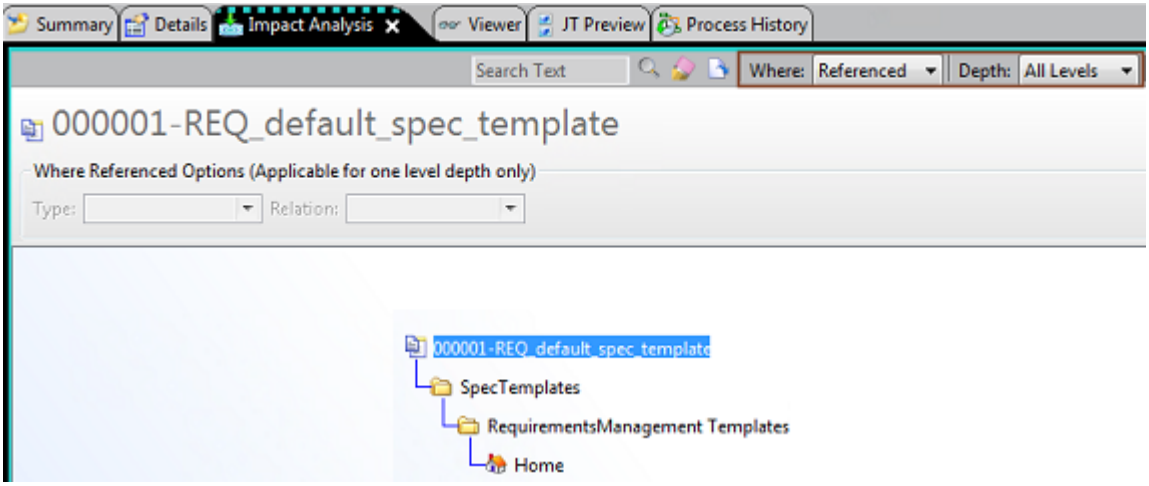
5. Click **OK**.

   The new **Spec Template** item and **Spec Template Revision** are created. The **Full Text** dataset is copied as a new object of the same name as the original dataset.

6. (Optional) Rename the **Full Text** dataset.

Delete a specification template

1. Ensure that the **Spec Template** item is not referenced by other objects in the Teamcenter database by performing a where-referenced search.



2. In the **Home** view, select the **Spec Template** item.

3. Choose **Edit→Delete**.

4. Click **Yes** in the **Delete** dialog box.

   The **Spec Template** item, **Spec Template Revision** item revision, and **Full Text** dataset are deleted from the database.

Working with object templates

What is an object template?

Note:

Microsoft Office Word is required for creating and modifying object templates.

Object templates control the content and format when you export Teamcenter objects to Word. Object templates are Word documents that:

- Use *property keywords* to define which properties of the selected objects are exported.

  Caution:

  - Do not use property keywords in the Word content of requirement, paragraph, or block objects. Keywords in object content (for example, **{%body_text}**) cause an infinite loop in the export process, and error messages are not displayed.
  - In an object template that is intended for export to the **Live integration with Word** output mode, the **{%object_name}** property keyword is required. Otherwise, structure data cannot be edited in live Word.
  - Properties with dynamic or cascading lists of values (LOVs) cannot be modified with live Word.

- Specify the format of the exported property information.

  For example, you can specify that the properties are exported in tabular format.

- Specify static text to be included in the export document.
- Can specify *content rules* that determine which objects are exported to the output document.
- Can be associated with specific business objects types using item revision definition configuration (IRDC).
- Can be selected manually when you export data.
- Can be viewed and edited in Word or in the Teamcenter embedded viewer.

Caution:

- Modify Word styles only in the template. Style modification through Word's associated XML files can corrupt the template and its output documents.

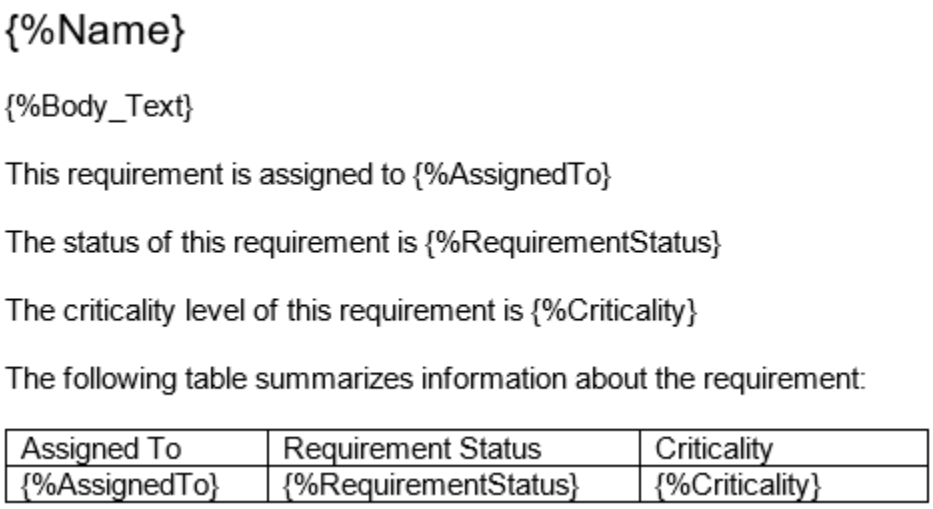
- In export documents based on the default object template, Word's **Normal** style can overwrite styles that are applied to object content. To work around this condition, do the following in the default object template:

  1. Add a dummy paragraph, and then apply the overwritten style to the dummy.
  2. Save the changes.
  3. Delete the dummy paragraph, and then save and close the specification template.

Object Template items and **Object Template Revision** item revisions contain datasets of the **FullText** subclass. **Full Text** datasets represent Word content by named references to XML files.

Example: Object template without rules

Keywords must follow a keyword syntax and the exported body content depends on body content rules.



{%Name}

{%Body_Text}

This requirement is assigned to {%AssignedTo}

The status of this requirement is {%RequirementStatus}

The criticality level of this requirement is {%Criticality}

The following table summarizes information about the requirement:

Assigned To	Requirement Status	Criticality
{%AssignedTo}	{%RequirementStatus}	{%Criticality}

Example: Object template with rules

{%object_name}

Property	Value
Type	{%object_type}
ObjectDescription	{%object_desc}
Last_mod_date	{%last_mod_date}

{%body_text}

{%RULE:SELECTION.EXPORT.TM.My_Transfer_Mode}

Warning:

The **{%object_name}** property keyword is required in object templates that are intended for export to the **Live integration with Word** output mode. Without this property keyword in the template, such a structure cannot be edited in live Word.

Create an object template

1. In your **Home** view, choose **Tools→Microsoft Office Templates→Create Object Template**.
2. Select **Object Template** from the list of item types, and then click **Next**.
3. Type an item ID and revision ID in the **Item ID** and **Revision ID** boxes, or click **Assign** to automatically assign the IDs.
4. Type a name for the object template in the **Name** box.

   You can type an optional description in the **Description** box.
5. Click **Finish**.

   The **Object Template** item, **Object Template Revision** item revision, and associated **Full Text** dataset are created in the **ObjectTemplates** folder, which is located in the **RequirementsManagement Templates** folder in your **Home** folder.
6. (Optional) To update the object template content, double-click the **Full Text** dataset.

   The template opens in Word.
7. (Optional) Modify the content, save the Word document, and close Word.

Caution:

If the object template is intended for export to the **Live integration with Word** output mode, the **{%object_name}** property keyword is required. Otherwise, structure data cannot be edited in live Word.

Import an object template

Note:

When exporting a Requirement Management template to PLMXML, use the **ConfiguredRequirementDataExport** mode, and when importing a Requirement Management template from PLMXML, use the **REQ_IMPORT_TEMPLATE** mode.

1. In the **Home** view, choose **Tools→Import→Templates→Object Template**.
2. In the **Import Object Template** dialog box, click **Browse** and locate the Word file (**.docx**) containing the template you want to import.
3. Select the file and click **Save**.
4. (Optional) Type a description in the **Template Description** box.
5. Click **OK**.

   The template is imported in to the Teamcenter database.
6. Click **OK** in the **Import Template** dialog box.

   The **Object Template** item, **Object Template Revision** item revision, and associated **Full Text** dataset are created in the **ObjectTemplates** folder, which is located in the **RequirementsManagement Templates** folder in your **Home** folder.

Modify an object template

1. In the **Home** view, check the **Full Text** dataset associated with the **Object Template Revision** item revision out of the database.
2. Double-click the **Full Text** dataset to open it in Microsoft Word.
3. Edit the content of the Word document.

Caution:

- If the object template is intended for export to the **Live integration with Word** output mode, the **{%object_name}** property keyword is required. Otherwise, structure data cannot be edited in live Word.
- Properties with dynamic or cascading lists of values (LOVs) cannot be modified with live Word.

4. Save the file in Word.
5. Close the file and check it in to the Teamcenter database.

Create an object template by copying an existing one

1. In the **Home** view, select the **Object Template** item containing the template that you want to duplicate, and choose **File→Save As**.
2. Click **Assign** in the **Item Details** pane of the **Save Item As** dialog box.

   The system assigns an item ID and revision ID for the new item.
3. Type a name for the new item in the **Name** box.
4. (Optional) Type a description of the new template item in the **Description** box.
5. (Optional) Select the **Open on Create** check box to open the item after it is created.
6. Click **OK**.

   The new **Object Template** item and **Object Template Revision** are created. The **Full Text** dataset is copied as a new object of the same name as the original dataset.
7. (Optional) Rename the **Full Text** dataset.

Delete an object template

1. Ensure that the **Object Template** item is not referenced by other objects in the Teamcenter database by performing a where-referenced search.
2. In the **Home** view, select the **Object Template** item.
3. Choose **Edit→Delete**.
4. Click **Yes** in the **Delete** dialog box.

The **Object Template** item, **Object Template Revision** item revision, and **Full Text** dataset are deleted from the database.

Content rule for specification templates and object templates

You can specify a rule in the specification export template documents and object templates. This rule determines which objects are exported.

Rule	Purpose	Rule format
Export dynamic structure using transfer mode	Processes the specified transfer mode to obtain a list of secondary linked objects when a selected object is processed for export. Secondary objects are exported as children of the selected object.	**{%Rule:Selection.Export.TM.***My-Transfer-Mode***}**

You can add text of the property of a referenced object with rule with pattern as follows:

{%Rule:Selection.Export.TM.*My-Transfer-Mode*.*<property_of_found_item>***}**

If more related objects are found, the line with the rule is repeated completely for every found object.

Example:

If the template contains a line such as "Prefix {%Rule:Selection.export.TM.Req_default_TM.item_id} tail.", the output for an object with two related objects is two lines:

"Prefix 0001234 tail"

"Prefix 0000345 tail"

This duplication occurs also in case such a phrase is in a table cell.

Syntax rules for content rules in specification templates and object templates:

- Do not include spaces.
- Multiple content rules are allowed in a template.
- Only the transfer mode name is case sensitive; the other parts of the rule are not case sensitive.
- Ensure that you spell the transfer mode name and the other parts of the rule correctly.
- The transfer mode name must exist.

- Text following the rule on the same line is not exported.

In case you specify multiple rules:

- Ensure that you specify each rule on a separate line.
- If the content rules are created in a table, then only one content rule is allowed per cell.
- If a rule is entered twice in the same template, it is executed only once. The second occurrence is ignored.
- If a rule is specified in the specification template and in the object template, it is executed twice.

Property name syntax in specification templates and object templates

Caution:

- Do not use property names in the Word content of requirement, paragraph, or block objects. Property names in object content (for example, **{%body_text}**) cause an infinite loop in the export process, and error messages are not displayed.

  You can find property names in the Business Modeler IDE.

- Properties with dynamic or cascading lists of values (LOVs) cannot be modified with live Word.

The following rules apply when property names are used in export templates:

- The property being specified must be enclosed by braces (**{ }**).
- The property name must be preceded by a percent sign (**%**).
- The property name must be the real property name specified in Business Modeler IDE.

Example

To include the **object name** property in a template, the format is:

{%object_name}

Caution:

- The **{%object_name}** property name is required in object templates that are intended for export to the **Live integration with Word** output mode. Without this property name in the template, such a structure cannot be edited in live Word.

- Styles applied to the **{%body_text}** property name in the object template do not affect the output in the export document. To specify formatting in the export document, ensure that the specification template contains the same styles that are applied to the Word content of the object itself.

Using macros in Microsoft Office Word templates

When working with Microsoft Office Word templates in Teamcenter, keep in mind the following:

- Teamcenter supports user-defined macros only for **Spec Templates**.
- You can add macros in **Spec Template** and save the template.
- Teamcenter preserves the macro only for *static* export to Word. If you export the macro using the Word live option, Teamcenter does not preserve the macro.
- Object templates do not support the macro capability. If you add any user-defined macro in the object template, it is not preserved in the output.
- Teamcenter does not support ActiveX controls for **Spec Templates** and **Object Templates**.
- Teamcenter does not support event level macros in the Teamcenter viewer pane. If a file needs to be viewed in Teamcenter, the event level macros should be renamed to nonevent macros to allow the storage and manual launch when the workbook is opened in Excel.

Associating export templates with business objects

Object export templates and specification export templates can be associated with an item revision using an item revision definition configuration (IRDC) object.

IRDCs standardize item revision behavior at specific times in the life cycle, such as at item creation, checkin, checkout, save as, and revise.

Create an IRDC for every item revision business object for which you want standardized behavior.

Microsoft Office Excel export templates

What are Microsoft Office Excel export templates?

Excel templates define the property data and output format used when you export objects from Teamcenter to Excel. Property data can be exported to live Excel or to a static Excel spreadsheet.

Note:

Live Excel provides direct connectivity to the Teamcenter server, which allows you to modify the properties of Teamcenter objects without installing the rich client. At this time, direct connectivity is only supported for objects exported from My Teamcenter using an Excel export template. Direct connectivity for objects exported from structure editors is not yet supported.

Microsoft Excel is required to use live Excel.

Excel Template items and **Excel Template Revision** item revisions are Teamcenter data objects that act as containers for the **MS ExcelX** dataset. The named reference of the **MS ExcelX** dataset is an Excel (**.xlsm** or **.xlsx**) file.

Excel export templates:

- Contain a rule table that specify rules that control which objects are exported.
- Define the overall layout of the export spreadsheet, including header, footer, and style.
- Can be selected manually when you export data.
- Can be viewed and edited in Excel.

Note:

An item revision definition configuration (IRDC) defines how an item revision is handled. IRDCs standardize item revision behavior at specific times in the life cycle, such as at item creation, checkin, checkout, save as, and revise. Create an IRDC for every item revision business object for which you want standardized behavior.

Working with multiple sheets in Excel export templates

Excel export templates support multiple-sheet workbooks:

- You can create multiple sheets in the template.
- Each sheet needs its own **<start>** tag, **<end>** tag, and rule table.
- Each sheet can specify different objects, rules, and formatting for export to Excel.

Cell formatting in Excel export templates

You can use standard Excel tools to apply cell formatting and styles, such as color, font, and shading, to the export template. The format and styles are applied to the property data in the export spreadsheet. In addition to applying formatting and styles to the Excel template, you can also add static text, such as column headers and titles.

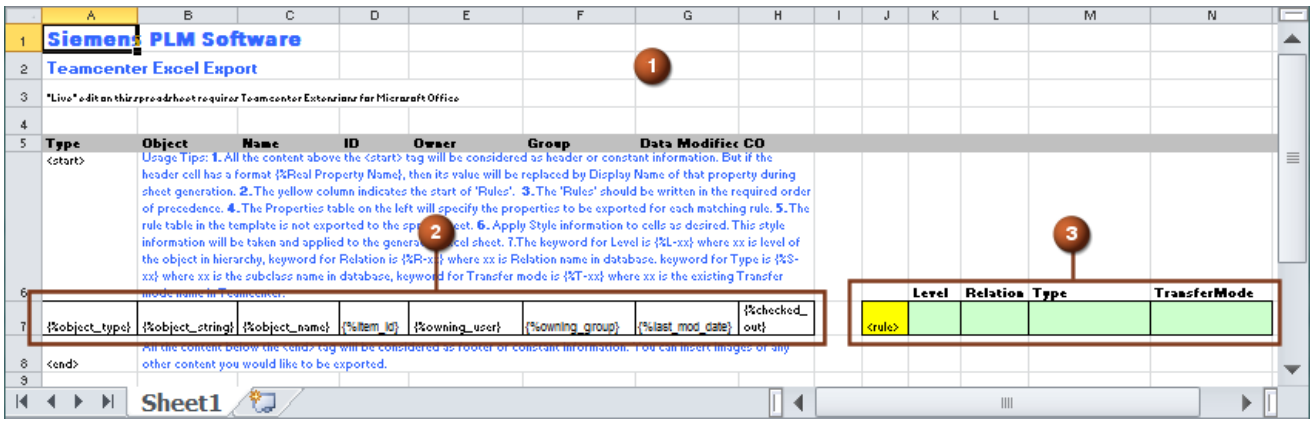
Note:

Siemens Digital Industries Software does not recommend applying formatting, conditional or otherwise, to empty cells.

Working with Excel template components

Excel template components

An Excel template consists of a *property table* and a corresponding *rule table*.



1	**REQ_default_excel_template.xlsm**	Located in the *TC-ROOT*/**install/data** directory.
2	Property table	Contains tags that call the properties whose values are exported to the output files that are based on the template.
3	Rule table	Contains key fields that define rules for filtering the properties that are tagged in the same row. The key fields are **Level**, **Relation**, **Type**, and **TransferMode**. They are not exported to the output files.

Caution:

Properties with dynamic and cascading lists of values (LOVs) cannot be modified with live Excel.

In addition to properties and rules, an Excel template controls the overall appearance of its output files. You can use the related Excel features to set up the template's page layout, including the header and footer, and to apply cell formatting and styles, such as color, font, and shading.

The layout, formatting, and styles are applied to the exported data in the output files. You can also add static text, such as corporate boilerplate and column headings, to appear in the output files.

Note:

Siemens Digital Industries Software does not recommend applying formatting, conditional or otherwise, to empty cells.

General guidelines for Excel templates

- Users must have at least **Read** access to all Excel export templates for their projects. A user without such access to one template cannot select any template in the **Export to Excel** dialog box.
- Excel export templates support multiple-sheet workbooks. Each sheet can specify different objects, rules, and formatting for export to Excel.

  Each sheet needs its own **<start>** tag, **<end>** tag, and rule table.
- User-defined macros are preserved both in static and live Microsoft Office Excel sheets, but not in workbooks.
- Excel templates do not support ActiveX components, OLE embeddings, charts, formulas, and so on. They do, however, support free text and images in the header and footer.
- Out-of-the-box templates have Teamcenter macros. Be sure that when you edit the macro to add custom code, you do not remove the Teamcenter macros. Removing the macros can cause issues.
- The template must contain only one **<start>** tag and only one **<end>** tag. The **<start>** tag must appear to the left of the **<rule>** column.
- All rows containing data tags must be located below the **<start>** tag and above the **<end>** tag. Cell content outside the area delimited by these tags is static text.
- All property columns must be located to the left of the rule table. Any cell content to the right of the rule table is not exported to the spreadsheet.

  Caution:

  Properties with dynamic and cascading lists of values (LOVs) cannot be modified with live Excel.
- A cell containing a **<rule>** tag must precede the headings (**Level**, **Relation**, **Type**, and **Transfer Mode**) in the rule table.
- The sequence of the **Level**, **Relation**, **Type**, and **Transfer Mode** columns in the rule table must not be changed.
- Additional columns cannot be inserted into the rule table.

Defining rules for filtering properties

The following key fields can be specified in the rule table and are applied to the properties being exported:

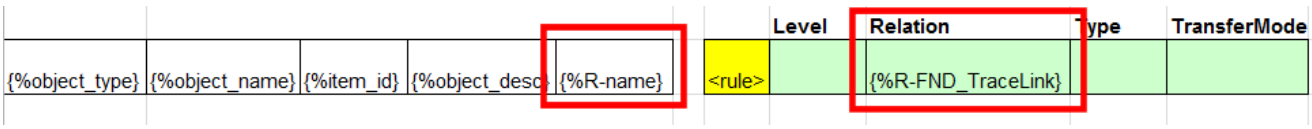
Key field	Syntax	Example	Description
Level	**{%L-***Level***}**	**{%L-2}**	Filters data based on the hierarchical level of the object. For example, when **2** is the argument, objects at the second level in the structure are exported to the Excel spreadsheet.
Relation	**{%R-***Relation-Name***}**	**{%R-FND_TraceLink}**	Filters data based on the relation traversed to the object found.
Type	**{%S-***Type-Name***}**	**{%S-Requirement}**	Filters data based on workspace object type.
Transfer Mode	**{%T-***Transfer-Mode-Name***}**	**{%T-Req_default_TM}**	If a transfer mode is found that matches the one specified in the template, the secondary objects associated with the object are also exported.

Example:

For a **Relation** key field, assume the following:

- The row contains property tags for a primary object.
- One of the property tags is **{%R-name}**, which calls the names of all relation objects for the primary object.
- You want to export only the names of the object's trace links.

When you enter the **{%R-FND_TraceLink}** rule tag in the **Relation** column, the export process filters the relations and passes only trace link names to the output file.



Tagging properties for export to Excel

Properties are specified for export by tags to the left of the rule table. These properties are filtered by the rules defined in the rule table.

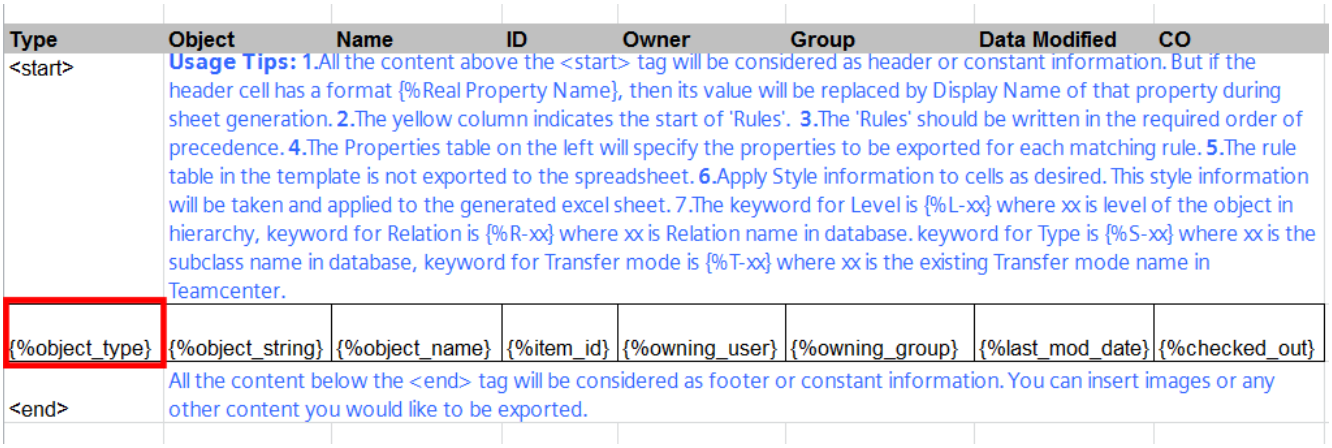
Property tags have the following syntax:

{%*property_name***}**

Replace **property_name** with the internal name of the property you want to export, for example, **{%object_type}**.

Caution:

- Use the property's internal name from the Business Modeler IDE, not the display name in the rich client.
- Properties with dynamic and cascading lists of values (LOVs) cannot be modified with live Excel.



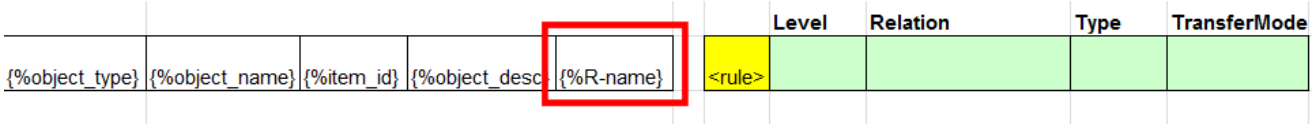
Type	Object	Name	ID	Owner	Group	Data Modified	CO
<start>	**Usage Tips: 1.**All the content above the <start> tag will be considered as header or constant information. But if the header cell has a format {%Real Property Name}, then its value will be replaced by Display Name of that property during sheet generation. **2.**The yellow column indicates the start of 'Rules'. **3.**The 'Rules' should be written in the required order of precedence. **4.**The Properties table on the left will specify the properties to be exported for each matching rule. **5.**The rule table in the template is not exported to the spreadsheet. **6.**Apply Style information to cells as desired. This style information will be taken and applied to the generated excel sheet. 7.The keyword for Level is {%L-xx} where xx is level of the object in hierarchy, keyword for Relation is {%R-xx} where xx is Relation name in database. keyword for Type is {%S-xx} where xx is the subclass name in database, keyword for Transfer mode is {%T-xx} where xx is the existing Transfer mode name in Teamcenter.							
{%object_type}	{%object_string}	{%object_name}	{%item_id}	{%owning_user}	{%owning_group}	{%last_mod_date}	{%checked_out}
<end>	All the content below the <end> tag will be considered as footer or constant information. You can insert images or any other content you would like to be exported.						

Tip:

To export properties of relations, use the prefix **R-** in the property tags.

{%R-*relation_property***}**

If the property tag is **{%R-name}**, for example, the names of the object's relations are exported to the same row in the output file.



						Level	Relation	Type	TransferMode
{%object_type}	{%object_name}	{%item_id}	{%object_desc	{%R-name}	<rule>				

Using transfer modes in Excel templates

Transfer modes combine rules and property sets to define what additional data is exported when an object is selected for export.

- You can specify transfer modes in the Excel template rule table.
- If the object being exported has an associated transfer mode that matches the transfer mode specified in the template, the secondary objects indicated in the transfer mode are processed.
- The secondary objects are processed through the rule table and objects that match the rules are exported.

The following syntax is used to specify transfer modes in the template:

{%T-*Transfer-Mode-Name***}**

Administrators create transfer modes by using the PLM XML/TC XML Export Import Administration application. You can specify a transfer mode for each rule in a Microsoft Excel rule table.

Syntax rules to be followed when specifying transfer modes in an Excel template:

1. Do not include any spaces when specifying a transfer mode.
2. In the syntax, only the *Transfer-Mode-Name* is case sensitive.

Rule table processing

Rules in Excel template rule tables are processed as follows:

1. From top to bottom, beginning with the topmost rule in the table.
2. If an object is found that matches all criteria in the current rule, the data is extracted to the spreadsheet. Rule processing ends for that object and begins for another object, starting with the topmost rule in the table.
3. If any portion of a key field is blank, the criterion is matched by default.
4. If the current rule is completely blank, all criteria are matched by default.

   Siemens Digital Industries Software recommends that you place rules containing more key field criteria above rules with fewer key field criteria.
5. If an object does not match any rule in the template, no data is extracted for the object.

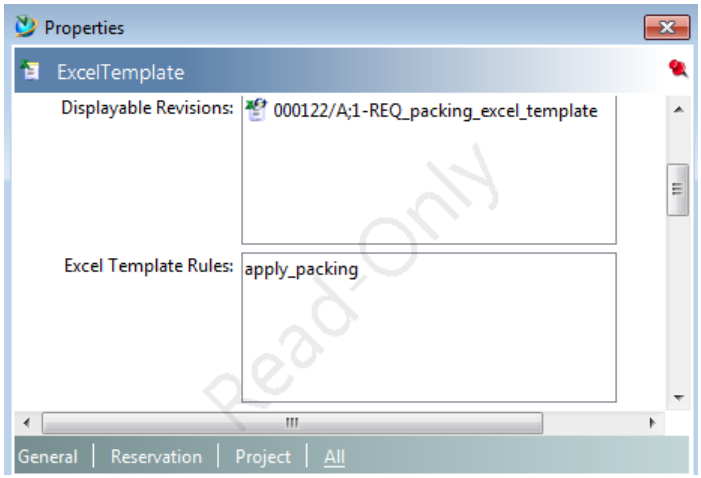
When the spreadsheet is generated, a warning message states that objects that do not match any template criteria have been ignored.

Setting up an Excel template for packed structure elements

Setting up an Excel template for object packing

In Teamcenter structures, two or more identical elements can be grouped, or *packed*, in a single level. To preserve that packing in one row of the Excel output file, set up the export template as follows:

- For the template's **Excel Template Rules** property value, choose **Apply Packing**.



- In the corresponding **Level** key field of the rule table, enter the level number as the argument.

  For example, enter **{%L-2}** for packed elements that are direct children of the top-level element.

Caution:

- Packing applies only if you export all elements in the structure.

  If you select individual elements, each one is exported to a separate row in the output file.

- Cell formatting may be processed as content, which in turn may prevent the desired data placement.

  Siemens Digital Industries Software recommends that you do not apply any formatting to a cell where you intend to include one object on the same row as another object (apply packing).

- A cell that contains a space is considered to be empty.

  To keep a cell from being overwritten, you must enter a visible character in the cell.

How objects are packed in Excel export files

When users export objects to Excel, objects in the database are examined to determine if their levels match the key fields in the rule table of the selected template. For the object that is currently being processed, data is placed in the export file as follows:

- If the level of the current object is lower than the level of the previous object, data for the current object is placed in a new row.
- If the level of the current object is greater than or equal to the level of the previous object:
  - If the cells following the previous object are empty, data for the current object is placed in the same row.
  - If the cells following the previous object contain any character, data for the current object is placed in a new row.

> Note:
> - Objects are packed only if the user exports all objects in the view.
>
> If the user exports selected objects only, each object's data is placed on a separate row.
> - A cell that contains a space is considered to be empty.
>
> To keep a cell from being overwritten, you must enter a visible character in the cell.
> - Siemens Digital Industries Software recommends that you do not apply any formatting to a cell where you intend to begin one object on the same row after another object.
>
> Cell formatting may be processed as content, which in turn may prevent the desired object placement.

For example, assume the following:

- Objects **A**, **D**, and **E** are at level one.
- Objects **B**, **C**, **F**, and **G** are at level two.

Assume also that tags for properties named **1**, **2**, **3**, and **4** are entered in the property columns of the template, with key fields entered in the rule table as in the following example.

					Level	**Relationship**	**Subtype**
{%1}	{%2}	{%3}			{%L-1}		
			{%1}	{%4}	{%L-2}		

By default, the property tags and key fields in the preceding example produce the following output in the export file.

A1	**A2**	**A3**		
			B1	**B4**
			C1	**C4**
D1	**D2**	**D3**		
E1	**E2**	**E3**		
			F1	**F4**
			G1	**G4**

When the **Apply Packing** value is applied to the template's **Excel Template Rules** property:

- **B1**, **B4**, **F1**, and **F4** are placed in the row above with the parent objects.
- **C1**, **C4**, **G1**, and **G4** are placed in the row immediately below.

A1	**A2**	**A3**	**B1**	**B4**
			C1	**C4**
D1	**D2**	**D3**		
E1	**E2**	**E3**	**F1**	**F4**
			G1	**G4**

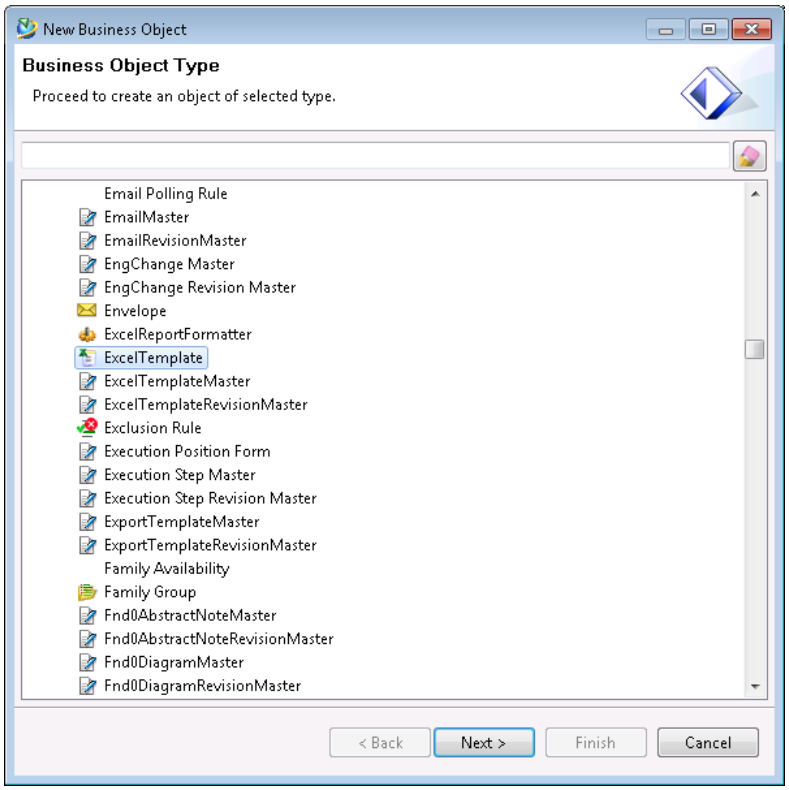
Apply packing to an Excel template

1. Select the Excel export template.
2. In the **Properties** tab, double-click the value for the **Excel Template Rules** property to display the **Multi-Choice** dialog box.
3. Select the **Apply Packing** check box.
4. In the **Level** column of the rule table, enter key fields according to the way you want to fill the rows in the export file.

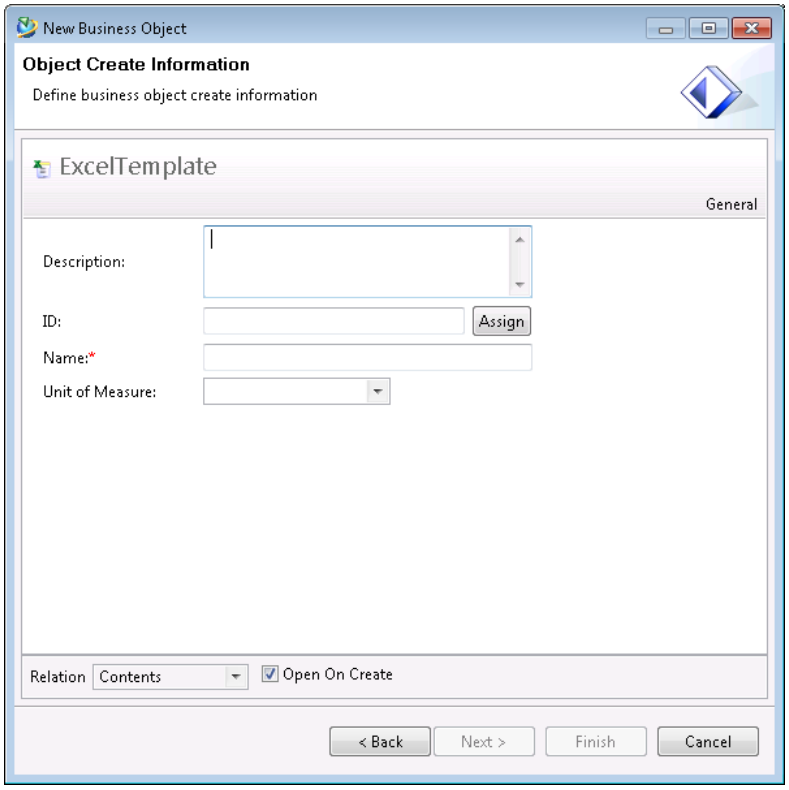
Create an Excel template

1. In your **Home** view, choose **Tools→Microsoft Office Templates→Create Excel Template** to open the **New Business Object** wizard.

2. Under **Most Recently Used** or **Complete List**, select **Excel Template**.



3. Click **Next** to advance to the **Object Create Information** page.



4. Type an item ID and revision ID in the **Item ID** and **Revision ID** boxes or click **Assign** to automatically assign the IDs.

5. Type a name for the Excel export template in the **Name** box.

6. (Optional) Type a description in the **Description** box.

7. Click **Finish**.

   The **Excel Template** item, **Excel Template Revision** item revision, and associated **MS ExcelX** dataset are created in the **ExcelTemplates** folder, which is located in the **RequirementsManagement Templates** folder in the **Home** view.

   > Note:
   >
   > The **REQ_default_excel_template.xlsm** file is associated with the **MS ExcelX** dataset by default.

8. (Optional) To update the contents of the specification export template document, double-click the **MS ExcelX** dataset.

9. (Optional) Modify the contents of the template document, save the Excel file, and close Excel.

Modify an Excel template

1. In the **Home** view, check out the **Excel Template Revision** item revision from the database.
2. Click the **MS ExcelX** dataset.
3. Click **Open** in the **File Download** dialog box.

   The template file is opened in Excel.
4. Modify the data in the template spreadsheet.

   > Caution:
   >
   > Properties with dynamic and cascading lists of values (LOVs) cannot be modified with live Excel.
5. Save and close the spreadsheet.

Create an Excel export template by copying an existing one

1. In My Teamcenter, click the **Excel Template** item containing the template that you want to duplicate, and choose **File→Save As**.
2. Click **Assign** in the **Item Details** pane of the **Save Item As** dialog box.

   The system assigns an item ID and revision ID for the new item.
3. Type a name for the new item in the **Name** box.
4. (Optional) Type a description of the new template item in the **Description** box.
5. (Optional) Select the **Open on Create** check box to open the item after it is created.
6. Click **OK**.

   The new **Excel Template** item and **Excel Template Revision** are created The **MS ExcelX** dataset is copied as a new object with the same name as the original dataset.

Import an Excel template

> Note:
>
> When exporting a Requirement Management template to PLM XML, use the **ConfiguredRequirementDataExport** mode, and when importing a Requirement Management template from PLM XML, use the **REQ_IMPORT_TEMPLATE** mode.

1. In the **Home** view, choose **Tools→Import→Templates→Excel Template**.
2. In the **Import Excel Template** dialog box, click **Browse** and locate the Excel file (**.xlsm** or **.xlsx**) containing the template you want to import.
3. Select the file and click **Save**.
4. (Optional) Type a description in the **Template Description** box.
5. Click **OK**.

   The template is imported in to the Teamcenter database.
6. Click **OK** in the **Import Template** dialog box.

   The **Excel Template** item, **Excel Template Revision** item revision, and associated **MS ExcelX** dataset are created in the **ExcelTemplates** folder, which is located in the **RequirementsManagement Templates** folder in the **Home** view.

Delete an Excel export template

1. Ensure that the **Excel Template** item is not referenced by other objects in the Teamcenter database by performing a where-referenced search.
2. In My Teamcenter, select the **Excel Template** item.
3. Choose **Edit→Delete**.
4. Click **Yes** in the **Delete** dialog box.

   The **Excel Template** item, **Excel Template Revision** item revision, and **MS ExcelX** dataset are deleted from the database.

Setting processing options for Excel templates

The Microsoft Excel template business object contains a list of values (LOV) property called **excel_template_rules**, which you can use to further configure Excel template exports. This LOV

property controls several additional behaviors that you can apply to the Excel template being used during the export to Excel by modifying the property in the rich client.

You set the **excel_template_rules** property on the template object itself. Because you do not tag it in the template's property table, its value is not exported to Excel output files and does not affect live Excel modifications.

The following are the possible LOV values on the **excel_template_rules** property.

LOV value	Description
disable_outline	Disables the outlining feature in Excel. If you set this LOV value on the Excel template and export a structure using this template, outlining is disabled and the exported structure is a flat structure.
apply_bom_crawling	Enables editing requirement properties in a Microsoft Excel export. By default, parents of a structure are not exported. But if you set this LOV value on the template and export a structure using the template, all of the parents of a selected element are exported and the export output in Excel matches the structure order.
apply_packing	Packs multiple rows into a single row.
absolute_level	Level of the object is considered as the absolute level within the structure. The level of the object is calculated as the actual level within the structure irrespective of the starting level of the top line object. This means that if any intermediate structure is exported, the starting level of the object is not zero, but it is considered as the absolute level of the object within the structure. For example, if you export an intermediate line in a structure, which is at level 4, the current export to Excel considers the level of the selected object as 0. If you want the actual level to be considered while exporting the structure to Excel, use the template and apply the **absolute_level** property so that the actual level, which is level 4, is considered.
disable_export_child_lines	Disables automatic traversal of the child lines. By default, when a structure is exported, you traverse deep into the structure. But if you set this LOV value on the template and export a structure using the template, automatic deep traversal of child lines is disabled. In this

LOV value	Description
	case, you can use a transfer mode to traverse deep into the structure.
audit_templates	Used for templates used for audit purposes (Audit Manager).
budget_templates	Used for templates in budget use cases (Budget scenarios).

Note:

You can set multiple LOV values for the same property.