



Teamcenter 2412

Systems Engineering — Deployment and Rich Client Usage

Teamcenter 2412

Unpublished work. © 2025 Siemens

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

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

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

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

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

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

About Siemens Digital Industries Software

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

Support Center: support.sw.siemens.com

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

Contents

Getting started with Systems Engineering

Getting started with Systems Engineering	1-1
Installing Systems Engineering components	1-1
Installing the requirements management microservice	1-2
Install Systems Engineering	1-3
Before you begin	1-4
Basic concepts about Teamcenter Systems Engineering	1-7
What is Systems Engineering?	1-7
Requirements management	1-8
Functional design	1-9
Logical modeling	1-9
Traceability	1-9
Performance measurement	1-10
Custom notes and supplemental information	1-10
Data export	1-10
Basic tasks using Systems Engineering	1-11
Systems Engineering interface	1-13
Systems Engineering interface	1-13
Systems Engineering views	1-14
Systems Engineering menus	1-46
Systems Engineering buttons	1-56
What are perspectives and views?	1-59
Working in the Systems Engineering interface	1-60
Working in requirement and architecture structures	1-60
Working with columns displayed in views	1-73
Working with multiple object selections	1-80

Developing requirements

Developing requirements	2-1
Building requirement specification structures manually	2-2
What is a requirement specification?	2-2
Create a requirement specification	2-2
Create a requirement	2-4
Create a derived requirement	2-5
Create a child requirement or paragraph	2-6
Create a paragraph	2-7
Importing requirement specifications from structured Microsoft Word documents	2-8
How Word styles determine structure hierarchy in Teamcenter	2-8
Generating separate requirements within a Word outline level	2-11
Special characters in Word import documents	2-12
Import a requirement specification from Word	2-12
Tracking the source of imported requirements	2-14

Resolve server time-out issues	2-14
Importing requirements from unstructured Microsoft Office files	2-15
Setting up a hierarchy for unstructured files	2-15
Accessing the flat file functionality using the Teamcenter ribbon	2-17
What are the steps you perform to import unstructured requirements?	2-19
Installing the flat file functionality	2-20
Customizing the flat file configuration files for your environment	2-21
Defining the requirement objects and trace links in the unstructured document	2-41
Search for existing requirement objects in Teamcenter	2-51
Previewing import file for unstructured documents	2-52
Importing the unstructured document to Teamcenter	2-60
Import objects in the Teamcenter format	2-77
Import objects in the STEP format	2-78
Import objects in the PLM XML format	2-79
Maintaining requirement structures and content	2-81
Navigating and organizing requirement specification structures	2-81
Editing requirement content	2-82
Comparing requirement content	2-89
Marking up requirement content	2-91
Create requirement variants	2-95
Managing parametric requirements	2-96
Duplicating requirement structures	2-101
Duplicating requirement structures	2-101
Duplicate a requirement structure	2-102
Find the source or duplicates of a requirement object	2-106
Editing a duplicated requirement specification structure	2-107
Cloning requirement structures	2-108
About cloning structures	2-108
Create a structure from a template	2-109
Designing and validating NX parts with requirements	2-112
Requirement types for NX design and validation	2-112
Create an NX design requirement	2-113
Create an NX validation requirement	2-115
Writing formulas for validation conditions	2-117
Defining variables for validation condition formulas	2-117
Editing design and validation requirements	2-118
 Editing requirements and structures with Teamcenter live Word	
How you can use live Word	3-1
Interactive live Word documents	3-3
Offline live Word documents	3-4
Open the Requirements pane in an interactive live Word document	3-6
Save a live Word document for offline editing	3-8
Accumulate live Word changes offline	3-9
Editing requirements and paragraphs in Live Word	3-10
Modifying requirement structures in live Word	3-12
Building requirement specifications in Word	3-12

Create a requirement specification	3-13
Create a requirement or paragraph object	3-14
Promote or demote an object	3-16
Move an object within a level	3-16
Remove a requirement or paragraph object from a structure	3-17
Moving and copying requirements between structures	3-17
Revising released objects in a requirements structure	3-28
Copying released objects in a requirements structure	3-38
Create trace links between objects	3-49
Save interactive live Word changes in Teamcenter	3-50
Synchronize offline live Word changes with Teamcenter	3-52
Accessing requirement content through URLs in live Word documents	3-55
Disconnect an offline live Word document from Teamcenter	3-55
 Designing functional and logical architectures	
Identifying the component subsystems of a product	4-1
Extending functional requirements to subsystems	4-1
Abstracting logical implementations	4-2
Building architectures in the Function or Logical Block view	4-3
Illustrating subsystems with object hierarchies	4-3
Organizing functional and logical hierarchies	4-4
Create a function	4-4
Create a logical block	4-6
Create a trace link between functions or logical blocks	4-8
Create an interface in a structure	4-9
Create a connection between two interfaces	4-10
Change interface direction in a structure	4-10
Remove elements from a functional or logical structure	4-11
Diagramming architectures in Microsoft Visio	4-13
Illustrating subsystems with diagrams	4-13
How diagram templates guide diagram content	4-17
Master shapes in Teamcenter stencils	4-18
Adding structure detail with containers	4-20
Create a diagram	4-21
Open a diagram from the Function or Logical Block view	4-24
Open a diagram from another diagram	4-25
Previewing diagram snapshots	4-26
Navigate from a diagram shape to the underlying Teamcenter object	4-26
Editing diagrams	4-26
Measuring technical progress for design solutions	4-42
Defining budgets	4-42
Reporting budget values upstream	4-43
Upstream reporting example	4-43
Create a budget definition	4-44
Create a budget on a structure	4-46
Edit budget values for a structure	4-47
Remove a budget from a structure	4-48
Reusing standard components from centralized libraries	4-49

Using data dictionaries	4-49
Adding a signal component to a functional or logical structure	4-49
Creating a data dictionary	4-50
Searching the data dictionary	4-56
Create a signal	4-71
Add a signal to a data dictionary	4-72
Add a signal to a functional or logical structure	4-72
Associate signals with an interface or a connection	4-73
View or navigate to signals associated with an interface or a connection	4-74
Remove signals from an interface or a connection	4-75
Set effectivity for a signal revision	4-75
Managing connections using Connection Manager	4-76
Create a connection	4-76
Remove a connection	4-77

Allocating requirements and design architectures

Traceability in product design	5-1
How trace links associate system elements	5-2
Trace link ownership	5-2
Trace link inheritance	5-3
How new item revisions affect traceability	5-3
How object modifications affect traceability	5-5
How object copying affects traceability	5-7
Rules for trace links	5-9
Defining trace link rules	5-9
Enforcing trace link rules	5-9
View trace link rules for object type pairs	5-10
Create a trace link rule	5-11
Set preference values for trace link rules	5-13
Creating trace links	5-14
Selecting defining and complying objects	5-14
Customizing trace links with subtypes	5-15
Create trace links	5-15
Default trace link name	5-19
Create a custom note on a trace link	5-20
Copy a custom note to a trace link	5-22
Remove a custom note from a trace link	5-25
Delete trace links	5-27
Viewing traceability relationships	5-28
Viewing downstream and upstream paths	5-28
What does the Trace Links view provide?	5-29
Open the Trace Links view	5-29
Column settings in Trace Links view	5-31
How certain objects appear in the Trace Links view	5-32
View defining and complying relationships	5-34
Display linked and unlinked objects in the Trace Links view	5-35
View the context of trace links on absolute occurrence objects	5-35

Navigate to a trace linked object	5-37
Open a parent requirement in the Trace Links view	5-38
Filter traceability information by configured structure or element	5-39
Filter traceability information by trace link type	5-41
Filter traceability information by object type	5-43
Filter traceability information by property	5-45
Set expansion levels in the Trace Links view	5-47
Highlighting requirement objects from the Trace Links view	5-48
Set scope from custom dialog box for the Trace Links view	5-50
Reporting traceability relationships	5-51
Generate a traceability matrix	5-51
Create trace links in the Traceability Matrix	5-54
Create trace links to occurrences in the Traceability Matrix	5-55
Navigate to a requirement in the Traceability Matrix	5-57
Delete trace links in the Traceability Matrix	5-58
Export the Trace Links view to Excel	5-59
Export traceability information to Excel in a structure view	5-61
Navigate to an occurrence by specifying a variant rule and perspective	5-62
Tracing relationships among Teamcenter domains	5-64
Determining change effect on downstream and upstream targets	5-64
Setting options for cross-domain tracing	5-64
Open views for cross-domain tracing	5-67
Run a cross-domain trace	5-69
Checking accountability between structures	5-70
Checking requirement structures against product and process structures	5-70
About the Accountability Check view	5-71
Run an accountability check	5-72
Set the result reporting options	5-75
Set the equivalence options	5-76
Specify partial match criteria	5-77
Clear display	5-78
Generate the results grouped by anomaly	5-78
Create a printable Excel report	5-79
 Recording supplemental information	
What are custom notes?	6-1
Associating custom notes to Systems Engineering objects	6-2
Create a custom note	6-2
Edit the contents of a custom note	6-4
Attach a custom note to an item or item revision	6-5
Create a new revision of a custom note	6-6
Delete a custom note	6-8
 Capturing data for downstream disciplines	
Exporting product data	7-1
Exporting data to Microsoft Office Word	7-2
Exporting data to Microsoft Office Word	7-2

Setting the structure context for export to Word	7-3
Override the default object template for frequent Word exports	7-4
Creating new objects in Word using object templates that contain formatting	7-4
Capturing a fixed data record in Word	7-6
Display static snapshot datasets	7-7
Changing property values interactively with live Word	7-8
Sharing comments from review participants	7-9
Export data to Word	7-10
Exporting data to Microsoft Office Excel	7-14
Exporting data to Microsoft Office Excel	7-14
Capturing a fixed data record in Excel	7-14
Editing properties interactively in Excel	7-15
Processing Excel edits in bulk	7-15
Preparing structure changes offline in Excel	7-16
Working around Excel's eight-level limitation	7-16
Export data to Excel	7-17
Using transfer modes with Excel export	7-22
Excel template rule tables	7-24
Exporting objects to Teamcenter email	7-28
Exporting objects to Teamcenter email	7-28
Send structure data by email	7-29
Access structure data in email	7-30
Export objects in the PLM XML format	7-31
Export objects to an application interface	7-32
Export objects in the Teamcenter format	7-33
Export objects in the STEP format	7-34

Using allocations

Understanding allocations	8-1
Using allocations between multiple sources and targets	8-1
Choosing allocations or trace links	8-4
Implementing allocations in Teamcenter	8-5
Understanding allocations in Teamcenter	8-5
Making allocations between configured structures	8-5
Making conditional allocations	8-7
Creating allocations	8-8
Revising allocations	8-9
Configuring allocations	8-10
Releasing allocations	8-12
When can I release allocations?	8-12
Releasing an allocation	8-13
Releasing an item with an allocation to a physical component	8-13
Releasing allocation maps	8-13

Configuring Systems Engineering

Configuring Systems Engineering	A-1
Overview of Systems Engineering options	A-1

Configuring Systems Engineering naming rules	A-3
Naming Systems Engineering objects	A-3
Autoname requirements	A-3
Autoname custom notes	A-4
Defining allowed child item types for Systems Engineering structures	A-4
Enable moving objects between levels in structures	A-5
Migrate Microsoft Excel templates from versions prior to Teamcenter 8.3	A-5
Exclude specification types from the Import Spec wizard	A-6
Configure keywords for importing requirement specification structures	A-7
Configure object checkout for export to Microsoft Office Excel	A-7
Configure trace link features	A-8
Configuring defining and complying object calculation	A-8
Configuring display of the Summary view and Properties dialog box	A-9
Configure baseline features	A-9
Setting up a custom template for requirements	A-10
Process for setting up a custom template	A-10
Add the FullText subclass for datasets	A-10
Create a requirement template	A-11
Associate a requirement template revision with Microsoft Office Word	A-12
Release a requirement template revision	A-13
Apply a requirement template revision to a business object	A-14

Customizing Word export templates

Exporting object data to Microsoft Word	B-1
Specification templates	B-2
What is a specification template?	B-2
Specification template components	B-3
Create a specification template	B-7
Import a specification template	B-9
Modify a specification template	B-10
Create a specification template by copying an existing one	B-10
Delete a specification template	B-11
Object templates	B-12
What is an object template?	B-12
Example: Object template without rules	B-13
Object export template document with rules	B-14
Create an object template	B-14
Import an object template	B-15
Modify an object template	B-15
Create an object template by copying an existing one	B-16
Delete an object template	B-16
Content rule for specification templates and object templates	B-17
Property name syntax in specification templates and object templates	B-18
Using macros in Microsoft Office Word templates	B-19
Associating export templates with business objects	B-19
Applying Microsoft Office Word templates to exported objects	B-19

Customizing Excel export templates

Excel template components	C-1
Create an Excel template	C-2
Import an Excel template	C-4
Modify an Excel template	C-4
Copy an Excel template	C-5
Tagging properties for export to Excel	C-5
Defining rules for filtering properties	C-6
Rule table processing	C-7
Using transfer modes in Excel templates	C-8
Setting processing options for Excel templates	C-8
Setting up an Excel template for object packing	C-10
How objects are packed in Excel export files	C-11
Insert macros in an Excel template	C-12
General guidelines for Excel templates	C-13
Delete an Excel template	C-14

Customizing Visio diagram templates

Diagram template components	D-1
Default Teamcenter stencils	D-1
Default mapping files	D-2
Designating application domains for diagram templates	D-3
Setting diagram template membership rules	D-4
Setting diagram template relation rules	D-5
Activating trace link subtypes for diagrams	D-6
Setting the diagram template availability flag	D-8
Enable connections without interfaces in diagrams	D-8
Hiding interfaces in diagrams	D-9
Create a diagram template	D-9
Edit a diagram template	D-11
Defining master shapes	D-12
Define a block master shape	D-12
Define an interface master shape	D-15
Define a connection master shape	D-18
Define a relation master shape	D-20
Define a container master shape	D-22
Defining compound master shapes for showing multiple properties	D-26
Mapping master shapes to Teamcenter	D-30
Teamcenter mapping file contents	D-30
Declaring a master shape	D-31
Relating a master shape to a Teamcenter object type	D-32
Making a master shape unique	D-34
Specifying required properties on a master shape	D-35
Specifying the property for the master shape text box	D-36
Additional mapping for creating occurrences and copying workspace objects	D-37
Making Teamcenter properties available in Microsoft Visio	D-39

Assigning interface names automatically	D-40
Setting interface direction for new connections	D-41
Localizing Teamcenter stencils and mapping files	D-42
Extending Microsoft Visio events	D-42



1. Getting started with Systems Engineering

Getting started with Systems Engineering

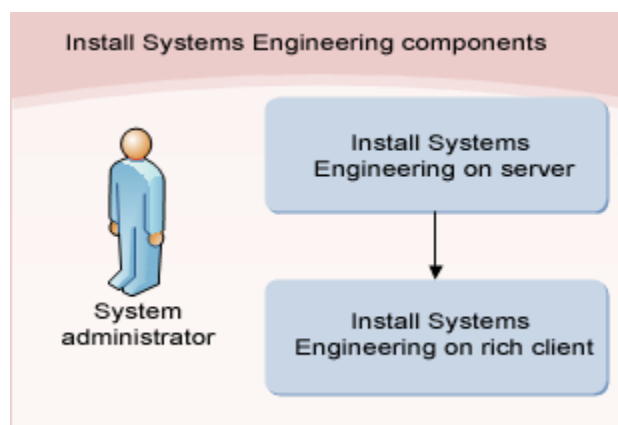
Teamcenter Systems Engineering provides tools for managing and developing complex, multidomain products or systems. Systems engineering is:

- Where product development begins.
- Where requirements are defined.
- When architecture decisions are made.
- Where interfaces are defined.
- Where globally optimized product decisions are made.
- Where you define what requirements, assemblies, and components can be reused.

Teamcenter Systems Engineering supports standard systems engineering processes such as IEEE 1220, ISO 15288, and EIA 632. It allows you to develop requirements as well as to create functional, logical, and physical models that you can use to define interfaces between elements and understand the relationships between them. You can also use it for ad hoc systems engineering and product development activities unique to your organization.

Installing Systems Engineering components

As a system administrator, you must install and configure Systems Engineering using Teamcenter Environment Manager (TEM). You install Systems Engineering on the server and on the rich clients that use the application.



Installing the requirements management microservice

1. Log on to Deployment Center.
2. Click the **SOFTWARE REPOSITORIES** tile and verify availability of the following: Teamcenter Foundation, Active Workspace, and the Microservice Framework software kits. Alternately, you can choose an existing environment containing the software to update.
3. Click **Update Software**.
4. On the **Components** tab, click **Start** on the **Microservice Node** to configure it. Update the fields including the machine name, container path, and service password.

Also specify the number of instances to run for each service.

5. Click **Save Component Settings** when done.
6. To deploy services on multiple nodes, click ⊕
on the **Components** tab, and choose **Microservice Node** and **Update Selected Components**.
7. Click **Start** on the new **Microservice Node** row to configure it.
8. Click **Generate Install Scripts** on the **Deploy** tab.

This creates a zip file for each service node, and generates instructions on deploying it on that node.

Deployment Instructions

Instructions for deploying on each of the service nodes are provided. As an example:

Deploy Instructions for Machine Scripts

- 1) Ensure that the software listed in the "Software Needed for Install" section is copied to a shared directory location available to all target machines.
- 2) Copy each zip file listed in the table above under "Zip File Name" to the corresponding machine listed under "Target Machine".
- 3) Unzip the zip file into any location on the target machine.
- 4) Open a command window on the target machine and execute the "deploy.bat" script located inside the unzipped folder location. Use the -softwareLocation command argument to provide the location of the shared directory that contains software. Provide the Deployment Center authentication parameters using the -dcusername and -dcpassword command arguments

Deployment Example

On the machine specified as the service host, the following command is executed:

```
unzip 20190423_164928EDT_TC_Foundation_and_ActiveWorkspace_Host1.zip
deploy -softwareLocation \\tserver\kits -dcusername admindc -dcpassword *****
```

If the output is successful, the following displays:

```

Executing Deployment Steps
Deploy Task: 1 of 15 Component-->Microservice Node-->analyseDiagnosticChecks...Success
Deploy Task: 2 of 15 Package-->Requirements Management Compare Service-->fileProcessing...Success
Deploy Task: 3 of 15 Package-->Requirements Management Export Service-->fileProcessing...Success
Deploy Task: 4 of 15 Package-->Requirements Management Import Service-->fileProcessing...Success
Deploy Task: 5 of 15 Component-->Microservice Node-->fileProcessing...Success
Deploy Task: 6 of 15 Package-->Requirements Management Compare Service-->preDatabaseUpdate...Success
Deploy Task: 7 of 15 Package-->Requirements Management Export Service-->preDatabaseUpdate...Success
Deploy Task: 8 of 15 Package-->Requirements Management Import Service-->preDatabaseUpdate...Success
Deploy Task: 9 of 15 Component-->Microservice Node-->preDatabaseUpdate...Success
Deploy Task: 10 of 15 Component-->Microservice Node-->databaseUpdate...Success
Deploy Task: 11 of 15 Package-->Requirements Management Compare Service-->postDatabaseUpdate...Success
Deploy Task: 12 of 15 Package-->Requirements Management Export Service-->postDatabaseUpdate...Success
Deploy Task: 13 of 15 Package-->Requirements Management Import Service-->postDatabaseUpdate...Success
Deploy Task: 14 of 15 Component-->Microservice Node-->postDatabaseUpdate...Success
Deploy Task: 15 of 15 Interoperability started...Success
Location of software installed/updated directory: D:\apps\lsc\TC\TR\
Status: Deploy Script Execution Successful !!!

```

The output also displays additional deployment steps:

Deployment Notes:

Action for User:

The files in the signer_config directory are required to sign authentication tokens provided when sending a request to a microservice. This directory should be copied to the Teamcenter web tier host machine prior to executing the web tier deployment script.

Please refer to documentation for details on J2EE Webtier deployment. It should also be copied into \$TC_DATA of the Teamcenter installation.

The files in the D:\apps\lsc\TC\TR\microservices\secrets directory are required by microservices to validate authentication tokens in a request. This directory should be copied to other Microservice Node machine.

Install Systems Engineering

1. Start Teamcenter Environment Manager (TEM).
2. Select appropriate options until you reach the **Features** pane.

Note:

The configuration installed is dependent on the options that you choose in the **Solutions** panel of the TEM installation windows, such as **Corporate Server**, **Rich Client 2-Tier** or **Rich Client 4-Tier**, and **Business Modeler IDE**. You can also add any of the options available in the **Solutions** panel from the **Features** pane.

3. In the **Features** panel under **Extensions**, expand **Systems Engineering and Requirements Management** and select the following features.

Feature	Description	Prerequisite features
Teamcenter Extensions for Microsoft Office	Installs Teamcenter Extensions for Microsoft Office.	Teamcenter Foundation
Requirements Manager	Installs the optional Requirements Manager support in the Systems Engineering application.	Teamcenter Foundation
Systems Engineering	Installs the optional support for functional modeling and budgets in the Systems Engineering application.	Teamcenter Foundation

4. (Optional) To configure your system using the Business Modeler IDE, do the following:
- Click **Next**, and then select appropriate options until you reach the **Business Modeler IDE Templates** pane.
 - In the **Business Modeler IDE Templates** pane, select the following options:

Template	Description
Teamcenter Foundation	Installs the Teamcenter Foundation template.
Systems Engineering	Installs the Systems Engineering templates.

5. Click **Next**, and then select appropriate options until you complete the installation.

Before you begin

Prerequisites	You need a basic understanding of the systems engineering discipline to use Systems Engineering. You do not need special permissions to use Systems Engineering. Standard Teamcenter user and administrator accounts are adequate. You need valid feature licenses to work with Systems Engineering. Certain menu commands are disabled if these licenses are not installed. Contact your Siemens Digital Industries Software representative for more information.

To use Teamcenter Extensions for Microsoft Office, the following must be installed on your computer:

- Teamcenter Extensions for Microsoft Office
- Microsoft .NET Framework

Note:

For certified versions of Microsoft products, see the hardware and software certifications matrix.

Enable Systems Engineering

Systems Engineering is an optional module installed as part of the standard Teamcenter core installation procedure. It does not need to be enabled before you use it.

During installation in Teamcenter Environment Manager, select the following check boxes under **Extensions**→**Systems Engineering and Requirements Management** in the **Features** panel:

- **Requirements Management**
- **Systems Engineering**

If you have trouble gaining access to Systems Engineering, consult your Teamcenter system administrator. It may be a licensing issue.

Note:

You can log on to Teamcenter only once. If you try to log on to more than one workstation at a time, an error message appears.

Configure Systems Engineering

Before using Systems Engineering, you must:

- Configure the necessary business rules.
- Set the appropriate preferences.

You can optionally configure some Systems Engineering features by setting user preferences.

- The **MS Word** view
- Keywords for parsing Word import documents
- Baselines

- Property columns
- Trace links

Note:

Trace link features are not displayed until you set the corresponding preferences.

- Display of defining and complying objects as properties
- The **Summary** and **Properties** panes

Start Systems Engineering

Do one of the following:

- Select **Systems Engineering** in the Teamcenter navigation pane.

Note:

You may need to enter your user name and password to log on.

The Systems Engineering main window is displayed. You can begin using Systems Engineering by doing any of the following:

- Choose **File→Import spec** to display the Import Spec wizard.
- Double-click a requirement specification, requirement, paragraph, function, or logical block to display a new tab containing the item.
- Click the **Most recently used structures** button on the toolbar to select an object from the list.
- In another Teamcenter application, such as My Teamcenter, Structure Manager, or Multi-Structure Manager:
 1. Navigate to a requirement specification, a requirement, a paragraph, a function, or a logical block that you want to send to Systems Engineering.
 2. Right-click the object and choose **Send To→Systems Engineering**.

Systems Engineering opens with the object displayed in a new tab.

Tip:

When you start a Systems Engineering session, the view you had open in the previous session is restored. The object you previously selected is highlighted, and the column configuration and revision rule are also restored. The previous session's sorting sequence is not retained, and the level expansion is retained only for the child object selected when the previous session ended. If multiple structure views were previously open, content is restored only for the last active view. If the client cache is cleared, the previous session settings are not restored.

Basic concepts about Teamcenter Systems Engineering

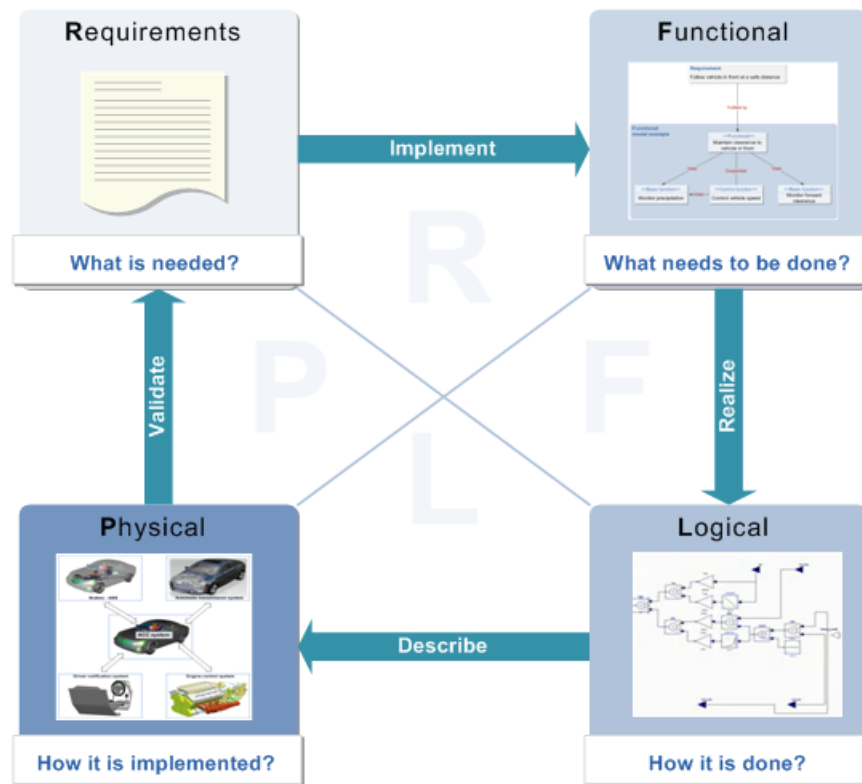
What is Systems Engineering?

The systems engineering discipline is a method of managing the design of complex, multi-domain engineering products throughout the life cycle of a project. Teamcenter Systems Engineering provides an object model and tools for developing (authoring) and managing the data related to such projects.

A product development project typically starts with a mixture of business objectives such as:

- Customer expectations and specifications
- Compliance with government regulations and industry standards
- Consideration of current technology and market factors

These objectives lead to a set of requirements that communicate the product design to the various disciplines in the development process. Teamcenter Systems Engineering delivers complete capability throughout the process of requirements management, functional design, logical modeling, and physical design (RFLP).



Systems Engineering primary capabilities include the following:

- Requirements management
- Functional design
- Logical modeling
- Traceability
- Performance measurement
- Custom notes and supplemental information
- Data export

Requirements management

To manage requirements, an organization may use spreadsheets, linked documents, custom databases, and document-oriented tracing tools. Typical problems in such methods are that requirements are isolated on individual computers with limited access, stored in databases with little resemblance to the product structure, or maintained through complicated user interfaces with significant learning curves.

Systems Engineering addresses this problem by bringing requirements into the life cycle and simplifies requirement development and access, and substantially reduces the learning curve. Requirements are developed with Microsoft Office Word and Excel, tools that are readily available in most organizations.

Functional design

Functional design allows users to capture what the product is supposed to do in an implementation independent way. Functional design enables users to describe what the product does without dictating how it is to be done. Functional design isolates implementation from what the product is supposed to do allowing organizations to evaluate many possible implementation alternatives without locking themselves into a particular alternative too soon.

Teamcenter Systems Engineering allows users to capture functions (with their inputs and outputs) while allocating requirements to functions and deriving additional requirements. For such functional design architectures, implementation alternatives can be evaluated through logical modeling.

In Systems Engineering, functional designs are developed through an interface with Microsoft Office Visio.

Logical modeling

Logical modeling allows users to capture a physical product alternative that addresses the functions. Allocating functions to logical blocks implements the interfaces described by the function inputs and outputs eventually showing up and physical interfaces that are implemented in the product.

The logical model describes the solution elements in abstract terms, and how they interact with each other logically rather than physically. For example, for an electromechanical product, you can create a logical model that describes the product or system in terms of input ports, output ports, interfaces, and connections.

The logical model depends on a data dictionary as a central repository for common signals and interfaces. For this, Classification provides the central repository.

The working and definition of the logical model in Teamcenter is managed in the same way as the functional model.

In Systems Engineering, logical models are developed through an interface with Microsoft Office Visio.

Traceability

Traceability provides a connection between objects in Teamcenter. For Systems Engineering, this means connecting our structure elements together. For example, creating links between requirements or between logical blocks. You can also create links between functions and requirements or logical blocks.

When the requirements, functional design and logical modeling are complete, users have complete traceability from requirements, to functions, to logical and physical objects allowing users to see the decision, requirements and alternatives that lead to a particular design alternative.

Trace links are used to provide traceability between structure elements and in which one object takes precedence over another. The predecessor, the defining object, is the trace link source. The successor, the complying object, is the target. Trace links are defined for both revisions and occurrences.

Performance measurement

Performance measurement includes tracking costs, weights, power consumption, or other measurable elements of your design. In Teamcenter, these measurements are defined and tracked using budget definitions. You assign a budget to any structure line that represents a significant element of the system. Budgets also contain rollout capabilities so you can view total performance for your structure.

Early in a product's life, goals and targets for critical parameters are set. For example, how much power a system uses, how much it weighs, and how much it costs. Distributing and managing these targets throughout the product life cycle is known as Technical Performance Management.

Once targets are established, Systems Engineering distributes those values across functional and logical structures, establishing performance goals for subsystems, which in turn distribute their targets to their subsystems. Periodically, actual values are *rolled up* to check progress towards meeting the goals, allowing users to correct their trajectory towards the goals in progress versus trying to figure out ways to change the product late in the life cycle.

In Systems Engineering, performance measurements are implemented through an interface with Microsoft Office Excel.

Custom notes and supplemental information

Custom notes contain supplemental information for elements in your structure, including requirements. Once a custom note is attached to your structure, you use the **Custom Notes** view to manage the notes.

As users make decisions, hold meetings, and discover problems, they need a way of documenting issues, actions and questions. Custom notes provide the ability to attach a note to anything to document the issue. Custom notes is an important way to capture issues they happen and directly associate issues with an object. For example, *why is this requirement this value?* can be directly associated with a requirement along with an answer for future reference.

In Systems Engineering, custom notes are developed through an interface with Microsoft Office Word.

Data export

Exporting Systems Engineering data to Microsoft Office Word and Excel help to easily manage requirements, budgets and notes. Teamcenter provides a live Office integration in most cases so

updates refresh in real time. Bulk loads are available when significant changes are required outside of Teamcenter.

Getting information into and out of Teamcenter Systems Engineering is important as other environments outside of Teamcenter are both sources and destinations for that information. A number of means for importing and exporting data from Teamcenter are available including:

- Document import/export
- Excel import/export
- XML-based import/export

Basic tasks using Systems Engineering

- Develop and manage requirements.

Use the **Requirements** view to perform the following tasks:

- Build requirement specification structures.
- Edit requirement content.
- Design and validate NX parts with requirements.
- Describe system functions with functional decompositions.

Use the **Function** view to perform the following tasks:

- Define system functions.
- Build functional decompositions.
- Diagram functional decompositions.
- Define functional solutions with logical models.

Use the **Logical Block** view to perform the following tasks:

- Design solution alternatives.
- Build logical decompositions.
- Diagram logical decompositions.

- Allocate requirements to design elements.

Use the **Trace Links** view to perform the following tasks:

- View defining and complying relationships.
 - Check requirement structures against product and process structures.
 - Navigate to a linked object.
 - Delete a trace link.
- Measure system performance.

Use the **Budgets** view to perform the following tasks:

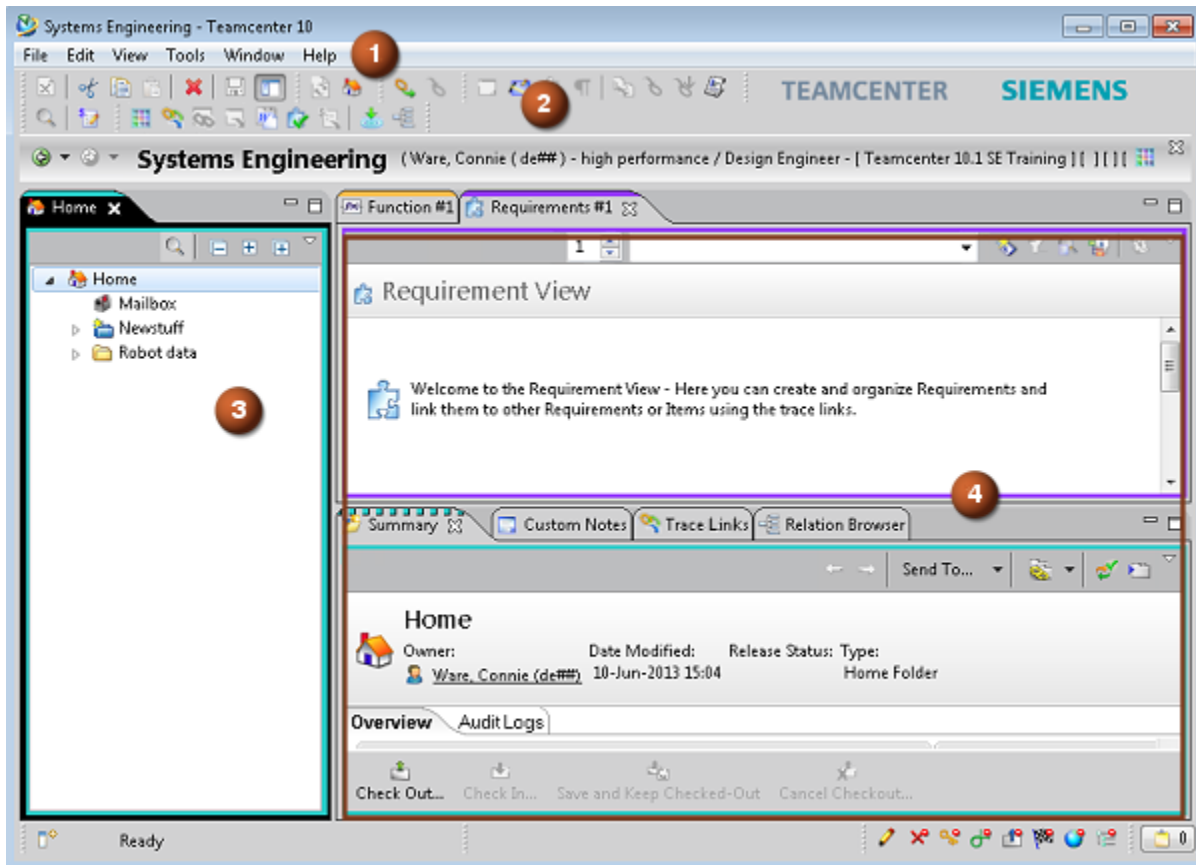
- Define technical performance measures through budgets.
 - Apply budget values to the structure.
 - Roll up budget data in the structure.
 - View, edit, and delete budgets.
- Record supplemental information.

Use the **Custom Notes** view to perform the following tasks:

- Add or edit custom note text.
 - Revise custom notes.
 - Delete custom notes.
- Export data from Systems Engineering.

Systems Engineering interface

Systems Engineering interface



1	Systems Engineering menu	Displays menus to support Systems Engineering actions. Some menu commands are specific to Systems Engineering.
2	Systems Engineering toolbar	Displays buttons to support Systems Engineering actions. Some toolbar buttons are specific to Systems Engineering.
3	Home view	Displays your Home folder structure. You place the objects you want to work with within your Home folder or within some folder structure beneath the Home folder.
4	Systems Engineering views	Displays multiple views of your Systems Engineering data. When you send Systems Engineering data to the Systems Engineering perspective, the data opens in the applicable view.

Example:

When you send a requirement specification to Systems Engineering, the requirement specification opens in the **Requirements** view. The **Requirement**  symbol displays in the tab.

Also, Systems Engineering has some interface elements, for example, the **Trace Links** view, that are not displayed in other Teamcenter perspectives.

Note:

You can open each view in a separate window by right-clicking the tab and choosing **Detached**.

Tip:

When you start a Systems Engineering session, the view you had open in the previous session is restored. The object you previously selected is highlighted, and the column configuration and revision rule are also restored. The previous session's sorting sequence is not retained, and the level expansion is retained only for the child object selected when the previous session ended. If multiple structure views were previously open, content is restored only for the last active view. If the client cache is cleared, the previous session settings are not restored.

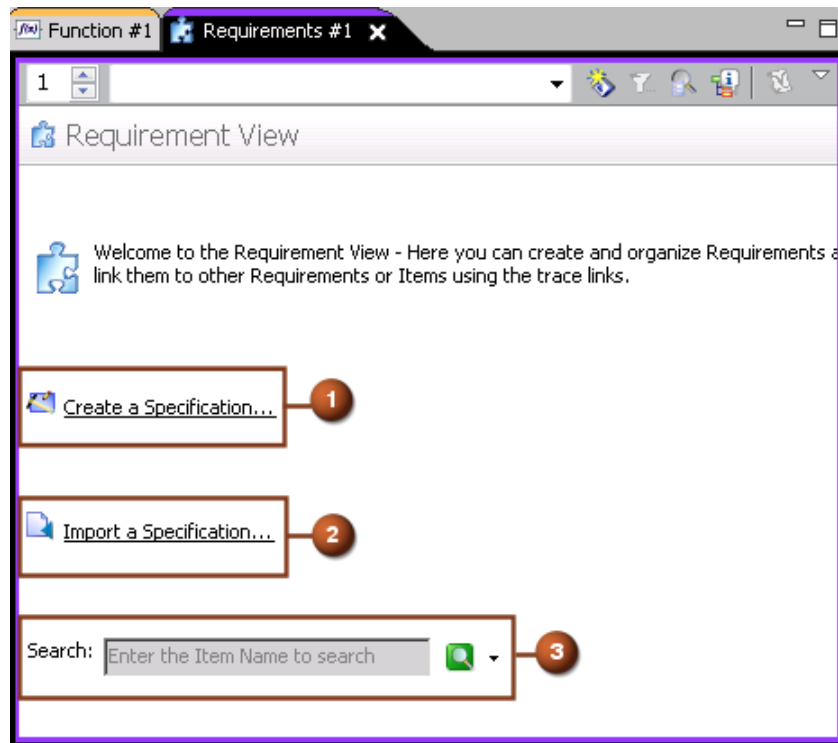
Other elements of the Systems Engineering interface are common to multiple Teamcenter applications.

Systems Engineering views

Requirement welcome view

When you open the Systems Engineering application, a default requirement welcome view appears that allows you to create a requirement specification, import a requirement specification, and search for Teamcenter data.

A *requirement specification* is a container for requirement and paragraph objects. Within a requirement specification, these objects can be structured in a hierarchy of parent, child, and sibling relationships.



1	Create a Specification	Allows you to create a new requirement specification.
2	Import a Specification	Allows you to import a new requirement specification from a word document.
3	Search	Allows you to perform standard and advanced searches for Teamcenter data.

Requirements view

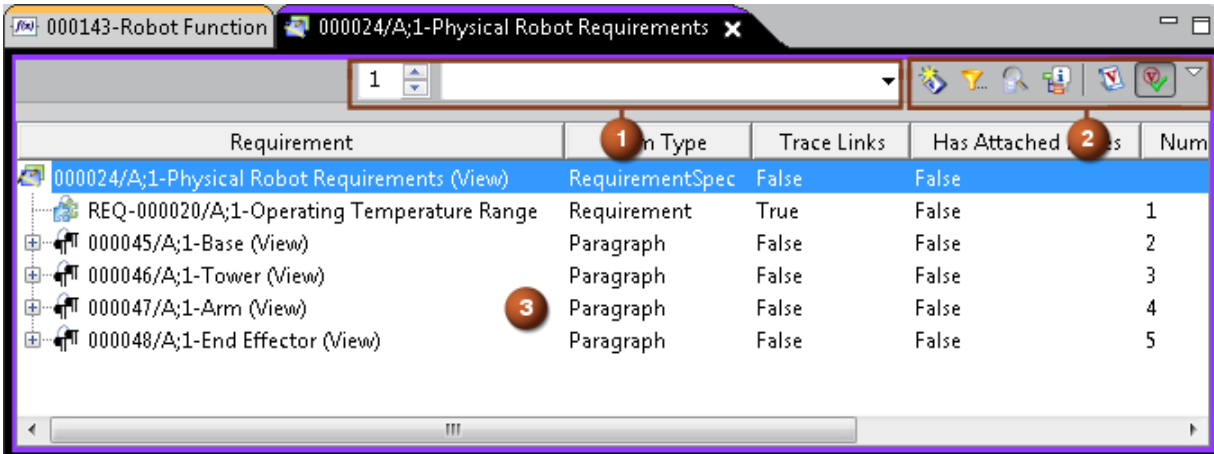
Requirements describe the product that the customer will buy. They communicate the customer's specifications to the various disciplines involved in the product's development.

The **Requirements** view displays hierarchical structures of requirements. The top level object, a *requirement specification*, is a container for requirement and paragraph objects. Within a requirement specification, objects can be structured as parents, children, and siblings.

The **Requirements** view is displayed when:

- You create a requirement specification or requirement structure.
- You import a requirement specification.
- You open an existing requirement specification or requirement structure.

A tab is created for the requirement specification and requirement structures that you create or open. You click the tab to display the structure.



1 Table display controls

The number box allows you to select the number of rows per requirement. The default setting is one row per object. Additional rows are added per requirement to view multiple lines of text in the table.

The column configuration box allows you to select a saved column configuration from the list. Column configurations change the tree table according to the defined columns in the saved configuration.

2 Requirements view toolbar

Displays buttons to support configuration of the requirements.

3 Structure pane

Contains a tab for each requirement specification that you send to Systems Engineering. You click a tab to display that structure in the pane.

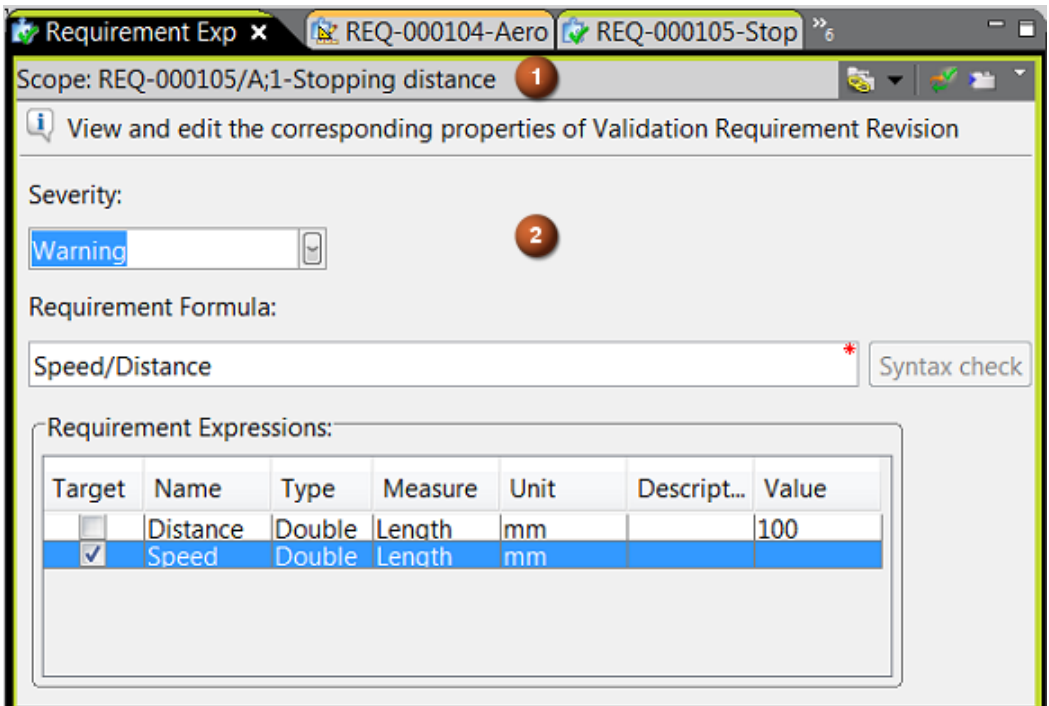
- The requirement specification object occupies the root level on each tab.
- Below the root level, requirements and paragraphs can be structured in a hierarchy of parent, child, and sibling items.
- A plus sign (+) is shown to the left of each item with children at the next lower level.

You can split the structure pane into upper and lower or left and right portions. For example, you can:

- Drag and drop a tab to the left, right, or bottom of the structure pane to see views side by side.
- Resize the upper and lower panes.
- Close the lower pane.

Requirement Expression view

The **Requirement Expression** view is used to display and edit the parameters for validation and design requirements. Some types of requirements that are used with NX have parameters that you can define. These parameters are then used by the NX validation processes.

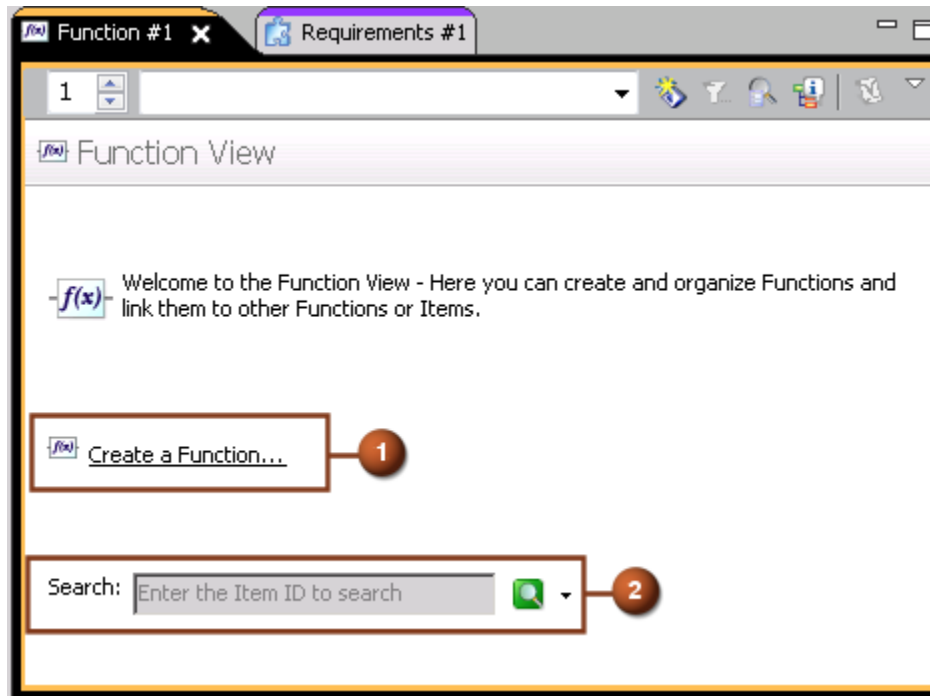


1	Requirement Expression view toolbar	Displays buttons to support the display of a requirement for design expression.
2	Requirement Expression view pane	Displays input parameters to CAD models for NX parts for a requirement design. Displays condition formula, variables, and severity level for NX parts for a requirement validation.

Function welcome view

When you open the Systems Engineering application, a function welcome view appears that allows you to create a new function and search for Teamcenter data.

A *function* object represents a purpose or function of the system or product. You create functional models by creating a set of functions that define what the product or system performs or delivers.



1	Create a Function	Allows you to create a new function.
2	Search	Allows you to perform standard and advanced searches for Teamcenter data.

Function view

The **Function** view displays the function models. A *function* object represents a purpose or function of the system or product. You create functional models by creating a set of functions that define what the product or system performs or delivers.

The **Function** view is displayed when you create a functional model or open an existing functional model.

A tab is created for the functional models that you create or open. You click the tab to display the functional model.

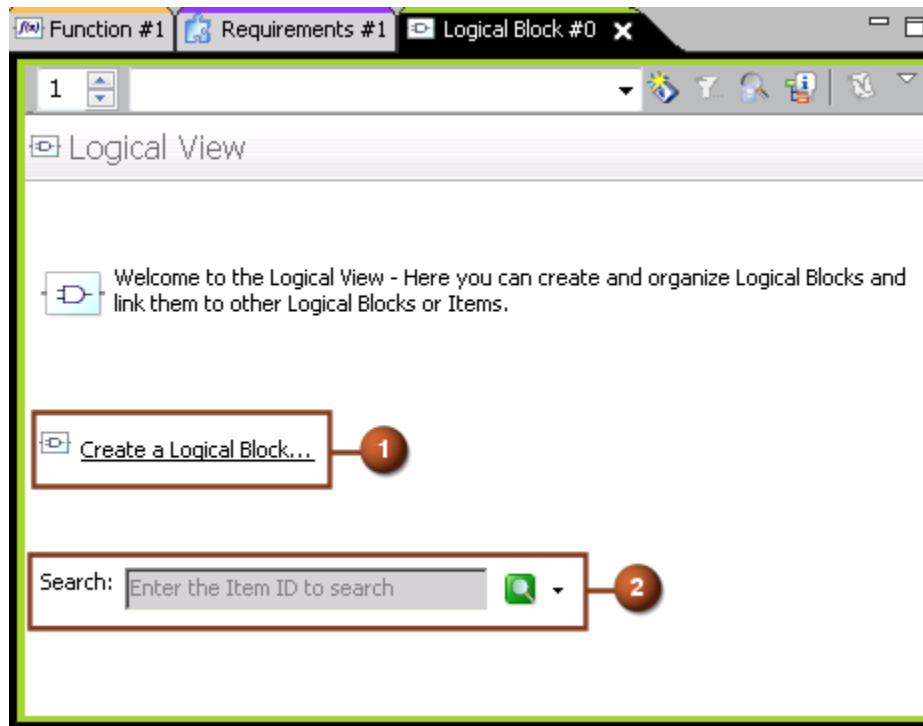
Item Id	Rev Name	Trace Links	Has Attached Notes	Body Cleartext	fnd0bl_hiera
000143	Robot Function	False	False		
000225	Sensor	False	False		
000161	Manage Communications	False	False	1	
000145	Manage Sensors	False	False	2	
000144	Control	False	False	3	
000309	Manage IO	False	False	4	

1	Table display controls	The number box allows you to select the number of rows per BOM line. The default setting is one row per object. Additional rows are added per function to view multiple lines of text in the table. The column configuration box allows you to select a saved column configuration from the list. Column configurations change the BOM tree table according to the defined columns in the saved configuration.
2	Function view toolbar	Displays buttons to support configuration of the functional model.
3	Structure pane	Displays the BOM lines for the functional model.

Logical welcome view

When you initially open the **Logical Block** view, a logical welcome view appears that allows you to create a new logical block and search for Teamcenter data.

A *logical block* object represents a solution component. You create logical models by creating a set of logical blocks that define abstract physical architectures for implementing system functions. For example, you create a logical model to define an implementation approach for implementing a specific function in the functional model.



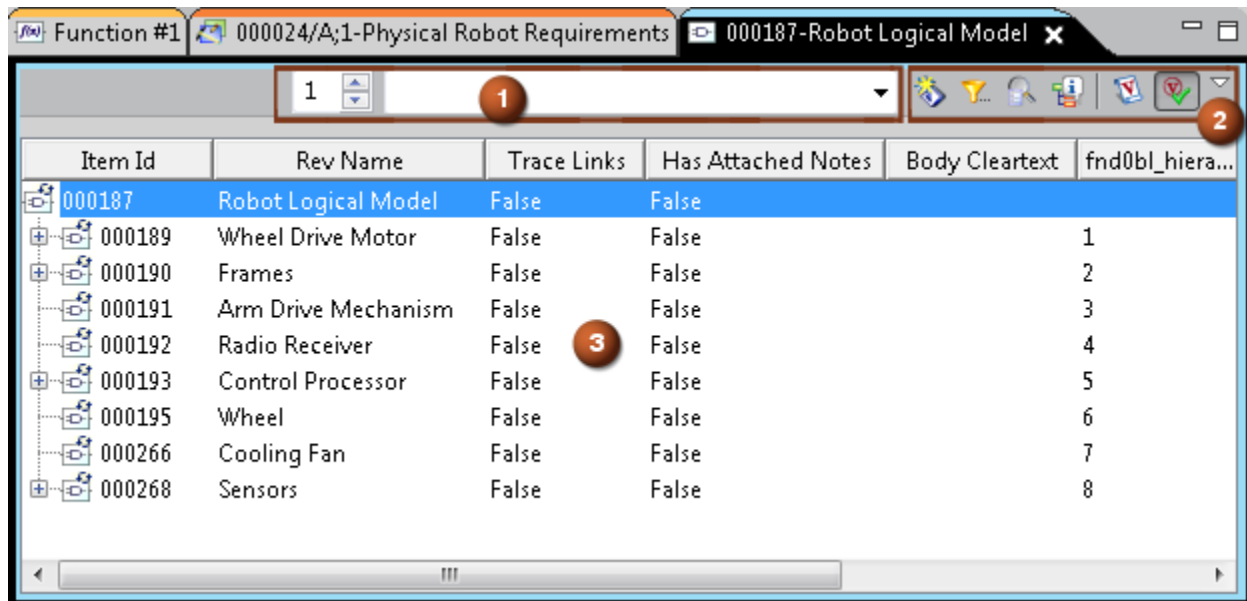
1	Create a Logical Block	Allows you to create a new logical block.
2	Search	Allows you to perform standard and advanced searches for Teamcenter data.

Logical Block view

The **Logical Block** view displays the logical models. A *logical block* object represents a solution component. You create logical models by creating a set of logical blocks that define abstract physical architectures for implementing system functions. For example, you create a logical model to define an implementation approach for implementing a specific function in the functional model.

The **Logical Block** view is displayed when you create a logical model or open an existing logical model.

A tab is created for the logical models that you create or open. You click the tab to display the logical model.



1	Table display controls	The number box allows you to select the number of rows displayed for each object. The default setting is one row per object. Additional rows are added per logical block to view multiple lines of text in the table. The column configuration box allows you to select a saved column configuration from the list. Column configurations change the BOM tree table according to the defined columns in the saved configuration.
2	Logical Block view toolbar	Displays buttons to support configuration of the logical model.
3	Structure pane	Displays the elements of the logical model.

MS Word view

MS Word view user interface

Note:

To use the **MS Word** view, you must enable it to open Word files in a browser.

The **MS Word** view embeds Microsoft Office Word in the Systems Engineering perspective. In the embedded window, you can work with the Word content of requirements and paragraphs, functions, logical blocks, and custom notes.

The **MS Word** view displays rich-text content by default. Rich-text content can include headings, body paragraphs, bulleted and numbered lists, special characters such as equations and symbols, hyperlinks, tables, and graphics.

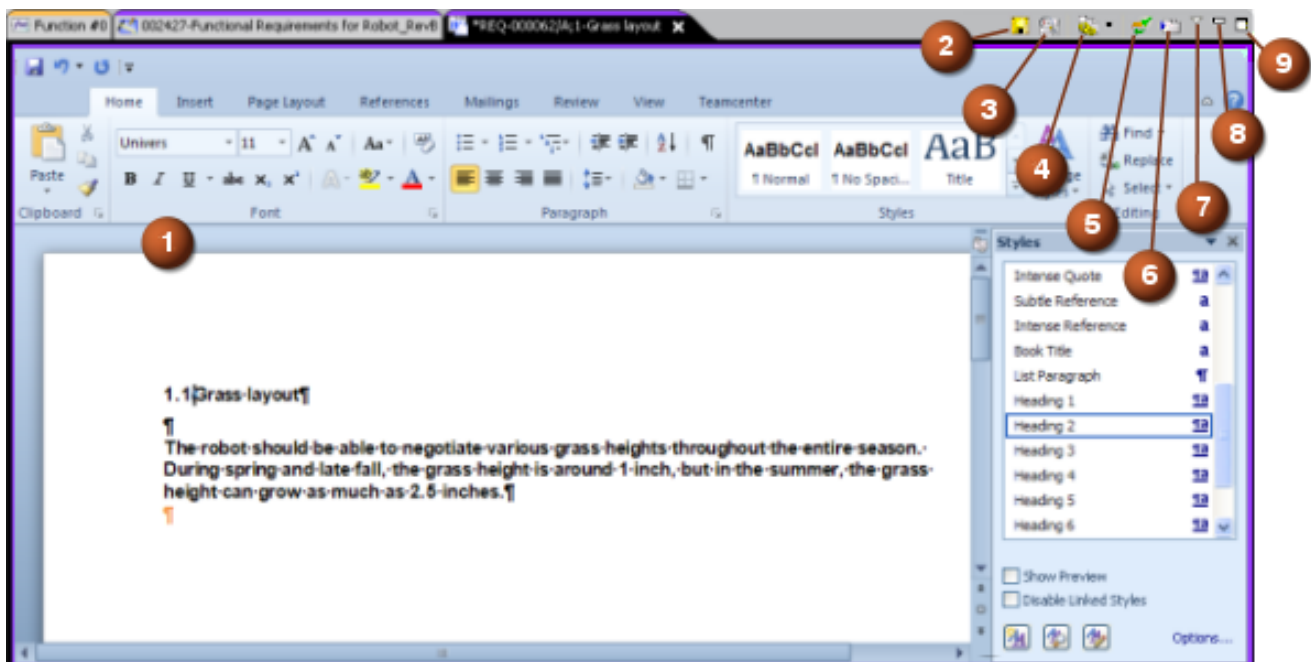
To have the **MS Word** view open a Requirement object embedded in the rich client rather than open in an external instance of Microsoft Word, follow these requirements:

- The rich client bit version must match the bit version of Microsoft Office. Therefore, if you are using a 32-bit rich client, then your Microsoft Office is also 32-bit. Similarly, if you are using a 64-bit rich client, then your Microsoft Office is also 64-bit.
- Do the following:
 1. **Enable the MS Word view to apply document templates.**
 2. **Enable the BrowserFlags Registry entry for Microsoft Word.**
 3. **Switch the MS Word view to plain text.**

You can open multiple instances of the **MS Word** view, with separate views for different selected objects. For each selection, choose **View→Show MS Word View**, or click **MS Word View**  on the Systems Engineering toolbar.

Note:

Because of limitations in third-party software, the toolbar can be disabled in one or more of the multiple instances. If a toolbar does not respond, click anywhere outside that view in the Systems Engineering main window. Then, click inside the view to enable its toolbar.



1	Embedded Word window	Contains the Office ribbon and the document pane, which displays the content of the selected object.
2	Save button	Commits content changes to Teamcenter.
Note: You can also use Word's Save button and features such as Ctrl-S.		
3	Hide/Show MS Office ribbon button	Minimizes or restores the Office ribbon in the embedded Word window.
4	Associate this 'Secondary' view to a different 'Primary' view button	Displays a list of primary views with which the MS Word view can be associated. The input to the MS Word view is synchronized with other open views that use the selected primary view. For example, to synchronize the MS Word view with the selected object in the Trace Links view, do one of the following: • In the list for the MS Word view, select the same primary view that is used by the Trace Links view. • In the list for the Trace Links view, select the same primary view that is used by the MS Word view. • In the list for each view, select None. Then, you can select an object in the Trace Links view and display its Word content simultaneously.
5	Disable response to selections button	Retains the currently displayed content regardless of the selected object.
6	Set input to recent selection button	Displays the content of the last object selected.
7	View Menu button	Displays a menu containing commands that correspond to the three previous buttons.
8	Minimize button	Reduces the MS Word view to the background. Minimizes all other views displayed in the same pane. At the extreme right of the main window, the minimized pane is represented by a vertical toolbar containing the following buttons:

	Restore	Redisplays all previous views in the same pane.
	Function #0	Displays the Quick Create bar for a function object.
	Requirement specification	Redisplays the Requirements view that contains the object's parent structure.
	Word document	Redisplays the previous MS Word view.
9	Maximize button	Enlarges the MS Word view to occupy the entire Systems Engineering main window.

Enable the Microsoft Word View to apply document templates

1. Choose **Edit→Options** to display the **Options** dialog box.
2. Select **Systems Engineering**.
3. Select the **Apply Templates (for embedded editing)** check box.

Note:

When you select the **Apply Templates (for embedded editing)** check box, select the requirement, and open the MS Word View, you can edit the requirement's content only if you have explicitly checked out the requirement or if the **TC_Enable_Implicit_CO** preference is set to **true**.

4. Click **Apply** to commit your edits and keep the **Options** dialog box open, or click **OK** to apply edits and close the dialog box.

Enable the BrowserFlags Registry entry for Microsoft Word

1. Create and open a text file with any text editor.
2. Paste the following content into the text file:

```
Windows Registry Editor 5.00
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Word.Document.8]
```

```
"BrowserFlags"=dword:80000024
```



```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Word.RTF.8]
```

```
"BrowserFlags"=dword:80000024
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Word.Document.12]
```

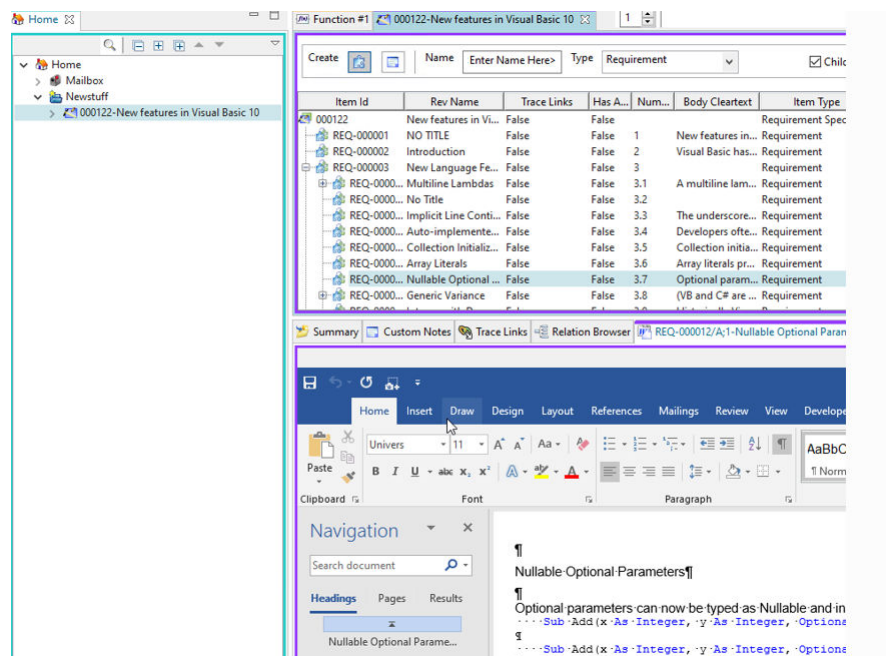
```
"BrowserFlags"=dword:80000024
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Word.DocumentMacroEnabled.12]
```

```
"BrowserFlags"=dword:80000024
```

3. Save the file as **EnableBrowserFlags.txt**, and then close the file.
4. Rename the text file to **EnableBrowserFlags.reg**.
5. Double-click **EnablebrowserFlags.reg** to merge the file into the registry.

When you open a file in the Word View, it opens in the embedded Word View as shown in the following graphic.



Switch the Microsoft Word View to plain text

You can switch the Microsoft Word View to plain text, which loads faster than rich text and speeds up content browsing.

The **Body Cleartext** property is displayed in the **Summary** view, on the **Overview** tab under **Item Properties**. You can insert the **Body Cleartext** property column in the **Requirements** view. Use the **Set the number of rows per line** list to display multiple text lines as necessary.

Caution:

- On Teamcenter servers using the ISO 8859-1 character set (**TC_CHARACTER_ENCODING_SET**), some characters of the rich-text format cannot be translated to plain text. Such characters include, for example, *em* dashes, symbols such as Ohm signs (Ω), and Word's smart quotation marks. These characters are replaced by question marks (?) in plain text and in the Systems Engineering interface, for example, in object names and in certain property values.
- Requirement content includes the text of any Word callout shapes that are inserted in the document. If callout text occurs in the first 2,000 characters, the requirement's **Body Cleartext** property shows the callout text inline with the overall content.

1. Choose **Edit→Options** to display the **Options** dialog box.
2. In the tree on the left, select **Systems Engineering** to display the preferences.
3. In the preferences pane on the right, select the **Browse Clear Text** check box.

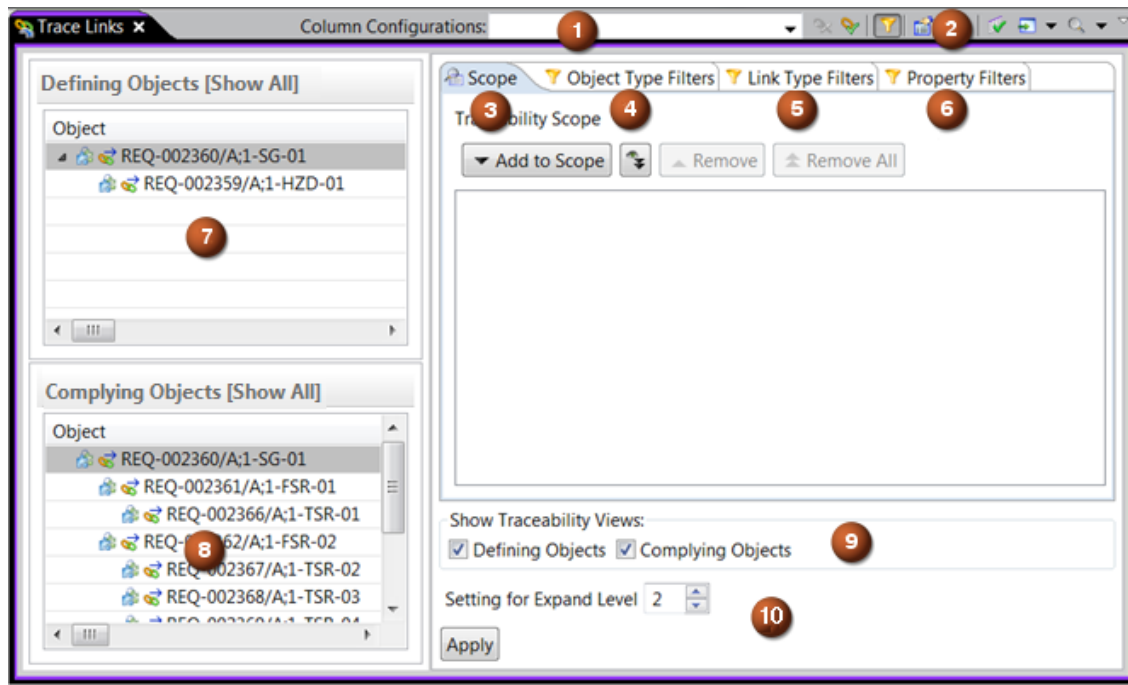
Note:

The **Browse Clear Text** check box is cleared by default.

Trace Links view

You can display traceability information using the **Trace Links** view. You can also display traceability information using the **Traceability** view, which is used by different applications such as Manufacturing Process Planner or Structure Manager. However, working in the **Trace Links** view, you can tightly control the displayed traceability information by setting expansion levels and filtering traceability information by scope, object type, trace link type, and property.

Some tasks can also be performed in the **Traceview Settings** dialog box, which appears when you open the **Trace Links** view by clicking  or choosing **View→Show Trace Links.....**



1 **Column Configurations** Allows you to apply saved column configurations.

2 **Toolbar** Allows you to perform the following:

Button

Purpose



Delete Trace Link

Deletes the selected trace link.



Show/Hide Trace Links

Toggle to display or hide the trace links.



Orphan Report

Allows you to display any of the following:

- Display all unlinked and linked objects.
- Display only linked objects.
- Display only unlinked objects.



Show/Hide Trace Link Criteria

Allows you to display or hide the trace view settings.



Properties

Displays the properties of the selected object.

Button	Purpose
 Export to Excel	Exports the information in the Trace Links view to Excel.
 Configuration	Allows you to change the configuration settings by selecting revision rules, variant rules, and effectivity.
 Go To Object	Allows you to open up the selected objects in any of the following applications: • CAE Manager • Manufacturing Process Planner • Multi-Structure Manager • My Teamcenter • Schedule Manager • Systems Engineering
 Find in views	Allows you select the visible views to display the selected object.
 Disable/Enable response to selections	Enables or disables other views' response to selections in the current view. If enabled, other views respond to selections in the current view. If disabled, other views do not reflect the selection changes in current view
 Associate...	Allows you to associate a secondary view with a different primary view.
 Set input to recent selection	Sets the input for a view to recently selected objects.
 Save Column Configuration	Saves the current column configuration with a name and description.
 Apply Column Configuration	Applies the column configuration you select

Button	Purpose
	from a list of saved column configurations.
 Mark as Publishable	Marks the column configuration as publishable so it is available for other Teamcenter users to use.
 Publish Column Configuration	Publishes the column configuration so it is available for other Teamcenter users to use.

Note:
Click the **View Menu** button  to display the menu commands.

3	Scope tab	Filter traceability information by configured structures.
4	Object Type Filters tab	Filter traceability information by object type.
5	Link Type Filters tab	Filter traceability information by trace link type.
6	Property Filters tab	Filter traceability information by properties.
7	Defining Objects	Displays the defining traceability information.
8	Complying Objects	Displays the complying traceability information.
9	Show Traceability Views	Show or hide defining or complying objects.
10	Setting for expand level	Set number of levels to expand for traceability information.

Traceability view

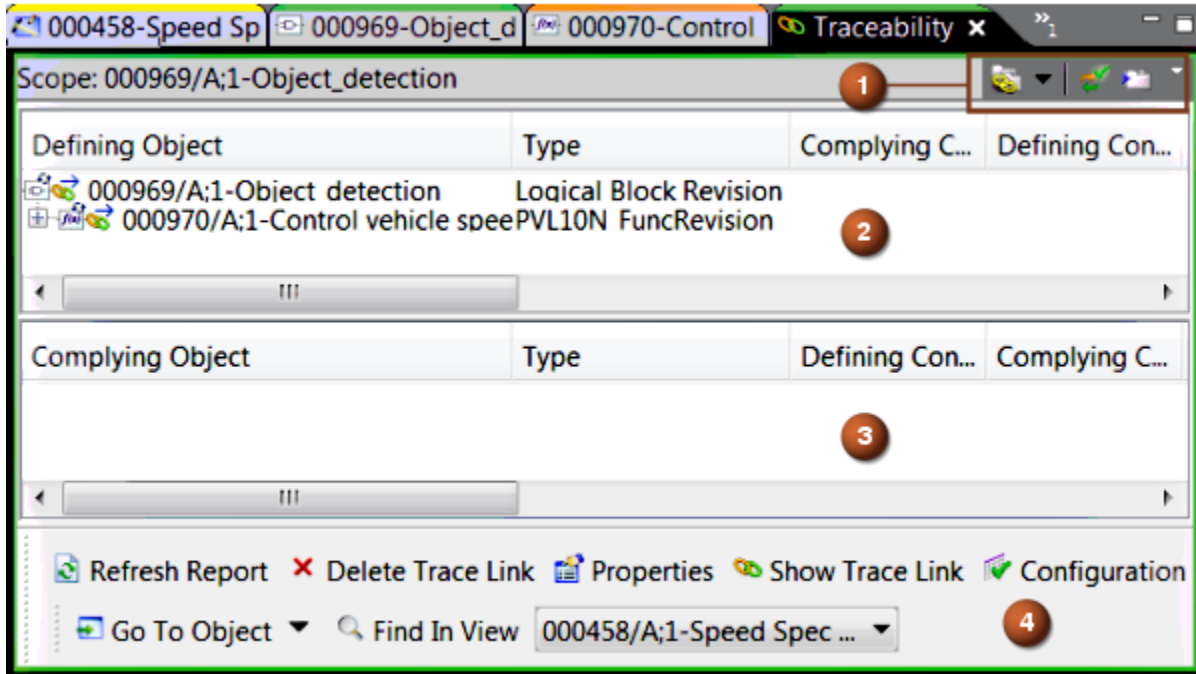
Traceability view interface

The **Traceability** view displays each object that is connected by a trace link to the object selected in the structure pane.

Traceability provides a connection between objects in Teamcenter. For Systems Engineering, this means connecting the structure elements together, for example, creating links between requirements or between logical blocks. You can also create links between functions and requirements or logical blocks. When the requirements, functional design, and logical modeling are complete, users have complete traceability from requirements, to functions, and to logical and physical objects allowing users to see the decision, requirements, and alternatives that lead to a particular design alternative.

You create *trace links* to provide traceability between structure elements and to define which object takes precedence over another. The predecessor, the defining object, is the trace link source. The successor, the complying object, is the target. Trace links are defined for both revisions and occurrences.

You can also display traceability information using the **Trace Links** view, which displays more information and provides additional capabilities such as filtering by configured structure, object type, trace link type, and property.



- 1 **Traceability** view toolbar Displays buttons to support the display of an object's traceability.
- 2 **Defining Object** pane Displays the defining object and its traceability to the complying object.

The **Defining Object** pane shows the defining objects that directly link to the selected object in the defining path. These objects drive the selected object. A plus sign (+) is shown for each defining object that complies with other objects, thus continuing the defining path upstream.
- 3 **Complying Object** pane Displays the complying object and its traceability to the selected object.

The **Complying Object** pane shows the complying objects that directly link to the selected object in the complying path. These objects are driven by the selected object. A plus sign (+) is shown for each complying object that defines

		other objects, therefore continuing the complying path downstream.
4	Traceability view buttons	Displays buttons to support configuration of the traceability.

Each pane contains a table in which the left column displays the linked objects in a hierarchy. This hierarchy shows the sequence of the objects in the defining or complying path from the selected object.

Click a plus sign to see the next objects in a continuing path. In addition, you can view the trace links themselves, which are separate objects, by showing the trace links.

Arrange the Traceability view layout

In the **Traceability** view, you can arrange the **Defining Object** and **Complying Object** panes horizontally or vertically.

- For the horizontal layout, select the **Left-Right View** option.

The **Traceability** view displays the **Defining Object** pane on the left and the **Complying Object** pane on the right.

Tip:

You can also do the following to arrange the view horizontally:

- Use the **Options** dialog box.
 1. To display the dialog box, choose **Edit→Options**.
 2. In the left pane, select the **Systems Engineering** option.
 3. In the right pane, select the **Left-Right Trace View** check box.
- Set the **SE_Trace_Report_Layout** preference value to **true**.

- For the vertical layout (the default layout), clear the **Left-Right View** option.

The **Traceability** view displays the **Defining Object** pane at the top and the **Complying Object** pane at the bottom.

Tip:

You can also do the following to arrange the view vertically:

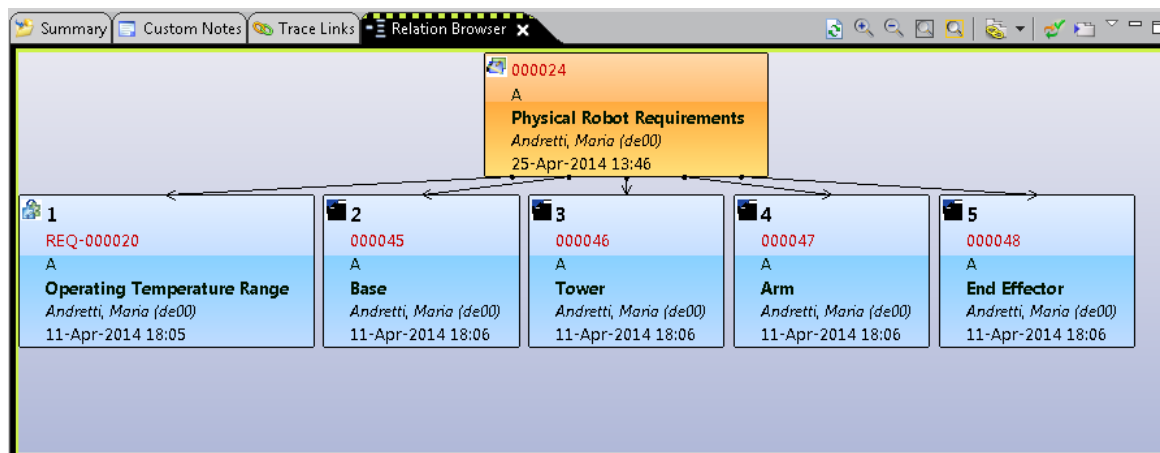
- Use the **Options** dialog box.
 1. To display the dialog box, choose **Edit→Options**.
 2. In the left pane, select the **Systems Engineering** option.
 3. In the right pane, clear the **Left-Right Trace View** check box.
- Set the **SE_Trace_Report_Layout** preference value to **false**.

Relation Browser view

The **Relation Browser** view provides the following features and functionality:

- When you select an object in the navigation pane, the **Relation Browser** view displays the newly selected object as the root object, and the **Properties** view displays properties of the object.
- When you open an object in Systems Engineering, and activate the **Relation Browser** view, the object you select appears in the **Relation Browser** view as the root object.

You can select any node to browse further related objects one level at a time.



- Zoom in , zoom out , and fit content  buttons are available.
- Choose a context from the **Relation Browser** view menu .

Menu command	Description
Contexts→Default	View structure relations in the Relation Browser view. In this context, defined

Menu command	Description
	structure relations display when you show successors or predecessors. This is the default view context.
Contexts→Traceability	View trace links in the Relation Browser view. In this context, defined trace links appear when you show successors.
Contexts→Genealogy	When a requirement object is duplicated, Teamcenter remembers its source. In this context, you can view the source or duplicate objects, if any, related to a requirement object selected in the Relation Browser view. To view the source of the selected object, right-click and choose Hide/Show Predecessors. To view the duplicate or duplicates of the selected object, right-click and choose Hide/Show Successors. Note: This feature is enabled by setting IMAN_based_on as a shown relation for the requirement revision item type. 1. In the toolbar, choose Edit→Options→General→Item Revision. 2. In the Options dialog box, General tab: a. Select Requirement Revision in the Item Revision type selected box. b. Select Specifications in the Default paste relation box. c. Add the IMAN_based_on relation type to the Shown Relations list. d. Click OK.

Note:

Context display is controlled by the **TC_RelationBrowser_Context** preference.

To view or edit preference information, use the rich client **Edit→Options** dialog box.

- Choose a layout orientation from the **Relation Browser** view menu ▼.

Menu command	Description
Hierarchical →  Top-to-Bottom	Creates a graph beginning with a top node and flowing to the bottom.
Note: The structure in the preceding example graphic uses the Top-to-Bottom orientation. The requirement specification appears at the top-level and the associated requirements appear at a lower level of the structure.	
Hierarchical →  Right-to-Left	Creates a graph beginning with node at the right and flowing to the left.
Hierarchical →  Left-to-Right	Creates a graph beginning with a node at the left and flowing to the right.
Hierarchical →  Bottom-to-Top	Creates a graph beginning with a node at the bottom and flowing to the top.
Hierarchical →  Incremental	Creates a graph beginning with a top node in the upper-left corner and flowing to the bottom and to the right.
 Circular	Displays group and tree structures within a network in a ring and star pattern.
 Orthographic	Creates compact graphs with no overlaps, few crossings, and few bends. It is most appropriate for medium-sized sparse graphs.
 Organic	Creates a layout with the nodes spaced as far away from each other as possible, still while minimizing this distance. It is most appropriate for undirected, complex graphs.
 Balloon	Creates a layout with a central node and child nodes extending from this in a radial fashion.

Menu command**Description**

It is most appropriate for large, hierarchical structures.

- You can select any node to browse further related objects one level at a time.
- Each node has a type symbol and object string to represent the component on the graph. If any node has an associated 2D preview image, the preview image can be displayed as a thumbnail image along with the text string in the **Relation Browser** layout and is also displayed in the **Image Preview** view.
- When you right-click a node in the graph, the shortcut menu displays the same commands as when you select a node on the navigation pane tree, based on the type of object.
- Each node can have predecessors and successors, based on the application with which the object is associated. You can select an object and use the shortcut menu to expand or collapse predecessors and successors. For example, an application can show predecessors as where-referenced objects and can show successors by showing default children context.

Traceability Matrix view

The **Traceability Matrix** view displays the trace links in a matrix format. A traceability matrix is used to analyze trace link allocations between two selected structures, for example, a requirement specification and a functional model.

	Total	000199/A; 1-Robot Func...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...	REQ-00199/A; 1-Park Requ...
000199/A; 1-Robot Func...	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
000197/A; 1-Robot Func...	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001919/A; 1-Park Requ...	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001920/A; 1-Park Dime...	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001921/A; 1-Park Layo...	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001922/A; 1-Grass layout	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001923/A; 1-Concrete ..	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001924/A; 1-Playground	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001925/A; 1-Fountain	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001926/A; 1-Ramp	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001927/A; 1-Flower Ga..	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001928/A; 1-Robot Sta..	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001929/A; 1-Recycling...	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001930/A; 1-Can Cleanup	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
REQ-001931/A; 1-Can sifting	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

1	Source heading	Displays the name of the structure selected as the source in the Show Traceability Matrix dialog box.
2	Target heading	Displays the name of the structure selected as the target in the Show Traceability Matrix dialog box.

Note:

Source and target structures can be the same to show traceability within the same structure.

3	Total column	Displays the number of trace links for each source element.
4	Total row	Displays the number of trace links for each target element.
5	Leftmost cells in rows	Display the names of source elements.

Tip:

To open a requirement in the **MS Word** view, right-click the name and choose **Open bodytext**.

6	Column headings	Display the names of target elements.

Tip:

To open a requirement in the **MS Word** view, right-click the heading and choose **Open bodytext**.

7	Status flag cell	Displays one of the following: • A solid red fill if one or more source or target elements are not allocated to the other structure. • A solid green fill if each source and target element has at least one trace link to the other structure.
8	Body cells	Display the number of trace links between a pair of source and target elements.

Note:

To keep specific cells visible while you scroll through the **Traceability Matrix** view, certain rows and columns are locked by default.

For horizontal scrolling, the following cells remain in their current positions:

- Cells in the **Total** column (the status flag and the number of trace links for individual source elements)
- The leftmost cells in rows (the names of source elements)

For vertical scrolling, the following items remain in their current positions:

- Cells in the **Total** row (the status flag and the number of trace links for individual target elements)
- Column headings (the names of target elements)

Tip:

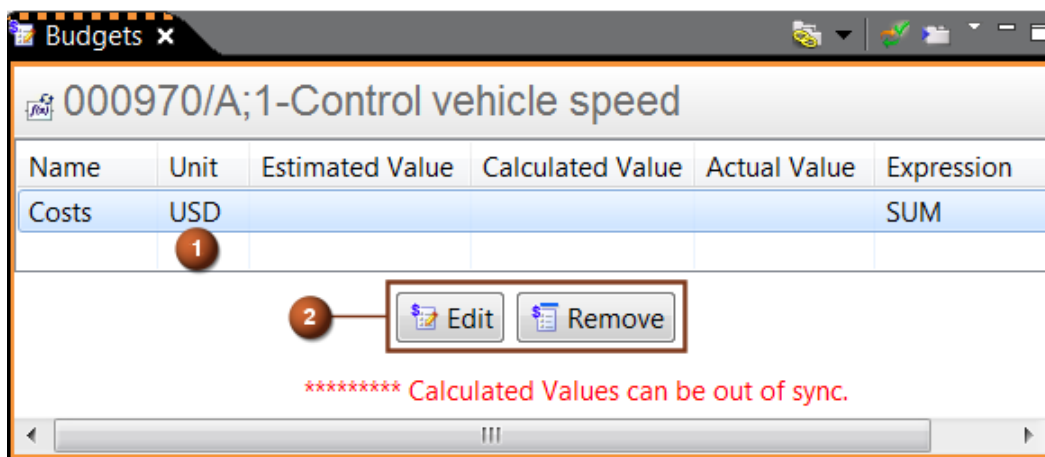
- You can unlock cells by selecting the row or column and pressing Shift+U.
- To relock the cells, select the row or column and press Shift+F.

Budgets view

In the **Budgets** view, you can monitor the performance of downstream targets by applying technical measurements, such as Technical Performance Measures (TPMs), to structures and their members. Teamcenter Systems Engineering implements technical measurements through *budgets*.

Budgets work through Microsoft Office Excel, in which you define and quantify parameters like *height*, *weight*, or *power*. With parameter values expressed in appropriate units (for example, *inches*, *pounds*, or *watts*), aggregates can be reported upstream to the system level.

You use Teamcenter live Excel to apply new budgets and initial values, and to edit existing values, for individual structure elements.



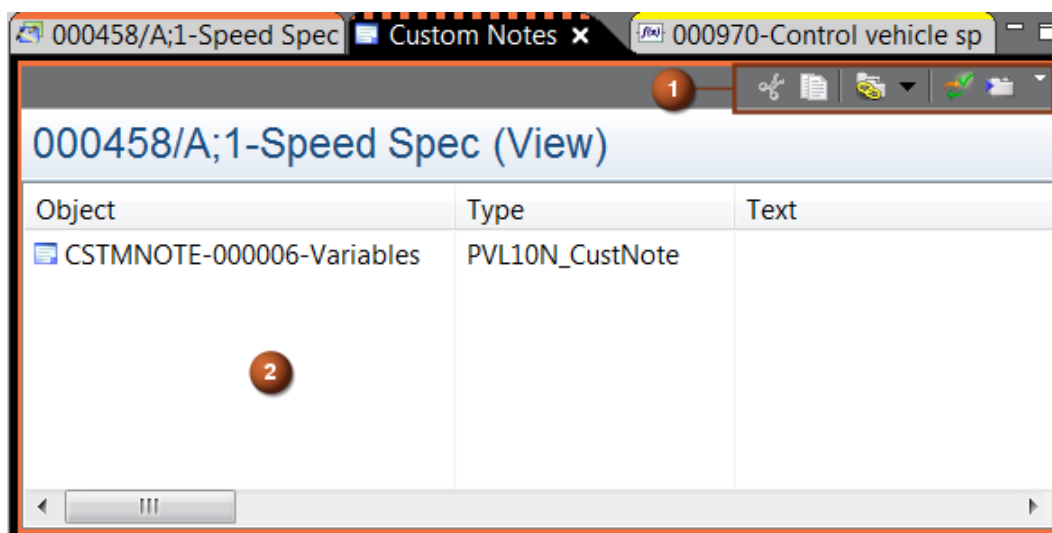
1	Budgets view data	Displays the actual values of each budget for the selected structure element.
2	Budgets view buttons	Allow you to edit a selected budget's values in live Excel and to remove the structure element from participation in the budget.

Custom Notes view

Notes are used to capture issues, actions, questions, or other ad hoc types of information. They act much like *yellow sticky notes* on an object. For example, as issues arise a user can attach a note to the object to capture or document the issue.

Notes currently come in two types (although users can define other subtypes): custom notes and standard notes.

- Custom notes contain any content and are edited with Microsoft Word.
- Standard notes contain parameters and are typically the types that appear as standard notes on drawings.



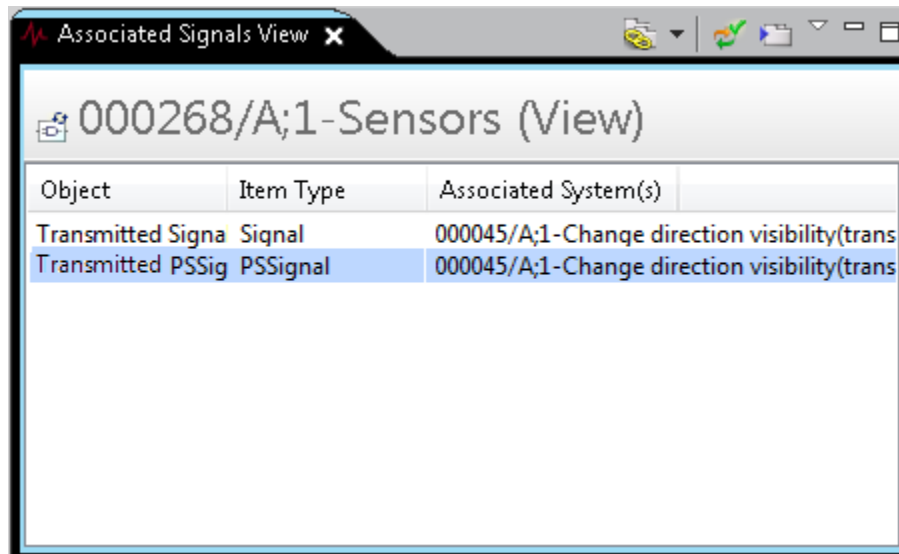
1	Custom Notes view buttons	Displays the buttons for managing your custom notes.
2	Custom Notes view data	Displays all the notes created for the selected object.

Associated Signals view

Signals represent information or messages transmitted between interfaces. Signals can be instantiated in a product structure and associated with product constituents such as connections, terminals, and

connectors to capture various relationships, for example, the transmitter of the signal and the receiver of the signal. You can also define custom signal types to meet specific needs.

The **Associated Signals** view displays information about the associated signal related to the selected interface.



The following information is displayed in the view.

Column name	Description
Object	Displays the signals associated with the selected object.
Item Type	Displays the item type of the signal.
Associated System(s)	Displays the interface or connection associated with the signal.

Visio Diagram view

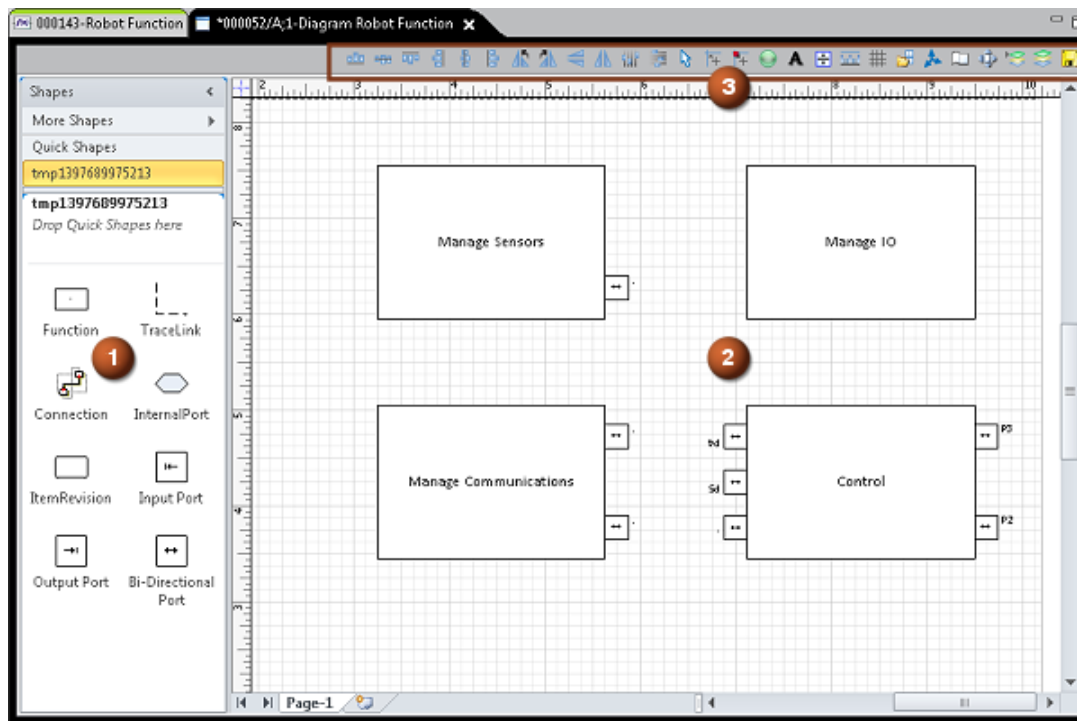
Note:

Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

The **Visio Diagram** view is an embedded Visio window in the **Systems Engineering** perspective. In this view, diagrams of functional and logical structures show how a product's subsystems relate to one another.

The shapes in a diagram represent elements in the structure below a selected *diagram owner*. The diagram owner is the structure's root object in the **Function** or **Logical Block** view.

Teamcenter stencils contain master shapes that are mapped to Teamcenter object types. As you drop the shapes in the drawing pane, corresponding objects are created in the underlying structure.



1	Shapes pane	Contains the master shapes from the selected stencil.
2	Drawing pane	Contains the shapes that represent the structure elements below the diagram owner.
Note: The diagram owner does not appear in the diagram.		
3	Teamcenter custom toolbar	Contains buttons for access to drawing features of Visio.

Note:

Systems Engineering embeds the Visio window through the Microsoft Visio ActiveX control. Due to certain aspects of the control, the **Visio Diagram** view does not contain the Office ribbon.

Office ribbon features are available through shortcut menus in the view. For example, you can select shape editing features by right-clicking a shape and choosing **Format** from the shortcut menu.

Quick Create bar

Quick Create bar user interface

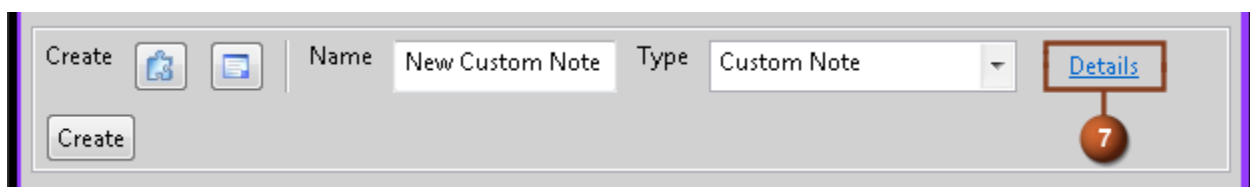
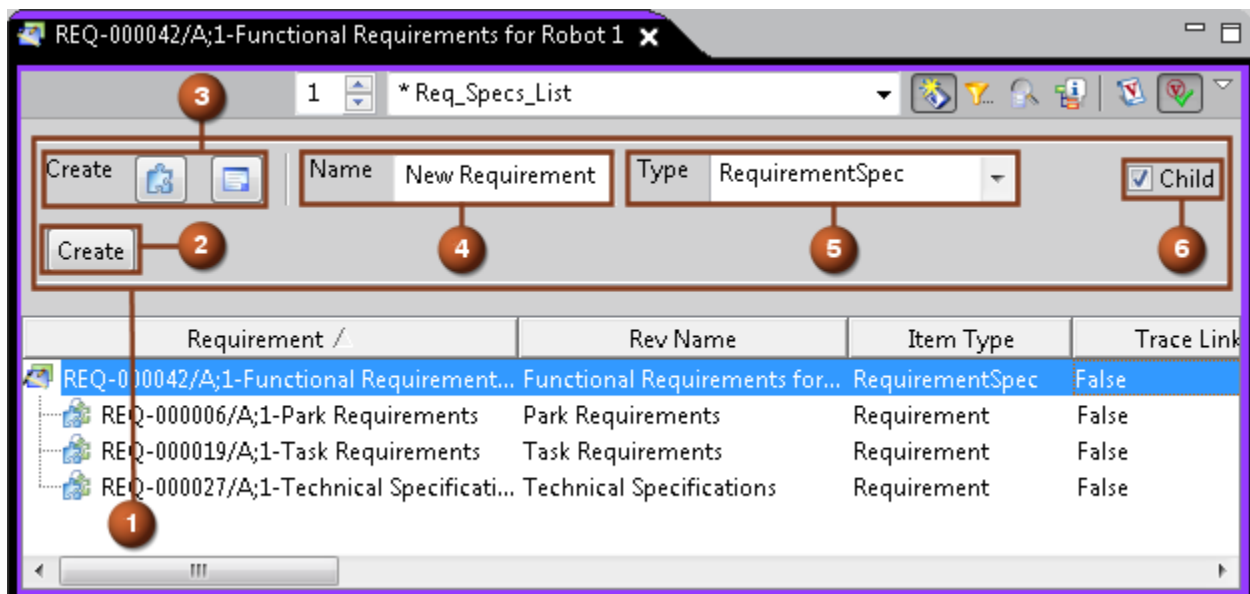
When you enable the Quick Create bar, it is available in all **Systems Engineering** views.

The Quick Create bar provides all of the necessary features for creating custom notes, requirements, paragraphs, functions, and other blocks whose object types meet one of the following conditions:

- Additional property values are not required.
- All required property values are supplied by default.

Note:

The **ItemID** and **ItemRevID** property values are assigned automatically.



1	Quick Create bar	Displayed below the object tab when you enable the Quick Create bar.
2	Create	Starts the creation process, and, if necessary, displays the Enter Required Attributes wizard.
3	Object type selection buttons	 Enables you to create a custom note. Note: In order for you to create a custom note, the TcNotesAllowedType_item_type preference must include the Fnd0CustomNote value.  Enables you to create all other Systems Engineering objects available for your installation.
4	Name	Specifies the name of the Systems Engineering object that you want to create. The Name box activates the other elements when it contains an object name.
5	Type	Specifies the type of Systems Engineering object that you want to create. The Type list shows either Custom Note or the requirement object types and subtypes for your Teamcenter installation. Note: The Type list displays subtypes if they are added as values of the TCAssowedChildTypes_item_type preference.
6	Child	When creating an object other than a custom note, determines whether the new object is a child or a sibling of the object selected in the structure pane.
7	Details	When creating a custom note, enables you to type the text for the note.

For creating such objects, the Quick Create bar saves time because you can bypass the New Requirement wizard, for example, when building requirement specification structures.

Note:

You can also use the Quick Create bar for object types that have required properties without default values. In these cases, the Quick Create bar automatically launches a wizard where you enter the required values.

Note:

The elements are unavailable when Systems Engineering displays the initial welcome window.

Enable or disable the Quick Create bar

1. Choose **Edit→Options** to display the **Options** dialog box.
2. In the tree on the left, select **Systems Engineering** to display the Systems Engineering options.
3. Select or clear the **Quick Create Panel** check box to enable or disable the Quick Create bar.
4. Click **OK** or **Apply**.

Use the Quick Create bar**Note:**

In order for you to create a custom note, the **TcNotesAllowedType_item_type** preference must include the **Fnd0CustomNote** value.

1. In the Systems Engineering views, choose **Quick Create Panel** from the **View Menu** to display the Quick Create bar for the view.
2. Click one of the following for the type of object you want to create:



Custom Note



All other Systems Engineering objects available for your installation

3. Enter the new object name in the **Name** box.
4. Do one of the following:
 - To create a custom note, click **Details**, and type the note text in the box.
 - To create an object other than a custom note:
 - a. Select the object type in the **Type** list.
 - b. To set the new object's level in the structure, select the target object, and then do one of the following:
 - Clear the **Child** check box to create a sibling of the target object.

- Select the **Child** check box to create a child of the target object.

Note:

The **Child** check box is selected by default.

5. Click **Create**.

Note:

If the Enter Required Attributes wizard is displayed, follow the instructions in the subsequent windows. Click **Finish** to start the process.

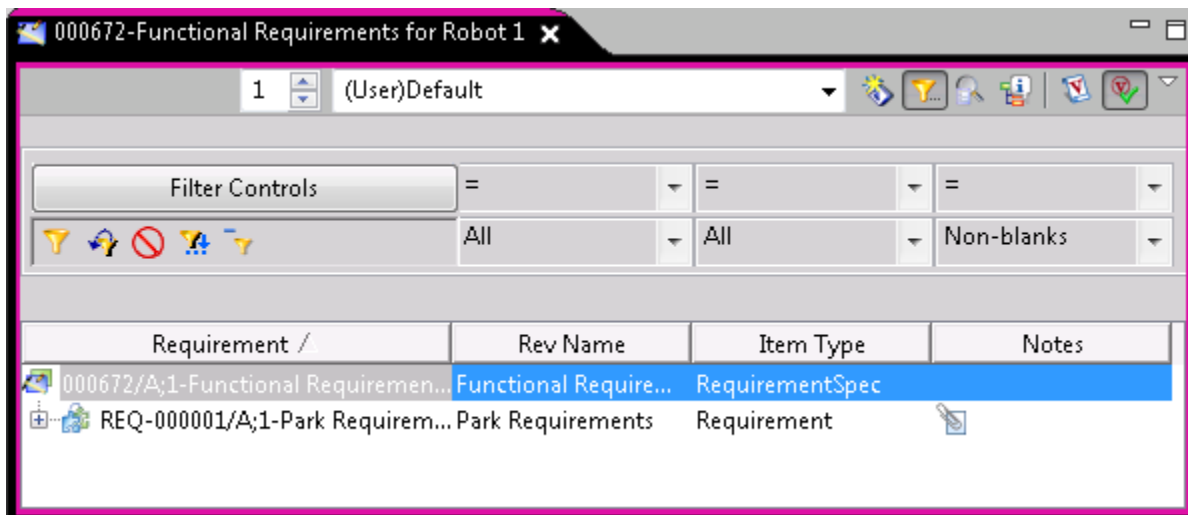
The new object is highlighted at the specified level in the structure pane.

Filtering column properties

About filtering column properties

You can use filter controls to set operators that filter the data in the property columns that have relevance to you. When you enable filter controls, they are available on all Systems Engineering views.

The following example shows the results of filtering all the requirements that have custom notes by setting the filtering column above the **Notes** column to **Non-blanks**.



Filter criteria is set for each individual property column. The combined criteria define the filter applied to the table data.

- The rows for property data that meet the filtering criteria are included in the table display.
- The rows for property data that do not meet the filtering criteria are removed from the table display.

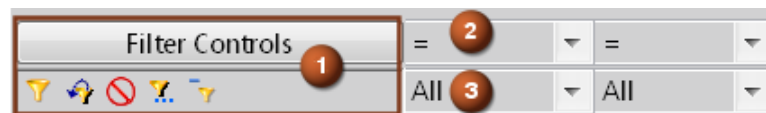
Filtered rows are not hidden in the results of processes such as structure searches and accountability checks. By revealing filtered rows in their results, such processes void the filter settings.

Use the filter controls

1. Do one of the following:

- Click **Show Filters**  from the toolbar.
- Click the **View Menu** button  from the view and choose **Show Filters** .

The filter controls appear directly above the column headings in the table display.



2. Use the following filter controls to filter the data in the columns.

1 Filter Controls buttons

- Use the **Filter Controls** buttons to apply and clear the filter settings:

-  Filter the data.

Applies the filter settings to the table data. Filtering the data determines which rows appear in the table, based on the filter criteria.

-  Unfilter the data.

Cancels the filter operation and displays all rows in the table data. This action does not clear the settings in the filter controls.

-  Clear all filters.

Clears the filter controls back to the original settings.

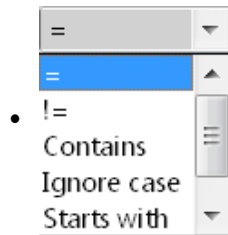
-  Hide filter operator options.

Removes the row of operators from the **Filter Controls** display.

-  Hide filters.

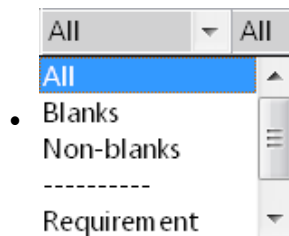
Removes the **Filter Controls** options from the Systems Engineering view.

2 Filter operators



Select the conditional operator for filtering the property column data.

3 Filter criteria



Select the criteria value for the property column data, to which the conditional operator is applied.

Systems Engineering menus

File menu

Command	Description
New→Requirement Specification	Creates an empty requirement specification, which is displayed as the root object on a new tab in the structure pane. You build the requirement specification structure by creating individual requirements and paragraphs inside a specification.
New→Requirement	Creates a requirement within a requirement specification structure. The requirement can be created at the primary level, directly below the requirement specification or, the requirement can be created as a child of any other item in the structure. You can also use this command to create the following types of requirements: <i>Validation requirements</i> can be used to check NX parts against the specifications of selected requirements. Validation requirement content typically includes measurable values that can be validated in the actual design with model validation tools.

Command	Description
	<i>Design requirements</i> can contain variables and details to be used as the input parameters to a CAD model. These variables and details can be used to drive design parameters in the actual mechanical design.
New→Paragraph	Creates a paragraph within a requirement specification structure. The paragraph can be created at the primary level, directly below the requirement specification. Or, the paragraph can be created as a child of any other item in the structure.
New→Custom Note	Creates a custom note on a selected item. The custom note is created as an attachment to the selected object.
New→Derived Requirement	Creates a requirement derived from a parent requirement. The derived requirement is a child of the selected requirements and starts with the existing content of the parent requirements. The process also creates a defining requirement from the parent requirement to the new derived requirement.
New→Blocks→Function	Creates a function object in the Function view.
New→Blocks→Logical Block	Creates a logical block object in the Logical Block view.
New→Blocks→Custom Block	Creates an object of the SE Block type in the Function or Logical Block view. These objects can represent generic block types such as process blocks and system blocks. Custom subtypes can be created.
New→Connection	Creates a new connection for a function or logical block. The connection uses interface objects defined on the structure elements to make the physical connection.
New→Interface	Creates a new interface for a function or logical block. The interface object allows for a connection to another structure element.
New→Signal	Creates a new signal for a function or logical block. The signal uses connection objects defined on the structure elements to represent the communication between structure elements.
New→Add Signal from Library	Adds a signal from the data dictionary to the structure.
New→Diagram	Creates a Microsoft Visio diagram representing the structure elements immediately below the selected function or logical block. The diagram is displayed in the Visio Diagram view.

Command	Description
New→Item from Template	Creates a new item using an existing item as a template allowing users to start a new object from an existing one. The selected template can be a requirement specification with an existing structure, a parent requirement or paragraph, or a single requirement or paragraph without a lower level structure.
New→Workflow Process	Submit the selected object to a Teamcenter workflow.
Open in MSWord Application	Opens a requirement object in Microsoft Office Word.
Import Spec	Imports an entire requirement specification structure from a Word document.
Collaboration Context→Open Collaboration Context	Opens a collaboration context.
Collaboration Context→Open Structure Context	Opens a structure context.
Collaboration Context→Save as New Collaboration Context	Saves the selected structure context as a new collaboration context. You can assign a name and description for the new collaboration context.
Collaboration Context→Save as New Structure Context	Saves the currently selected structure as a new structure context. The original structure view is replaced with the new structure context view. In the Collaboration Context Tree view, the original structure is removed and the new structure context is displayed.
Collaboration Context→Save Configuration	Saves the configuration of the currently selected structure.
Collaboration Context→Save as New Configuration Context	Saves the current configuration of the structure pane to a new configuration context and adds it to the structure context. This menu command is available only if a structure context is opened.
Collaboration Context→Apply Configuration Context	Applies the configuration of the selected configuration context to the currently opened structure.
Collaboration Context→Restore Configuration	Resets a structure view's configuration back to its previously saved configuration context. This menu command is available only if a structure context is opened.
Save As	Creates a new object of the same type as the selected source object.
Revise	Creates a new revision of the selected object.
Duplicate	Clones the selected requirement structure. The selected line and everything below it will be copied into the clone. Datasets and attachments may be copied to the clone, depending on Business Modeler IDE deep copy rules.

Command	Description
Close	Exits Systems Engineering without ending the Teamcenter session.
Unload All	Unloads all structures currently open in Systems Engineering.
Exit	Ends the Teamcenter session.

Edit menu

Command	Description
Copy Append	Copies the selected object to the Teamcenter clipboard with the intention of appending the copied object to the destination object, and retaining the current contents of clipboard.
Paste As New	Pastes a duplicate of the object from the Teamcenter clipboard as a new object and a child of one selected object.
Copy URL	Captures the URL to the selected object in Systems Engineering. This URL can be embedded in e-mails or used in any application where the object and its context are required.
Move Up	Moves the selected structure element up one position in the structure.
Move Down	Moves the selected structure element down one position in the structure.
Promote One Level	Moves the selected structure element up one level in the structure. The selected structure element becomes a sibling of its parent structure element.
Demote One Level	Moves the selected structure element down one level in the structure. The selected structure element becomes a child of the immediately preceding sibling at the previous level.
Attach Requirements/ Notes→Replace Parametric Requirement Revision	Attaches a parametric requirement revision to an item or item revision.
Attach Requirements/ Notes→Custom Note	Attaches a custom note to an item or item revision.
Edit Number	Rearranges the sequence of the selected object in a structure, according to the line number you specify.
Replace	Replaces a selected component or all components of the selected item revision that exist in the parent structure.
Insert Level	Creates an object and inserts it in the current structure as a new level below the selected line.

Command	Description
Create Child	Creates a child of the selected object as the same object type.
Remove	Removes the selected line from the structure.
Toggle In-Context Mode	Enables or disables editing in context mode, allowing creation and edit of absolute occurrences for the selected structure.
Toggle Precise / Imprecise	Changes the precision of the selected structure line.
Variant Condition	Displays variant equations that describe variant conditions on a structure.
Select Below	Selects all the lower lines or only the lowest lines.
Change Ownership	Changes ownership of the selected object or objects.
Add to Favorites	Adds the selected object or objects to your Favorites list in the navigation pane.
User Setting	Changes your group, role or volume assignments, also your application logging options.
Options	Displays the Options dialog box, where you can set Systems Engineering preferences to configure trace link mode and to configure the MS Word pane for requirement viewing and editing. You can also change your user interface settings for all other Teamcenter applications.

View menu

Command	Description
Show Trace Links	Displays the Traceview Settings dialog box, from which you can set filters to limit the display of trace links associated with the selected object. When you click OK in the Traceview Settings dialog box, the Trace Links view appears. The same filtering capabilities are available in the Trace Links view.
Show Traceability Matrix	Displays the Show Traceability Matrix dialog box, from which you can select the trace link types you want to filter. When you click OK , the Traceability Matrix view appears.
Show MS Word View	Displays the MS Word view, where you can view and edit requirement content in an embedded Word window.
Show Custom Notes Manager	Displays the Custom Notes view, where you can view note content of a selected object.
Show Requirements Expression	Displays the Requirement Expression view, where you can view and edit the parameters for validation and design requirements.

Command	Description
Show Budget	Displays the Budgets view, where you can monitor the performance of downstream targets by applying technical measurements.
Show Associated Signals	Displays the Associated Signals view, where you can view information about the signals associated with the selected object.
Show Diagram Explorer	Displays the Diagram Explorer view.
Systems Engineering	Displays the Requirements view.
Cross Probing Mode	Initiates a trace of a selected object's downstream and upstream relationships across structures in different Teamcenter applications.
Occurrence Notes	Displays the occurrence notes for the selected object.
Signal Explorer→Show Source	Displays the logical block origination of the signal.
Signal Explorer→Show Target	Displays the logical block receiver of the signal.
Signal Explorer→Show Transmitter(s)	Displays the logical blocks that originate signals.
Signal Explorer→Show Transmitted	Displays the logical blocks that have sent signals.
Signal Explorer→Show Received	Displays the logical blocks that have received signals.
Refresh Window	Reads information from the database and updates the information displayed in the application.
Refresh Current Structure	Updates information in the currently open and active structure.
Expand Options→Expand	Expands the substructure immediately below the selected lines
Expand Options→Expand Below	Expands the complete substructure below the selected lines
Expand Options→Expand Below...	Expands the substructure below the selected lines to a user-selected level. You can also collapse an expanded substructure when you choose this command (hide certain child lines).
Expand Options→Expand To Type	Expands the substructure below the selected lines to a user-selected type.
Collapse Below	Hides the complete substructure below the selected object.

Tools menu

Command	Description
Trace Link→Start Trace Link	Initiates a trace link from the selected objects. If multiple defining objects are selected, a separate trace link is created from each object.
Trace Link→End Trace Link	Completes a trace link to the selected object. If multiple complying objects are selected, a separate trace link is created to each object. This menu command assigns the default subtype, Trace Link , to the new trace link.
Trace Link→End Trace Link...	Completes a trace link to the selected complying object of a given subtype. If multiple complying objects are selected, a separate trace link is created to each object. This button allows you to assign a custom subtype to the new trace link.
Microsoft Office Templates→Create Specification Template	Displays the New Business Object dialog box, where you can create a specification template. A specification template is a Microsoft Office Word document that controls the content and format of the data that is exported to Word.
Microsoft Office Templates→Create Object Template	Displays the New Business Object dialog box, where you can create an object template. An object template controls the content and format when you export Teamcenter objects to Word.
Microsoft Office Templates→Create Excel Template	Displays the New Business Object dialog box, where you can create a Microsoft Office Excel template. An Excel template defines the property data and output format used when you export Teamcenter objects to Excel.
Microsoft Office Templates→Override Word Templates	Displays the Override Word templates dialog box, where you can configure the specification template to override the default for selected business objects.
Budgets→Create Budget Definition	Displays the New Budget Definition dialog box, where you can create a budget definition.
Budgets→Edit Budget	Displays the Edit Budget dialog box, where you can select the Excel template for a selected budget definition.
Diagram→Open Diagram	Opens a Visio diagram.
Diagram→Create Template	Displays the New Diagram Template dialog box, where you can create a new Visio diagram template.
Diagram→Edit Template	Displays the Edit Diagram Template dialog box, where you can edit a Visio diagram template.
Connection Manager→Connect	Creates a connection between a network port and one or more network connections in a functional or logical model structure.

Command	Description
Connection Manager→Disconnect	Removes a connection between a network port and one or more network connections in a functional or logical model structure.
Connection Manager→Show Connected Lines	Displays the Show Connected Lines dialog box for the selected function or logical block.
Signal Manager→Associate Signal To Source	Associates a signal to the source, transmitter, or origin of the signal.
Signal Manager→Associate Signal To Target	Associates a signal to the target, receiver, or destination of the signal.
Signal Manager→Associate Signal To Transmitter	Associates a signal to a transmitter, or connection.
Signal Manager→Remove Signal Source	Removes a signal from the source.
Signal Manager→Remove Signal Target	Removes the signal from the target.
Signal Manager→Remove Signal Transmitter	Removes a signal from a transmitter, or connection.
Import→Objects	Displays the Import dialog box, where you select an import file containing object data in the Teamcenter format or the STEP format.
Import→From PLMXML	Creates new objects in Teamcenter from a previously exported PLM XML file.
Import→Templates→Specification Template	Available only in the Home view. Displays the Import Specification Template dialog box, where you select the template to import. Used in exporting object data to Microsoft Office Word, a specification template defines the Word styles that are applied to the data in the output document.
Import→Templates→Object Template	Available only in the Home view. Displays the Import Object Template dialog box, where you select the template to import. Used in exporting object data to Microsoft Office Word, an object template designates the properties that are exported for the selected objects.
Import→Templates→Excel Template	Available only in the Home view. Displays the Import Excel Template dialog box, where you select the template to import. Used in exporting object data to Microsoft Office Excel, an Excel template specifies which properties are exported and defines property filtering rules for the selected objects.
Export→Objects	In the Requirements , Function , and Logical Block views, displays the Application Interface Export dialog box. By exporting the

Command	Description
	selected objects to an application interface, you can store one set of parameters for sharing PLM XML data repeatedly with other sites.
	In the Home view, displays the Export dialog box. You can export the selected objects to an application interface or an output file in the Teamcenter or STEP format.
Export→To PLMXML	Generates a PLM XML file containing properties of the specified objects.
Export→Objects To Excel	Generates a Microsoft Office Excel file containing the objects and properties of the specified objects. You can create a static Excel file, or you can create an interactive live Excel file that is synchronized with the Teamcenter database. Changes made to cells in the spreadsheet are applied to the database.
Export→Objects To Word	Generates a Microsoft Office Word file containing the objects and properties of the specified objects. You can create a static Word file, or you can create an interactive live Word file that is synchronized with the Teamcenter database. Edits in a live Word document are applied to the database at exit and file save.
Export→To Remote Site via Global Services	Available only in the Home view. Displays the Remote Export to Global Services dialog box, where you select the destination site to receive the exported data.
Compare→Structure Compare→BOM Compare	Displays the BOM Compare Structure dialog box. You can compare a source structure and a target structure at the lowest level, at a single level, and at multiple levels. You can also generate a comparison report, which opens in the BOM Compare Report view.
Compare→Structure Compare→Clear Compare	Closes the BOM Compare Report view and removes the highlighting in the structures.
Compare→Document Compare→Start Compare	Sets a requirement or specification as the source for comparison with a subsequently selected target requirement or specification.
Compare→Document Compare→End Compare	Sets a requirement or specification as the target for comparison with a previously selected source requirement or specification.
Compare→Document Compare→Compare With Prior	Initiates a comparison of the selected requirement or specification with its preceding version.
Compare→Accountability Check→Accountability Check	Displays the Accountability Check dialog box, where you can compare two structures to ensure all lines in the source structure are consumed in the target structure.

Command	Description
Compare→Accountability Check→Clear Accountability Check Display	Removes color highlights created by the accountability check.
Compare→Accountability Check→Select Unused	Runs the accountability check and shows all occurrences in the source pane that do not match an occurrence in the target pane in an occurrence group.
Compare→Accountability Check→Select Overused	Runs the accountability check and shows all occurrences in the source pane that match a corresponding occurrence in the target pane in an occurrence group.
Compare→Accountability Check→Select Completely Used	Runs the accountability check and shows all occurrences in the source pane that match more than one occurrence in the target pane in an occurrence group.
Compare→Accountability Check→Find in Other Structure	Finds the corresponding item in the other panel and selects it. This also expands the tree automatically.
Generate Report	Creates item reports in the context of one or more selected objects. Item reports generate in multiple output formats and follow PLM XML standards allowing integration with third-party reporting tools. Runs the Report Generation Wizard from which you can select a style sheet for the type of report to generate. For example, the TL Complying and Defining Report.

Window menu

Command	Description
Show View	Displays the list of views available from Systems Engineering, from which you can select a view to open. Tip: Choose Show View→Other and then expand Systems Engineering to access other Systems Engineering views.
Reset Perspective	Choose Reset Perspective if you want to start over with the default setup for the Systems Engineering perspective window, or the current perspective.

Command	Description
	Note: This command unloads all of the open objects and structures from the current views before resetting the perspective to its default configuration.

Systems Engineering buttons

The following table describes the buttons that are specific to Systems Engineering.

Button	Purpose
The following buttons are located on the main toolbar at the top of the Systems Engineering window.	
	Cut Moves the selected object or objects to the Clipboard.
	Copy Places a duplicate of the selected object or objects in the Clipboard. Copy can also be used to initiate a trace link from the selected defining object or objects.
	Paste Inserts the Clipboard contents below the selected item.
	Paste as New Pastes a duplicate of the object from the Teamcenter clipboard as a new object and a child of one selected object.
	Delete Deletes the selected objects.
	Refresh Refreshes the display in the selected window.
	Start Trace Link Initiates a trace link from the selected defining object. If multiple defining objects are selected, a separate trace link is initiated from each object.
	End Trace Link Completes a trace link to the selected complying object. If multiple complying objects are selected, a separate trace link is created to each object. This button assigns the default subtype, Trace Link , to the new trace link.
	End Trace Link... Completes a trace link to the selected complying object of a given subtype. If multiple complying objects are selected, a separate trace link is created to each object. This button allows you to assign a custom subtype to the new trace link.

Button		Purpose
	Remove	Removes the selected items from the requirement specification structure, without placing the items on the Clipboard. If the requirement specification is selected, the entire structure is removed. The items are not removed from the database.
	Requirement Specification	Creates an empty requirement specification, which is displayed on a new tab in the structure pane. You build the requirement specification structure by creating individual requirements and paragraphs.
	Requirement	Creates a requirement within a requirement specification structure. The requirement is created at the primary level, directly below the requirement specification. Or, the requirement is created as a child of any other item in the structure.
	Paragraph	Creates a paragraph within a requirement specification structure. The paragraph is created at the primary level, directly below the requirement specification. Or, the paragraph is created as a child of any other item in the structure.
	Custom Note	Creates a custom note within a requirement specification structure. The custom note is created at the primary level, directly below the requirement specification. Or, the custom note is created as a child of any other item in the structure.
	Function	Opens the New Function dialog box for you to create a new function object.
	Logical Block	Opens the New Logical Block dialog box for you to create a new logical block object.
	Custom Block	Opens the New Business Object dialog box for you to create other types of systems engineering blocks. Other systems engineering blocks are configured by your systems administrator.
	Interface	Creates an interface port for connecting inputs/outputs to/from blocks. The interface object allows for a connection to another block.
	Connection	Creates a connection between a selected interface port. The connection uses interface objects defined on the blocks to make the physical connection.
	Signal...	Opens the New Signal dialog box in which you can create a signal to add to the Signals library in Classification. After

Button	Purpose
	Associated Signals View a signal is available in the Signals library, you can add the signal to a functional or logical structure and associate it with interfaces and connections in the structure.
	Add Signals From Library Displays information about the signals associated with the selected object.
	Add Signals From Library Add a signal from the Signals library to the selected level in the functional or logical structure.
	Connect Creates a connection on a network port in a functional or logical structure.
	Disconnect Removes a connection on a network port in a functional or logical structure.
	Toggle In-Context Mode Enables or disables editing in context mode, allowing creation and edit of absolute occurrences for the selected structure.
	Structure Search Opens the Structure Search view.
	Cross Probing Mode Initiates a trace of a selected object's downstream and upstream relationships across structures in different Teamcenter applications.
	Derived Requirement Creates a new requirement as a child of the selected requirement with a copy of the information of parent requirement.
	Create Child Creates a child of the selected object of the same object type.
	Move Up Moves the selected structure element up one position in the structure.
	Move Down Moves the selected structure element down one position in the structure.
	Promote One Level Moves the selected structure element up one level in the structure. The selected structure element becomes a sibling of its parent structure element.
	Demote One Level Moves the selected structure element down one level in the structure. The selected structure element becomes a child of the immediately preceding sibling at the previous level.
	Custom Notes Manager Opens the Custom Notes view for the selected object.

Button		Purpose
	MS Word View	Opens the content of the selected requirement or paragraph for viewing or editing in Microsoft Word.
	Export objects to Word	Exports the content of selected requirements and paragraphs to Microsoft Office Word.
	Export objects to Excel	Exports the content of selected requirements and paragraphs to Microsoft Office Excel.
	Variant Condition	Displays variant equations that describe variant conditions on a structure.
	Edit Budget	Opens the Edit Budget dialog box for the selected object.
	Trace Links View	Opens the Trace Links view.
	Traceability Matrix	Opens the Traceability Matrix view.
	Show Diagram Explorer	Opens the Diagram Explorer view.
	Requirement Expression View	Opens the Requirement Expression view.
	Budget View	Opens the Budgets view.
	Impact Analysis	Opens the Impact Analysis view.
	Relation Browser	Opens the Relation Browser view.

The following buttons are located at the top of the embedded Word window in the **MS Word** pane.



Save

Saves the requirement or paragraph content in the Teamcenter database.

Note:

To view and edit requirement content in the **MS Word** pane, Word 2010 or 2013 must be installed on your computer. Otherwise, the **MS Word** pane displays the properties of the object selected in the structure pane.

What are perspectives and views?

Within the rich client user interface, application functionality is provided in *perspectives* and *views*.

View The basic display component that displays related information in a UI window.

Perspective A collection of one or more views and their layout.

Some applications use a perspective with multiple views to arrange how functionality is presented. Other applications use a perspective with a single view.

You can use the **HiddenPerspectives** preference to prevent the display of some Teamcenter perspectives in the rich client.

If your site has online help installed, you can access the application and view help from the rich client **Help** menu.

Working in the Systems Engineering interface

Working in requirement and architecture structures

Understanding requirement and architecture structures

Requirement specification and block structures define a hierarchical structure of requirements, paragraphs, functions, logical blocks, and other blocks. These items can be nested from the primary level, directly below a specification or block structure, to progressively subordinate levels.

A plus sign (+) appears to the left of each item that has one or more subordinates, or children, at the next lower level. You can view only these direct children, which may themselves have children that extend the structure to even lower levels. Or, you can view the entire structure below the item, displaying its direct children and all lower level items simultaneously.

You can set the number of structures of the same type that you can open by setting the **MEMaxOpenViewsSameType** preference. The default value is 10. Set this value to a higher number to open more than 10 structures of the same type in the Systems Engineering application. For example, you can open 10 or more requirement specification documents simultaneously for editing and viewing.

Systems Engineering structure views display icons that indicate the object types of the structure elements. In the Business Modeler IDE, default icons can be customized for existing object types, and icons for new object types can be defined. Other images can be overlaid on icons to show the states of objects according to specific property values. For example, an overlay can show whether or not objects of that type are currently checked out.

Search by scope and criteria for elements in a structure view

1. Open the structure and select the top line against which you want to search.

You can also select multiple top lines in the same structure or substructure.

2. Do one of the following:

- Click **Structure Search**  on the toolbar.
- Choose **Tools**→**Structure Search**.

Teamcenter displays the **Structure Search** view.

The structure line you select represents the scope of the search and is listed in the **Scopes** box.

3. (Optional) Click **...** to add more lines to the search scope or clear the scope.
 - Select a line in the structure and click **+** in the **Scopes** dialog box to add a scope.
 - Select a line in the **Scopes** dialog box and click **—** to remove the line from the scope.
 - Click **Clear**  in the **Scopes** dialog box to clear all scopes.

The columns names and widths that are initially displayed in the scope list are configured by the **MEAdvancedSearchResultsViewColumnsShownPref** and **MEAdvancedSearchResultsViewColumnsShownWidthPref** preferences.

4. (Optional) Load an existing search by clicking **Open Search**  to load a saved search.
5. From the **Query type** list, select a search type.

Note:

The available search types depend on the structure type you are searching.

6. Use the criteria sections to specify the search criteria pertaining to the type of search you selected.

Click the ... button to the right of each individual search criteria box and enter search criteria in the resulting dialog box. All the criteria you enter are combined so that you can, for example, search for all bolts whose name starts with **bo** and are within 3 centimeters of another part.

7. Select **Show results in a new tab** if you want to display each set of results in its own tab in the **Structure Search Results** view. If this check box is not selected, Teamcenter overwrites the results of the last search in the same pane.

Note:

The **Show results in a new tab** check box is not visible on the **Structure Search** view until after the first structure search is performed.

8. Click **Search**  to begin the search.

Teamcenter starts the search and loads the lines into the **Structure Search Results** view.

You can manage the search results in the **Structure Search Results** view.

9. (Optional) Select a line in the **Structure Search Results** view to highlight the corresponding item or item revision in the structure tree and the embedded viewer.
10. (Optional) Click **Clear**  to clear individual search criteria from the search, or click **Clear All** to clear all criteria from the **Structure Search** view.
11. (Optional) Save the structure search.

Search by property for elements in a structure view

In the **Requirements**, **Function**, and **Logical Block** views, you can locate specific elements quickly by using displayed properties as search criteria. The criteria consist of property names, operators, and property values.

Note:

Before you start this procedure, ensure that the view contains a column for each property by which you intend to search. Insert those columns if necessary.

1. Select the parent element, and then click **Find within the displayed objects**  on the toolbar.

The **Find in Display** dialog box is displayed.

2. Double-click the **Property Name** text box to select a property from the list of available columns.
3. Double-click the = box to select a different relational operator.

Relational operator	Definition
=	Equal to without case sensitivity
==	Exact match including case sensitivity
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
!=	Not equal to
-	Range

Relational operator	Definition
	OR
""	Exact string
&	AND

Note:

The operators <, <=, >, and >= apply to numeric properties only.

- Click the **Searching Value** text box and type the value you want to search for.

Note:

Partial matches are supported for string properties. Wildcard characters are not supported, and the search is not case sensitive.

- Add further clauses to the search condition by clicking +.

You can also double-click **AND** to change it to **OR**.

If necessary, reorder clauses by clicking ▲ and ▼.

- When the search condition definition is complete, click **Find**.

Teamcenter highlights the first line it finds that matches the search condition, and the **Find** dialog box shows a count of the total number of matching lines found.

- Click **Find next one**  and **Find previous one**  to step through the matching lines one at a time.

Tip:

You can click **Load All**  to select all matching lines at once.

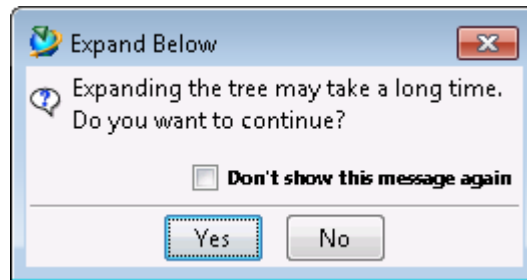
View only the direct children of an item

- Click the plus sign (+) to the left of the item.

The direct children are displayed at the next lower level. The item's plus sign becomes a minus sign (–), which you can click to hide the children.

View the entire structure below a selected line

1. Select the structure line and choose **View→Expand Options→Expand Below** or right-click the structure line and choose **Expand Below**.
2. Click **Yes** to expand the structure.



Items at all lower levels are displayed.

You can reverse this action by using the **View→Collapse Below** menu command.

Close structures

To close the view for a single structure, do one of the following:

- Right-click the view tab and choose **Close**.
- Click the **X** in the view tab.

To close the views for multiple structures, choose **File→Unload All**.

Navigating requirement structures using Windows shortcut keys

Within the **Requirements** view, you can navigate through the structure using the following keys:

Key	Definition
Page Up	Scroll up within the requirement specification structure.
Page Down	Scroll down within the requirement specification structure.
Home	Move to the first column in the selected line.
End	Move to the last column in the selected line.
↑	Move up one row in the structure.
↓	Move down one row in the structure.

Key	Definition
←	Move left one column in the structure.
→	Move right one column in the structure.
Ctrl-Q	Creates a new child or sibling in the structure based on the selected object. Enables the edit mode on the Rev Name column of the object if that column is displayed.
F2	Enables the edit mode on the Rev Name column of the selected object if that column is displayed.
Ctrl-Up Arrow	Promotes the selected object in the structure.
Ctrl-Down Arrow	Demotes the selected object in the structure.
Ctrl-A	Selects all objects in the view.
Ctrl-C	Copies the selected lines to the clipboard.
Ctrl-X	Cuts the selected object.
Ctrl-V	Inserts the Clipboard contents below the selected item.
Delete	Deletes the selected object from the database.
Ctrl-N	Displays the Quick Create bar.

Capture a Teamcenter object URL

You can capture the URL to an object in Systems Engineering. This URL can be embedded in e-mails or used in any application where the object and its context are required.

1. From Systems Engineering, open the structure (requirement/item/function/logical block or any item type) and select the object that you want to capture the URL for.
2. Right-click the object and choose **Copy URL**.

The URL is placed in your Windows Clipboard.

3. Paste the URL into an e-mail or other application.

When you click the URL, the object is opened in the same state as it was in at the time the URL was copied.

Opening objects in context

You can open objects in other views with the **Go To Object** shortcut menu command or button in the **Trace Links** view.

- Right-click any of the defining or complying objects and choose the **Go To Object** shortcut menu command to open it in its own tab.
- Click the object, choose the menu button ▼ next to the **Go To Object** button, and then select the perspective you want to open the object in.
- If the trace link points to an occurrence, **Go To Object** opens the entire structure that contains the occurrence.

In the **Visio Diagram** view, right-click the shape and choose **Go to Teamcenter** to open the object in the **My Teamcenter** perspective.

Expand a requirement structure in Relation Browser

1. In Systems Engineering, click the requirement specification you want to expand.
2. Select **Window→Show View→Relation Browser**.
3. In the **Relation Browser** view, right-click the requirement specification node and select the **Show Successors→All Levels** shortcut menu command.

Relation Browser displays all levels of the requirement's structure. For each structure element, Relation Browser displays the **Number** property value, which indicates the element's position in the hierarchy.

Print a requirements structure in Relation Browser

- In the **Relation Browser** view, choose **Print** from the **View Menu** ▼.

Organizing structures

How the Number property is used in structures

The **Number** property automatically shows each object's level and sequence within a Systems Engineering structure. For example, assume that objects A, B, and C occupy the first level, directly under the top object in the structure. Also, each object has three children at the next lower level.

In this example, the **Number** column values are:

- **1** for object A, with **1.1**, **1.2**, and **1.3** for its children
- **2** for object B, with **2.1**, **2.2**, and **2.3** for its children
- **3** for object C, with **3.1**, **3.2**, and **3.3** for its children

The **Number** column values are adjusted automatically when objects move up and down in a structure through:

- Promote and demote structure elements to higher and lower levels.
- Move structure elements up and down within their current levels.
- Add or remove structure elements.
- Manually updating the numbers directly.

Multiple levels in a structure allow flexibility in the organization of requirements and models. For example, all requirements can occupy the top level, directly below the requirement specification. Or, requirements can be structured in a hierarchy, in which the levels convey relationships among the requirements.

Working with hierarchical structures in Systems Engineering is similar to working with numbered paragraphs in Word's outline view. A given structure element may occupy the highest level, as a sibling of all other structures at level 1. Or, it may occupy a lower level, as a child of the structure element at the next higher level. In turn, a child structure element may also be a parent of children at even lower levels.

Structure elements can be organized in multiple levels of parents, children, and siblings. The **Number** column shows this hierarchy, with a plus sign (+) for an item that has one or more children at lower levels.

You can move items to higher and lower levels, changing their relationships to other items in the structure. When you move a parent item, you also move its direct children and lower-level descendants.

Promote structure elements

1. In Systems Engineering, double-click a requirement, functional model, or logical model to open it in its own view.
2. Click a plus (+) sign to expand the structure to find the target structure element.
3. Select the structure element to promote.
4. Choose **Edit→Promote One Level**.

You can also click **Promote One Level**  on the toolbar, or drop the element one level higher in the structure.

The selected structure element and its children move to the next higher level, as a sibling of its parent structure element.

Note:

If the structure element to promote is directly under the top level, you receive an error message stating that the selected object cannot be promoted.

If the structure element to promote has a parent that is released, you receive an error message stating that the select object cannot be promoted.

Demote structure elements

1. In Systems Engineering, double-click a requirement, functional model, or logical model to open it in its own view.
2. Click a plus (+) sign to expand the structure to find the target structure element.
3. Select the structure element to demote.
4. Choose **Edit→Demote One Level**.

You can also click **Demote One Level**  on the toolbar, or drop the element one level lower in the structure.

The selected structure element and its children move to the next lower level, as a child of the immediately preceding sibling at the previous level.

Note:

If the structure element to demote is the first element at any level, you receive an error message stating that the selected object cannot be demoted.

If the structure element to demote has a parent that is released, you receive an error message stating that the select object cannot be demoted.

Rearrange structure elements within a level

1. In Systems Engineering, double-click a requirement, functional model or logical model to open it in its own view.
2. Click a plus (+) sign to expand the structure to find the target structure element.
3. Select the structure element to reorder in its current level.
4. Choose one of the methods to rearrange the structure:
 - Choose **Edit→Move Up** or **Edit→Move Down**.

You can also click **Move Up**  or **Move Down**  on the toolbar.

The selected structure element and its children move up or down one position in its current level.

- Double-click the value in the **Number** column for the object you want to rearrange.
 - a. In the **Edit Number** dialog box, type the new number position in the **New Number** box.
 - b. Click **OK**.

The selected structure element is moved to the new position.

Note:

If the structure element to move has a parent that is released, you receive an error message stating that the select object cannot be moved.

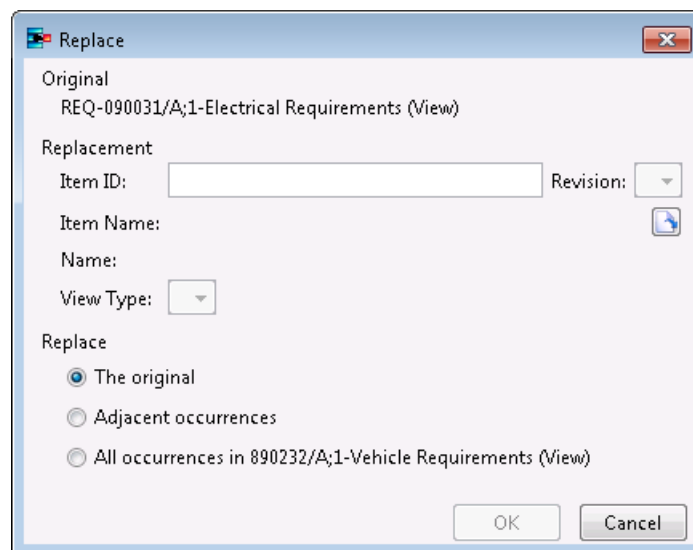
Replace an element in a structure

By replacing a structure element, you retain the occurrence data so that it is assumed by the replacement. Occurrence data (such as properties, trace links, and notes) is lost if you cut the old element out of the structure and paste in a new element.

Tip:

To automatically obtain the replacement's item identifier, you can copy the replacement to the clipboard. The item identifier is required in the **Replace** dialog box.

1. Select the occurrence to replace, and then choose **Edit→Replace...** to open the **Replace** dialog box.

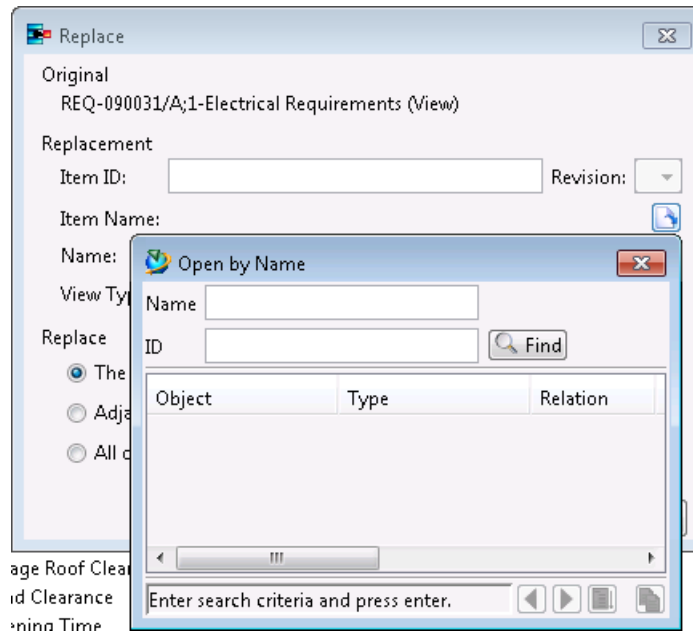


2. In the **Item ID** box, enter the item identifier of the replacement occurrence.

The item identifier is populated from the clipboard if you copied the replacement before selecting the old occurrence.

Tip:

To search for the identifier, you can click the **Item Name** button to use the **Open by Name** dialog box.



3. Under **Replace**, select one of the following:

The original Replaces the selected occurrence only.

Adjacent occurrences Replaces all occurrences of the selected item revision that appear in the parent structure

All occurrences Replaces all occurrences of the selected item revision, regardless of their position in the structure

Caution:

You cannot cancel the replacement of multiple occurrences once the replacement is in progress.

4. Click **OK** to start the replacement.

A progress indicator appears until the replacement is complete.

Note:

If any replacement action fails, the process continues with other occurrences. Elements not replaced are reported when the progress indicator closes.

Copy a structure element as a new object

In the **Requirements**, **Function**, and **Logical block** views, you can save time in building a structure by copying an existing element as a new object. Using this method, you create a unique object rather than a reference to the source object.

1. Right-click the source object and choose **Copy**.

Note:

Children of the source object are copied with the parent.

2. Select the object in the structure below which you want to paste the new object.

Note:

Select only one target object. If you select multiple objects, the duplicated object is pasted only to the first selected object.

3. Click **Paste as New**  on the application toolbar, or right-click the target object and choose **Paste as New**.

The new object appears below the selected object. Any children of the source object are included as new objects below the new parent.

Remove elements from a structure

Objects remain in the Teamcenter database after you remove them from a structure in Systems Engineering. For example, they stay in My Teamcenter if you previously save them there. You can also use the **Search** feature to recall the objects.

Note:

- When you delete items and structures from the database, they are also deleted from other Teamcenter applications, and they cannot be recalled with the **Search** feature. In some structures, you cannot delete multiple objects; you must delete them individually.

- When you delete an item revision, only that occurrence is deleted. Teamcenter does not delete other revisions of the item or the item itself.

Caution:

If you delete the only revision of an item, Teamcenter also deletes the item.

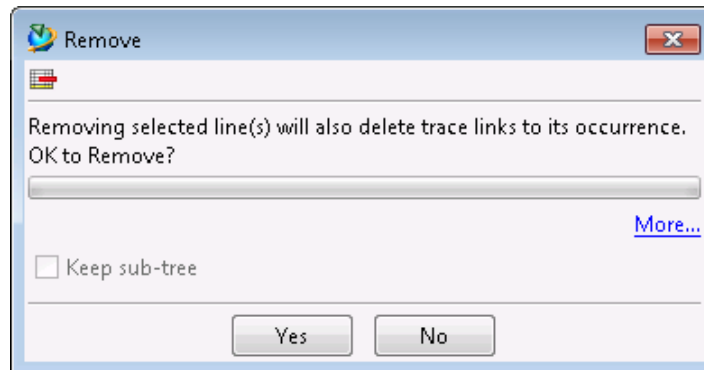
- You can delete an item or item revision from a structure even if it is duplicated elsewhere in Teamcenter.

1. Select the objects that you want to remove.

You can select multiple objects by using the standard Windows functions. Or, select the top object to remove the entire structure.

2. Choose **Edit→Remove**, or click , or press Ctrl-R.

The **Remove** dialog box appears.



Caution:

If any trace links are related to the object selected for remove, a message states that trace links related to the removed object are deleted.

3. Click **Yes** to continue or click **No** to cancel the remove operation.

Note:

A progress indicator appears during the remove operation. You can abort the operation by clicking **Stop**. However, the remove operation may not be fully recoverable.

Working with columns displayed in views

Understanding property column configuration

Configuring property columns includes setting the display of property columns shown in the view table.

You can insert, remove, and reposition columns, as well as apply and save column configurations.

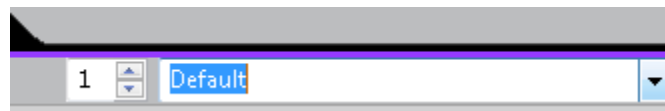
The changes you make to a column configuration override the default column configuration in the Systems Engineering views, but you can restore the default column configuration to the view at any time. Only administrators can change the default column configuration. Also, you cannot remove the default column configuration from the **Select a saved configuration** list.

In Systems Engineering you can mark a saved column configuration as publishable. The Teamcenter administrator can then publish the saved column configuration so it is available to other users.

Default property columns in Systems Engineering views

When you open a view in Systems Engineering, the default column configuration for the top-level object appears. While you can make changes to the current property columns displayed and create your own saved column configurations, only administrators can change the stored version of the default column configuration. Also, you cannot remove the default column configuration from the **Select a saved configuration** list.

You can restore the default column configuration at any time by selecting **Default** from the **Select a saved configuration** list.



The default column configurations are defined by the administrator, using the following preferences to set the default columns in a view based on the top-level object.

Preference	Sets default columns when topic-level object is:
Fnd0LogicalBlockRevisionColumnsShownPref	Logical block
FunctionalityRevisionColumnsShownPref	Function
Paragraph RevisionColumnsShownPref	Paragraph
Requirement RevisionColumnsShownPref	Requirement
RequirementSpec RevisionColumnsShownPref	Requirement specification

Insert property columns in a view

1. Right-click in the header of a column and choose **+ Insert Column(s)**.

The system displays the **Change Columns** dialog box.

2. Select attributes from the **Available Columns** list and click the **Add** button **+**.

Tip:

To locate attributes in the **Available Columns** list, type the attribute name in the box above the list.

3. Set the display order of the columns using the **Up arrow** button **▲** and the **Down arrow** button **▼**.
4. (Optional) To save the column configuration:
 - a. Click **Save**.
 - b. In the **Save Column Configuration** dialog box, type a name and description for the configuration, for example **Requirement properties**.
 - c. Click **Save**.
5. In the **Change Column** dialog box, click **Apply** and then click **Close**.

The changes you made to the column configuration override the default column configuration; however, you can restore the default column configuration at any time by selecting **Default** from the **Select a saved configuration** list.

Remove columns from a view

Do one of the following to remove columns from the view:

- **To remove one column:**

1. Right-click the header of the column that you want to remove and choose **Remove this column**.
2. Click **Yes** to remove the column.

The changes you made to the column configuration override the default column configuration; however, you can restore the default column configuration at any time by selecting **Default** from the **Select a saved configuration** list.

- **To remove multiple columns:**

1. Right-click in the header of any column and choose **+ Insert column(s)**.
2. In the **Change Columns** dialog box, select attributes from the **Displayed Columns** list and click the **Remove** button .
3. (Optional) To save the column configuration.
 - a. Click **Save**.
 - b. In the **Save Column Configuration** dialog box, type a name and description for the configuration, for example **Requirement properties**.
 - c. Click **Save**.
4. In the **Change Columns** dialog box, click **Apply** and then click **Close**.

The changes you made to the column configuration override the default column configuration; however, you can restore the default column configuration at any time by selecting **Default** from the **Select a saved configuration** list.

Reposition columns in a view

1. Click the header of the column that you want to move and hold the mouse button.
2. Drag the column to the desired position in the table and release the mouse button.

The column is displayed in the new position.

Save column configuration

1. Right-click the column header and choose **Save Column Configuration**.

The **Save Column Configuration** dialog box appears.

2. Type values in the **Name** and **Description** boxes.

Caution:

If the name exceeds 32 characters, the column configuration cannot be marked for publishing, deleted, or renamed.

Note:

Column configuration names should be unique.

3. Select the **Scope** option as **User**, **Group**, or **Site**.

Note:

The options you see enabled depend on your role. If you are a site administrator, you see all three options enabled. If you are a group administrator, you see the **User** and **Group** options enabled. If you are not an administrator, no option displays and the column configuration is saved at the user level.

4. Click **Save**.

Teamcenter creates a column configuration and saves it for the chosen scope.

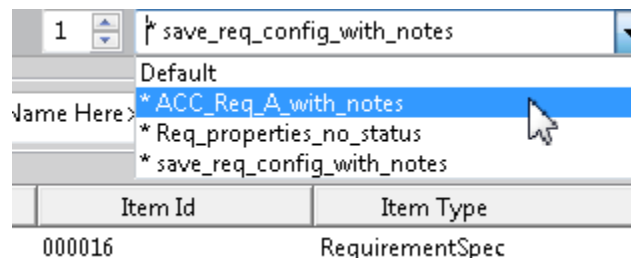
Apply a saved column configuration

To display table data in a previously saved column configurations you can:

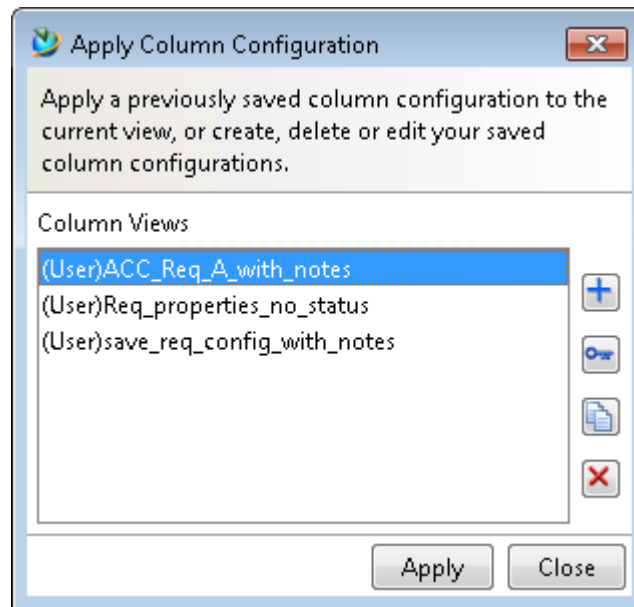
- Select the name of the saved column configuration from the **Select a saved column configuration** list.

Note:

You can also restore the default column configuration by selecting **Default**.



- Select the name of the saved column configuration from the **Apply Column Configuration** dialog box.
 1. Right-click in the header of a table column and choose **Apply a Column Configuration**.
 2. In the **Apply Column Configuration** dialog box, select the name of the saved column configuration and then click **Apply**.



Note:

You can also use the **Apply Column Configuration** dialog box to add a new column configuration, modify a saved column configuration, create a new configuration based on a saved configuration, or delete a saved configuration.

Update column configuration

1. Right-click the column header and choose **Apply Column Configuration**.

The **Apply Column Configuration** dialog box opens.

2. Select the column configuration name depending on your role.

Note:

If you are a site administrator, select the site column configuration name. If you are a group administrator, select the group column configuration name. If you are not an administrator, select the user column configuration name.

3. Click **Modify** to modify the column configuration.

The **Change Columns** dialog box opens, allowing you to update the selected configuration.

4. When you have updated the column configuration, click **Apply**.

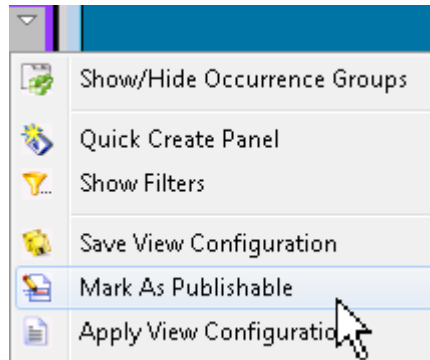
Teamcenter updates the column configuration for the current and future sessions.

Mark saved column configurations as publishable

Note:

You cannot mark a column configuration whose name exceeds 32 characters.

1. Choose **Mark as Publishable** from the view menu.



2. Select the name of the saved column configuration you want to mark as publishable.
3. Click **OK**.

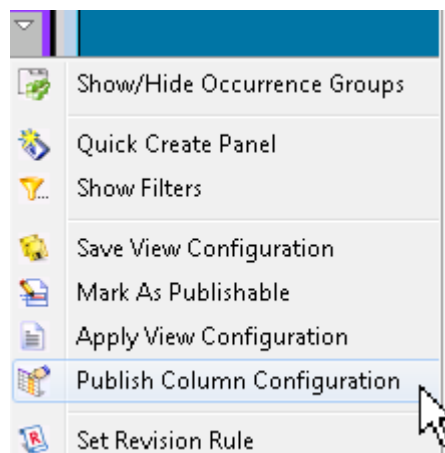
Publish saved column configurations for site level reuse

Note:

A saved column configuration must be marked as publishable before it can be published.

You must have administrator privileges to publish a saved column configuration.

1. Choose **Publish Column Configuration** from the view menu.



2. In the **Available** pane, select the name of the column configuration you want to publish.
3. Click the right-arrow button to move the name of the column configuration to the **Published** pane.
4. Click **OK**.

Delete column configuration

Note:

You cannot delete a column configuration whose name exceeds 32 characters.
You cannot delete the default column configuration.

1. Right-click the column header and choose **Apply a Column Configuration**.

The **Apply Column Configuration** dialog box appears.

2. Select the column configuration name depending on your role.

Note:

If you are a site administrator, select the site column configuration name. If you are a group administrator, select the group column configuration name. If you are not an administrator, select the user column configuration name.

3. Click **Delete**  to delete the column configuration.

Teamcenter displays the **Delete Column Configuration** dialog box.

4. Click **Yes** to confirm deletion of the selected column configuration.

Display multiple text lines in the structure view

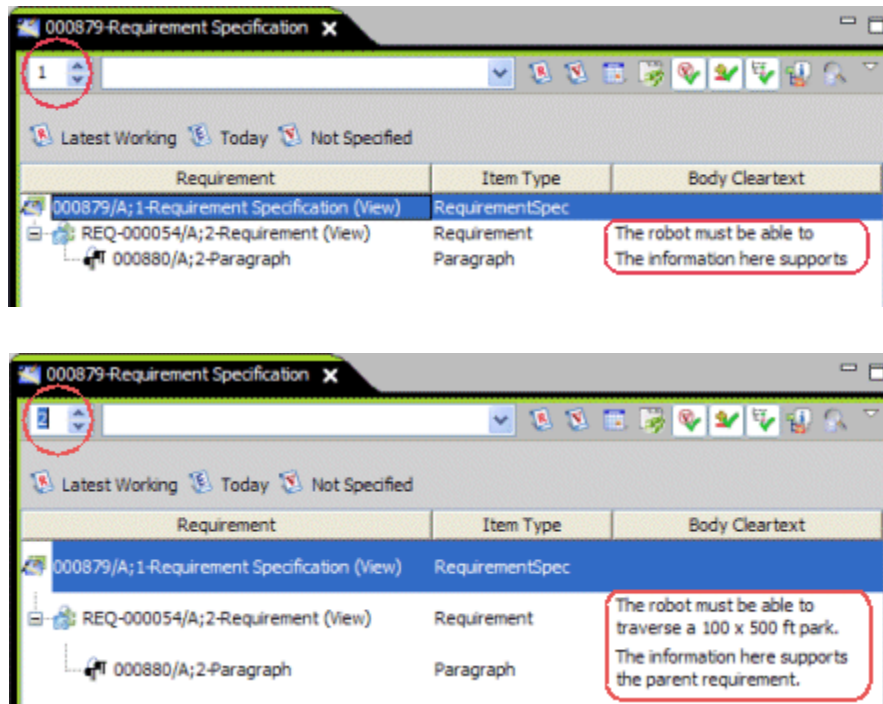
The **Body Cleartext** column displays the text of an object's Word content.

- To set the number of text lines to view in the **Body Cleartext** column, select a number from the **Set the number of rows per line** list, located above the column headings.

This enables the lines of text in the **Body Cleartext** column to wrap so you can view all, or a larger portion of, the text in the property column.

For example, in the first image, the number of rows per line is set to 1 and the system only shows one line of the text in the **Body Cleartext** property column. In the second image, the number of rows per line is set to 2. Therefore, the system shows two lines of text in the **Body Cleartext** property column,

which happens to be all the text in the requirement and the paragraph content. You can increase the number of rows per line until you can see all or most of the text.



Note:

Using the **MultilineColumnProperties** preference, Teamcenter administrators can configure for which column properties multiple lines of text can be displayed. The preference is set to the **Body Cleartext** property, **bl_fulltext_body_cleartext**, by default.

Displaying graphic values in property columns

Values for some Systems Engineering object properties appear as icons in the property columns display.

The **IconShownColumnProperties** preference setting specifies the property columns in which an icon value appears. The default settings for the preference are **bl_has_trace_link** and **bl_rev_Fnd0ListsCustomNotes**.

As a result of the **bl_has_trace_link** setting, for example, the trace link symbol  appears as the value in the **Trace Link** property column for any objects that are trace linked.

Working with multiple object selections

In Systems Engineering you can:

- Modify the properties of one object at a time or modify the common properties of multiple selected objects in a single action.

For example, you can select one or more multiple requirement objects, right-click, and then choose **Properties**.

If you select one requirement, the **Properties** dialog box appears.

If you select multiple requirements, the **Common modifiable properties** dialog box appears.

- Submit a single selected object or submit multiple selected objects to a workflow process.

For example, you can select one or more requirements objects and then choose **File→New→Workflow Process**. The object or objects you selected appear in the **Targets** folder on the **New Process Dialog** dialog box.

- Revise one selected object or revise multiple selected objects in a single action.

For example, you can select one object and then choose **File→Revise** to access the **Revise** wizard. You can use the steps in the wizard to further define the new revision of the object.

Or, you can select multiple objects and then choose **File→Revise** to access the **Multiple Revise** dialog box. Using the **Multiple Revise** dialog box, you are limited to editing the name of the new revision.

Note:

To enable the ability to revise multiple selected objects using the **Multiple Revise** dialog box, the **Multiple_revise_dialog_visible** preference must be set to **true**.

2. Developing requirements

Developing requirements

Insufficient planning and misunderstood expectations affect more than two-thirds of all product development projects.¹ The systems engineering process solves both problems in the most crucial phase of development, the decision-making stage, when plans and expectations are consolidated as requirements. Requirements are used to communicate those decisions to the people responsible for creating the design.

Requirements describe the product that the customer will buy. They communicate the customer's specifications to the various disciplines involved in the product's development. To ensure that the finished product meets those specifications, developers follow the requirements throughout the development process. When the product conforms to all requirements, it is ready for delivery, and more importantly, it has the functions and the quality that the customer demands.

Projects that inadequately communicate requirements to developers are prone to run over budget and behind schedule, with late-cycle changes and heroic integration efforts. Also, all aspects of testing are usually more time consuming and post-delivery problems, such as warranty claims and recalls, are often much higher.

In avoiding such difficulties, two principles are vital. The highest level requirements must be identified and developed at the project's inception and requirements engineering processes must be followed throughout product development, so that problems are revealed and understood as early as possible in the product development process. Requirements must be associated with the product design, and that connection must be maintained through all successive stages—while drawings, parts, and assemblies are developed, tested, and changed.

Incorporating both principles, Teamcenter Systems Engineering provides for:

- Development of requirements in the initial stage of product development.
- Early association of requirements with the product design and individual components.
- Ongoing maintenance of requirements, each as a separate item with specific properties.

Typically, systems engineers use trace links for allocating requirements to elements in downstream or upstream structures, such as:

- Other requirements in the same structure or in different **Requirement** views.
- Functional models, logical models, and physical models in Systems Engineering.

¹ *Forrester Report*, March 2000, in *Collaborative Solutions for Product Lifecycle Management* brochure (M-1000), published March 2002 by EDS PLM Solutions.

Trace links set up paths in which elements precede and succeed one another across the system's various domains. By tracing these defining and complying relationships, systems engineers can analyze change impact both upstream and downstream.

Tip:

Trace links can also exist between Systems Engineering objects and subsets in 4th Generation Design. Collaborative designers can analyze these relationships through the Systems Engineering *cross-domain tracing* feature.

The systems engineering process extends customer involvement to all phases of product development. The customer can actively influence the design from requirements development and association to product testing and acceptance.

Though a traditional requirements specification can include hundreds or thousands of requirements, it is typically managed as a single document. In Teamcenter Systems Engineering, however, each requirement is managed as a separate item with specific properties.

Note:

Teamcenter Systems Engineering provides two types of requirement items, **Requirement** and **Paragraph**. A paragraph can be extended to include other subtypes of requirements, such as regulatory or performance requirements. This guide uses the term *requirement* for both types.

Building requirement specification structures manually

What is a requirement specification?

A requirement specification is a container for requirement and paragraph objects. Within a requirement specification, these objects can be structured in a hierarchy of parent, child, and sibling relationships.

To build a structure manually:

- Create a requirement specification object at the top level.
- Add the structure elements by creating requirements and paragraphs at lower levels.

Create a requirement specification

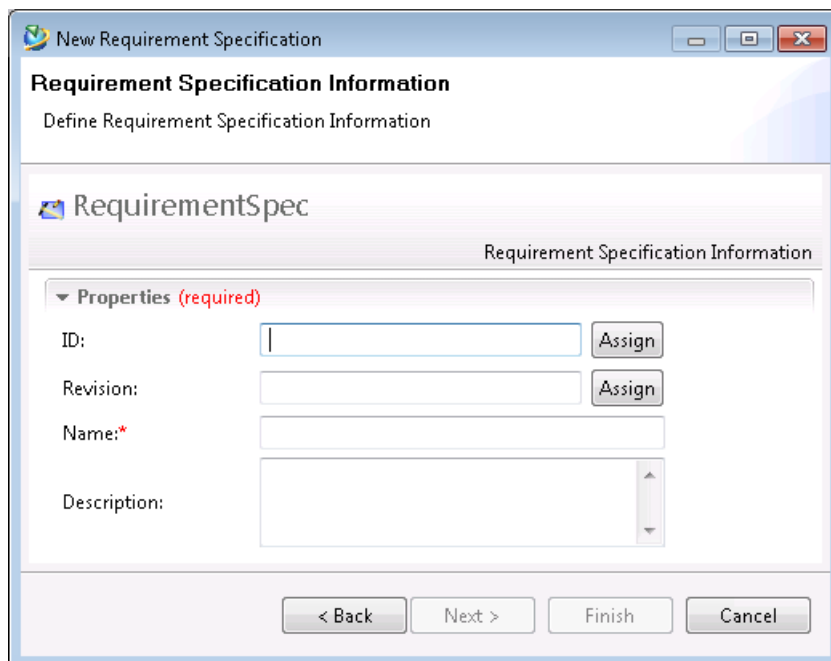
1. In Systems Engineering, do one of the following to create a requirement specification:
 - Click the **Create a specification** link in the **Requirement View** welcome pane.

- Choose **File→New→Requirement Specification**.
- Click **Requirement Specification**  on the toolbar.

Note:

You can also create a requirement specification from the Quick create bar in the **Requirements** view.

2. In the New Requirement Specification wizard, select **RequirementSpec** as the requirement specification type, and then click **Next**.
3. In the **Requirement Specification Information** dialog box, enter the following information for the new requirement specification:



Field name	Definition	Valid values
ID	Requirement specification ID (required). To fill in the ID box automatically, leave it empty or click the Assign button. Systems Engineering inserts the next identifier in the sequence.	Alphanumeric (1–128 characters)
Revision	Requirement specification revision (required).	Alphanumeric (1–128 characters)

Field name	Definition	Valid values
	To fill in the Revision box automatically, leave it empty or click the Assign button. Systems Engineering inserts the current revision.	
Name	Name of the requirement specification (required).	Alphanumeric (1–128 characters)
Description	Description of the requirement specification.	Alphanumeric (1–240 characters)

- (Optional) Click **Next** to assign the requirement to a workflow process template and a project.
- Click **Finish**.

Teamcenter creates the requirement specification object and opens it in a **Requirements** view. The new object occupies the top level in the requirements structure.

You can click **Cancel** to close the New Requirement Specification wizard, or you can create another requirement specification.

- (Optional) Edit the content of the requirement specification in either the **MS Word** view or in a separate Word window.

Create a requirement

After you create a specification, you can create requirements and paragraphs to develop the structure of the specification.

- In Systems Engineering or Requirements Manager, select the node where you want to create the requirement at the next lower level, and do one of the following:
 - Choose **File→New→Requirement**.
 - Click **Requirement**  on the toolbar.

Note:

You can also create a requirement specification from the Quick create bar in the **Requirements** view.

- In the New Requirement wizard, select **Requirement** as the requirement type, and then click **Next**.

3. In the **Requirement Information** dialog box, enter the following information for the new requirement:
4. (Optional) Click **Next** to assign the requirement to a workflow process template and project.
5. Click **Finish** to complete the new requirement and close the wizard.

The new requirement is created and occupies the last position at the selected level.

You can click **Cancel** to close the New Requirement wizard, or you can create another requirement.

6. (Optional) Edit the content of the requirement specification in either the **MS Word** view or in a separate Word window.

Create a derived requirement

A derived requirement is a child of an existing requirement. It inherits the existing requirement's description and is related to the parent requirement with a trace link. This enables you to use the existing information to quickly create a derivative of the existing requirement as a child requirement.

1. In Systems Engineering, select an existing requirement or paragraph in the **Requirements** view, and do one of the following:
 - Choose **File→New→Derived Requirement**.
 - Click **Derived Requirement**  on the toolbar.

Note:

You can also create a requirement specification from the Quick create bar in the **Requirements** view.

2. In the New Derived Requirement wizard, select the type of derived requirement that you want to create, and then click **Next**.
3. In the **Derived Requirement Information** dialog box, enter the following information for the new requirement.

Field name	Definition	Valid values
Requirement ID	Requirement ID (required). To fill in the Requirement ID box automatically, leave it empty or click the Assign button. Systems Engineering inserts the next identifier in the sequence. The	Alphanumeric (1–128 characters)

Field name	Definition	Valid values
	default prefix REQ- precedes the identifier. Note: The default prefix is set by the Default Requirement ID Rule naming rule.	
Revision	Requirement revision (required). To fill in the Revision box automatically, leave it empty or click the Assign button. Systems Engineering inserts the current revision.	Alphanumeric (1–128 characters)
Name	Name of the requirement (required).	Alphanumeric (1–128 characters)
Description	Description of the requirement. Text typed into this box becomes the initial body of the requirement. Note: You use Microsoft Word to add and edit the text that constitutes the requirement body.	Alphanumeric (1–240 characters)

- Optionally, click **Next** to assign the requirement to a project.
- Click **Finish**.

Teamcenter copies the value of the **Description** property from the parent requirement and creates the derived requirement as a new occurrence with a trace link to the parent requirement.

- (Optional) Edit the content of the requirement specification in either the **MS Word** view or in a separate Word window.

Create a child requirement or paragraph

In contrast to a derived requirement, a child requirement does *not* inherit the properties of the parent and there is no trace link relationship.

- In the **Requirements** view, select an existing requirement or paragraph, and do one of the following:
 - Click **Create Child**  on the toolbar.
 - Press Ctrl-Q.
 - Choose **Edit→Create Child**.

Teamcenter creates the child requirement or paragraph and adds it under its parent, as a new occurrence.

Create a paragraph

1. In Systems Engineering or Requirements Manager, select the node where you want to create the paragraph at the next lower level, and do one of the following:
 - Choose **File→New→Paragraph**.
 - Click **Paragraph**  on the toolbar.

Note:
 You can also create a requirement specification from the Quick create bar in the **Requirements** view.

2. In the New Paragraph wizard, select **Paragraph** as the type, and then click **Next**.
3. In the **Paragraph Information** dialog box, enter the following information for the new paragraph.

Field name	Definition	Valid values
ID	Paragraph ID (required). To fill in the ID box automatically, leave it empty or click the Assign button. Systems Engineering inserts the next identifier in the sequence.	Alphanumeric (1–128 characters)
Revision	Paragraph revision (required). To fill in the Revision box automatically, leave it empty or click the Assign button. Systems Engineering inserts the current revision.	Alphanumeric (1–128 characters)

Field name	Definition	Valid values
Name	Name of the paragraph (required).	Alphanumeric (1–128 characters)
Description	Description of the paragraph. Text typed into this box becomes the initial body of the paragraph.	Alphanumeric (1–240 characters)
	Note: You use Microsoft Word to add and edit the text that constitutes the paragraph body.	

4. (Optional) Click **Next** to assign the paragraph to a workflow process template and a project.
5. Click **Finish**.

The new paragraph is created and occupies the last position at the selected level.

You can click **Cancel** to close the New Paragraph wizard, or you can create another paragraph.

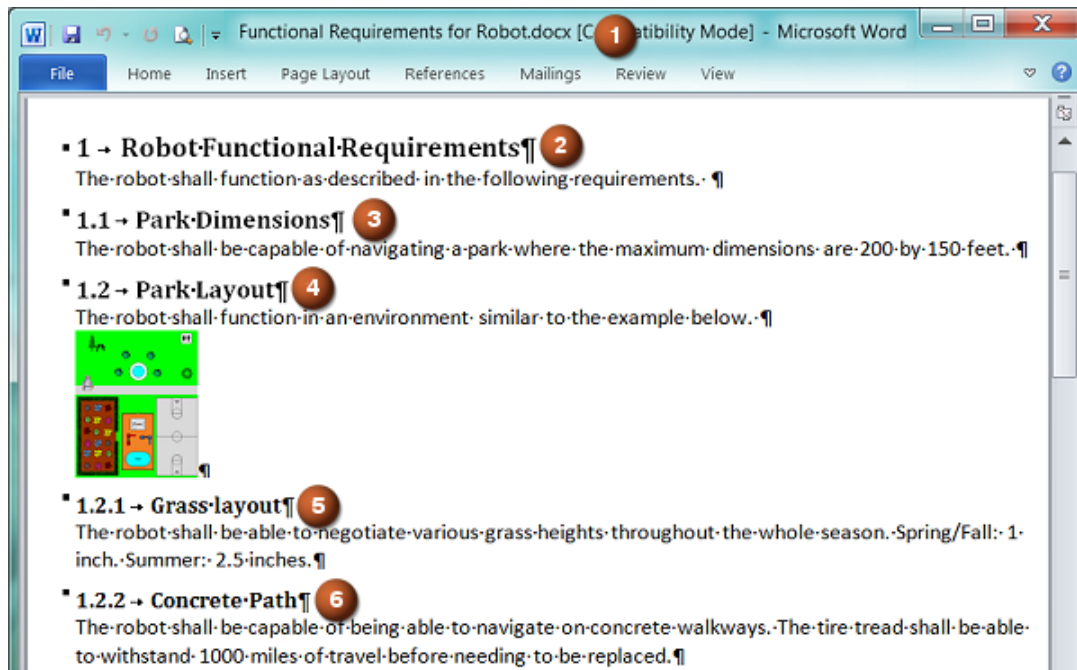
6. (Optional) Edit the content of the requirement specification in either the **MS Word** view or in a separate Word window.

Importing requirement specifications from structured Microsoft Word documents

How Word styles determine structure hierarchy in Teamcenter

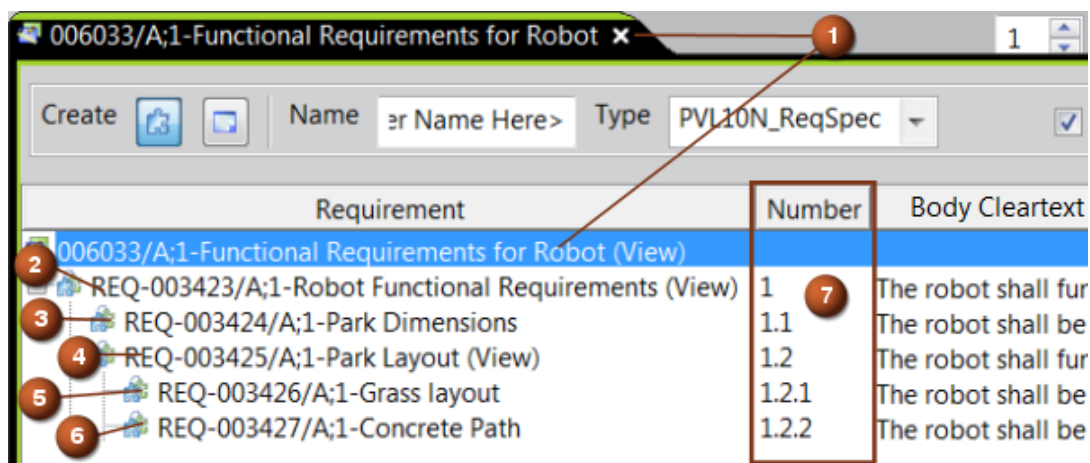
You can create an entire requirement structure in one action by importing a specification document from Word. Teamcenter uses the document's Word heading styles to generate a new structure in the Systems Engineering **Requirements** view. Together, the sequence of heading styles and their numbered outline levels (1 through 9) determine the parent, child, and sibling hierarchy in the imported specification.

For example, assume that you create the following requirement specification:



- 1 Word document name
- 2 Word **Heading 1** style (outline level 1)
- 3 Word **Heading 2** style (outline level 2)
- 4 Word **Heading 2** style (outline level 2)
- 5 Word **Heading 3** style (outline level 3)
- 6 Word **Heading 3** style (outline level 3)

When you import the previous example document, Teamcenter generates a matching specification in the **Requirements** view:



- 1 The name of the Word document becomes the name of the new requirement specification. By default, the requirement specification object occupies the top level of the structure.

Tip:

You can change the top-level object type by editing the **TcAllowedImportExportPeakSpecObject** preference.

- 2 The text of the first heading becomes the name of the first new requirement. The heading style's outline level places the requirement at the first level below the specification.
- 3 The text of the second heading becomes the name of the second new requirement. The heading style's outline level places the requirement at the second level as a child of the first requirement.
- 4 The text of the third heading becomes the name of the third new requirement. The heading style's outline level places the requirement at the second level as a sibling of the second requirement.
- 5 The text of the fourth heading becomes the name of the fourth new requirement. The heading style's outline level places the requirement at the third level as a child of the third requirement.
- 6 The text of the fifth heading becomes the name of the fifth new requirement. The heading style's outline level places the requirement at the third level as a sibling of the fourth requirement.
- 7 In the **Number** property column, each value matches the numbered outline level of that requirement's heading in the Word document.

All Word content below a given heading style becomes the content of the corresponding requirement. Requirement content can include elements such as graphics and tables.

Note:

The **Body Cleartext** property column displays only the text portion of the content.

Caution:

- Special characters in Word heading styles are replaced by question marks (?) in the imported requirements. If those requirements later are exported to Word, the question marks are included in the output document.
- If the document does not begin with a Word heading style, the import process generates a requirement named **NO TITLE** immediately below the top-level object. This requirement receives all content from the beginning of the document to the first Word heading style, or the entire content if there is no heading style.

Generating separate requirements within a Word outline level

Note:

Teamcenter Extensions for Microsoft Office must be installed on your computer.

Within an outline level in the import document, further separation of requirements can be achieved through *keywords*. You can enter one or more keywords in the Import Spec wizard when you import a requirement specification.

Starting with the first Word heading style, the parser searches the body content for the keyword or keywords.

- If one keyword is found before the next outline level, the requirement starts after the nearest outline level preceding the keyword:
 - The text of the immediately preceding outline level paragraph becomes the requirement name.
 - All material under the outline level paragraph becomes the requirement content. The paragraph itself does not appear in the content.
 - The requirement has the subtype that you assign.
- If two or more keywords are found before the next outline level, the requirements are separated as follows:
 - The first requirement starts after the nearest outline level preceding the first keyword.
 - The second requirement starts after the nearest paragraph separator preceding the second keyword. This requirement is created as a child of the first requirement.
 - For each subsequent keyword, the nearest preceding paragraph separator begins the requirement, which becomes a child of the first requirement and a sibling of the preceding one.

Note:

If keywords are found within the same content paragraph, sentence separators are used instead of paragraph separators. If two or more keywords are found in the same sentence, a requirement is created only for the first keyword.

The name of each requirement is the text of the immediately preceding outline level paragraph. All requirements have the same subtype, which you assign.

Note:

Keywords in tables are ignored. Separate requirements are not created for this content.

If no keyword is found in the content before the next heading style the heading text becomes the requirement name, and contains all material under that heading. The document heading does not appear in the requirement content.

At the next outline level, the parser repeats the processing described.

Special characters in Word import documents

For the following reasons, Siemens Digital Industries Software recommends that you consider special characters before you import objects from a Word document.

- Because of Teamcenter server and database configurations, special characters in Word heading styles (outline levels 1 through 9) are replaced by question marks (?) in the imported objects. If those objects later are exported to Word, the question marks are included in the output document.
- On Teamcenter servers using the ISO 8859-1 character set (**TC_CHARACTER_ENCODING_SET**), some characters of the rich-text format cannot be translated to plain text. These characters include, for example, *em* dashes, symbols such as Ohm signs (Ω), and Word's smart quotation marks.

Such characters are replaced by question marks (?) in plain text and in the Systems Engineering interface, for example, in object names and in certain property values. To avoid this problem, do one of the following:

- Before you enter content for an object, switch the **MS Word** view to plain text.
- Run the Teamcenter server on Windows, using a database compatible with CP1252 encoding.
- Use UTF-8 encoding for the Teamcenter server and database.

Import a requirement specification from Word

Caution:

- In .NET Web tier (middle tier) configurations, importing large documents can cause the server to time out. You can edit the **Web.config** file to resolve server time-out issues.
- The specification document must carry the **.docx** file name extension.
- To generate separate requirements within Word outline levels, Teamcenter Extensions for Microsoft Office must be installed on your computer.

Tip:

Before you start this procedure, you can do the following:

- Specify the object type for the top line of the imported structure, by editing the **TcAllowedImportExportPeakSpecObject** preference.
- Exclude unwanted specification types from the **Spec type** list in the Import Spec wizard.

Note:

You can assign a subtype to the imported requirements during this procedure.

1. In the **Home** view, select the folder to which you want to import the structure.

If no folder is selected, the specification is imported to the **Newstuff** folder.

2. Choose **File→Import Spec**.

The Import Spec wizard appears.

3. For the **Enter Information for Import** step, do the following:

- a. Designate the import document by doing one of the following in the **File name** box:
 - A. Click the browse button (...) to the right, navigate to and select the document, and then click **Import**.
 - B. Enter the document's full path directly in the box.
- b. In the **Spec type** list, accept the default value or select a different specification type.

You can also do the following optional actions for the **Enter Information for Import** step:

- Use the **Description** box to enter general information about this import.
- Check the **Import as child of selected element** box to add the structure to an existing specification.

4. Click **Next** to advance to the **Select Import Options** step.

- To parse the document by outline levels only:
 - a. Click **Import as single subtype** to display a list of subtypes.

- b. Select **Requirement**, **Paragraph**, or a custom subtype defined in your Teamcenter environment.

The subtype is assigned to all requirements that are imported from the document.

- To parse the document by outline levels and keywords, do the following:
 - a. Click **Use keywords for import**.

The **Keywords** box and the = list become available.

- b. Enter the keyword or keywords in the **Keywords** box.

Tip:

- Use a comma to separate two keywords.
- Special characters and partial words are not supported.

- c. In the = list, select the subtype to assign to each requirement imported from the document.

Note:

Keywords in the = list set the value of the **REQ_import_keywords** user preference. You cannot set the **REQ_import_keywords** site preference from the Import Spec wizard.

5. Click **Finish**.

Systems Engineering displays the specification in a new **Requirements** view. Below the top level, the structure consists of parent, child, and sibling requirements according to the outline levels in the import document.

Tracking the source of imported requirements

For each requirement in the imported structure, two properties track the requirement's source:

- The **Source Document** property stores the file name of the Word document from which the requirement was imported.
- The **Source Document Paragraph** property stores the number of the Word document paragraph from which the requirement content was extracted.

Resolve server time-out issues

1. Browse to the middle tier installation directory.

2. In the **Web.config** file, increase **executionTimeout** to **7200** and **maxRequestLength** to **131072**, as follows:

```
<httpRuntime executionTimeout="7200"maxRequestLength="131072"/>
```

3. Clear the cache.
4. Restart the IIS server, the Teamcenter server, and the Teamcenter rich client.

Importing requirements from unstructured Microsoft Office files

Setting up a hierarchy for unstructured files

During your business process you may originate or receive unstructured requirements documents created in Microsoft Word or Excel. Unstructured documents (or flat files) are files that contain no Word defined outline levels. Word defined outline levels are required in documents that are imported to Teamcenter, as these are used by Teamcenter to define requirements structures.

If you manage requirements and traceability information in Teamcenter, and you want to import unstructured documents, you must follow these key steps:

1. Customize the flat file configuration files, which are XML files used to define rules for adding identifiers to the content in unstructured documents. With these identifiers, content can be read and marked up by a parser to be imported as structured requirements to Teamcenter. The configuration files also define Teamcenter connection and parser settings, object property assignments, and import preview color settings.
2. Assign identifiers in the unstructured document for defining requirement objects such as requirements, paragraphs, and trace links that you want to maintain in Teamcenter.
3. Import the unstructured document to Teamcenter to create and maintain the requirement objects and trace links.

During your business process you may originate or receive unstructured requirements documents created in Microsoft Word or Excel. Unstructured documents (or flat files) are files that contain no Word defined outline levels. Word defined outline levels are required in documents that are imported to Teamcenter, as these are used by Teamcenter to define requirements structures.

If you manage requirements and traceability information in Teamcenter, and you want to import unstructured documents, you must follow these key steps:

1. Customize the flat file configuration files, which are XML files used to define rules for adding identifiers to the content in unstructured documents. With these identifiers, content can be

read and marked up by a parser to be imported as structured requirements to Teamcenter. The configuration files also define Teamcenter connection and parser settings, object property assignments, and import preview color settings.

2. Assign identifiers in the unstructured document for defining requirement objects such as requirements, paragraphs, and trace links that you want to maintain in Teamcenter.
3. Import the unstructured document to Teamcenter to create and maintain the requirement objects and trace links.

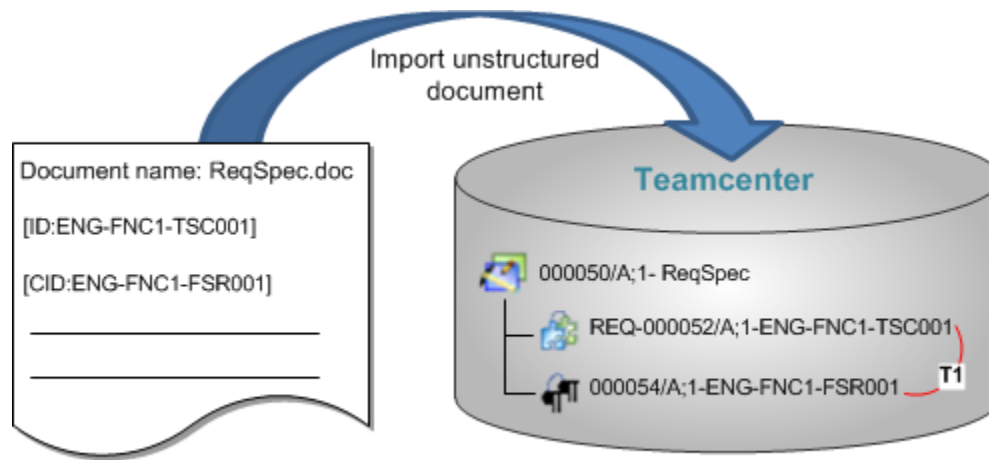
Example:

These documents originated from an external supplier and contain no Word defined outline levels to define a structure.

Functional Requirements for Robot <u>Park Requirements</u> Dimensions: The robot should be capable of navigating a park where the maximum dimensions are 100 by 50 feet. Layout: Grass layout - The robot should be able to navigate throughout the whole season, around 1 inch, but in the summer months it should be able to navigate around 2 inches. Concrete Path - The robot should be able to navigate on concrete walkways in the park for up to 1000 miles of travel before needing to be replaced.													
<table border="1"> <thead> <tr> <th colspan="2">Fuel Tank Specification</th></tr> <tr> <th>Functional Requirements</th><th>Business Requirements</th></tr> </thead> <tbody> <tr> <td>Size: Size will be 8 inches x 9 inches.</td><td>Size should meet ISO specification.</td></tr> <tr> <td>Capacity: Capacity of fuel tank will be 16 liters to accommodate gas expansion.</td><td>As per government law, fuel tank must have a minimum capacity of 14 liters.</td></tr> <tr> <td>Surface: Surface will be made of an alloy of aluminum and iron to sustain a high temperature.</td><td>Use anti-rust, high temperature sustained material.</td></tr> <tr> <td>Pressure: Tank will be able to sustain pressure of 50 psi and will be leak-proof tested as per the</td><td>Tanks should be 100% pressure tested for</td></tr> </tbody> </table>		Fuel Tank Specification		Functional Requirements	Business Requirements	Size: Size will be 8 inches x 9 inches.	Size should meet ISO specification.	Capacity: Capacity of fuel tank will be 16 liters to accommodate gas expansion.	As per government law, fuel tank must have a minimum capacity of 14 liters.	Surface: Surface will be made of an alloy of aluminum and iron to sustain a high temperature.	Use anti-rust, high temperature sustained material.	Pressure: Tank will be able to sustain pressure of 50 psi and will be leak-proof tested as per the	Tanks should be 100% pressure tested for
Fuel Tank Specification													
Functional Requirements	Business Requirements												
Size: Size will be 8 inches x 9 inches.	Size should meet ISO specification.												
Capacity: Capacity of fuel tank will be 16 liters to accommodate gas expansion.	As per government law, fuel tank must have a minimum capacity of 14 liters.												
Surface: Surface will be made of an alloy of aluminum and iron to sustain a high temperature.	Use anti-rust, high temperature sustained material.												
Pressure: Tank will be able to sustain pressure of 50 psi and will be leak-proof tested as per the	Tanks should be 100% pressure tested for												

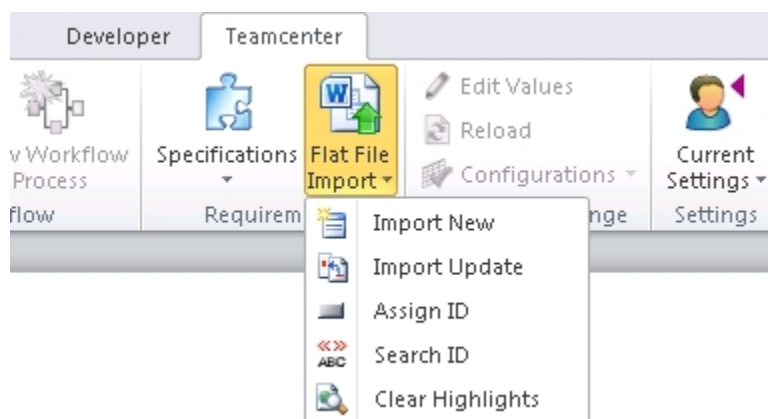
After customizing the configuration files and assigning identifiers to the content in the unstructured document, you can preview the document *before* the import to check for accuracy or errors. The preview document highlights new items, existing items, duplicate items, internal links, and external links.

After you have resolved any errors in the document, you import it to Teamcenter to create or update the requirement and traceability information.



Accessing the flat file functionality using the Teamcenter ribbon

From the Teamcenter ribbon in Word or Excel, you access the flat functionality from the **Flat File Import** list in the **Requirements** group.

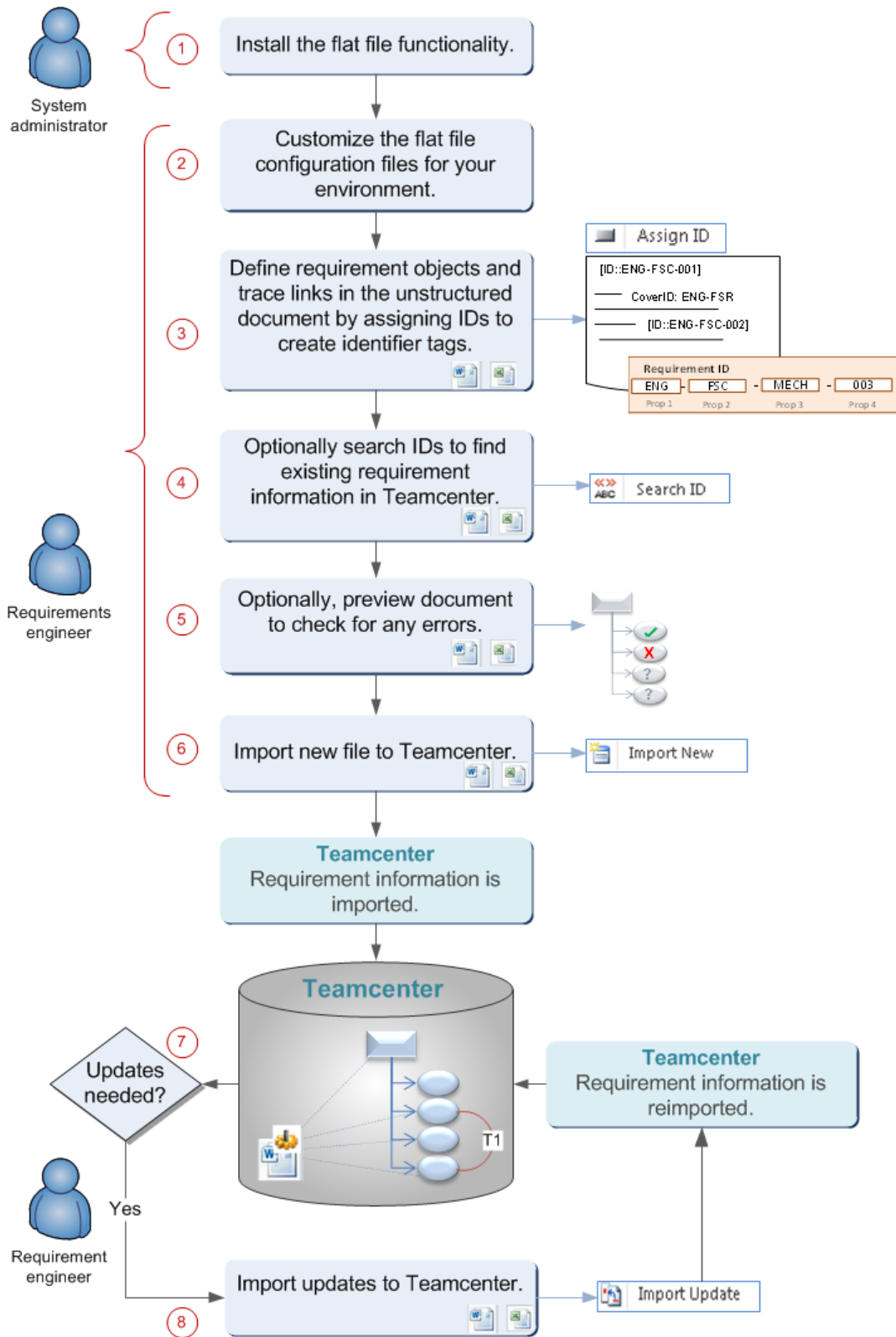


From the Teamcenter ribbon, you can perform the following for unstructured documents.

Function	Description
Import New	Imports the unstructured document for the first time as a new file to Teamcenter.
Import Update	Reimports any changes made to the unstructured document to Teamcenter after initial import.
Assign ID	Assigns IDs in Microsoft Word or Excel to create object identifier tags to specify requirements, paragraphs, and trace links In Teamcenter when you import the file. You also may set property values or create links when you import requirements of a certain type. You can use <i>regular</i>

Function	Description
	<i>expressions</i> for matching identifiers that can be linked or assigned to specific properties in the database. Before the import, you can preview the requirement structure for accuracy or errors.
Search ID	Search IDs to find existing requirements in Teamcenter.
Clear Highlights	Clears highlights after document preview.

What are the steps you perform to import unstructured requirements?



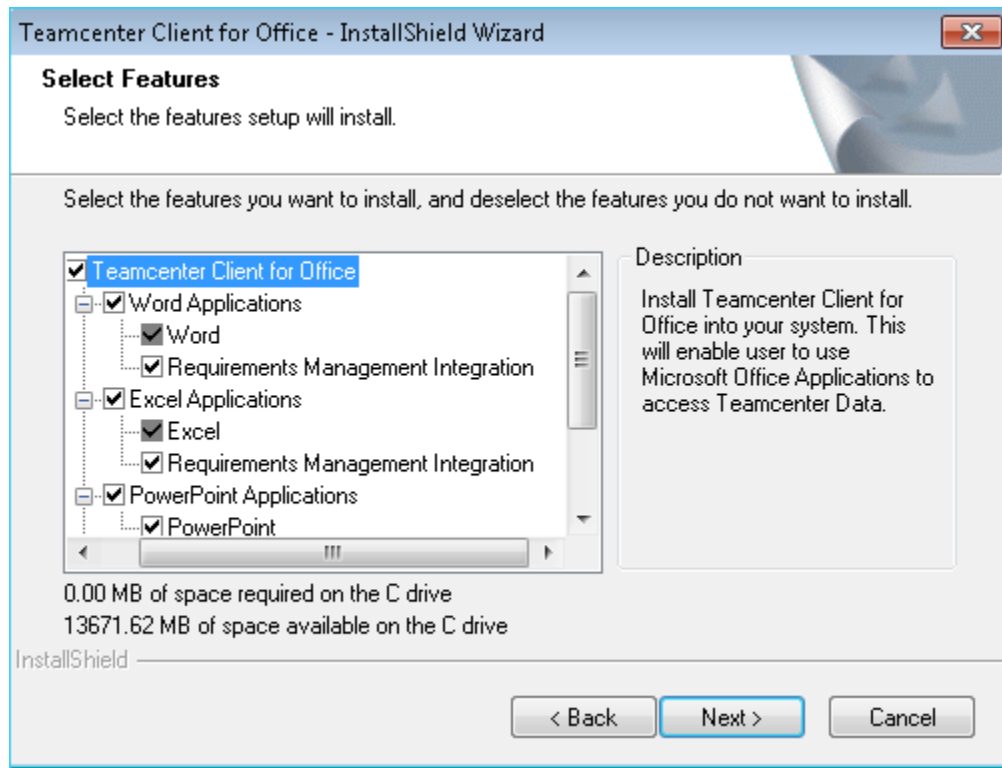
Step number	Step information
1	As the system administrator, you install the flat file functionality. Flat file functionality is installed as part of the Client for Office installation by selecting the Requirements Management Integration options.
2	As the requirements engineer, you customize the flat file configuration files for your environment. You modify the common settings, parser rules, and property settings section of the configuration files to configure such items as the server name, valid identifier tags and the parser rules.
3	As the requirements engineer, you define requirement objects and trace links in the unstructured document by assigning IDs to create identifier tags. Identifiers are regular expressions that are used to identify the object in the unstructured document. From the Teamcenter ribbon from Word or Excel, you assign IDs to create the requirement information.
4	As the requirements engineer, you optionally search IDs to find existing requirement information in Teamcenter. From the Teamcenter ribbon from Word or Excel, you can search for existing requirement information.
5	As the requirements engineer, you optionally preview the unstructured document to check for errors. From the Teamcenter ribbon from Word or Excel, you can preview the document before the import to check for accuracy or any errors. For example, you may want to preview the document to validate the identifiers are unique and don't exist in Teamcenter.
6	As the requirements engineer, you import the unstructured document to Teamcenter. From the Teamcenter ribbon from Word or Excel, you import the unstructured document as a new file to create the dataset and the requirement information in Teamcenter.
7	As the requirements engineer, you can update the unstructured document to modify, delete, or add requirement information.
8	As the requirements engineer, you import updates to Teamcenter. From the Teamcenter ribbon from Word or Excel, you reimport the unstructured document to update or revise the dataset and the requirement information in Teamcenter.

Installing the flat file functionality

The flat file functionality is installed as part of the Client for Office installation.

Client for Office prerequisites must be installed.

To install the flat file functionality, in the **Teamcenter Client for Office** dialog box, select the option **Requirements Management Integration** under **Word Applications** and **Excel Application**.



Customizing the flat file configuration files for your environment

Overview of the flat file configuration files

The first step for the requirements manager in importing unstructured requirements is customizing the flat file configuration file(s) to be used for the import process. The flat file configuration files are XML files used to define rules for adding identifiers to the content in unstructured documents. With these identifiers, content, including requirements, paragraphs, and trace links, can be read and marked up by a parser to be imported as structured requirements to Teamcenter. The configuration files also define Teamcenter connection and parser settings, object property assignments, and import preview color settings.

Two flat file configuration files are included with the Teamcenter installation. One is used for importing Word files and one is used for importing Excel files. These are files you customize to specify the import parameters for your environment.

Office product	Location and name of XML configuration file
Word	<i>Client for Office install location/TcFFWordConfig.xml</i>
Excel	<i>Client for Office install location/TcFFExcelConfig.xml</i>

The configuration files contain the following four sections:

- Common settings: configure parser settings and flat file global settings.
- Parser rules: configure rules that create objects and trace links.
- Property assignments: configure property rules.
- Color assignments: configure color assignments for highlighting the identifiers when previewing the unstructured document.

In the configuration files, you define the rules for the format and behavior of object identifiers. Identifiers are defined in the configuration files by using *regular expressions*. A regular expression is a set of characters that defines a pattern, used for searching for specific lines of text containing the particular pattern. You create regular expressions for each of the different object types you are importing, for example, requirements revisions and paragraph revisions.

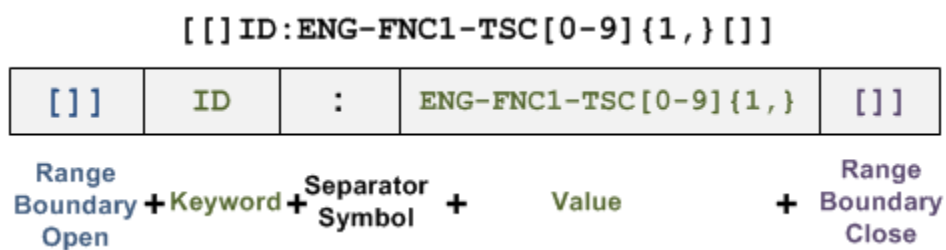
In Word or Excel, you use the **Assign ID** command to assign identifiers to the content in the unstructured requirements, for example on requirements revisions and paragraph revisions. The identifiers are validated against and created based on the regular expressions in the configuration file.

You also define the rules in the configuration files for creating trace links. The cardinality of a relation, whether it is a primary or secondary relation, is defined by the corresponding parser rule in the configuration file that applies between the two object identifiers.

An object and trace link identifier tag is a regular expression in the following syntax:

Range Boundary Open + Keyword + Separator Symbol + value + Range Boundary Close

Here is an example of a regular expression breakdown in Word.



Flat files in Excel

To create a requirement under a requirement specification in Excel using flat files, the *TcFFExcelConfig.xml* file in the *%TC_Root%/officeClient* directory must contain the following:

```
<add RuleID="1" LinkType = "FND_TraceLink" PrimaryKeyword = "[[]ID:.*[]"
PrimaryKeywordType = "Requirement Revision" CreatePrimaryKeyword =
```



```
"true" SecondaryKeyword = "[[ ]CID:PAR-[0-9]{6} <CID:PAR-[0-9]%7b6%7d>[ ]]"
SecondaryKeywordType = "Paragraph Revision" CreateSecondaryKeyword = "false"
LinksAs = "Defining"/>
```

Note:

In the PrimaryKeyword line, there must be a period before the asterisk.

How are regular expressions used?

For the import of unstructured requirements in Word or Excel, you can use regular expressions to assign object identifiers and property identifiers in the imported unstructured flat file. You specify these regular expressions in the flat file configuration files.

A *regular expression*, a set of characters that defines a pattern, is used for searching for specific lines of text containing the particular pattern. Regular expressions search for patterns on a single line and not for patterns that start on one line and end on another. You define these patterns in the flat file configuration files to identify the object's type for the identifier that matches the pattern.

To match strings when assigning identifiers in unstructured Word documents, you can create regular expressions by combining one or more wildcard characters supported in Word along with literal text. The regular expressions that you use for Excel files are similar but with some minor differences. For Excel files, you can use the same syntax for regular expressions that are used for regular expressions in Microsoft .NET Framework.

For more information about using regular expressions and the supported wildcard characters in Word, and for regular expressions used in .NET Framework, refer to Microsoft help.

With regular expressions, you can perform actions such as:

- Searching for words of a certain size.
- Searching for a word with four or more vowels that end with a specific letter, number, or a punctuation mark.
- Matching strings such as the strings "Handel", "Händel", and "Haendel". The regular expression **H(ä|ae?)ndel** specifies the pattern that is used to match all three strings.

Example:

The following information displays some examples of regular expression patterns for Word and what strings or characters would match.

Pattern	What it matches	Why does it match?
/[0-9a-zA-Z]/	Matches all alphanumeric strings.	Matches any number, any lower case letters in the alphabet, and any upper case letters in the alphabet.
/b[aeiou]t/	Matches words such as "bat", "bet", "bit", "bot" and "but", "baseball bat", and "bitter taste".	Matches any expression that contains the letter "b", followed by any one of the vowels (a,e,i,o, and u), and followed by the letter "t".
/b*t/	Matches words such as "bat" and "bait", "but" also part of the string "best time".	Matches any expression beginning with the letter "b" and ending with the "t".
/<(boot)/	Matches words such as "booth", but not "reboot".	Matches any word that begins with (<) "bat".

Example: Configuring identifier tags for requirement objects and trace links

You customize the flat file configuration file for Word to set the **PrimaryKeyword** to the `[[]ID:ENG-FNC1-TSC[0-9]{1,}[]]` regular expression and the **SecondaryKeyword** to the `[[]CID:ENG-FNC1-FSR[0-9]{1,}[]]` regular expression.

Using the **Assign ID** dialog box, you create a requirement revision identifier to define a requirement in the unstructured document:

- **Type** is set to **Requirement Revision**
- **Prefix** is set to **ID**
- **ID** is set to **ENG-FNC1**
- **Name** is set to **TSC001**

The resulting valid identifier for the paragraph revision is set to **[ID:ENG-FNC1–TSC001]**.

You also assign the ID for the paragraph revision using the following values:

- **Type** is set to **Paragraph Revision**
- **Prefix** is set to **CID**
- **ID** is set to **ENG-FSC1**
- **Name** is set to **FSR001**

The resulting valid identifier for the paragraph revision is set to **[CID:ENG-FNC1–FSR001]**.

The unstructured document now contains the two identifiers:

[ID:ENG-FNC1–TSC001]

[CID:ENG-FNC1–FSR001]

Configuring the common settings

The *common settings* section of the configuration file specifies the settings used by the add-ins such as the connection to Teamcenter and the parser that is used.

The following describes the parameters you can configure.

Key setting name	Description
Parser	Specifies the name and location of the DLL that is used to parse the contents of the unstructured document. You set the [APPATH] to the folder location of the parser DLL. By default, the folder location of the parser DLL is located in the path where you installed Client for Office. Default value for Word: Teamcenter.Client.TcReq_IDToIDParser.WordIDParser; [APPPATH]\TcReq_IDToIDParser.dll" Default value for Excel: Teamcenter.Client.TcReq_IDToIDParser.ExcelIDParser; [APPPATH]\TcReq_IDToIDParser.dll"
PropDebugComments	Generates debug information while parsing properties in the flat file. Value is set to true or false. Default value: false Warning: Set this value to true only when identifying problem issues for technical support.

Key setting name	Description
LinkDebugComments	Generates debug information while parsing links in the flat file. The value is set to true or false. Default value: false Warning: Set this value to true only when identifying problem issues for technical support.
DefaultSpecType	Specifies the top-level default specification type to be created in Teamcenter when you import the document. The RequirementSpec type or a subtype of RequirementSpec can be used. Default value: RequirementSpec
IDValidationPattern	Specifies the regular expression pattern used to validate the ID in the Assign ID dialog box. This pattern should encompass <i>all</i> the patterns used in this file for all the parser rules. For example, if your file contains rules for the identifiers RDV-SPR-NNNNN, RMV-SOR-NNNN, and RDV-SRR-NNNNNN, set this value to R[A-Z]{1}V-S[A-Z]{1}R-[0-9]{1,6} to ensure all patterns are covered. Default value: [[]*:[A-Za-z0-9_]{1,}-[0-9]{6}[]]
IDPrefix	Specifies the identifier prefixes that are used to match a specific rule. The prefixes specified here are populated in the Assign ID dialog box. Default value: ID, CID
IDProperty	Specifies the property the identifier tag is set to after import to Teamcenter. It is also the default property used for searching requirement information such as requirements and paragraphs. You set this value to Item ID to ensure the identifier tag is unique. Default value: object_name
DefaultSavedQuery	The default property used for searching a requirement specification that you want to reimport for update or revise. Default value: Item ID
Search	The value of the search class and the .NET assembly you want to use when searching existing requirement objects in Teamcenter. You can customize the search behavior by

Key setting name	Description
	creating your own search class and .NET assembly. To add a key value for the customized search and assembly, use the following format: <pre><add key="Search" value="<Namespace of Search Class>; <Full Path to assembly>" /></pre> Default value: OOTBSearch

Example: How are the common settings used?

This example uses the following values for the common settings section of the configuration file for Word.

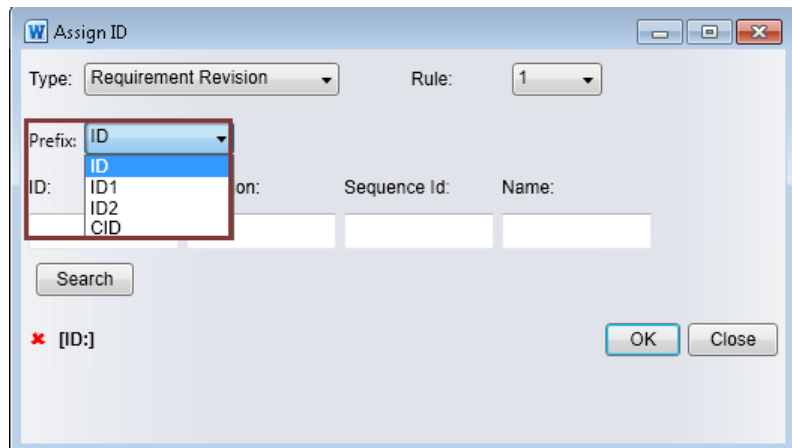
```
<Common Settings>
<add key = "Parser" value = "Teamcenter.TcIDToIDParser.WordIDParser;
C:\Program Files (x86)\Siemens\Teamcenter
9\TcWord2K7DocImp\TcIDToIDParser.dll"/>
<add key = "PropDebugComments" value = "false"/>
<add key = "LinkDebugComments" value = "false"/>
<add key = "DefaultSpecType" value="RequirementSpec" />
<add key = "IDValidationPattern" value="[[]*:[A-Za-z0-9_]{1,}-[0-9]{6}
[]]"/>
<add key = "IDPrefix" value = "ID,ID2,ID3,CID" />
<add key = "IDProperty" value = "object_name" />
<add key = "Default Saved Query" value = "Item ID" />
<add key = "Search" value="OOTBSearch" />
</Common Settings>
```

The following information displays how some of the key values are used in the common settings section.

Text in common settings section	How is the setting used?
<add key = "IDPrefix" value = "ID,ID2,ID3,CID" />	In the unstructured Word or Excel document, you assign an identifier for the requirement information you want to import to Teamcenter. In the Assign ID dialog box, the list of valid prefixes from the IDPrefix key is displayed when you click the Prefix box. The identifier prefixes are used to match a specific rule in the parser rules.

Text in common settings section

How is the setting used?



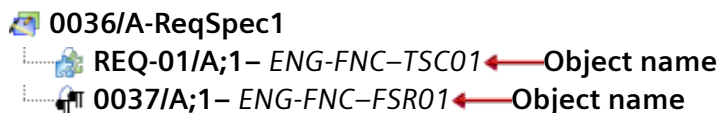
```
<add key = "IDProperty" value =
object_name" />
```

In the unstructured Word or Excel document, you assign the following identifiers for a requirement revision and a paragraph revision in an unstructured Word or Excel document.

```
[ID:ENG-FNC-TSC01]
[CID:ENG-FNC-FSR01]
```

You import the unstructured document to Teamcenter. After the import, identifier is set to the **IDProperty** value. The object name does not contain the identifier prefix values such as **ID** or **CID**.

As the following requirement structure shows, the identifiers are set to the object names in Teamcenter.

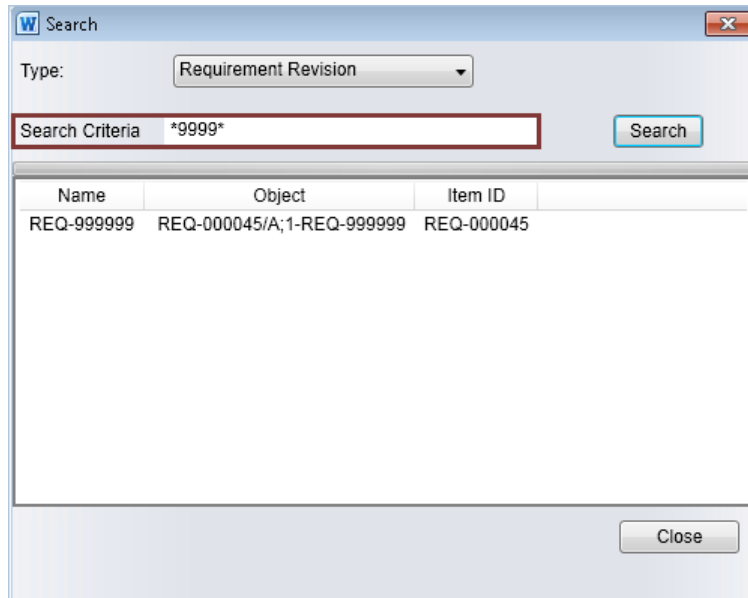


The **IDProperty** value is also used for the search criteria when searching for existing requirement information in Teamcenter from an unstructured Word or Excel document.

In the following example, in the **Search** dialog box, you type ***9999*** in the **Search Criteria** box to search for all requirement revisions that have the string **9999** as part of the object name.

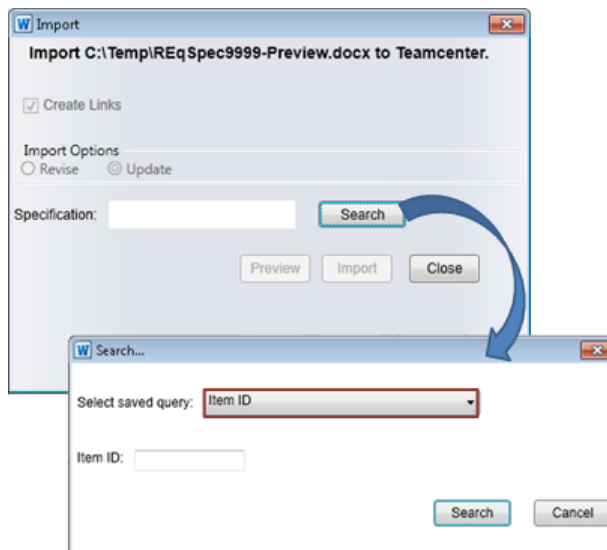
Text in common settings section

How is the setting used?



```
<add key="DefaultSavedQuery"
value="Item ID" />
```

The **Item ID** value is used as the default when searching requirement specifications that you want to reimport for **Update** or **Revise**.



Configuring the parser rules

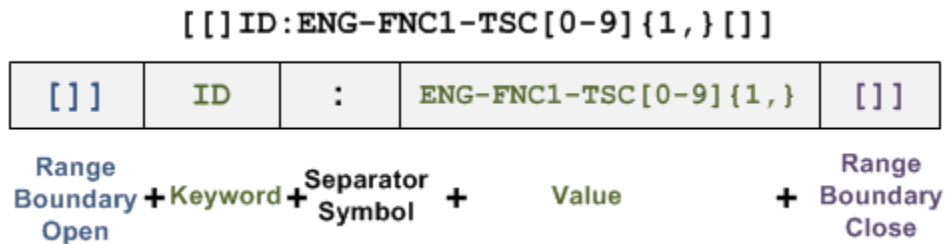
The *parser rules* section of the configuration file specifies the rules used by the parser to identify objects and their relations in the file. A rule defines the primary and secondary object identifiers and the relations between them. You can define multiple rules to match your environment needs.

The object identifier is a regular expression that is used to identify the object in the unstructured document.

The regular expression is in the following format:

Range Boundary Open + Keyword + Separator Symbol + value + Range Boundary Close

Here is an example of a regular expression breakdown for a **PrimaryKeyword** or **SecondaryKeyword** setting in Word:



You use delimiters to specify the start and end value of the identifier. The delimiter is enclosed in square brackets in the range boundaries. By default, the start delimiter is set to square brackets. You can change the value of the delimiter in the configuration file. The following example shows using parentheses as the delimiters for the regular expression:

`[()D:ENG-FNC1-TSC[0-9]{1,}()]`

Caution:

Some characters such as **<**, **>**, and **&** cannot be used in XML.

Delimiters are also used in the regular expression **Value** to designate the valid property values for the identifier.

The following describes the parameters you can configure.

Setting name	Description
LinkType (optional)	Specifies the type of link you want to create between the object identifier matching the PrimaryKeyword rule and the SecondaryKeyword rule. Default value: FND_TraceLink
RuleID (required)	Specifies the unique identifier for the rule. Each rule must have a unique identifier.

Setting name	Description
	Note: Siemens Digital Industries Software recommends that you set the RuleID using digits.
	Default value: 1
PrimaryKeyword (required)	Specifies the pattern for the object identifier for the primary object. This pattern is used to match objects in the document for this rule. The object identifier is a regular expression that is used to identify the object in the unstructured document. Any string that matches the regular expression is treated as the ID of the object of the type specified in PrimaryKeywordType. Keyword should be one of the values that you set for IDPrefix in the common settings section. The separator symbol is a colon by default. Default value: [[]ID:REQ-[0-9]{6}[]]
PrimaryKeywordType (required)	Specifies the type to be used when creating the object matching the rule defined for PrimaryKeyword. Default value: Requirement Revision
CreatePrimaryKeyword (required)	Specifies whether you want to create the object matching the rule defined for PrimaryKeyword, if it does not exist in Teamcenter. Valid values are true or false. Default value: true
SecondaryKeyword (optional)	Specifies the pattern for the object identifier for the secondary object. This pattern is used to match objects in the document for this rule. Any string that matches the regular expression is treated as the ID of the object of the type specified in SecondaryKeywordType. Default value: [[]CID:PAR-[0-9]{6}[]]
SecondaryKeywordType (optional)	Specifies the type to be used when creating the object matching the rule defined for SecondaryKeyword. Default value: Paragraph Revision
CreateSecondaryKeyword (optional)	Specifies whether you want to create the object matching the rule defined for SecondaryKeyword, if it does not exist in Teamcenter. Valid values are true or false.

Setting name	Description
	Default value: false
LinksAs (optional)	Specifies the cardinality of the trace link between the object identifiers. If LinksAs is set to Defining , then the object identifier matching the rule using the SecondaryKeyword is <i>primary</i> in the relation. If LinksAs is set to Complying , then the object identifier matching the rule using the SecondaryKeyword is <i>secondary</i> in the relation.
	Default value: Defining

Example: How are the parser rules used?

This example uses the following values for the parser rules section of the configuration file for Word.

```
<ParserRules>
<add
  RuleID="1"
  LinkType = "FND_Tracelink"
  PrimaryKeyword = "[[]ID:REQ-[0-9]{6}[]]"
  PrimaryKeywordType = "Requirement Revision"
  CreatePrimaryKeyword = "true"
  SecondaryKeyword = "[[]CID:PAR-[0-9]{6}[]]"
  SecondaryKeywordType = "Paragraph Revision"
  CreateSecondaryKeyword = "true"
  LinksAs = "Complying"/>

<add
  RuleID="2"
  LinkType = "BZ_FND_Tracelink"
  PrimaryKeyword = "[[]ID:B-FNC-TSC[0-9]{1,}[]]"
  PrimaryKeywordType = "Requirement Revision"
  CreatePrimaryKeyword = "true"
  SecondaryKeyword = "[[]ID:B-FNC-TSR[0-9]{1,}[]]"
  SecondaryKeywordType = "Paragraph Revision"
  CreateSecondaryKeyword = "false"
  LinksAs = "Defining"/>
</ParserRules>
```

The following information displays how some of the settings are used in the parser rules section.

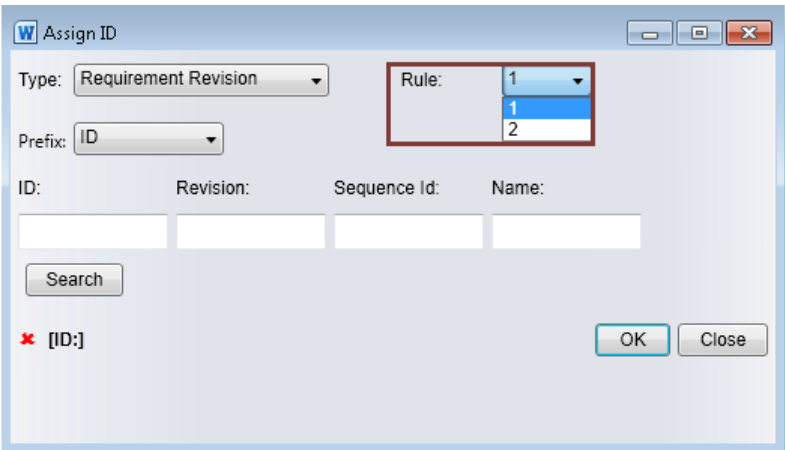
Text in parser settings section	How is the setting used?
RuleID="1"	In the unstructured Word or Excel document, you assign an identifier for the requirement information you want to import to
RuleID="2"	

Text in parser settings section

How is the setting used?

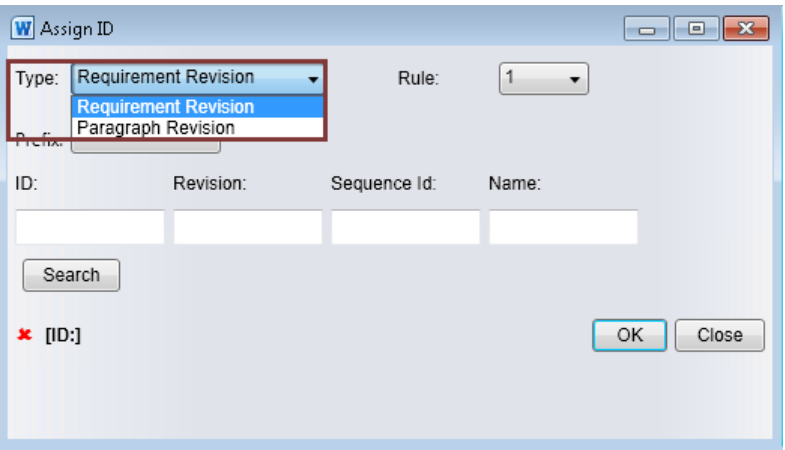
Teamcenter. In the **Assign ID** dialog box, the list of valid rules from the parser settings is displayed when you click the **Rule** box. You specify the rule that you want to use for assigning the requirement identifiers and types, and determining the trace link behavior.

Note:
If there is only one rule defined for a specific type, the **Rule** box is not displayed in the **Assign ID** dialog box.



PrimaryKeywordType =
"Requirement Revision"
SecondaryKeywordType =
"Paragraph Revision"

In the **Assign ID** dialog box, the list of valid types is displayed when you click the **Type** box. The valid types are determined by the rule you specify in the **Rule** box. In this example, **Requirement Revision** and **Paragraph Revision** are valid types for rule 1.



PrimaryKeyword = "[[ID:REQ-[0-9]
{6}]]"

In the **Assign ID** dialog box, you type the valid values to create a requirement or paragraph identifier. In this example, using

Text in parser settings section

How is the setting used?

rule 1 and **Type** set to **Requirement Revision** the identifier must match the "[ID:REQ-[0-9]{6}]" regular expression value set for the **Primary Keyword** setting. The green check mark is displayed to indicate the [ID:REQ-000100] identifier is a valid identifier.

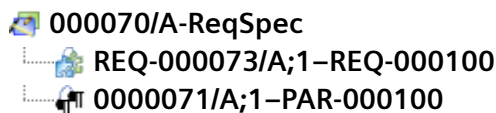
LinkType = "FND_Tracelink"

LinksAs = "Complying"

In the unstructured document, you import the following identifiers validated from rule 1 to Teamcenter:

- [ID:REQ-000100] requirement revision
- [CID:PAR-000100] paragraph revision

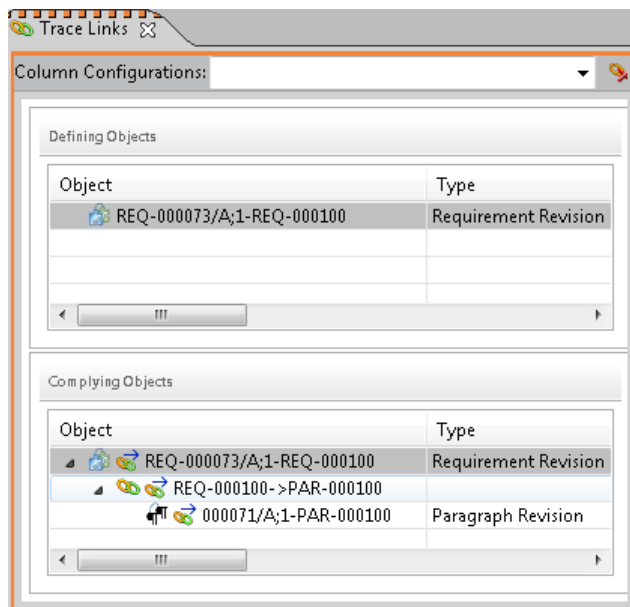
In this example, the **REQ-000100** requirement revision is the primary object and the **PAR-000100** paragraph revision is the secondary object. After importing the unstructured document to Teamcenter, the following requirement structure is created.



A **FND_Tracelink** trace link is created between the **REQ-000100** requirement revision and the **PAR-000100** paragraph revision. Because **LinksAs** is set to **Complying**, the requirement revision is the defining link and the paragraph revision is the complying link. The trace link is classified as an *internal* trace link since the paragraph revision is within the same structure as the requirement revision.

Text in parser settings section

How is the setting used?



Note:

If the **CreateSecondaryKeyword** setting was set to **false**, the **PAR-000100** paragraph revision would have to exist in Teamcenter for the trace link to be created. The trace link is then classified as an *external* trace link since the paragraph revision is not within the same structure as the requirement revision.

Configuring the property assignments

The *property assignments* section of the configuration file specifies the property assignments that are used by the parser. The property assignment determines what properties are associated to a particular object. This section can remain empty if no properties need to be updated.

A property assignment has the following format:

```
<add name = "[[]P:Description:*[]]" databasename = "object_desc"
AssignedTo = "[[]ID:REQ-[0-9]{6}[]]" />
```

Note:

Any property that is a *mandatory* property must be specified in the **property assignments** section, so it contains a valid value. The import fails if a mandatory property of the imported

item is a null value.

Properties on persistent objects such as item revision properties are supported. Properties on runtime objects are *not* supported.

The following describes the parameters you can configure.

Setting name	Description
Name	Specifies the regular expression to be used to match the property and its value. For example, if you want to set a property called Description on an object, you set the Name attribute in the unstructured document using the following format: <pre>[P:Description:This is test description.]</pre>
DatabaseName	Specifies the database name of the property. You should contact your Teamcenter administrator to get the database name for the property.
AssignedTo	Specifies the list of patterns for the object identifiers that you assign to the property in the database. The patterns are separated by ^ character. If a valid property assignment is found in the unstructured document that matches the object identifier, the property is then set for that object in the database.

Example: How are the property assignments used?

This example uses the following values for the property assignments section of the configuration file for Word:

```
<PropertyAssignments>
<add name = "[[]P:TXT:*[[]]" databasename = "body_text"
AssignedTo = "[[]ID:REQ-[0-9]{6}[[]]^[[[]CID:PAR-[0-9]{6}[[]]]"/]>
<add name = "[[]P:Description:*[[]]" databasename = "object_desc"
AssignedTo = "[[]ID:REQ-[0-9]{6}[[]]]"/>
</PropertyAssignments>
```

The following information displays how some of the settings are used in the property assignments section.

Text in property settings section

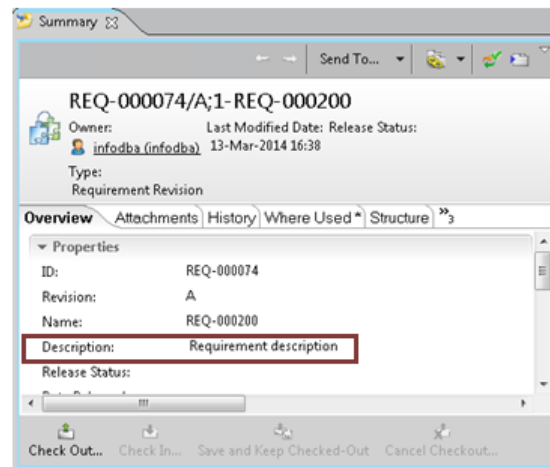
```
<add name = "[[P:Description:*[]]"
databasename = "object_desc" AssignedTo =
"[[ID:REQ-[0-9]{6}[]]"/>
```

How is the setting used?

In the unstructured document, you assign the [ID:REQ-000200] requirement revision identifier and the [CID:PAR-000200] paragraph revision identifier. You type the property setting in the following format in the unstructured document to set the **Description** property for the requirement revision.

```
[ID:REQ-000200]
[P:Description:Requirement
description]
[CID:PAR-000200]
```

Because the identifier matches the regular expression used for the property assignment, after the import to Teamcenter, the **Description** property of the requirement revision is set to **Requirement description**.



Configuring the color assignments

The *color assignments* section of the configuration file specifies the colors used for highlighting the identifiers when previewing an unstructured document.

A color assignment has the following format:

```
<add key ="NewItem" value ="0,255,0"/>
```

You use the RGB color model to specify the color value. The flat file import feature supports only Microsoft Office standard font colors.

Tip:

To determine the color value of a standard Office font color, choose to format a font color, choose the color on the **Standard** tab of the **Colors** dialog box, and then click the **Custom** tab to see the RGB value.

The following describes the parameters you can configure.

Key setting name	Description
NewItem	Highlights the object in the preview document if it doesn't exist in Teamcenter. Default color value: 0,255,0 (green)
ExistingItem	Highlights the object in the preview document if it exists in Teamcenter in the same requirement specification when you reimport the unstructured document to Teamcenter to revise or update requirement information. Default color value: 255,153,204 (pink)
DuplicateItem	Highlights the object in the preview document if there is a duplicate in Teamcenter that was created in a different requirement specification when you import the unstructured document. Default color value: 255,0,0 (red)
InternalLinks	Highlights the object in the preview document if it contains an internal trace link. Internal trace links are links to objects within the same structure. Default color value: 255,255,0 (yellow)
ExternalLinks	Highlights the object in the preview document if it contains an external trace link. External trace links are links to objects outside the structure. Default color value: 255,102,0 (orange)

Example: How are the color assignments used?

This example uses the following values for the parser rules and color assignments section of the configuration file for Word.

```
<ParserRules>
<add
  RuleID="1"
  LinkType = "FND_Tracelink"
  PrimaryKeyword = "[[]ID:REQ-[0-9]{6}[]]"
  PrimaryKeywordType = "Requirement Revision"
```



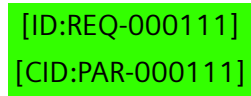
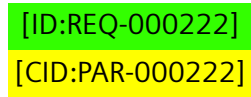
```

CreatePrimaryKeyword = "true"
SecondaryKeyword = "[[]CID:PAR-[0-9]{6}[]]"
SecondaryKeywordType = "Paragraph Revision"
CreateSecondaryKeyword = "true"
LinksAs = "Complying"/>
</ParserRules>

<ColorAssignments>
<add key = "NewItem" value = "0,255,0"/> <!--green-->
<add key = "ExistingItem" value = "255,153,204"/> <!--pink-->
<add key = "DuplicateItem" value = "255,0,0"/> <!--red-->
<add key = "InternalLinks" value = "255,255,0"/> <!--yellow-->
<add key = "ExternalLinks" value = "255,102,0"/> <!--orange-->
</ColorAssignments>

```

The following information displays how some of the settings are used in the color assignments section.

Possible scenarios	What colors are used for the Preview?
Scenario 1:	When you preview the document, the following colors are assigned:
1. In the unstructured document, you assign the [ID:REQ-000111] requirement revision identifier and the [CID:PAR-000111] paragraph revision identifier. The [ID:REQ-000111] requirement and [CID:PAR-000111] paragraph revisions do <i>not</i> exist in Teamcenter.	 Both objects are highlighted in green because:
2. You perform the Import New command to import the unstructured document to Teamcenter, click Preview and do <i>not</i> select the option to create trace links.	• The [ID:REQ-000111] requirement and [CID:PAR-000111] paragraph revisions do <i>not</i> exist in Teamcenter. • No trace links are created between the objects.
Scenario 2:	When you preview the document, the following colors are assigned:
1. In the unstructured document, you assign the [ID:REQ-000222] requirement revision identifier and the [CID:PAR-000222] paragraph revision identifier. The [ID:REQ-000222] requirement and [CID:PAR-000222] paragraph revisions do <i>not</i> exist in Teamcenter.	 The [ID:REQ-000222] requirement revision is highlighted in green because

Possible scenarios

What colors are used for the Preview?

2. You perform the **Import New** command to import the unstructured document to Teamcenter, click **Preview** and select the option to create trace links.

it is a new object and does not exist in Teamcenter.

The **[CID:PAR-000222]** paragraph revision is highlighted in yellow because it is an *internal* defining trace link and the requirement revision is the complying link. It is an internal trace link because:

- The requirement objects are created by the document.
- The trace links are created by the document.

Scenario 3:

1. In the unstructured document, you assign the **[ID:REQ-000333]** requirement revision identifier and the **[CID:PAR-000333]** paragraph revision identifier to the **ReqSpec1** requirement specification.

The **[ID:REQ-000333]** requirement revision already exists in a different requirement specification in Teamcenter.

The **[CID:PAR-000333]** paragraph revision already exists in a different requirement specification in Teamcenter.

2. You perform the **Import New** command to import the unstructured document to Teamcenter, click **Preview** and do *not* select the option to create trace links.

When you preview the document, the following colors are assigned:

[ID:REQ-000333]
[CID:PAR-000333]

The **[ID:REQ-000333]** requirement revision is highlighted in red because there is a duplicate object that exists in a different requirement specification in Teamcenter.

The **[CID:PAR-000333]** paragraph revision is highlighted in red because there is a duplicate object that exists in a different requirement specification in Teamcenter.

Scenario 4:

1. In the unstructured document, you assign the **[ID:REQ-000444]** requirement revision identifier and the **[CID:PAR-000444]** paragraph revision identifier to the **ReqSpec2** requirement specification.

The **[ID:REQ-000444]** requirement revision already exists in Teamcenter in the **ReqSpec2** requirement specification.

When you preview the document, the following colors are assigned:

[ID:REQ-000444]
[CID:PAR-000444]

Both objects are highlighted in pink because the **[ID:REQ-000444]** requirement revision and the **[CID:PAR-000444]** paragraph revision

Possible scenarios	What colors are used for the Preview?
The [CID:PAR-000444] paragraph revision already exists in Teamcenter in the ReqSpec2 requirement specification. 2. You perform the Import Update command to reimport the unstructured document to Teamcenter, click Preview and select the option to create trace links.	exists in Teamcenter in the same ReqSpec2 requirement specification.
Scenario 5: 1. In the unstructured document, you assign the [ID:REQ-000555] requirement revision identifier and the [CID:PAR-000555] paragraph revision identifier. In the parser rules in the configuration file, the CreateSecondaryKeyword setting is set to false so the secondary object is not created. The [ID:REQ-000555] requirement revision does <i>not</i> exist in Teamcenter. The [CID:PAR-000555] paragraph revision already exists in Teamcenter. 2. You perform the Import New command to import the unstructured document to Teamcenter, click Preview and select the option to create trace links.	When you preview the document, the following colors are assigned: [ID:REQ-000555] [CID:PAR-000555] The [ID:REQ-000555] requirement revision is highlighted in green because it does not exist in Teamcenter. The requirement revision is created after the import. The [CID:PAR-000555] paragraph revision is highlighted in orange because it is an <i>external</i> defining trace link and the requirement revision is the complying link. It is an external trace link because the paragraph revision resides in a different requirement structure than the requirement revision.

Defining the requirement objects and trace links in the unstructured document

Defining the requirement objects by assigning identifiers in the unstructured document

You define the requirement objects in the unstructured document by assigning identifiers. Teamcenter validates the identifiers by the rules you define in the flat file configuration files. When you assign an identifier that is not valid, an error displays that the identifier does not match the configuration rules. The configuration rules also specify if trace links are created between the primary and secondary requirement objects after the import to Teamcenter. An *internal* trace link is linked to requirement objects within the same requirement structure. An *external* trace link is linked to requirement objects in a different requirement structure.

Note:

In Excel, there is a limitation when assigning multiple IDs.

You can perform the following when you assign identifiers:

- Assign identifiers for new requirement objects that do not exist in Teamcenter.
If the configuration parser rules specify that trace links are created between the primary and secondary requirement objects, *internal* trace links are created after the import to Teamcenter.
- Search for and insert identifiers for requirement objects that already exist in Teamcenter.
If the configuration parser rules specify that trace links are created between the primary and secondary requirement objects, *external* trace links to existing objects in different structures are created after the import to Teamcenter.

Note:

To avoid errors during the import for requirement objects that already exist in Teamcenter, in the flat file configuration files, do the following:

- For existing primary requirement objects, set the **CreatePrimaryKeyword** setting to **false**.
- For existing secondary requirement objects, set the **CreateSecondaryKeyword** setting to **false**.

You use the **Assign ID** dialog box to create the identifiers. In the **Assign ID** dialog box, values such as **Type**, **Rule**, **Prefix**, **ID**, and **Name** are checked for validation according to the configuration rules set up in the flat file configuration files. Once the values are validated, a green check mark displays next to the identifier tag at the bottom of the dialog box.

Note:

You can customize these properties by modifying the **DisplayName** business constant using the Business Modeler IDE.

The **Assign ID** dialog is automatically resized based on the list of display properties.

Note:

Properties on persistent objects such as item revision properties are supported. Properties on runtime objects are *not* supported.

1	Type of requirement object.	The list of valid of requirement object types is specified by the PrimaryKeywordType and SecondaryKeywordType settings in the configuration parser rule section for a specified rule ID. You select the rule ID from the Rule box.
2	Parser rule used for validating identifiers for the requirement objects and creating trace links.	The list of valid rule IDs is specified by the RuleID setting for the configuration parser rule. If there is only one parser rule defined, the Rule box is not displayed.
3	Prefix of the requirement object.	The list of valid prefixes is specified by the IDPrefix setting in the configuration common settings section.
4	Property values for creating a valid Identifier for the requirement object.	The valid identifier is defined by the PrimaryKeyword or SecondaryKeyword settings in the configuration parser ruler section for the specified rule ID. You select the rule ID from the Rule box. If you set the PrimaryKeywordType setting as the type, the identifier is validated by matching the regular expression defined by the PrimaryKeyword setting. If you set the SecondaryKeywordType setting as the type, the identifier is validated by matching the regular expression defined by the SecondaryKeyword setting. You use delimiters in the regular expression Value to designate where a property begins and ends. The property values are validated by matching the regular expression for the property. In this example, in the Assign ID dialog box, the ID, Revision, Sequence Id, and Name properties are validated against the <code>[[]ID:REQ/[A-Z]{1};[0-9]{1}-[0-9]{6}[]]</code> regular expression.

[[ID:REQ/[A-Z]{1};[0-9]{1}-[0-9]{6}]]

[[	ID	:	REQ/[A-Z]{1};[0-9]{1}-[0-9]{6}	]]

Range Boundary Open + Keyword + Separator Symbol + Value + Range Boundary Close

REQ/[A-Z]{1};[0-9]{1}-[0-9]{6}

REQ	/	(A-Z){1}	;	[0-9]{1}	-	[0-9]{6}
ID		Revision		Sequence Id		Name

W Assign ID

Type: Requirement Revision Rule: 1

Prefix: ID

ID:	Revision:	Sequence Id:	Name:
REQ	A	1	000200

Search

✓ [ID:REQ/A;1-000200]

OK Close

- 5 Search for existing requirement objects in Teamcenter. Clicking **Search** launches a **Search** dialog box that enables you to search for existing requirement objects in Teamcenter and insert the existing objects in the unstructured document. The **IDProperty** setting in the configuration flat file is used for the search criteria property when searching for existing requirement objects in Teamcenter.

W Search

Type: Requirement Revision

Search Criteria *9999*

Search

Name	Object	Item ID
REQ-999999	REQ-000045/A;1-REQ-999999	REQ-000045

Close

- 6 Green check mark

indicates valid identifier.

Example: Assigning valid identifiers in the unstructured document

The following information displays some examples using regular expressions that you might specify in the configuration file rules and some corresponding valid identifier tags when assigning IDs in Word or Excel.

In these examples, the key values are set to the following values in the common settings section of the configuration file:

```
<add key = "IDPrefix" value = "ID,ID2,ID3,CID" />
<add key = "IDProperty" value = "object_name" />
```

Regular expression for identifier	Valid values when assigning an ID	Some valid identifier tags
[]ID:ENG-FNC-TSC[0-9]{1,}[]	Type: Requirement revision Prefix: ID ID: ENG-FNC Name: TSC followed by a sequence number	Valid identifier tag: [ID:ENG-FNC-TSC001] Object name in Teamcenter after import: ENG-FNC-TSC0001 Valid identifier tag: [ID:ENG-FNC-TSC123] Object name in Teamcenter after import: ENG-FNC-TSC123
[]CID:ENG-FNC-FSR[0-9]{1,}[]	Type: Paragraph revision Prefix: CID ID: ENG-FNC Name: FSR followed by a sequence number.	Valid identifier tag: [CID: ENG-FNC-FSR0001] Object name in Teamcenter after import: ENG-FNC-FSR0001 Valid identifier tag: :[CID: ENG-FNC-FSR111] Object name in Teamcenter after import: ENG-FNC-FSR111
[]ID:SYSCP[0-9]{1,}[]	Type: Requirement revision Prefix: CID ID: SYSOP followed by a sequence number.	Valid identifier tag: [ID:SYSOP0001] Object name in Teamcenter after import: SYSOP0001 Valid identifier tag: [ID:SYSOP1001] Object name in Teamcenter after import: SYSOP1001

Regular expression for identifier	Valid values when assigning an ID	Some valid identifier tags
[[]ID:REQ/[A-Z]{1}; [0-9]	Type: Requirement revision	Valid identifier tag: [ID: REQ/A;1–100001]
	Prefix: ID	Object name in Teamcenter after import: REQ/A;1–100001
; [0-9]{1} - [0-9]{6}	ID: REQ	Valid identifier tag: [ID: REQ/B;2–222222]
[]	Revision: Any letter A to Z	Object name in Teamcenter after import: REQ/A;1–222222
	Sequence: Any integer 0–9	
	Name: Six digit sequence number	

Assign identifiers for new requirement objects in the unstructured document

1. In the unstructured document, place the cursor where you where you want to insert the identifier. For example, you would place the cursor before the text that you want to be identified as a requirement.
2. Click the **Teamcenter** tab→**Requirements** group→**Flat File Import** and choose **Assign ID**.
3. If the **Teamcenter Login** dialog box is displayed, enter your Teamcenter user name and password, and then click **Login**.
4. In the **Assign ID** dialog box, enter the following information.

Field name	Definition	Valid values
Rule	The rule ID used for the identifier parsing. The field is unavailable if there is only one rule defined in the flat file configuration files.	Select from list of valid rule IDs. Rule IDs are defined in the flat file configuration files.
Type	Type of requirement object for the identifier.	Select from list of valid requirement object type for the selected rule. Requirement object types are defined in the flat file configuration files.
Prefix	The identifier prefix used to match a specific rule.	Select from list of valid prefixes. Prefixes are defined in the flat file configuration files.

5. Type the values in the properties boxes to create a valid identifier for the selected rule, type, and prefix.

Identifiers are validated if they match the regular expression defined in the flat file configuration files.

If you do not type a value for a property, the identifier is created without the property.

By default, the properties boxes for **Requirement revision** and **Paragraph revision** object types are:

- ID
- Revision
- Sequence Id
- Name

By default, the properties boxes for **Requirement** and **Paragraph** object types and subtypes are:

- ID
- Name

Once the values are validated, a green check mark displays next to the identifier tag at the bottom of the dialog box.

The screenshot shows the 'Assign ID' dialog box with the following configuration:

- Type: Requirement Revision
- Rule: 1
- Prefix: ID
- ID: REQ
- Revision: (empty)
- Sequence Id: (empty)
- Name: 000100

The 'Search' button is located below the input fields. At the bottom of the dialog, a green checkmark is displayed next to the identifier '[ID:REQ-000100]'. The 'OK' and 'Close' buttons are at the bottom right.

6. Click **OK** to create the identifier.

An identifier of the specified requirement type is created in the unstructured document. Depending on the configuration rules defined in the flat file configuration files, the identifier may be a complying or defining trace link.

7. Place the cursor where you want to create additional identifiers and repeat steps 1–5.

- Click **Close** to close the dialog box.

Insert identifiers for existing requirement objects in the unstructured document

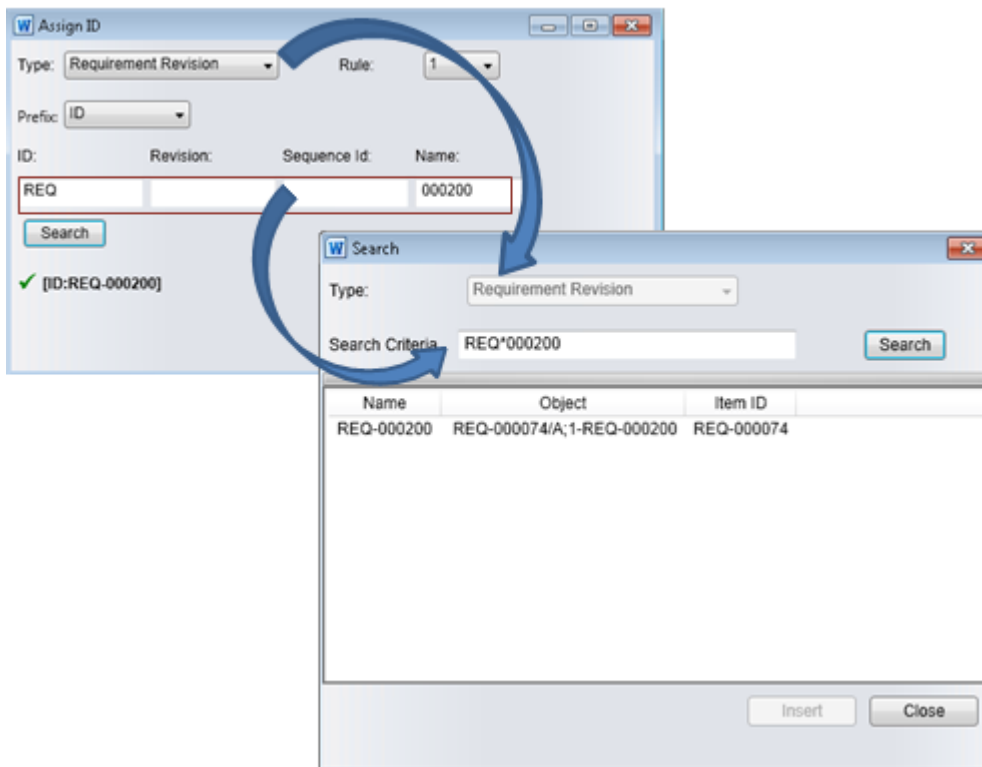
- In the unstructured document, place the cursor where you want to insert the identifier.
- Click the **Teamcenter** tab→**Requirements** group→**Flat File Import** and choose **Assign ID**.
- If the **Teamcenter Login** dialog box is displayed, enter your Teamcenter user name and password, and then click **Login**.
- In the **Assign ID** dialog box, enter the following information.

Field name	Definition	Valid values
Rule	The rule ID used for the identifier parsing. Unavailable if there is only one rule defined in the flat file configuration files	Select from list of valid rule IDs. Rule IDs are defined in the flat file configuration files.
Type	Type of requirement object for the identifier.	Select from list of valid requirement object type for the selected rule. Requirement object types are defined in the flat file configuration files.
Prefix	The identifier prefix used to match a specific rule.	Select from list of valid prefixes. Prefixes are defined in the flat file configuration files.

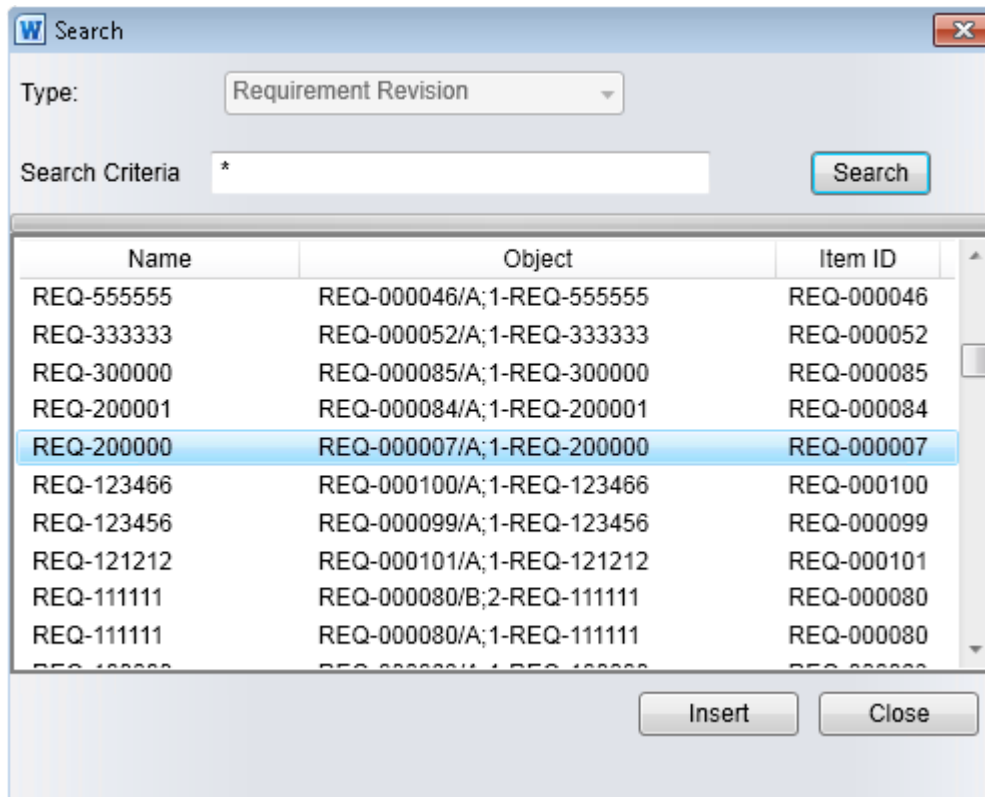
- Optionally, type the values in the properties boxes to search for a specific identifier.
- Click **Search**.

In the **Search** dialog box, the requirement object type is populated in the **Type** box.

If the properties for the identifier have values, the **Search Criteria** box is populated with the identifier from the **Assign ID** dialog box. If you do not type values for the properties in the **Assign ID** dialog, an asterisk (*) representing a wild card is used as the search criteria.



7. Optionally, type a different value in the **Search Criteria** box for searching identifiers of the specified type.
8. In the **Search** dialog box, click **Search**.
9. From the results list, select the requirement object you want to add to the unstructured document.



10. Click **Insert**.

The **Assign ID** dialog is displayed and validates the identifier for the inserted requirement object is valid.

11. Click **OK** to insert the identifier.

An existing identifier of the specified requirement type is inserted in the unstructured document. Depending on the configuration rules defined in the flat file configuration files, the identifier may be a complying or defining trace link.

12. Place the cursor where you want to create additional identifiers and repeat steps 1–10.
13. Click **Close** to close the dialog box.

Assign additional property values for the requirement objects in the unstructured document

The **Property assignments** section of the configuration file specifies the property assignments that are used by the parser. The property assignment determines what properties are associated to a particular object.

Note:

Any property that is a *mandatory* property must be specified in the **property assignments** section, so it contains a valid value. The import fails if a mandatory property of the imported item is a null value.

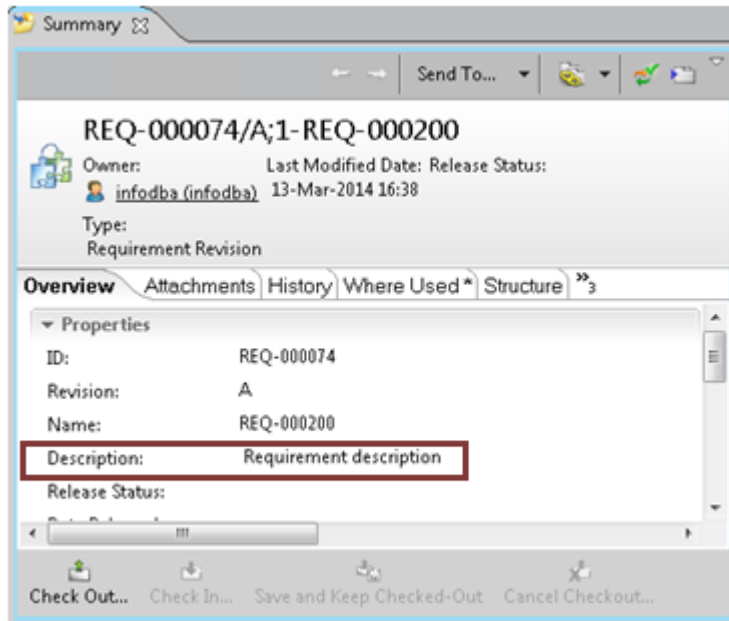
1. In the unstructured document, define new requirement objects or insert existing requirement objects using the **Assign ID** dialog box.
2. On the line following the requirement object identifier in the document, type the property assignment for the requirement object in the following format:
[P:Property name:Property value]

For example, the unstructured document contains the following requirement revision identifier and property assignment:

[ID:REQ-000200] ← Requirement revision identifier

[P:Description:Requirement description] ← Property assignment for requirement revision

After importing the unstructured document to Teamcenter, the **Description** property of the requirement revision is set to **Requirement description**.



Search for existing requirement objects in Teamcenter

1. In the unstructured document, click the **Teamcenter** tab→**Requirements** group→**Flat File Import** and choose **Search ID**.

2. In the **Search** dialog box, click the **Type** box and select the requirement object type you want to search for.
3. Type the value of the requirement object property you want to search for in the **Search criteria** box. Wildcards can be used as part of the search criteria.

Note:

The search criteria property is set to the **IDProperty** setting in the configuration flat files. The default property is set to **object_name**.

The search results are displayed.

4. Click **Close** to close the dialog box.

Previewing import file for unstructured documents

Previewing the import file for accuracy and errors

Before you import the unstructured document to Teamcenter, you can preview the import file to check for accuracy or errors. This enables you to make any corrective changes to the unstructured document before the import. You have the option of previewing the import file with or without trace links.

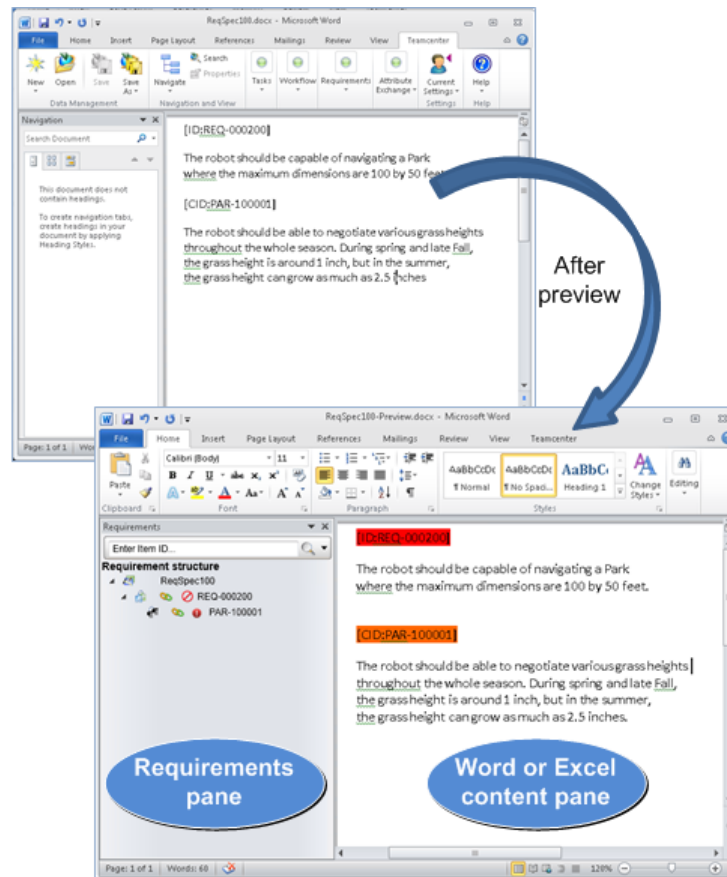
If you are importing the unstructured document for the first time, you preview the import file as a new import. If you are reimporting the unstructured document, you preview the import file as an updated or revised import.

Note:

If the **IDProperty** setting located in the flat file configuration files is not set to a unique property, but you still want to ensure identifier uniqueness when creating new requirement objects in Teamcenter, preview the import file *before* you import the unstructured document to Teamcenter after the first import. Any subsequent imports will display errors indicating the requirement objects already exist in Teamcenter.

For example, the **IDProperty** setting located in the flat file configuration files is set to **object_name** which is not a unique property. You preview the import file first before the actual import, and then import it as new unstructured document to Teamcenter. You make changes to the requirement objects and import it a second time as a new import. Teamcenter displays an error message indicating the requirement objects exist.

If you did *not* preview the import file first and import the unstructured document a second time as a new import, a new requirement specification containing new requirement objects are created in Teamcenter with the same object names but different item IDs than the first import.



The following occurs after the preview:

- A preview document is created with a requirements pane and has the text **-Preview** appended to the original filename and is stored in your local **temp** directory. When you import the unstructured document to Teamcenter, the preview document is also imported to maintain any corrective actions you made to the unstructured document after the preview.
- Requirement objects are highlighted with different colors in the content pane for the following conditions:

- **New requirement object**

Highlights the object in the preview document if it doesn't exist in Teamcenter.
Default color value: 0,255,0 (green)

- **Existing requirement object**

Highlights the object in the preview document if it exists in Teamcenter in the same requirement specification when you reimport the unstructured document to Teamcenter to revise or update requirement information.
Default color value: 255,153,254 (pink)

- **Duplicated requirement object**

Highlights the object in the preview document if there is a duplicate in Teamcenter that was created in a different requirement specification when you import the unstructured document.
Default color value: 255,0,0 (red)

- **Internal trace link**

Highlights the object in the preview document if it contains an internal trace link. Internal trace links are links to objects within the same structure.
Default color value: 255,255,0 (yellow)

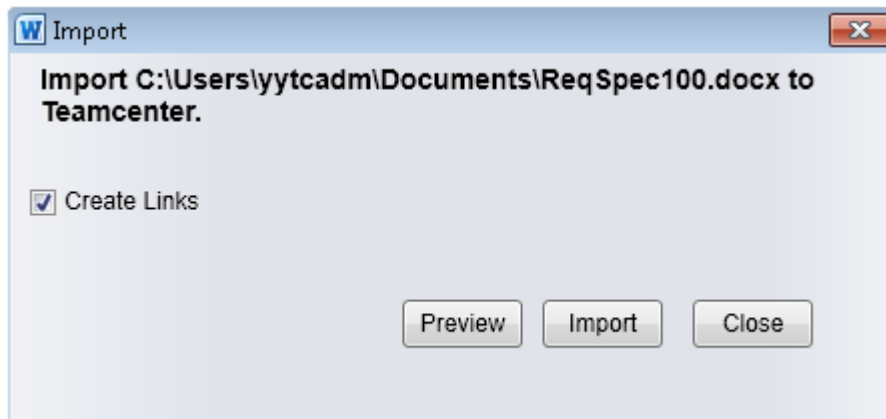
- **External trace link**

Highlights the object in the preview document if it contains an external trace link. External trace links are links to objects outside the structure.
Default color value: 255,102,0 (orange)

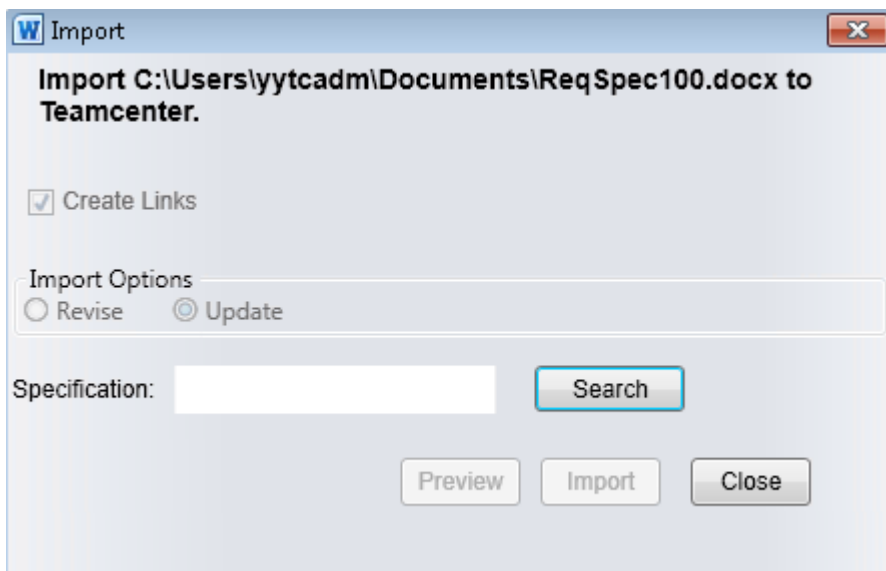
- Indicators are displayed next to the requirement object in the requirements pane for the following conditions:
 - A  symbol next to the requirement object indicates the object already exists in Teamcenter.
 - A red asterisk (*) next to the requirement object indicates the object does not exist in Teamcenter.
 - A  symbol next to the requirement object indicates the object is an unresolved trace link. You resolve the conflict by selecting the revision of the requirement object to resolve the trace link.
 - A  symbol next to the requirement object indicates the object trace link has been resolved.

Generate the preview document and resolve any conflicts

1. Choose one of the following options:
 - Preview the import file as a new import.
 - a. click the **Teamcenter** tab→**Requirements** group→**Flat File Import** and choose **Import New**.
 - b. Select the **Create Links** box to preview trace links.



- Preview the import file as an updated or revised import.
 - a. Click the **Teamcenter** tab→**Requirements** group→**Flat File Import** and choose **Import Update**.
 - b. Select the **Create Links** box to preview trace links.
 - c. Under **Import Options**, select **Revise** to preview the revised structure or **Update** to preview updated structure.

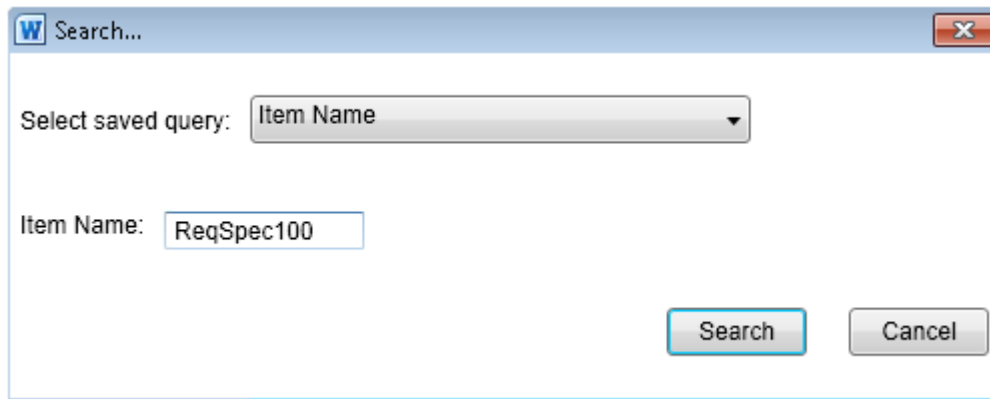


- d. Click **Search** to search for the specification you want to update or revise.
- e. In the **Search** dialog box, click the **Select saved query** box and select the property you want to search for.

The list of saved queries is displayed.

You can also create your own saved query in Teamcenter to search for a specific properties.

- f. Type the property value you want to search for.



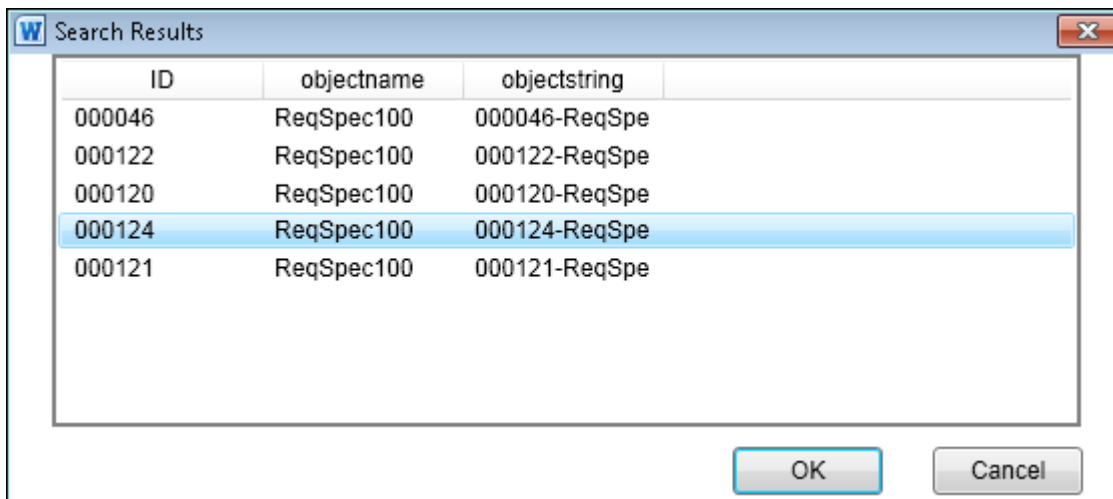
Search...

Select saved query: Item Name

Item Name: ReqSpec100

Search Cancel

- g. Click **Search**.
- h. In the **Search Results** dialog box, select the requirement specification you want to update or revise.



ID	objectname	objectstring
000046	ReqSpec100	000046-ReqSpe
000122	ReqSpec100	000122-ReqSpe
000120	ReqSpec100	000120-ReqSpe
000124	ReqSpec100	000124-ReqSpe
000121	ReqSpec100	000121-ReqSpe

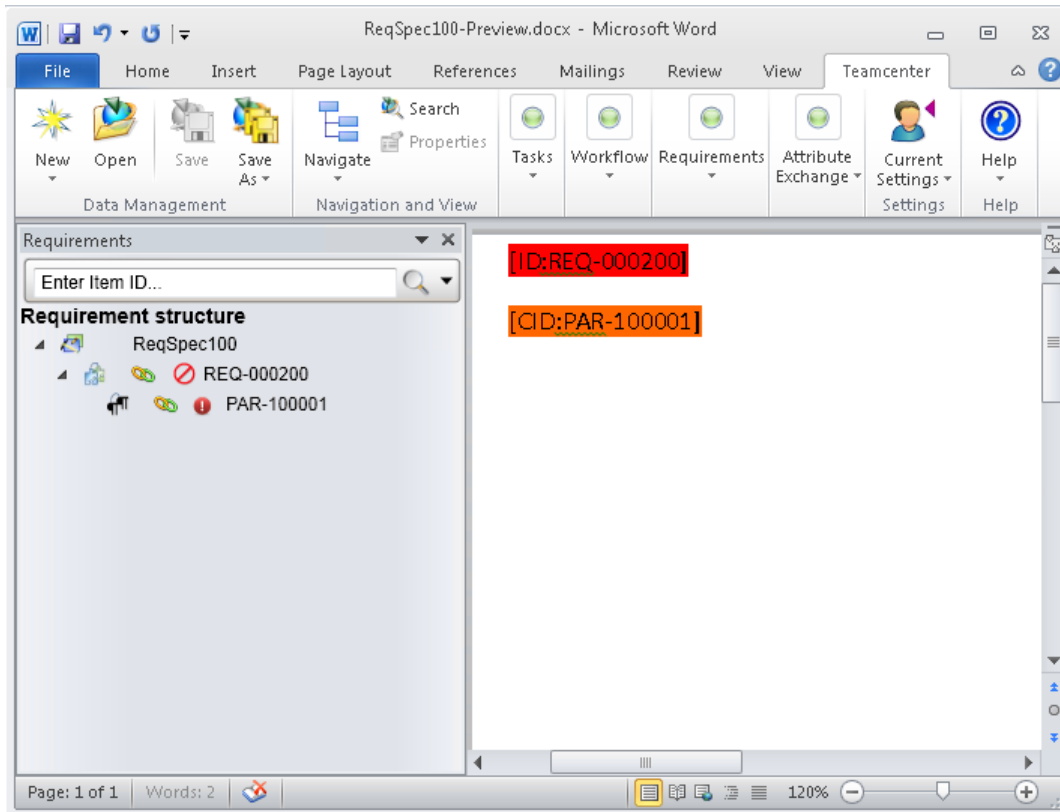
OK Cancel

- i. Click **OK**.

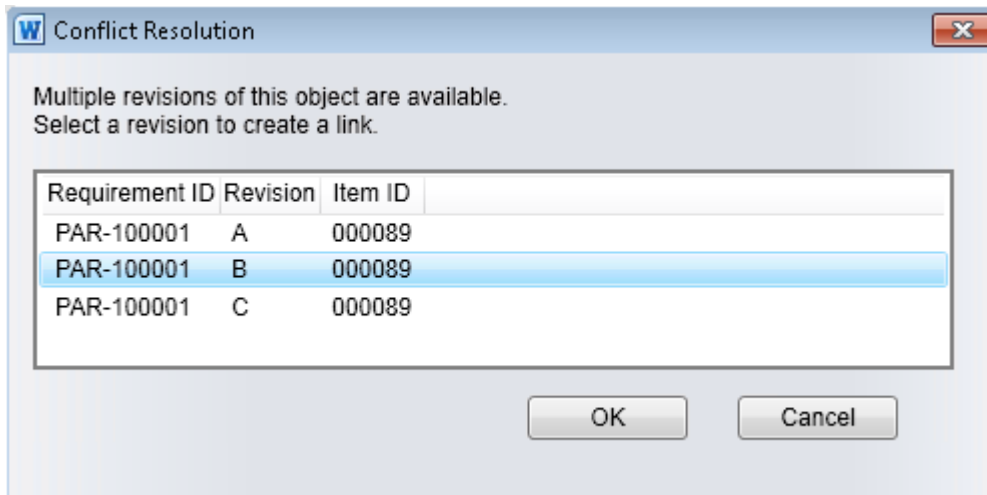
The **Specification** box displays the requirement specification you want to update or revise.

2. Click **Preview**.

The preview document is displayed and has the text **-Preview** appended to the original filename.



3. Click the indicator symbol in the requirements pane to display the warning or error.
4. If the  symbol is displayed next to the requirement object indicating the object is an unresolved trace link, perform the following steps to resolve the link:
 - a. Click the  symbol in the requirements pane.
 - b. In the **Conflict Resolution** box, select the revision of the requirement object to resolve the trace link.



- c. Click **OK**.

The  symbol is displayed next to the requirement object indicating the object trace link has been resolved.

Example: Previewing import file scenarios

Preview scenario	Highlighted colors in content pane	Indicators displayed in requirements pane
Scenario 1 The REQ-000200 requirement revision object already exists in Teamcenter. The PAR-000201 paragraph revision object does not exist in Teamcenter. You perform the following steps: - In the unstructured document, assign the [ID:REQ-000200] primary requirement revision identifier and the [CID:PAR-000201] secondary paragraph revision identifier. - Set the CreatePrimaryKeyword and CreateSecondaryKeyword parser settings to True. - Save the unstructured document as ReqSpec100.	[ID:REQ-000200] [CID:PAR-000201] The [ID:REQ-000200] requirement revision is highlighted in red because there is a duplicate object that exists in a different requirement specification in Teamcenter. The [CID:PAR-000201] paragraph revision is highlighted in yellow because it is an <i>internal</i> defining trace link and the requirement revision is the complying link. It is an internal trace link because the paragraph revision is contained within the same	▲  ReqSpec100 ▲  REQ-000200 ▲  PAR-000201 The  symbol displays next to the REQ-000200 requirement revision object indicates the object already exists in Teamcenter.

Preview scenario	Highlighted colors in content pane	Indicators displayed in requirements pane
4. Perform the Import New command to import the unstructured document to Teamcenter, click Preview and select the option to create trace links.	requirement structure as the requirement revision.	
Scenario 2 The REQ-000200 requirement revision object does not exist in Teamcenter. Multiple revisions of the PAR-000201 paragraph revision object exist in Teamcenter. You perform the following steps: 1. In the unstructured document, assign the [ID:REQ-000200] primary requirement revision identifier and the [CID:PAR-000201] secondary paragraph revision identifier. 2. Set the CreatePrimaryKeyword and CreateSecondaryKeyword parser settings to False. 3. Set the LinksAs parser setting to Complying. 4. Save the unstructured document as ReqSpec100. 5. Perform the Import New command to import the unstructured document to Teamcenter, click Preview and select the option to create trace links.	[ID:REQ-000200] [CID:PAR-000201] The [ID:REQ-000200] requirement revision is highlighted in green because it does not exist in Teamcenter. The [CID:PAR-000201] paragraph revision is highlighted in orange because it is an <i>external</i> defining trace link and the requirement revision is the complying link. It is an external trace link because the paragraph revision resides in a different requirement structure than the requirement revision.	▲ ReqSpec100 ▲  REQ-000200    PAR-000201 The  symbol displays next to the PAR-000200 paragraph revision object because it is an unresolved trace link. Because there are multiple revisions of the paragraph revision, Teamcenter does not know which revision of the paragraph revision to use as the defining trace link.
Scenario 3 You resolve the trace link from scenario 2 by performing the following steps:	[ID:REQ-000200] [CID:PAR-000201] The [ID:REQ-000200] requirement revision is	▲  ReqSpec100 ▲  REQ-000200    PAR-000201

Preview scenario	Highlighted colors in content pane	Indicators displayed in requirements pane
1. Click the  symbol displays next to the PAR-000201 paragraph revision in the requirements pane.	highlighted in green because it does not exist in Teamcenter.	The  symbol displays next to the PAR-000201 paragraph revision object because you have resolved the trace link conflict by selecting the specific paragraph revision.
2. In the Conflict Resolution dialog box, select the paragraph item revision you want to use to resolve the trace link.	The [CID:PAR-000201] paragraph revision is highlighted in orange because it is an <i>external</i> defining trace link and the requirement revision is the complying link. It is an external trace link because the paragraph revision resides in a different requirement structure than the requirement revision.	

Clear highlights from the preview document

After you preview the unstructured document, the requirement object identifiers in the Word or Excel content pane are highlighted in different colors depending on the condition of the requirement object.

To remove the highlights from the preview document, click the **Teamcenter** tab→**Requirements** group→**Flat File Import** and choose **Clear Highlights**.

The requirement object identifiers in the Word or Excel content pane are no longer highlighted.

Importing the unstructured document to Teamcenter

Importing a new requirement specification for the unstructured document

To create a new requirement specification containing requirement objects defined by the identifiers in the unstructured document, you import the unstructured document as a *new* import. After the import, the requirement specification *object name* is set to the *name* of the unstructured document. You have the option of importing the unstructured document with or without trace links.

If the import fails with an error, the entire import transaction is rolled back and the import operation is aborted.

The following occurs in Teamcenter after the import is successfully completed:

- A new requirement structure is created in Teamcenter. The requirement specification or specification subtype is set to the unstructured document name and contains the imported items. The imported items are created from the valid identifiers for the requirement information such as requirement and

paragraph revisions that are specified in the unstructured document.

The rules you define in the flat file configuration files validate the identifier tags and control their behavior.

- The requirement structure is imported as a *precise* structure.
- A dataset is created containing the contents of the unstructured document and is associated with the requirement specification. Any Word or Excel macros in the unstructured document are preserved in the dataset.

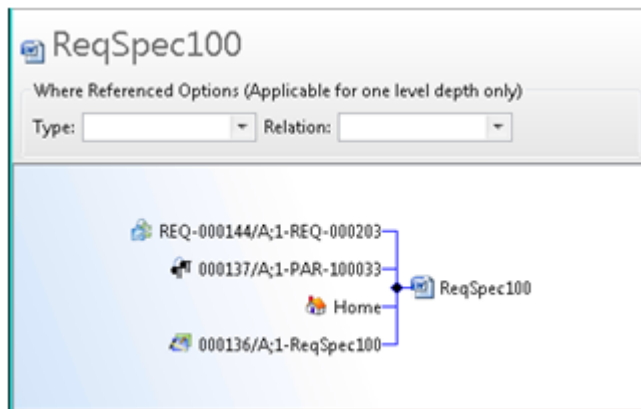
Note:

If you have previewed the unstructured document before the import, the unstructured document name and the dataset name has the text **-Preview** appended to the original name.

- The dataset is linked to the requirement objects contained in the requirement specification.

Example:

The **ReqSpec100** dataset and its linked requirement information displayed in the **Impact Analysis** view.



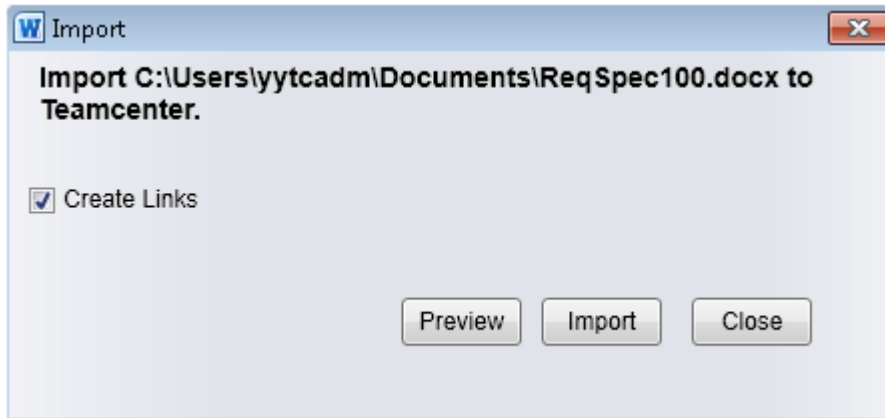
- External and internal trace links are created in Teamcenter that are defined in the unstructured document.
The rules you define in the flat file configuration files determine the trace link behavior.
- The body text of the requirement objects is set to the URL of the dataset.

Note:

When importing an unstructured document, the formatting for body text may not be preserved.

Import the unstructured document for the first time

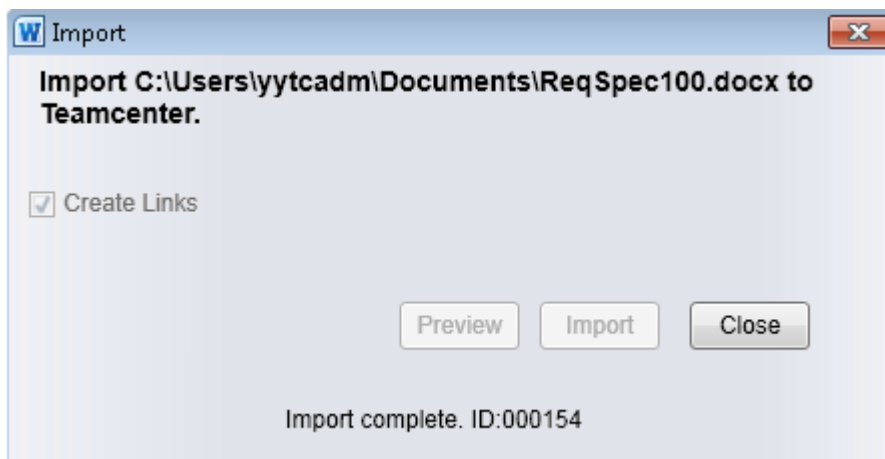
1. In the unstructured document, click the **Teamcenter** tab→**Requirements** group→**Flat File Import** and choose **Import New**.
2. In the **Import** dialog box, select the **Create Links** box to create the trace links in Teamcenter after the import.



If **Create Links** box is not selected, no trace links are created after the import.

3. Click **Import**.

When the import is complete, the item ID of the new requirement specification created in Teamcenter is displayed.



Example: Importing a new requirement specification for the unstructured document

The following example shows importing the **ReqSpec1.doc** unstructured Word document as a new import to Teamcenter. The option to create trace links for the import was selected. The rules you define in the flat file configuration files determine the identifier tag format and its behavior.

For this example, the flat file configuration file for Word contains the following common settings and parser rule settings:

```
Common Settings>
<add key = "Parser" value = "Teamcenter.TcIDToIDParser.WordIDParser;
C:\Program Files (x86)\Siemens\Teamcenter
9\TcWord2K7DocImp\TcIDToIDParser.dll"/>
<add key = "PropDebugComments" value = "false"/>
<add key = "LinkDebugComments" value = "false"/>
<add key = "DefaultSpecType" value="RequirementSpec" />
<add key = "IDValidationPattern" value="[[]*:[A-Za-z0-9_]{1,}-[0-9]{6}
[]]"/>
<add key = "IDPrefix" value = "ID,ID2,ID3,CID" />
<add key = "IDProperty" value = "object_name" />
<add key = "Default Saved Query" value = "Item ID" />
<add key = "Search" value="00TBSearch" />
</Common Settings>

<ParserRules>
<add
  RuleID="1"
  LinkType = "FND_Tracelink"
  PrimaryKeyword = "[[]ID:REQ-[0-9]{6}[]]"
  PrimaryKeywordType = "Requirement Revision"
  CreatePrimaryKeyword = "true"
  SecondaryKeyword = "[[]CID:PAR-[0-9]{6}[]]"
  SecondaryKeywordType = "Paragraph Revision"
  CreateSecondaryKeyword = "true"
  LinksAs = "Complying"/>

<add
  RuleID="2"
  LinkType = "FND_Tracelink"
  PrimaryKeyword = "[[]ID:ENG-[0-9]{6}[]]"
  PrimaryKeywordType = "Requirement Revision"
  CreatePrimaryKeyword = "true"
  SecondaryKeyword = "[[]CID:ENG-[0-9]{6}[]]"
  SecondaryKeywordType = "Paragraph Revision"
  CreateSecondaryKeyword = "false"
  LinksAs = "Complying"/>
</ParserRules>
```

The following steps are performed:

1. Using rule 1, the **REQ-000100** requirement revision and **PAR-000102** paragraph revision identifiers are assigned in the unstructured **ReqSpec1.doc** Word document

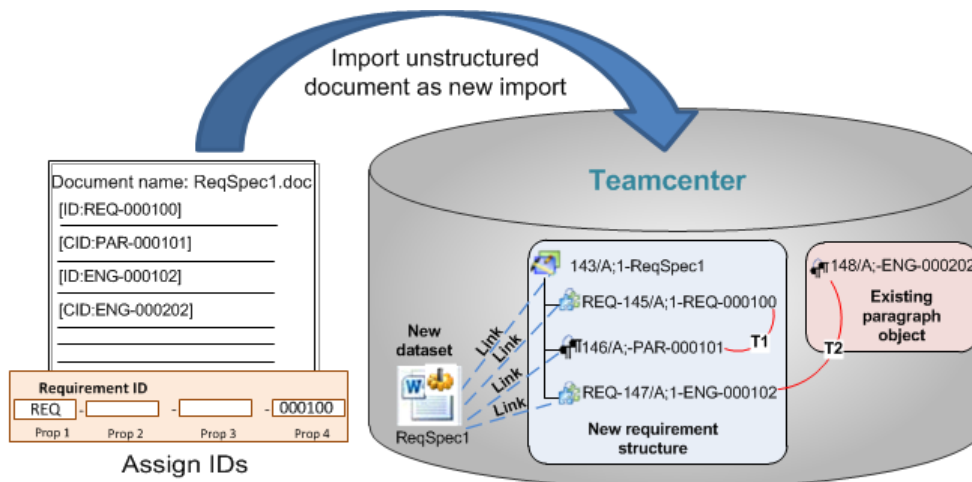
- Using rule 2, the **ENG-000102** requirement revision and the **ENG-000202** paragraph revision identifiers are assigned.

The contents of the unstructured document now contain the following identifiers:

```
[ID:REQ-000100]
[CID:PAR-000101]
[ID:ENG-000102]
[CID:ENG-000202]
```

- The unstructured document is imported as a *new* import. The option to create trace links for the import was selected.

After the import to Teamcenter, the following occurs:

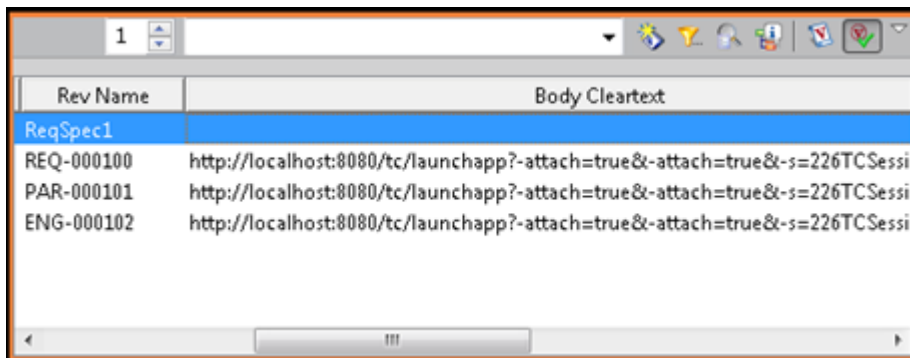


- The **ReqSpec1** dataset is created with links to the requirement objects in the **143/A;1-ReqSpec1** requirement specification.
- The **143/A;1-ReqSpec1** requirement specification is created in Teamcenter containing the following:
 - The new **REQ-145/A;1-REQ-000100** requirement revision is created.
 - The new **146/A;1-PAR-000101** paragraph revision is created.
 - The **REQ-147/A;1-ENG-000102** requirement revision is created.
 - Because the **CreateSecondaryKeyword** is set to **True** in the parser settings for rule 1 and the **146/A;1-PAR-000101** paragraph revision does not exist in Teamcenter, an *internal T1* trace link is created between the **REQ-145/A;1-REQ-000100** requirement revision and the **146/A;1-PAR-000101** paragraph revision.

- Because the **CreateSecondaryKeyword** is set to **False** in the parser settings for rule 2 and the **148/A;-ENG-000202** paragraph revision already exists in Teamcenter, an *external T2* trace link is created between the **REQ-147/A;1-ENG-000102** requirement revision and the **148/A;-ENG-000202** paragraph revision.

Because the **IDProperty** setting in the flat file configuration file is set to **object_name**, the item IDs for the requirement objects are automatically assigned by Teamcenter after the import.

- The body text of the requirement objects in the **ReqSpec1** requirement specification is set to the URL of the **ReqSpec1** dataset.



The screenshot shows a table with two columns: 'Rev Name' and 'Body Cleartext'. The table contains three rows of data. The first row is highlighted in blue and shows 'ReqSpec1' in the 'Rev Name' column and a URL in the 'Body Cleartext' column. The second row shows 'REQ-000100' and the same URL. The third row shows 'PAR-000101' and the same URL. The fourth row shows 'ENG-000102' and the same URL. The table is part of a software application window with a toolbar at the top.

Rev Name	Body Cleartext
ReqSpec1	http://localhost:8080/tc/launchapp?-attach=true&-attach=true&-s=226TCSessi
REQ-000100	http://localhost:8080/tc/launchapp?-attach=true&-attach=true&-s=226TCSessi
PAR-000101	http://localhost:8080/tc/launchapp?-attach=true&-attach=true&-s=226TCSessi
ENG-000102	http://localhost:8080/tc/launchapp?-attach=true&-attach=true&-s=226TCSessi

Make changes to an imported unstructured document

If you import the unstructured document and then need to make changes to it, you can modify the dataset and then reimport the unstructured document as an update or a revision.

You can modify the dataset as follows:

- Add identifier tags for new objects and trace links by assigning new identifiers.
- Delete identifier tags for existing objects and trace links.

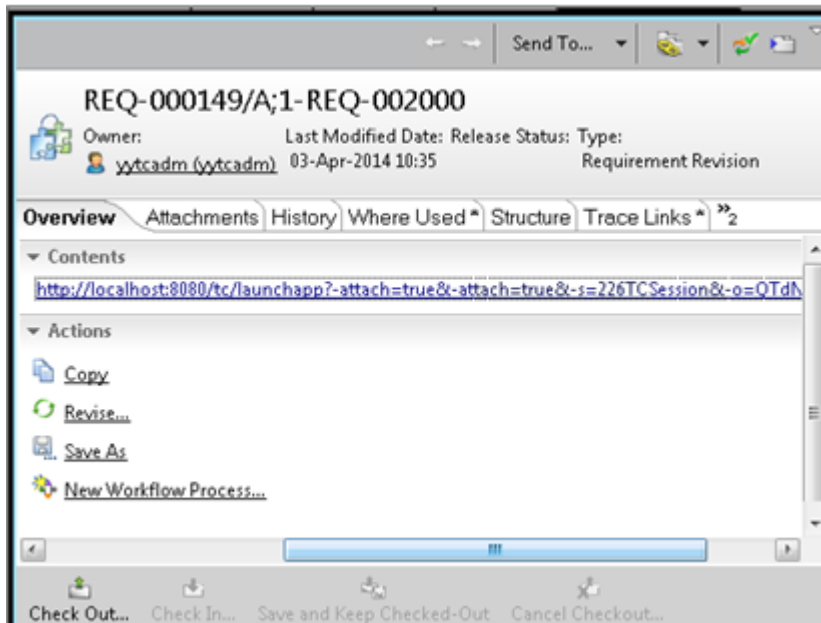
You use the standard Word or Excel features to delete identifier tags.

1. In Teamcenter, choose one of the following options to open the dataset that you want to modify:

- In the navigation pane, click **My Teamcenter**.
 - Search for the dataset you want to modify.
 - Double-click the dataset.

The dataset is automatically checked out and the stand-alone Microsoft Office Word or Excel is launched with the dataset.

- In the navigation pane, click **My Teamcenter** or **Systems Engineering**.
 - a. Search for any of the requirement objects that are contained within the dataset.
 - b. Right-click the requirement object and choose **Open With→Summary**.
 - c. In the **Summary** view, under the **Contents** heading, click the URL.



- d. In the **File Download** dialog box, click **Open** to launch stand-alone Microsoft Office Word or Excel.

The dataset is automatically checked out and the stand-alone Microsoft Office Word or Excel is launched with the dataset.

2. Modify the unstructured document as needed and reimport it.
3. Check in the dataset to unlock it in Teamcenter.

Reimporting a modified unstructured document

After you have modified an unstructured document, which is now a Teamcenter dataset, you can reimport it to Teamcenter in one of two ways:

- **As an update**

When you reimport the modified unstructured document as an update, the following occurs in Teamcenter when the import is complete:

- New requirement objects are created for new requirement identifiers added to the dataset.
- Existing requirement objects are removed for requirement identifiers deleted from the dataset.
- Trace links are created for new trace link identifiers added to the dataset.
- Trace links are removed for trace link identifiers deleted from the dataset.
- The dataset is updated with the modified requirement information.
- **As a revision**

When you reimport the modified unstructured document as a revision, the same results occur as when you reimport as an *update*, with these additional results:

- A new revision of the dataset is created in Teamcenter and the updated flat file is added to the revised dataset.

Any Word or Excel macros in the previous dataset are preserved in the revision.

- A deep copy for revise is used when creating existing requirement identifiers in the requirement specification. The existing requirement objects are revised to the next revision in Teamcenter.

If the reimport fails, the entire import transaction is rolled back and the import operation is aborted.

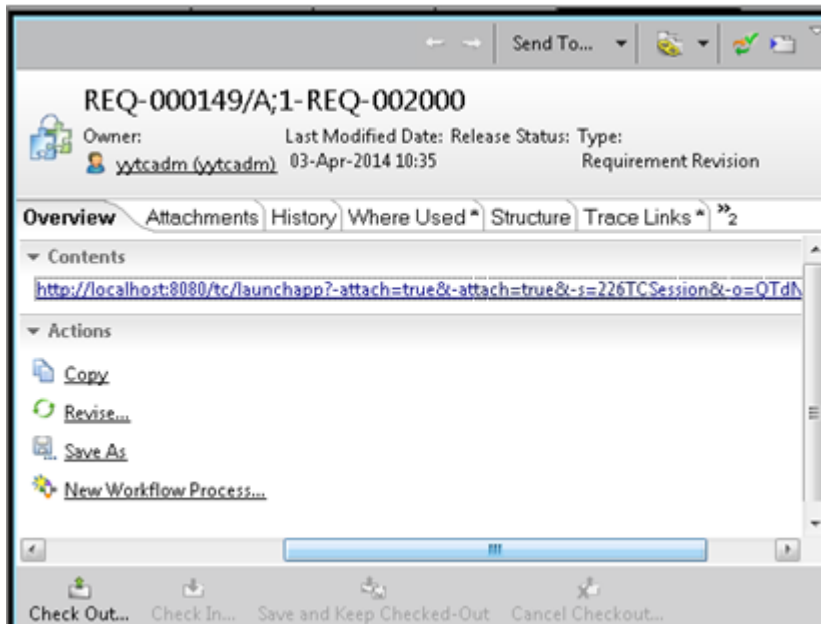
Reimport the modified dataset

1. In Teamcenter, choose one of the following options to open the dataset that you want to reimport:

- In the navigation pane, click **My Teamcenter**.
 - a. Search for the dataset you want to reimport.
 - b. Double-click the dataset.

The dataset is automatically checked out and the stand-alone Microsoft Office Word or Excel is launched with the dataset.

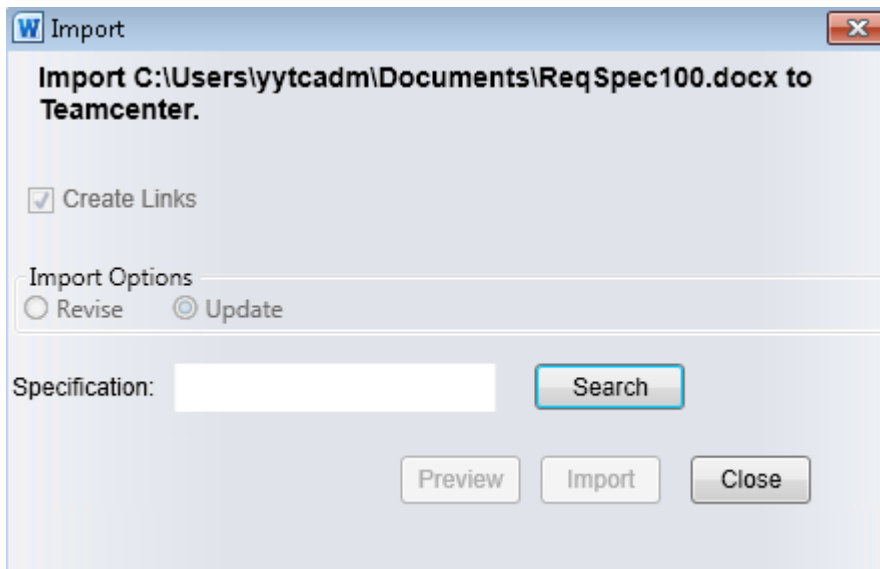
- In the navigation pane, click **My Teamcenter** or **Systems Engineering**.
 - a. Search for any of the requirement objects that are contained within the dataset.
 - b. Right-click the requirement object and choose **Open With→Summary**.
 - c. In the **Summary** view, under the **Contents** heading, click the URL.



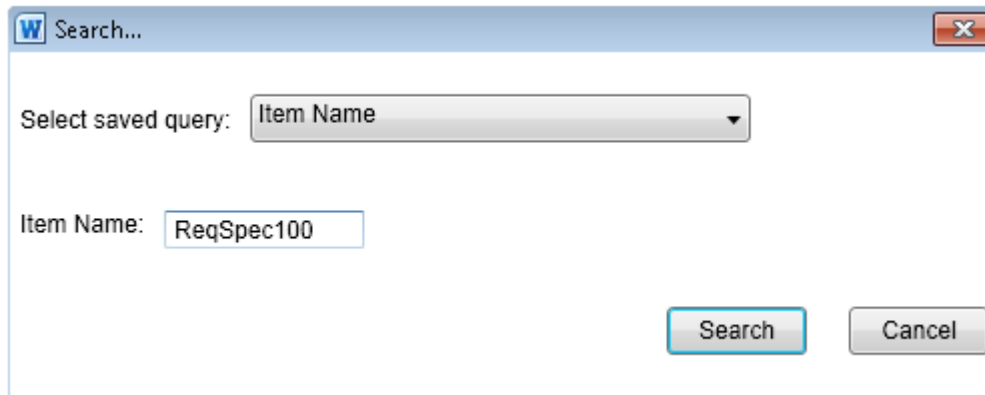
- d. In the **File Download** dialog box, click **Open** to launch stand-alone Microsoft Office Word or Excel.

The dataset is automatically checked out and the stand-alone Microsoft Office Word or Excel is launched with the dataset.

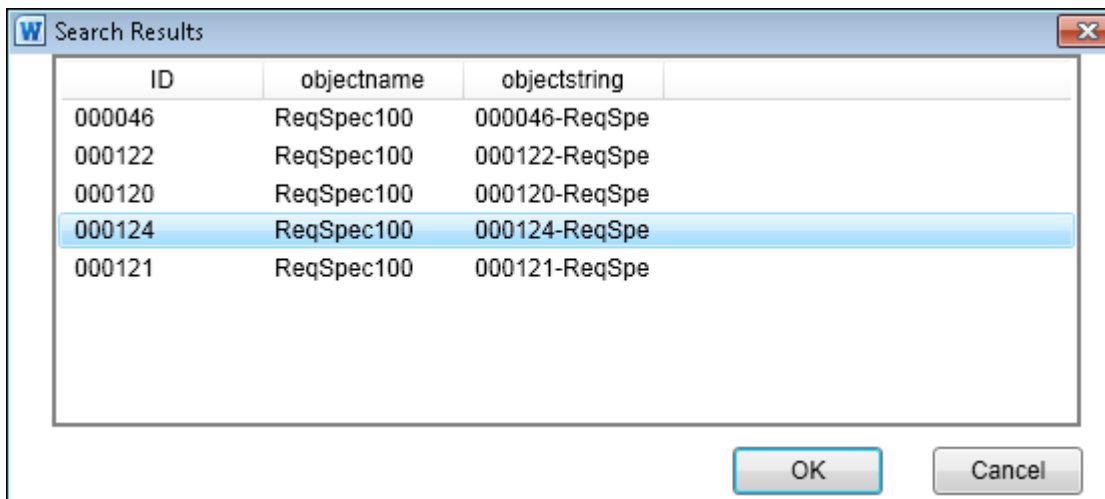
2. In the unstructured document, click the **Teamcenter** tab→**Requirements** group→**Flat File Import** and choose **Import Update**.
3. Click **Search** to search for the specification you want to reimport.



4. In the **Search** dialog box, click the **Select saved query** box and select the property you want to search for.
5. Type the property value you want to search for.



6. Click **Search**.
7. In the **Search Results** dialog box, select the requirement specification you want to reimport.



8. Click **OK**.

The **Specification** box displays the requirement specification you want to reimport.

9. Select the **Create Links** box to create trace links.
10. Under **Import Options**, select one of the following:
 - **Update**

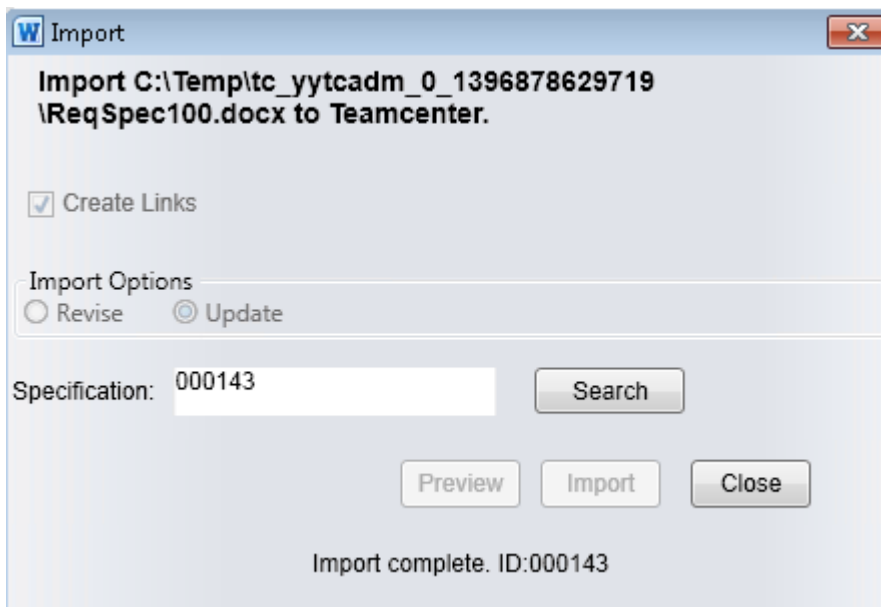
Adds new requirement objects and trace links and removes deleted requirement objects and trace links.

- **Revise**

Performs the same additions and deletions as the **Update** option, and in addition, creates a new revision of the dataset in Teamcenter.

11. Click **Import**.

When the import is complete, the item ID of the reimported requirement specification in Teamcenter is displayed.



Example: Reimporting a modified dataset as an update

For this example, the flat file configuration file for Word contains the following common settings and parser rule settings:

```
<Common Settings>
<add key = "Parser" value = "Teamcenter.TcIDToIDParser.WordIDParser;
C:\Program Files (x86)\Siemens\Teamcenter
9\TcWord2K7DocImp\TcIDToIDParser.dll"/>
<add key = "PropDebugComments" value = "false"/>
<add key = "LinkDebugComments" value = "false"/>
<add key = "DefaultSpecType" value="RequirementSpec" />
<add key = "IDValidationPattern" value="[[]*:[A-Za-z0-9_]{1,}-[0-9]{6}
[]]"/>
<add key = "IDPrefix" value = "ID,ID2,ID3,CID" />
<add key = "IDProperty" value = "object_name" />
```



```
<add key = "Default Saved Query" value = "Item ID" />
<add key = "Search" value="00TBSearch" />
</Common Settings>
```

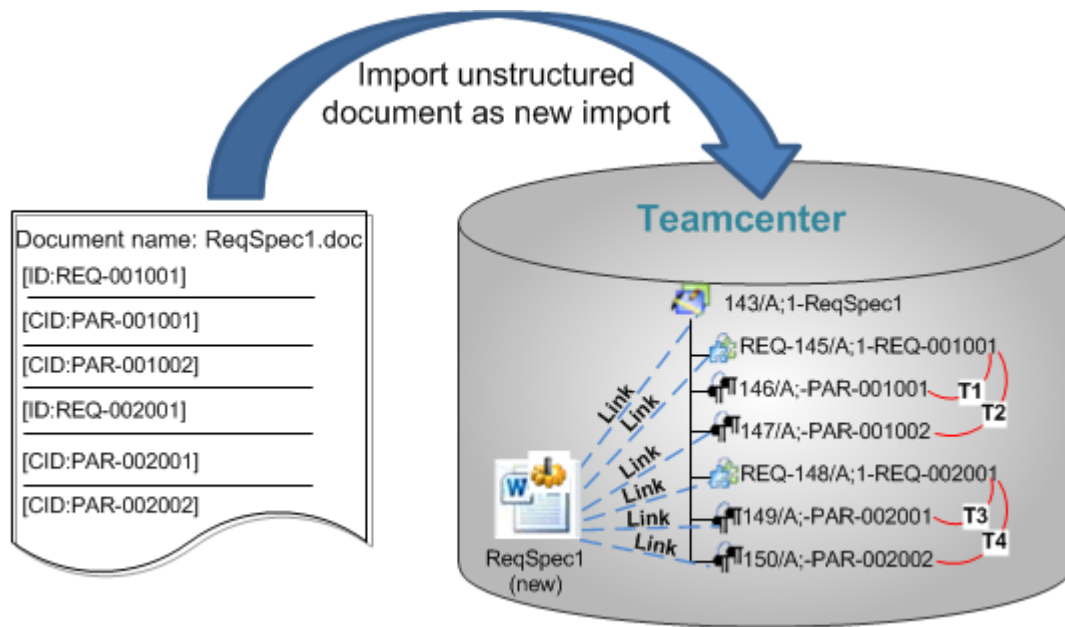
```
<ParserRules>
<add
  RuleID="1"
  LinkType = "FND_Tracelink"
  PrimaryKeyword = "[[]ID:REQ-[0-9]{6}[]]"
  PrimaryKeywordType = "Requirement Revision"
  CreatePrimaryKeyword = "true"
  SecondaryKeyword = "[[]CID:PAR-[0-9]{6}[]]"
  SecondaryKeywordType = "Paragraph Revision"
  CreateSecondaryKeyword = "true"
  LinksAs = "Complying"/>
</ParserRules>
```

The following steps are performed:

1. The following identifiers are assigned in the unstructured **ReqSpec1.doc** unstructured Word document:

```
[ID:REQ-0001001]
[CID:PAR-0001001]
[CID:PAR-0001002]
[ID:REQ-0002001]
[CID:PAR-0002001]
[CID:PAR-0002002]
```

2. The unstructured document is imported as a *new* import.
The option to create trace links for the import was selected.



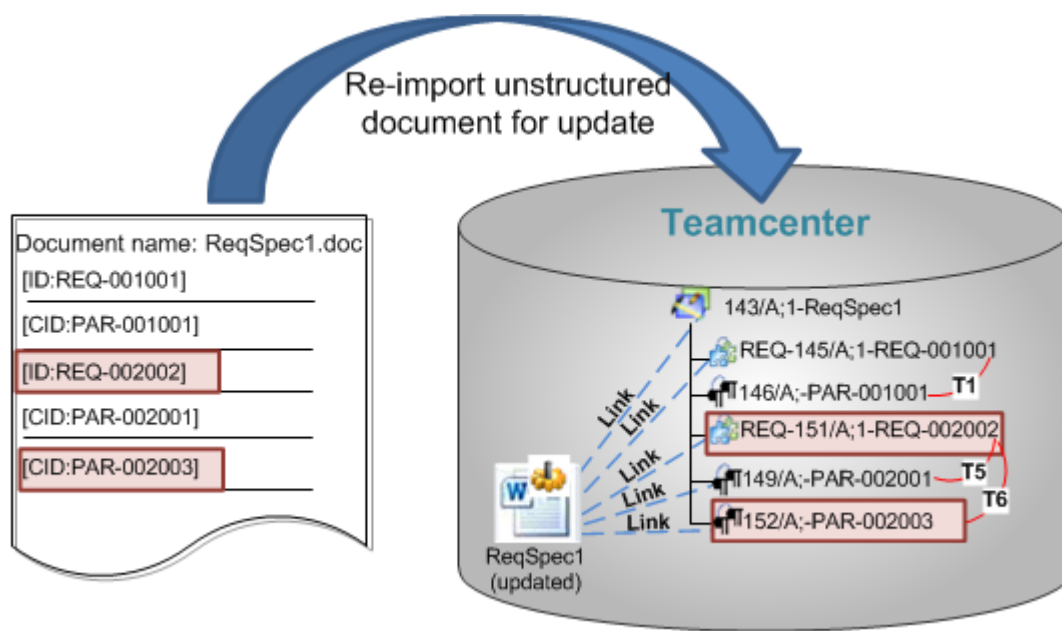
3. The following modifications to the **ReqSpec1** dataset are performed:
 - The [CID:PAR-0001002] paragraph revision identifier is deleted.
 - The [CID:PAR-0002002] paragraph revision identifier is deleted.
 - The new [ID:REQ-0002002] requirement revision identifier is assigned.
 - The new [ID:PAR-0002003] requirement paragraph identifier is assigned.

The updated contents of the **ReqSpec1.doc** unstructured document: contain the following identifiers:

[ID:REQ-0001001]
 [CID:PAR-0001001]
 [ID:REQ-0002002]
 [CID:PAR-0002001]
 [CID:PAR-0002003]

4. The unstructured document is reimported as a *update* import. The option to create trace links for the import was selected.

After the reimport to Teamcenter for update, the following occurs:



- The **ReqSpec1** dataset is updated with links to the requirement objects in the **143/A;1-ReqSpec1** requirement specification.
- The **143/A;1-ReqSpec1** requirement specification is updated in Teamcenter containing the following changes:
 - The new **REQ-151/A;1-REQ-002002** requirement revision is created in Teamcenter.
 - The new **152/A;-PAR-002003** paragraph revision is created in Teamcenter.
 - Because the **[CID:PAR-0001002]** paragraph revision identifier was deleted in the unstructured document, the **147/A;-PAR-001002** paragraph revision and the **T2** trace link are deleted in Teamcenter.
 - Because the **[CID:PAR-0002002]** paragraph revision identifier was deleted in the unstructured document, the **150/A;-PAR-002002** paragraph revision and the **T4** trace link are deleted in Teamcenter.
 - Because the **CreateSecondaryKeyword** is set to **True** in the parser settings for rule 1, the new *internal* **T5** trace link is created between the **REQ-151/A;1-REQ-002002** requirement revision and the **149/A;-PAR-002001** paragraph revision.
 - Because the **CreateSecondaryKeyword** is set to **True** in the parser settings for rule 1, the new *internal* **T6** trace link is created between the **REQ-151/A;1-REQ-002002** requirement revision and the **152/A;-PAR-002003** paragraph revision.

Example: Reimporting a modified dataset a revision

For this example, the flat file configuration file for Word contains the following common settings and parser rule settings:

```
<Common Settings>
<add key = "Parser" value = "Teamcenter.TcIDToIDParser.WordIDParser;
C:\Program Files (x86)\Siemens\Teamcenter
9\TcWord2K7DocImp\TcIDToIDParser.dll"/>
<add key = "PropDebugComments" value = "false"/>
<add key = "LinkDebugComments" value = "false"/>
<add key = "DefaultSpecType" value="RequirementSpec" />
<add key = "IDValidationPattern" value="[[]*:[A-Za-z0-9_]{1,}-[0-9]{6}
[]]"/>
<add key = "IDPrefix" value = "ID,ID2,ID3,CID" />
<add key = "IDProperty" value = "object_name" />
<add key = "Default Saved Query" value = "Item ID" />
<add key = "Search" value="00TBSearch" />
</Common Settings>

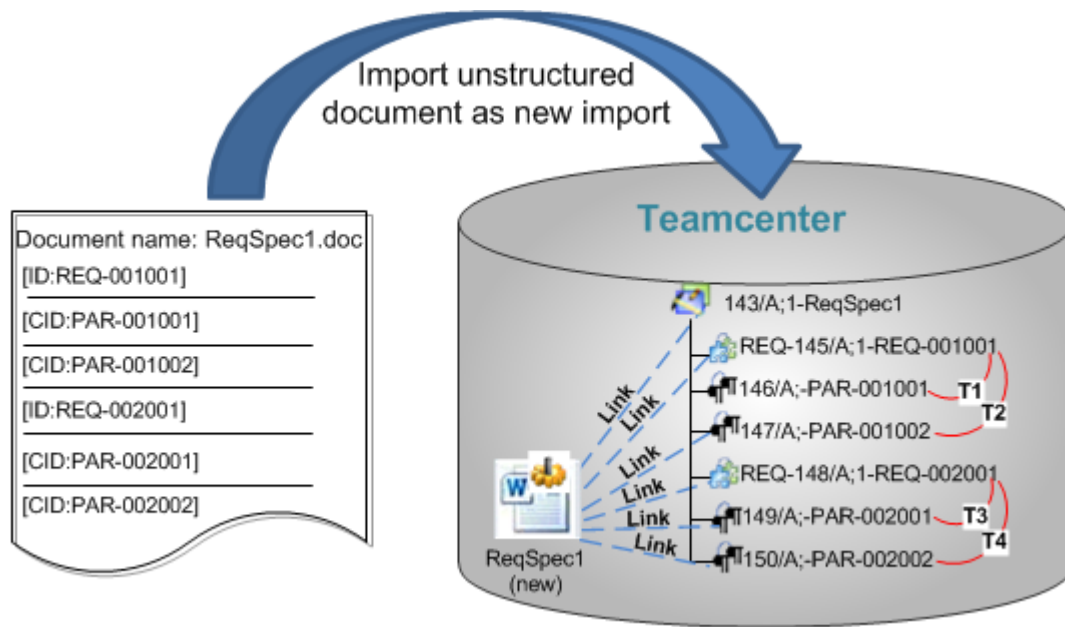
<ParserRules>
<add
  RuleID="1"
  LinkType = "FND_Tracelink"
  PrimaryKeyword = "[[]ID:REQ-[0-9]{6}[]]"
  PrimaryKeywordType = "Requirement Revision"
  CreatePrimaryKeyword = "true"
  SecondaryKeyword = "[[]CID:PAR-[0-9]{6}[]]"
  SecondaryKeywordType = "Paragraph Revision"
  CreateSecondaryKeyword = "true"
  LinksAs = "Complying"/>
</ParserRules>
```

The following steps are performed:

1. The following identifiers are assigned in the unstructured **ReqSpec1.doc** unstructured Word document:

```
[ID:REQ-0001001]
[CID:PAR-0001001]
[CID:PAR-0001002]
[ID:REQ-0002001]
[CID:PAR-0002001]
[CID:PAR-0002002]
```

2. The unstructured document is imported as a *new* import.
The option to create trace links for the import was selected.

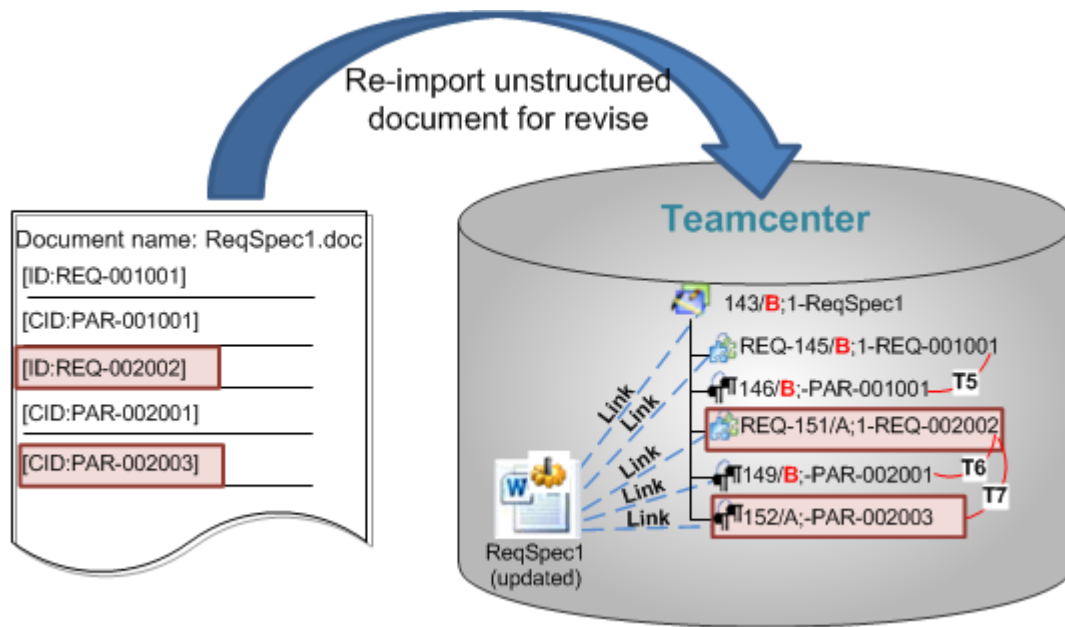


- The following modifications to the **ReqSpec1** dataset are performed:
 - The **[CID:PAR-0001002]** paragraph revision identifier is deleted.
 - The **[CID:PAR-0002002]** paragraph revision identifier is deleted.
 - The new **[ID:REQ-0002002]** requirement revision identifier is assigned.
 - The new **[ID:PAR-0002003]** requirement paragraph identifier is assigned.

The updated contents of the **ReqSpec1.doc** unstructured document: contain the following identifiers:

```
[ID:REQ-0001001]
[CID:PAR-0001001]
[ID:REQ-0002002]
[CID:PAR-0002001]
[CID:PAR-0002003]
```

- The unstructured document is reimported as a *revise* import. The option to create trace links for the import was selected.



After the reimport to Teamcenter for revise, the following occurs:

- The **ReqSpec1** dataset is updated with links to the requirement objects in the **143/B;1-ReqSpec1** requirement specification.
- The **143/B;1-ReqSpec1** requirement specification is revised to the next revision in Teamcenter and contains the following changes:
 - The **REQ-145/B;1-REQ-001001** requirement revision is revised to the next revision in Teamcenter.
 - The **146/B;-PAR-001001** paragraph revision is revised to the next revision in Teamcenter.
 - The **149/B;-PAR-002001** paragraph revision is revised to the next revision in Teamcenter.
 - The new **REQ-151/A;1-REQ-002002** requirement revision is created in Teamcenter.
 - The new **152/A;-PAR-002003** paragraph revision is created in Teamcenter.
 - Because the **[CID:PAR-0001002]** paragraph revision identifier was deleted in the unstructured document, the **147/A;-PAR-001002** paragraph revision and the **T2** trace link are deleted in Teamcenter.
 - Because the **[CID:PAR-0002002]** paragraph revision identifier was deleted in the unstructured document, the **150/A;-PAR-002002** paragraph revision and the **T4** trace link are deleted in Teamcenter.

- Because the **CreateSecondaryKeyword** is set to **True** in the parser settings for rule 1, the *internal T5* trace link is created between the **REQ-151/A;1-REQ-002002** requirement revision and the revised **149/B;-PAR-002001** paragraph revision.
- Because the **CreateSecondaryKeyword** is set to **True** in the parser settings for rule 1, the new *internal T6* trace link is created between the **REQ-151/A;1-REQ-002002** requirement revision and the revised **149/B;-PAR-002001** paragraph revision.
- Because the **CreateSecondaryKeyword** is set to **True** in the parser settings for rule 1, the new *internal T7* trace link is created between the **REQ-151/A;1-REQ-002002** requirement revision and the **152/A;-PAR-002003** paragraph revision.

Import objects in the Teamcenter format

1. Select the container object into which the objects will be imported.
2. Choose **Tools→Import→Objects**.

The system displays the **Import** dialog box.

3. Click the Teamcenter button in the left pane of the window.
4. In the **Importing Object** box, click **Browse**  to locate and select a folder or file from the system directory.
5. If you selected a directory or file containing Teamcenter objects, the objects are displayed in the **Object List**. Select the objects that you want to import using one of the following methods:
 - Click the check boxes to select individual objects in the list.
 - Click **Select All** to select all importable objects in the folder.
 - Click **All But** to select all objects except those that are currently selected.
 - Click **Select None** to clear all object selections.
6. (Optional) Choose from the following report options:

Note:

The report options are offered in addition to the operation log file.

Generate Import Report

Generates a report listing the objects that were imported. The system displays the report after the successful completion of the import operation.

**Preview
Import
Report**

Performs a "dry run" of the import operation. The resulting report lists the objects that would be imported, as well as any problems that are encountered.

**Continue
on Error**

Allows the import operation to continue if errors are encountered while importing optional objects. All objects are considered optional except those with the following relation types:

Requirement
Specification
Item Master
Item Revision Master

If errors are encountered, a completion report is generated even if the **Generate Import Report** option is not selected.

- Click **Apply** to start the import operation and retain the dialog box, or click **OK** to start the import operation and exit the dialog box. When the import operation has successfully completed, you can review the log file.

Import objects in the STEP format

Note:

You can view STEP files in the stand-alone viewer only. In addition, to work with STEP files in the stand-alone viewer, you must have the optional STEP geometry license.

- Select the container object into which the objects will be imported.
- Choose **Tools→Import→Objects**.

The system displays the **Import** dialog box.

- Click **STEP** in the left pane of the window.

The system displays the **STEP Import** dialog box.

Note:

When you import product data into Teamcenter from a STEP AP203 or AP214 physical file, it is always imported as a complete item structure.

- Type the path to the folder containing the STEP file that you want to import in the **Importing Object** box, or click **Browse**  to locate and select a folder from the system directory.

Once a STEP file is selected, the directory path of the selected file is displayed in the **Importing Object** box.

The STEP file and all included references are displayed in the **Include Reference** section of the dialog box.

5. Review the objects selected for import, as displayed in the **Include Reference** list.
 - If the list accurately reflects the objects that you want to import, click **Apply** to start the import operation and retain the dialog box, or click **OK** to start the import operation and exit the dialog box.
 - If the list does not accurately reflect the objects that you want to import, click **Cancel** and begin the process again.

When the import operation has successfully completed, you can review the log file.

Import objects in the PLM XML format

1. Choose **Tools→Import→From PLMXML**.

The system displays the **PLM XML Import** dialog box.

2. Click **Browse**  to the right of the **Importing Object** box.

The system displays the **Select Object** dialog box.

3. Select **PLM XML File (.xml)** from the **Files of type** list.
4. Navigate to the directory containing the file and select the file.

The system displays the file name in the **Directory** name box.

5. Click **Select**.

The system displays the path to the selected file in the **Importing Object** box of the dialog box.

6. Select the transfer mode to be used to configure the import operation.

Note:

To prevent undesirable results, use the **REQ_IMPORT_TEMPLATE** transfer mode if you are importing any of these file types:

- ExcelTemplate
- SpecTemplate
- ObjectTemplate
- HTML Object Template

7. (Optional) Create or select an incremental change object into which BOM components contained in the XML file will be imported. Importing BOM components into an incremental change object allows you to view changes in Structure Manager prior to making them effective.

Note:

BOM components imported into an incremental change object are appended to the list of incremental change components.

8. Click **Apply** to start the import operation and retain the dialog box, or click **OK** to start the import operation and exit the dialog box.

The system imports the data as follows:

- By default, the imported object has no container reference. You can use relevant search criteria to find the imported object and so you can copy it to a folder.
- To have Teamcenter import objects into your **Newstuff** folder, create the **PLMXML_put_objects_in_newstuff_on_import** preference and set the value to **TRUE**. Imported objects are placed in a folder with the same name as the imported XML file.

Note:

If object names or IDs are encountered that exceed the character limit for those boxes in Teamcenter, the name and/or ID is truncated when imported into Teamcenter.

When the import completes, a dialog box appears. You can click **Yes** in the dialog box to view the log file for details on the import and information about any errors that occurred.

Note:

Import results can vary, depending on the localization status of property values and the languages involved.

- If the **PIE_allow_import_with_different_sml** preference is set to **true**, import is allowed even though the site master language from the exporting site is not the same as at the importing site.

- If some of the languages represented in the PLM XML file are supported in the importing site, the nonsupported languages are ignored for import and are logged in the importer log.
- You can view unsupported languages in the import log file. If there is no language information in the XML file, the PLM XML file contents are imported into the database based on your data entry preferences.
- You can import localized property values of objects from a PLM XML file. The language marked with as primary is imported as attribute master language. All other languages are imported as translation values with respective language information.

Maintaining requirement structures and content

Navigating and organizing requirement specification structures

Use these methods for navigating and organizing requirements specification structures in the **Requirements** view.

Note:

If your top level object is a type of object that is not a child of the **Specification** business object in the data model, then you must configure the object to allow moving the first level objects up and down in your structure.

To do this	Use these options
Navigate up and down through the requirements structure.	Use the ↑ and ↓ keys or the Page Up and Page Down keys.
Move to the first and last columns in the row for the line in the requirement structure.	Use the Home and End keys. The Home key moves to the first column and the End key moves to the last column.
Reorder the child nodes of the structure.	• Choose Edit→Move Up or Edit→Move Down. • Click Move Up ↑ or Move Down ↓ on the toolbar. • Edit the Number property (for example, changing it from 2 to 4) by selecting the row and choosing Edit→Edit Number.
Move a child node up to become a sibling node within the structure.	• Choose Edit→Promote One Level. • Click Promote ↶ on the toolbar. • Press Ctrl+↑.

To do this	Use these options
Move a sibling node down to become a child node within the structure.	- Edit the Number property by removing a level (for example, changing it from 2.1 to 3) by selecting the row and choosing Edit→Edit Number. - Choose Edit→Demote One Level. - Click Demote  on the toolbar. - Press Ctrl+↓. - Edit the Number property by adding a level (for example, changing it from 3 to 2.1) by selecting the row and choosing Edit→Edit Number.

Editing requirement content

Elements of requirement content

Requirements can contain all of the familiar rich-text content elements of Microsoft Office Word, such as:

- Headings, body paragraphs, and hyperlinks
- Equations, symbols, and other special characters
- Bulleted and numbered lists
- Tables
- Graphics

You can apply paragraph formatting through Word styles and direct formatting options. Character formatting, such as boldface and italics, can be applied as well.

You can copy, move, and delete selected portions through the standard Word functions.

Caution:

Requirement content includes the text of any Word callout shapes that are inserted in the document. If callout text occurs in the first 2,000 characters, the requirement's **Body Cleartext** property shows the callout text inline with the overall content. The **Body Cleartext** property is displayed in the **Summary** view, on the **Overview** tab under **Item Properties**.

You can insert the **Body Cleartext** property column in the **Requirements** view. Use the **Set the number of rows per line** list to display multiple text lines as necessary.

Opening requirement content from the Requirements view

There are two ways to open a requirement's Microsoft Office Word content for editing:

- In the **MS Word** view, an embedded Word window in the Requirements Manager and Systems Engineering perspectives. The **MS Word** view must be enabled to open Word files in Windows.
- In a separate Word window.

Note:

- These methods differ from opening Word documents that are stored in Teamcenter and may be related to requirement objects or other objects. They also differ from opening requirement content from Client for Office using Teamcenter live Word.
- The My Teamcenter **Viewer** does not support editing or viewing rich-text content.

Caution:

On Teamcenter servers using the ISO 8859-1 character set (**TC_CHARACTER_ENCODING_SET**), some characters of the Microsoft Office Word rich-text format cannot be translated to plain text. These characters include, for example, Word's smart quotation marks, *em* dashes, and symbols such as Ohm signs (Ω).

Such characters are replaced by question marks (?) in plain text and in the Systems Engineering interface, for example, in object names and in certain property values. To avoid this problem, do one of the following:

- Before you enter content for an object, switch the **MS Word** view to display plain text.
- Run the Teamcenter server on Windows, using a database that is compatible with CP1252 encoding.
- Use UTF-8 encoding for the Teamcenter server and database.

Formatting content

Manual formatting is retained in the database when you close the temporary file. You can apply manual formatting through Word's **Format** menu and formatting toolbar.

The Word styles that the database retains are governed by the full text dataset that is attached to the **REQ_default_spec_template** item. This dataset can be opened in Word for editing styles, which are set

in Teamcenter when the dataset is saved. Those settings are applied to requirements and paragraphs when they are opened.

Changes to styles in the temporary file do not affect the content in the database. Although you can modify and create styles, and apply them with visible results in Word, such changes are not retained when you close the file. Nor is it effective to modify Word template settings or to attach other templates.

Furthermore, page setup modifications have no effect in the database. Changes to margins, paper, layout, or headers and footers are not retained beyond the temporary Word session.

Check out a requirement explicitly

1. In My Teamcenter, select the full-text dataset associated with the requirement object.

Tip:

To see if the dataset is checked out by another user, right-click the dataset in My Teamcenter and choose **Properties**. The **Checked Out** box displays either **Y** or **N**. If **Y** is displayed, the **Checked Out User ID** box displays the user.

2. Choose **Tools**→**Check-In/Out**→**Check-Out**.

Note:

If the dataset is already checked out by another user, a message states that the dataset will be opened in read-only mode and changes are not saved.

3. Right-click the requirement and choose **Send To**→**Systems Engineering**.

The requirement is highlighted in the Systems Engineering structure view.

Edit requirement content in the MS Word view

Caution:

Requirement content includes the text of any Word callout shapes that are inserted in the document. If callout text occurs in the first 2,000 characters, the requirement's **Body Cleartext** property shows the callout text inline with the overall content. The **Body Cleartext** property is displayed in the **Summary** view, on the **Overview** tab under **Item Properties**.

You can insert the **Body Cleartext** property column in the **Requirements** view. Use the **Set the number of rows per line** list to display multiple text lines as necessary.

1. To open the **MS Word** view, do one of the following:
 - Choose **View**→**Show MS Word View**.

- Click **MS Word View**  on the toolbar.

2. Select the requirement.

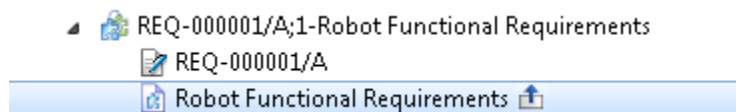
The **MS Word** view displays the content in an embedded Word window.

3. Place the cursor in the window, and use the standard Word functions to enter, change, or format the content.
4. To save the content in the database, click **Save**  at the top of the **MS Word** view.

Warning:

Do not use Word functions for this action.

If you did not check out the full-text dataset explicitly, the dataset remains checked out implicitly to your user ID. When you select another object in the structure view or exit Teamcenter, the dataset is checked in implicitly.



If you checked out the full-text dataset explicitly, it remains checked out. You must explicitly check in the dataset.

Tip:

When the **MS Word** view is active, only the **File** menu and the **Window** menu are available in the Systems Engineering perspective. To restore the other menus, deactivate the **MS Word** view by activating another view. For example, to activate a structure view, click the **Requirements Manager** tab or an item in the structure.

Edit requirement content in a separate Word window

Note:

Requirement content includes the text of any Word callout shapes that are inserted in the document. If callout text occurs in the first 2,000 characters, the requirement's **Body Cleartext** property shows the callout text inline with the overall content. The **Body Cleartext** property is displayed in the **Summary** view, on the **Overview** tab under **Item Properties**.

You can insert the **Body Cleartext** property column in the **Requirements** view. Use the **Set the number of rows per line** list to display multiple text lines as necessary.

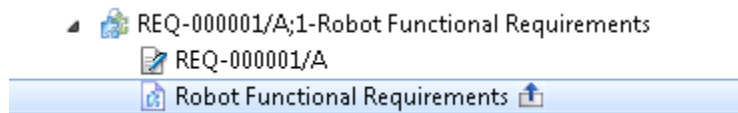
This functionality differs from opening Word documents that are stored in Teamcenter and may be related to requirement objects or other objects. It also differs from editing requirements and structures with Teamcenter live Word in Client for Office.

1. Select the requirement and choose **File→Open in MSWord Application**.

The requirement opens in Word.

2. Use the standard Word functions to enter, change, or format the content.
3. To commit the edits to Teamcenter, choose **File→Save** in Word or click **Save**  on the Word toolbar.

If you did not check out the full-text dataset explicitly, the dataset remains checked out implicitly to your user ID when you select another object in a structure view or exit Teamcenter. You must close the Word window to check in the dataset implicitly.



If you checked out the full-text dataset explicitly, it remains checked out. You must explicitly check in the dataset.

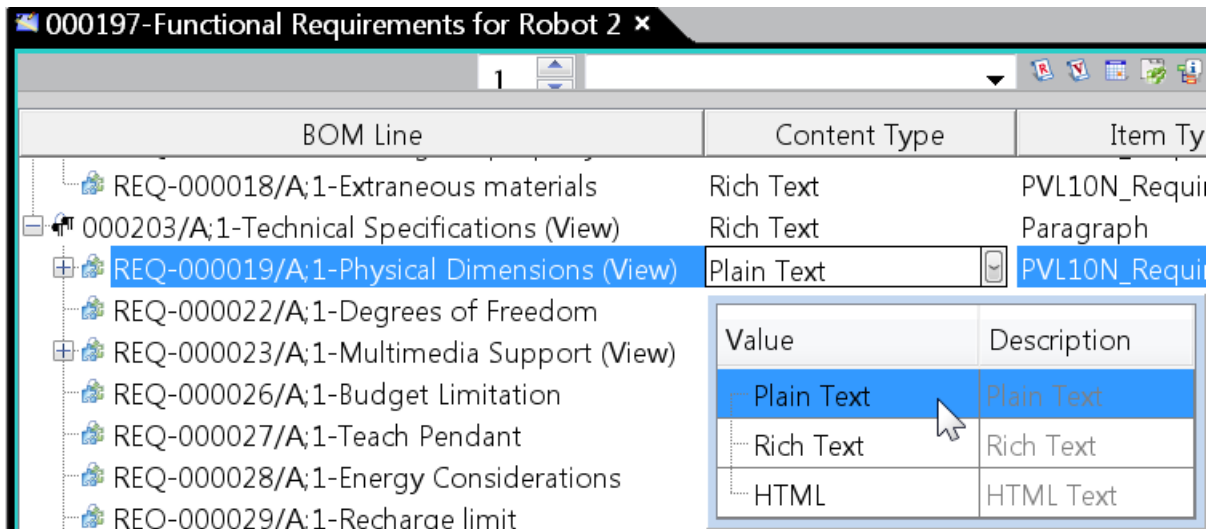
How to recover corrupt rich-text requirement content

You can recover corrupt rich-text requirement content that was originally created using Microsoft Word, by converting it to plain text.

Caution:

Any formatting or images included in the original rich-text requirement content are lost during the conversion to plain text. Only the text is retained in the requirement content.

1. In Systems Engineering, open the requirement structure that contains the requirement object or objects with the corrupt rich-text requirement content.
2. In the **Requirements** view, insert the **Content Type** property column.
3. For each requirement object that you want to recover the corrupted rich-text content:
 - a. Select the structure line for the requirement object.
 - b. Set the **Content Type** value to **Plain Text**.



The corrupted rich-text requirement content for the object is converted to plain text.

You can now view or edit the plain text requirement content.

Add property and other information to Word document headers and footers

You can add property information, rich text, keywords, and images to exported Word documents. Define a property by opening the specification, requirement, or paragraph template in Word and adding a percent sign (%) before the property name and enclosing it all in curly brackets. For example, to add the **created_date** property to the header, place the following in the Word header or footer area:

{%created_date}

Teamcenter uses the property value from the top line selected in the structure.

Insert a dataset reference in a Word document

1. In Teamcenter, select an Image, JPEG, GIF, or TIF dataset, and choose **Edit→Copy Reference**.
2. Open a Word document using either the **MS Word** view in Systems Engineering, the **File→Open in MSWord Application** menu command, or by exporting a Word document for live editing.
3. Place the cursor where you want to insert the reference, and then paste the contents into the body text of the requirement.
4. To convert the reference to a hyperlink, press the Enter key after inserting the reference.
5. Save the document and close it.

When you reopen the document, a URL to the parent of the dataset replaces the reference in the Word document. If you click the URL, the parent object opens in Teamcenter. If you are not logged on to Teamcenter, you are asked to log on.

In Teamcenter, when you upload a file as a named reference, it does not appear as a relation to the dataset unless **Fnd0_Symbolic_Link** is added as a value to the **Dataset_DefaultChildProperties** preference. If this value exists, the file appears under the dataset with the **Symbolic Link** relation.

Note:

If you revise the referenced object in Teamcenter, the reference in Word continues to point to the original version.

Insert a diagram hyperlink in requirement or paragraph content

1. Select the requirement or paragraph and choose **File→Open in MSWord Application**.

The requirement or paragraph opens in Microsoft Office Word.

2. In My Teamcenter, select the diagram and choose **Edit→Copy**.

The diagram identifier is placed in the Windows clipboard.

3. In the Word document, place the cursor where you want to insert the hyperlink, and then click **Paste** on the Word **Home** tab.

The diagram identifier is inserted as a hyperlink.

Tip:

To navigate to the diagram object in My Teamcenter, press CTRL and click the hyperlink.
In the **Summary** view, the **Overview** tab displays a static preview image of the diagram.

Check in a requirement explicitly

1. In My Teamcenter, select the full-text dataset associated with the requirement object.
2. Choose **Tools→Check-In/Out→Check-In**.

Comparing requirement content

Compare two requirements

Note:

Both Teamcenter and Microsoft Office must be either 32-bit or 64-bit. The comparison cannot function properly if the bit versions differ.

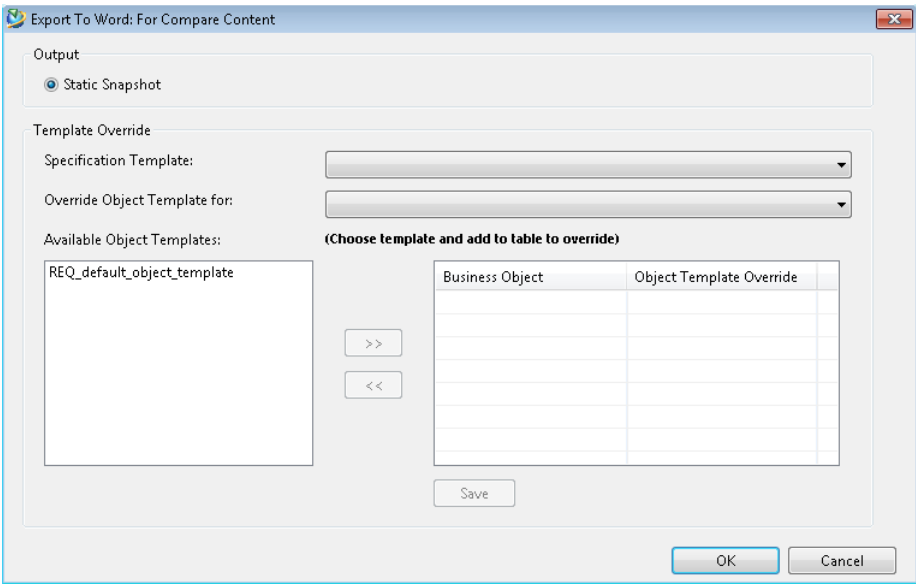
Caution:

Requirement content includes the text of any Word callout shapes that are inserted in the document. If callout text occurs in the first 2,000 characters, the requirement's **Body Cleartext** property shows the callout text inline with the overall content. The **Body Cleartext** property is displayed in the **Summary** view, on the **Overview** tab under **Item Properties**.

You can insert the **Body Cleartext** property column in the **Requirements** view. Use the **Set the number of rows per line** list to display multiple text lines as necessary.

1. In Systems Engineering, Requirements Manager, or My Teamcenter, select a source requirement or specification that you want to compare to another requirement or specification, and then choose **Tools→Compare→Document Compare→Start Compare**.
2. Select a target requirement or specification you want to compare to the first selection and choose **Tools→Compare→Document Compare→End Compare**.

The **Export to Word: For Compare Content** dialog box opens.



3. (Optional) Select a specification template to override the default specification template.

4. (Optional) Select a business object for which to override the object template.
5. (Optional) If you selected a business object for which to override the object template, select the object template to use for the selected business object, and then click the » button.

To configure the template to override for a business object, choose **Tools→Microsoft Office Templates→Override Word templates** in the **Home** view. In the **Configure Templates to override** dialog box, for each specification template, choose a business object.

6. Click **OK**.

Word opens a document showing a comparison of the requirements.

7. (Optional) Save the comparison results file using the Microsoft save function.

The comparison file is saved to the **Newstuff** folder under the source comparison selection that you chose in step 1.

Note:

You can configure the number of versions of Teamcenter datasets to keep before deleting old versions by changing the value of the **AE_dataset_default_keep_limit** preference. When you change the value, it only affects datasets created after the value is changed.

Compare a requirement with its preceding revision

Caution:

Requirement content includes the text of any Word callout shapes that are inserted in the document. If callout text occurs in the first 2,000 characters, the requirement's **Body Cleartext** property shows the callout text inline with the overall content. The **Body Cleartext** property is displayed in the **Summary** view, on the **Overview** tab under **Item Properties**.

You can insert the **Body Cleartext** property column in the **Requirements** view. Use the **Set the number of rows per line** list to display multiple text lines as necessary.

1. In Systems Engineering, Requirements Manager, or My Teamcenter, select the requirement or specification that you want to compare with its previous revision, and then choose **Tools→Compare→Document Compare→Compare With Prior**.

To compare revision sequences, select a revision sequence, and choose **Tools→Compare→Document Compare→Start Compare**. Select the other revision sequence to compare with and choose **Tools→Compare→Document Compare→End Compare**.

The **Export to Word: For Compare Content** dialog box opens.

2. (Optional) Select a specification template to override the default specification template.

3. (Optional) Select a business object for which to override the object template.
4. (Optional) If you selected a business object for which to override the object template, select the object template to use for the selected business object, and then click the » button.

To configure the template to override for a business object, choose **Tools→Override Word template** in My Teamcenter. In the **Configure Templates to override** dialog box, for each specification template, choose a business object.

5. Click **OK**.

Microsoft Word opens with a comparison of the two revisions of the requirement.

6. (Optional) Save the comparison results file using the Microsoft save function.

The comparison file is saved to the **Newstuff** folder under the source comparison selection that you chose in step 1.

Note:

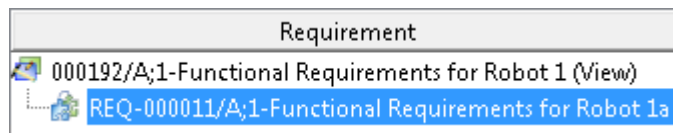
You can configure the number of versions of Teamcenter datasets to keep before deleting old versions by changing the value of the **AE_dataset_default_keep_limit** preference. When you change the value, it only affects datasets created after the value is changed.

Marking up requirement content

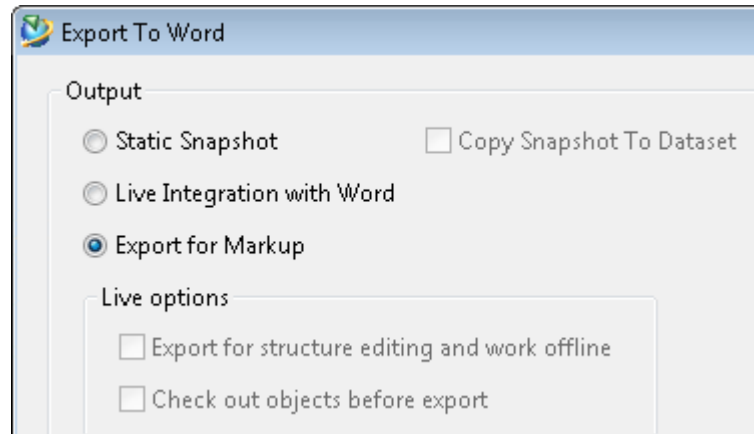
Display markup comments in a requirement object

You can add markup comments to requirement specifications, requirements, paragraphs, and custom notes using Microsoft Office Word as the interface to the content of the requirement objects. This functionality differs from adding markup comments to Word documents that are stored in Teamcenter and may be related to requirement objects or other objects. Adding markup comments to Word documents includes additional features to enhance the content review process, such as using roles and applying disposition comments and status.

1. From the **Requirements** view, select the requirement for which you want to display markup comments.



2. Click **Export objects to Word** .
3. In the **Export to Word** dialog box, select **Export for Markup**, and click **OK**.



4. If the **Teamcenter Login** dialog box appears, log on to Teamcenter.

The document opens with live Word, which means if you edit requirements, you are immediately updating the content stored in Teamcenter.

5. In live Word, click **Markup**  on the **Teamcenter** tab.

The **Teamcenter Markup Manager** pane appears, and any existing markup comments appear in the document and in the comment details list.

Tip:

If you have the document open while others are adding markup comments, click **Load** to see the new markup comments.

6. Select or clear the check boxes of the users whose comments you want to display or hide, respectively.

Add markup comments to requirement objects

You can add markup comments to requirement specifications, requirements, paragraphs, and custom notes using Microsoft Office Word as the interface to the content of the requirement objects. This is known as *live Word*. This functionality differs from adding markup comments to Word documents that are stored in Teamcenter and may be related to requirement objects or other objects. Adding markup comments to Word documents includes additional features to enhance the content review process, such as using roles and applying disposition comments and status.

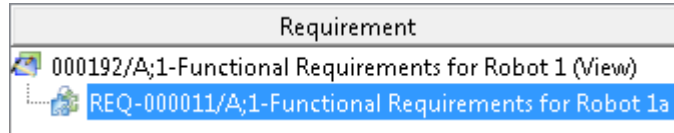
Note:

For Systems Engineering requirement structures, markups are supported only for the body text property.

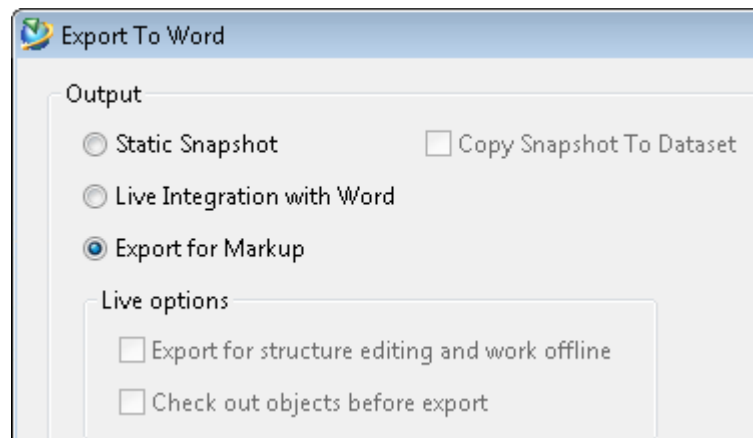
Caution:

Siemens Digital Industries Software does not support markups when different Office versions are used. For example, if a markup comment is created in Microsoft Office 2010, it may not open correctly in Microsoft Office 2013.

1. From the **Requirements** view, select the requirement to which you want to add markup comments.



2. Click **Export objects to Word** .
3. In the **Export to Word** dialog box, select **Export for Markup**, and click **OK**.



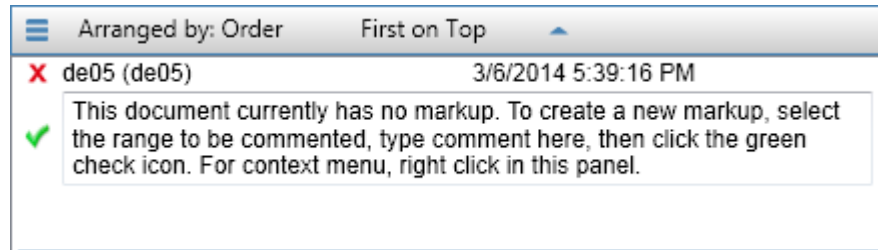
4. If the **Teamcenter Login** dialog box appears, log on to Teamcenter.

The document opens with live Word, which means when you edit requirements, you are immediately updating the content stored in Teamcenter.

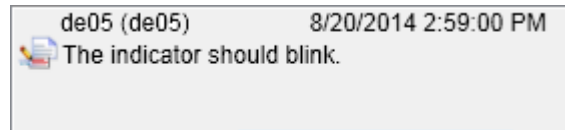
5. In live Word, click **Markup**  on the **Teamcenter** tab.

The **Teamcenter Markup Manager** pane appears.

If the document has no existing markup comments, the comment details list appears in the lower pane with instructions in the text box.



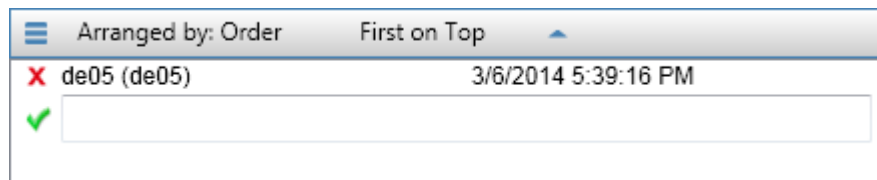
If the document has markup comments, they appear in the comment details list.



Tip:

If you have the document open while others are adding markup comments, click **Load** to see the new markup comments.

6. Select the text in the Word document that you want to mark up with a comment.
7. Do one of the following:
 - If the document has no existing markup comments, delete the instruction text in the text box.
 - If the document has existing markup comments, right-click anywhere in the details list, and choose **Create new markup**.



8. Type your comment in the blank text box.
9. To add the comment to the document, click  next to the text box.

To cancel the new comment, click  next to the text box.

10. To save the comments to Teamcenter, click **Save/Extract** in the **Teamcenter Markup Manager** pane.

The markup comments are saved to Teamcenter.

Note:

Although the markups are displayed in the document, the Teamcenter Markup Manager pane does not show the markups immediately after you click **Save/Extract**. To see these markups in the Teamcenter Markup Manager pane, close the document, reopen it, and click the **Markups** button in the **Teamcenter** tab of the Office ribbon.

Note:

If multiple users are marking up the same document concurrently and an error message appears indicating the base dataset is checked out by another user or is out of date, contact your administrator to review the **ITEM_relation_types_update_lmd** preference.

11. To save any other changes you make to the document, click **Save**  on the Word toolbar.
12. In the **Save** dialog box, click **OK**.
13. Close the document.

Create requirement variants

Variants allow you to create options (for example, **color**) and allowed values of those options (for example, **red** and **blue**) and associate them with a requirement revision. You use the **Variants** tab to create variant rules to filter the structure elements.

Variants in Systems Engineering work in a similar way as they do for product structures in Structure Manager.

1. Open the **Variants** tab.
2. At the bottom of the tab, click the **Show the legacy data tab panel** button .
3. Select the lines in the structure for which you want to variant and create variant options and values.
4. Set the variant conditions on the lines in the structure where you created variant options and values.
5. Ensure the **Show Unconfigured Variants** button  is cleared. (The button background is *not* white.)
6. Click the **Configure Variants** button  to define your variants.

Teamcenter displays only the lines in the structure that satisfy the variant conditions.

7. Click the **Show Unconfigured Variants** button  again. (The button background is white.)

Teamcenter displays the entire structure.

Managing parametric requirements

Parametric requirements overview

Parametric requirements represent the requirements established by the Standards Engineering group that conform to the design practices of a program. Parametric requirements are often categorized and maintained in a library for use by design engineers in the creation of parts lists.

The following concepts apply to parametric requirements:

- Parametric requirements specify parametric variable values that designers select when attaching the requirement to an item or an item revision.
- Parametric requirements are associated with items, such as parts or documents, through the **Parametric Requirements Lists** relation where the part or document revision is the primary object and the parametric requirement or parametric requirement revision is the secondary object.
- When a parametric requirement is created, a **FullText** dataset is created that contains the requirement content. The dataset is attached to the requirement revision.

Note:

You cannot delete the dataset or detach the dataset from the requirement revision.

- You enter the parameters for the requirement in the **Body Cleartext** property of the requirement revision.

Note:

You cannot edit the **Body Cleartext** property after you save it. To edit parameters later, you must create a new revision and edit the revision's **Body Cleartext** property.

- The **Body Cleartext** property contains text, parameters, and parameter values in the syntax shown below.

```
text [parameter_name: parameter_value_1 delimiter parameter_value_2
delimiter . . . parameter_value_n]
```

Example:

Round all sharp corners to [Radius: 0.4, 0.5, 0.6] inch
and heat treat to [Temperature: 200, 220, 230] degrees Fahrenheit.

The default delimiter is a comma (,). However, your Teamcenter administrator can configure the delimiter to use a different character.

- Parametric requirements are used to associate parts with documents.
- Parametric requirements and parametric requirement revisions can be related to multiple items or item revisions.

Create a parametric requirement

Note:

You must have **DBA** privileges set with administrative bypass, or you must fill a role that your Teamcenter administrator has designated as having parametric requirement creation privileges.

1. Choose **File**→**New**→**Item** to open the **New Item** wizard.
2. Select **Parametric Requirement** under **Business Object Type**, and then click **Next** to advance to **Edit Additional Properties**.
3. In the **Name** box, enter the requirement name.

Note:

This text becomes the value of the new requirement's **Rev Name** property.

You can also enter the following optional data:

ID	Object identifier You can specify the identifier explicitly, or you can click Assign to enter the next available identifier in the automated sequence. Or, leave the ID box blank.
Revision	Revision number You can specify the number explicitly, or you can click Assign to enter the next available number in the automated sequence. Or, leave the Revision box blank.
Description	Plain text to convey additional information about the requirement

For example, you can describe its purpose or its relationships to functional and logical structures. This text becomes the initial value of the **Rev Description** property.

Caution:

When selected, the **Open On Create** option automatically opens the new requirement for editing in Word. However, you cannot enter parameters in Word. You must enter parameters in the **Body Cleartext** property of the revision.

4. To create the item, click **Finish**.

Tip:

You can click **Next** instead of **Finish** to open an optional page for defining workflow information.

The parametric requirement item and item revision are created in the **Newstuff** folder of your **Home** view. A new **Requirements** view opens for the occurrence.

Enter parametric requirement parameters

Note:

You must have **DBA** privileges set with administrative bypass, or you must fill a role that your Teamcenter administrator has designated as having parametric requirement creation privileges.

1. Select the parametric requirement revision, and then choose **Tools→Check-In/Out→Check Out**.

Caution:

Do not open the revision's full-text dataset to enter parameters. You must enter the parameters in the **Body Cleartext** property of the revision.

2. In the **Check-Out** dialog box, click **Yes**.

Note:

You can enter optional data in the **Change ID** and **Comments** boxes.

The dialog box closes, and the checked-out symbol  appears next to the requirement.

3. Choose **View→Properties** to open the **Properties** dialog box.
4. In the lower left corner of the dialog box, click **All**.

5. In the **Body Cleartext** box, enter text, parameters, and parameter values in the following syntax:

*text [parameter name: parametric value1 delimiter parametric value2 delimiter
parametric value n]*

Example:

Round all sharp corners to [Radius: 0.01, 0.02, 0.03] inches
and heat treat to [Temperature: 100, 120, 130] degrees Fahrenheit.

Note:

The delimiter is defined by your Teamcenter administrator.

6. To save your entries and close the dialog box, click **OK**.

Caution:

You cannot modify the **Body Cleartext** property after you save it. To update parameters later, you must create a new revision of the requirement.

Revise a parametric requirement

To modify the data for a parametric requirement, you must first create a new revision of the parametric requirement.

1. Select the source requirement item or item revision, and then choose **File→Revise** to open the **Revise** wizard.
2. (Optional) In the **Define the basic information for the new item revision** pane, type the name of the new revision.
3. (Optional) Type a description of the revision and choose a unit of measure for the revision.
4. Click **Next** to open the **Define additional item revision information** pane.
5. (Optional) Enter values for the requirement revision attributes.
6. Click **Next**.

The **Define attached objects** pane displays the source revision and attachments and the destination revision and attachments. By default, all objects are copied forward to the new (destination) revision.

7. (Optional) Choose copy options for the objects related to the source revision by clicking the icon to the right of the object in the tree.

**Don't Copy**

Allows you to selectively copy objects from one revision to another. Objects that are designated as **Don't Copy** appear with a line through them in the **Destination** tree.

**Copy as Object**

Creates a new object of the same type and relation to the parent object. Objects created by this method are totally independent of the parent object. Therefore, any changes to the copied object are not reflected in the original object.

Copied objects appear in bold in the **Destination** tree and can be renamed.

**Copy as Reference**

Copies the object as a reference to the original object. All changes to the reference copy affect the original object. The names of attachments that are copied as references cannot be modified.

8. Click **Next**.

The **Select projects or programs that you want to assign new object to** pane is displayed.

9. (Optional) Assign the new revision to a different project or program.
10. Click **Next** to open the **Select open option and alternate id display option** pane.
11. (Optional) Set the open, display, or check-out option for the new revision.
12. Click **Finish**.

The new revision is created as a copy of the source revision and is opened in Systems Engineering. You can edit the new revision's **Body Cleartext** property to enter new parameters or change existing ones.

The new revision is independent of the source revision. Modifications to the new revision are not reflected in the source.

Replace a parametric requirement revision with another revision

By default, only one revision of a parametric requirement can be associated with an item or item revision. However, you can replace an existing requirement with a new revision of the requirement by using the **Replace ParametricRequirement Revision** option.

1. Right-click the replacement requirement revision, and then choose **Copy**.

2. Expand the item or item revision and select the parametric requirement revision that you want to replace.
3. Choose **Edit→Attach Requirements/Notes→Replace Parametric Requirement Revision**.

If parameter values are defined for the requirement, the **Input Parametric Values** dialog box opens.

4. Choose values for the parameters.
5. Click **OK**.

The previous requirement revision is replaced by the new revision.

View parametric values in standard notes

This is a para in the topic template.

Duplicating requirement structures

Duplicating requirement structures

A requirement specification can be duplicated to create a new structure. You use the **File→Duplicate** command to duplicate the structure.

- You can duplicate the entire requirement specification structure or a substructure.
- If you do not have access rights to a substructure in the original structure being duplicated, the substructure cannot be duplicated. Instead, Teamcenter creates references to the substructure in the duplicate.
- The duplication operation uses deep copy rules, defined by your Teamcenter administrator, to determine whether datasets and attachments to the original structure are copied to the duplicate.
- Trace links in the source structure are preserved in the duplicate.

The duplication process places the duplicate in a folder if one is selected. The new structure is also placed on the Teamcenter clipboard, unless the **Run in Background** command is selected for asynchronous processing. During asynchronous processing, Teamcenter runs the duplication process in a separate server thread and not in the current thread that initiated the duplication request, so you can perform other Teamcenter tasks while the duplication process is running. This is important when duplicating large structures that could have long processing times. You receive an Outlook email and a Teamcenter mail message to inform you when the processing starts and completes. Any errors in processing are reported in Teamcenter.

Two preferences are used for duplicating requirement structures:

- **DuplicateRunInBackground**

Set this user preference to **true** to automatically select the **Run in Background** check box in the **Duplicate...** dialog box.

- **DuplicateAssignToProject**

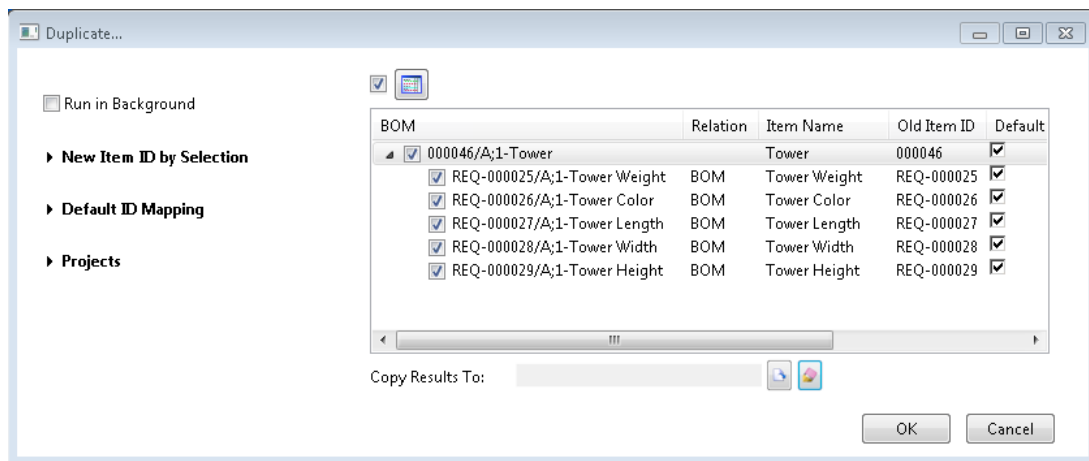
Set this user preference to **true** to automatically select the **Assign Projects to New Items** check box and expand the section under the **Projects** header in the **Duplicate...** dialog box.

Duplicate a requirement structure

You can create a new set of requirements from an existing set by duplicating the requirement specification structure at the level you select.

1. To duplicate the entire requirement specification structure, select the top line of the structure in the **Requirements** view. Otherwise, select the line for the substructure or the structure object you want to duplicate.
2. Choose **File→Duplicate**.

The **Duplicate** dialog box appears. You can expand or collapse the structure tree that is displayed in the dialog box.



3. (Optional) Select a folder in which to place the duplicate.

Note:

The folder must already exist in Teamcenter.

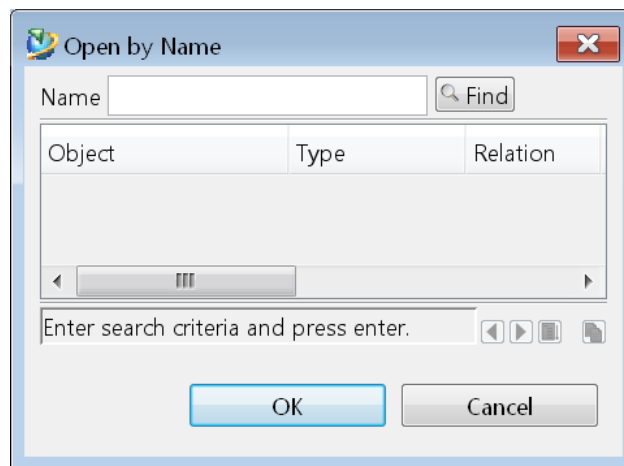
If you do not select a folder, the duplicate is placed on the Teamcenter clipboard, unless the **Run in Background** option was selected.

- a. In the **Duplicate** dialog box, click **Search a folder by name**  next to the **Copy Results To:** box.

Note:

If a folder name already exists in the **Copy Results To:** box, click **Clear option**  first, if you want to save to a different folder.

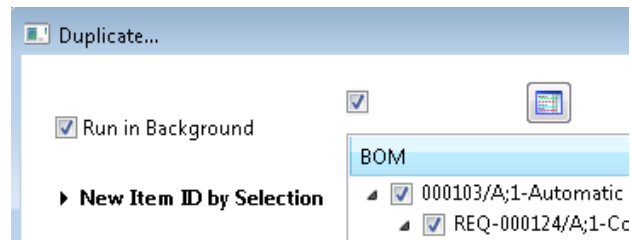
- b. In the **Open by Name** dialog box, type the folder name search criteria in the **Name** box, and then click **Find**.



- c. In the **Open by Name** dialog box, select the row with the folder name to copy to, and then click **OK**.

The folder name appears in the **Copy Results To:** box, in the **Duplicate** dialog box.

4. Select or clear the check box located at the left of each line in the structure to refine the contents of the duplicate you create.
 - Select the check box to copy the object, creating a new object in the duplicated structure.
 - Clear the check box to create a reference to the object in the duplicated structure.
5. Define how the duplicate is created:



Run in Background check box

If selected, the duplication process is run in a separate server thread allowing you to perform other Teamcenter tasks while the duplication process is running.

Note:

You receive an Outlook email and a Teamcenter mail message to inform you when the processing starts and completes. Any errors in processing are reported in Teamcenter.

Select All check box

- If selected, all requirements in the list are duplicated.
- If cleared, you can select the specific lines to be duplicated.

Note:

Items not selected are referenced in the duplicate structure, rather than duplicated.

Select rows assigned to projects in common with top item project assignments.

If selected, the rows for item revisions in the same project as the top item are automatically selected for duplication.



New Item ID by Selection check boxes

Click the **Apply** button to update the structure display if you change the default mapping:

Assign New IDs

If selected, you can specify the naming pattern of duplicated items that are included in the duplicate. The new identifiers are not applied until you create the duplicate.

Prefix	Allows you to define a prefix to the original name of each duplicated item that is included in the duplicate, for example, NEW_ .
Suffix	Allows you to define a suffix after the original name of each duplicated item that is included in the duplicate.
Replace With	Allows you to replace a specified substring in the original item name with another defined string in the duplicated item name. For example, you can replace TEMPLATE with NEW .

Default ID Mapping check boxes

Click the **Apply** button to update the structure display if you change the default mapping:

Assign New IDs	If selected, you can modify the naming pattern of original items before they are included in the duplicate.
Prefix	Allows you to define a prefix for the original item name before it is included in the duplicate.
Suffix	Allows you to define a suffix after the original item name before it is included in the duplicate.
Replace With	Allows you to replace a specified substring in the original item name with another defined string before the duplicate is created.

Projects check boxes

Assign Projects to New Items	If selected, items created during the cloning process can be automatically assigned to one or more projects.

Note:

The default setting for this check box is controlled by the **DuplicateAssignToProject** preference.

Add projects to be assigned	Access the Available Projects dialog box to select the project or projects to add to the assign projects list.



Delete selected projects

Remove the selected project from the assign projects list.



- Click **OK** to create the duplicate structure.

When the duplicate process is complete, Systems Engineering displays the **Duplicate Results** message box.

The duplicate is saved in the folder you selected to copy results to, if any. It is also placed on the Teamcenter clipboard unless the **Run in Background** option was selected.

The **Duplicate Results** message box indicates whether the duplicate operation was successful. It also lists any instances where, because of owner permissions, Teamcenter could not duplicate an object selected for cloning in the original structure but, instead, referenced the object.

Find the source or duplicates of a requirement object

Use this procedure to determine whether a requirement object is the source or duplicate of another requirement object and, if so, to which requirement object or objects it is the source or duplicate.

- In the Systems Engineering **Requirements** view, select the requirement object for which you want to determine the source or duplicate objects.

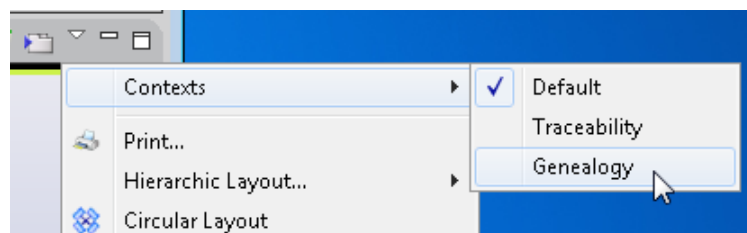
Note:

Requirement objects include requirement specifications, requirements, or paragraphs, and nested requirement specifications, requirements, or paragraphs.

- Click the **Relation Browser** tab, or open the **Relation Browser** view by choosing **Window→Show View→Relation Browser**.

The selected requirement object appears in the **Relation Browser** view.

- Choose **Contexts→Genealogy** from the **Relation Browser** view menu.



Note:

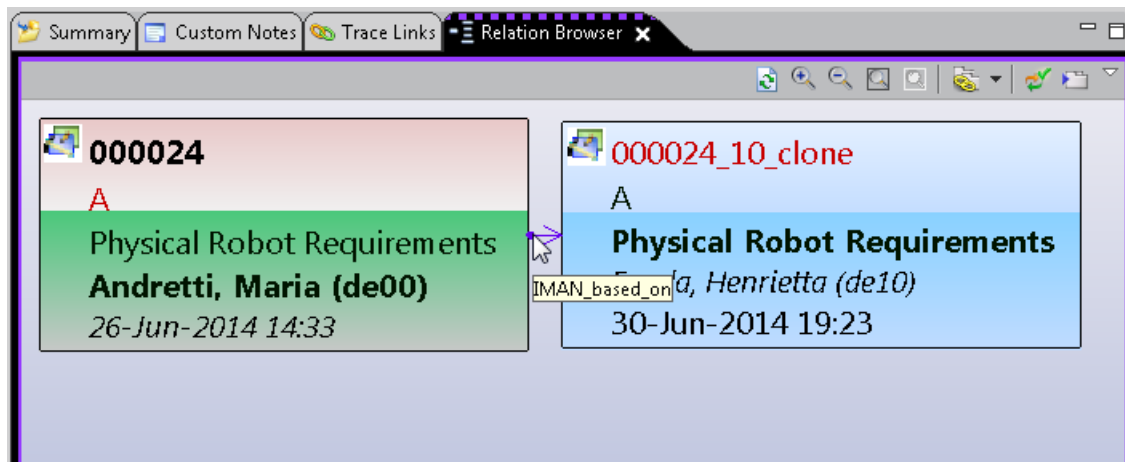
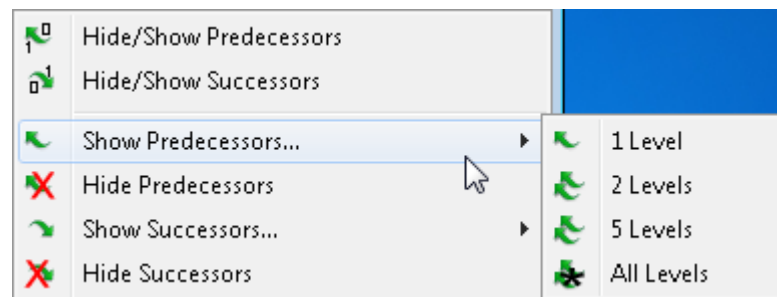
The genealogy feature is enabled by setting **IMAN_based_on** as a shown relation for the requirement revision item type.

Any duplicate requirement objects that are related to the selected requirement object appear in the **Relation Browser** view.

- To view the predecessor (source object) related to the selected requirement object, right-click the node in the **Relation Browser** view and choose **Hide/Show Predecessors**.

Tip:

You can control whether the source (predecessor) and duplicate (successor) objects appear for the selected node using the hide and show successor and predecessor commands in the **Relation Browser** view.



Editing a duplicated requirement specification structure

The following rules apply to editing a structure duplicate:

- The user who executes the **File→Duplicate** command to create the duplicate, owns the duplicate structure.
- Any elements of the original structure that were referenced rather than copied to the duplicate remain owned by their original owner.
- Access rights and ACL rules on structure lines are not carried forward from the original structure to the duplicate.
- You can edit any of the items, item revisions, datasets, and attachments to which you have permissions in the duplicated structure, without affecting the original structure.

Cloning requirement structures

About cloning structures

When creating structures, it can be convenient to create a structure that is based on one that already exists. For example, you may design standard requirement specifications that you can reuse as templates for typical products. Selecting an existing requirement specification or structure as a template is a convenient way to quickly generate another structure.

Use the **File→New→From Template** command to create a structure that is based on another structure that is not currently open. You also use this method to clone structures when you want to select configuration options for the cloning process.

Your administrator specifies *cloning rules* that determine the actions taken when you clone a structure, as follows:

- **Clone**

The new structure includes a copy of the object that is referenced by the source structure.

- **Reference**

The new structure references the same object that is included in the source structure.

- **Ignore**

The new structure does not include or reference the object.

- **Map**

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

Note:

Cloning is a core capability and is used in various functions, such as, create alternate, create from template, process synchronize, and isolated study. It is very recursive and database I/O intensive. To help reduce the number of database commits to the minimum needed, your Teamcenter administrator can create a preference named **ME_defer_save_in_clone** and set it to **true**.

IMPORTANT: The value of this preference should be made **true** only after running cloning usecases in a test environment. This is required as cloning may fail in specific data conditions when the preference is set to **true**.

Note:

By default, trace links in the source structure are not reproduced in the clone. To reproduce trace links from internal defining objects to internal or external complying objects, add one of the following naming rules to the **Product.Template.Cloning** preference value:

:.Relation.FND_TraceLink:Clone	Internal defining trace links are copied from the source to the clone.
:.Relation.FND_TraceLink:Reference	The clone contains references to the source's internal defining trace links.

You cannot reproduce trace links from defining objects that are external to the source.

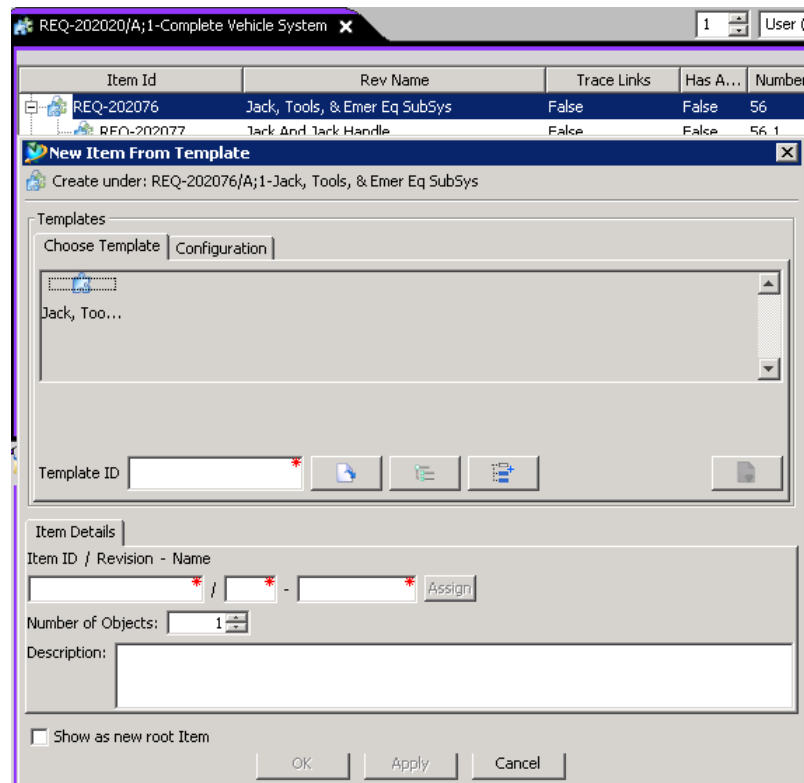
Create a structure from a template

1. In the **Requirement** view, select the parent specification or requirement whose structure you want to clone.

Tip:

If the **Requirement** view does not display the object you want, disregard the current selection. You can browse to and select the object after opening the **New Item From Template** dialog box.

2. Choose **File→New→Item From Template** to open the **New Item From Template** dialog box.



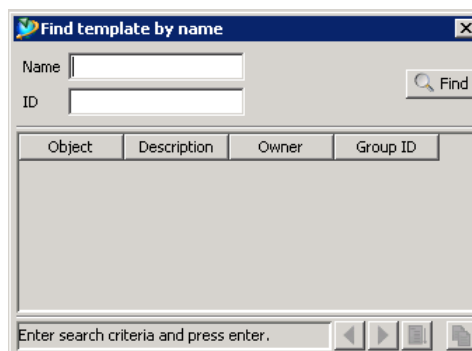
- On the **Choose Template** tab, click the **Set Current Selection** button to fill the **Template ID** box with the parent object identifier.

You can also type the identifier in the **Template ID** box, and then press Enter.

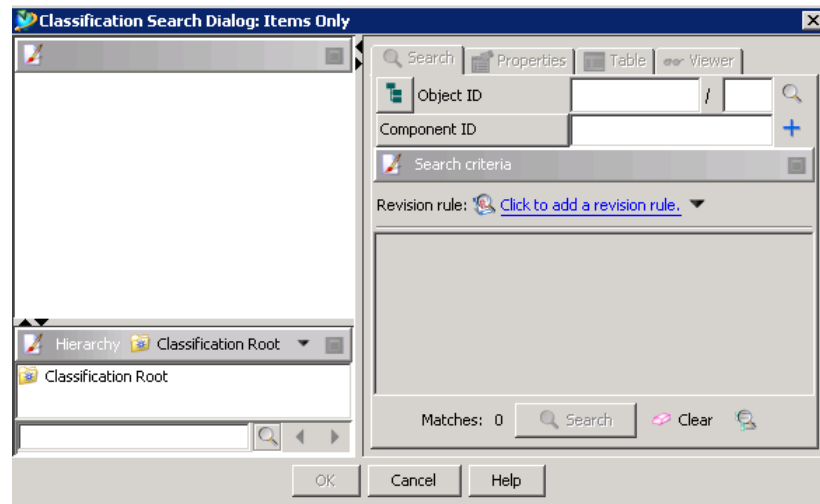
Tip:

To browse to a different parent and enter that object identifier, do one of the following:

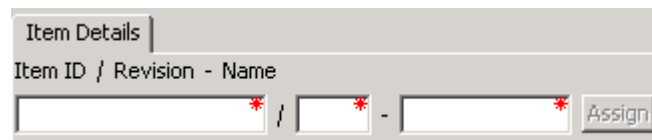
- To search by object name, click the **Find the template by name** button.



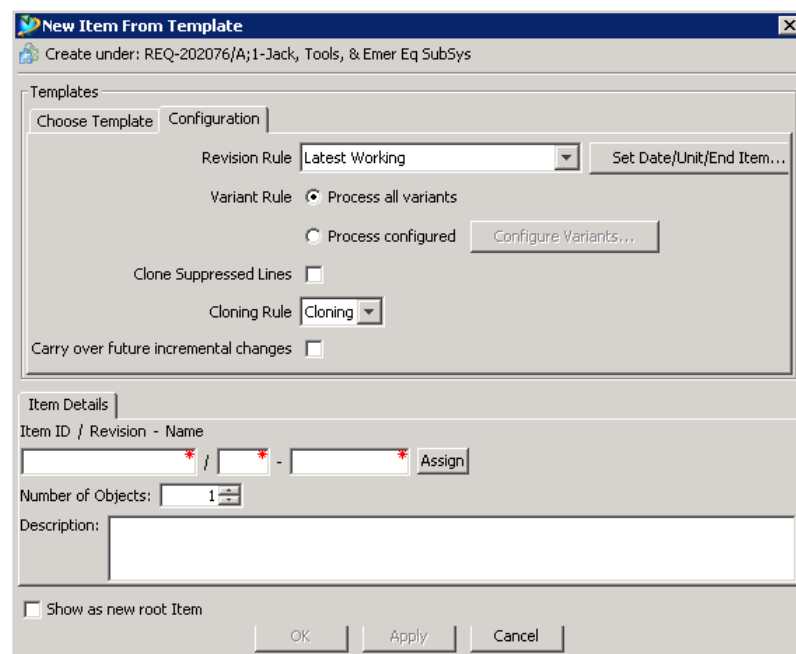
- If the object is classified, click the **Find the template by class** button to use the **Classification Search** dialog box.



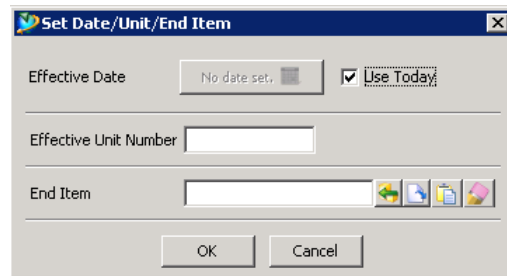
4. On the **Item Details** tab, enter the clone's object identifier, revision, and name.



5. Click the **Configuration** tab.



6. Select a revision rule from the **Revision Rule** list.
7. Click **Set Date/Unit/End Item** to open the **Set Date/Unit/End Item** dialog box and set an effectivity.



8. Do one of the following:
 - Click **Process all variants** to traverse the structure considering all unconfigured variants.
 - Click **Process configured** to only process the configured structure. You must specify a variant rule when you choose this option. The structure is configured using the variant rule and unconfigured variants are not considered during cloning.

9. From the **Cloning Rule** list, select a rule.

10. (Optional) Select **Carry over future incremental changes**.

If you select this option, you must set the date in the cloning dialog. If you do not select this option, cloning copies the configured structure only.

11. (Optional) Select **Show as new root** to create the cloned structure as a top-level object.

12. Click **OK** or **Apply** to create the new structure from the template you choose.

At the end of any cloning operation, if new incremental changes are created as part of cloning process, Teamcenter displays a dialog box listing the new incremental changes.

Designing and validating NX parts with requirements

Requirement types for NX design and validation

In Teamcenter, you can create design and validation requirements only in Requirements Manager and Systems Engineering. To bring design requirements into NX, use the NX **ug_requirement_read** built-in function in the **Expressions** dialog box. Import Teamcenter validation requirements to NX with the NX HD3D **Requirements Validation** tool.

Validation requirements in Teamcenter can be linked to the item revision or BOM line for an NX part. When the part is opened in NX, the linked requirements are automatically present and available for checking. The check results can be saved to Teamcenter.

From within NX, designers can create trace links to external requirements in Teamcenter. Designers can follow trace links to understand the reasons for design decisions, and potentially to alert engineers to the effects of changes to design requirements. Trace links between requirements and parts are shown in Teamcenter trace link reports.

Note:

For information and instructions on requirements validation, see NX Help.

Requirement type	Description
Design Requirement	Contains variables that you define as input parameters for CAD models of NX parts.
Validation Requirement	Contains the following elements: • A <i>condition formula</i> that expresses the validation parameters. • <i>Formula variables</i> that refer to expression definitions by name. • A <i>severity level</i> that specifies the degree to which a validation failure creates a problem.

Tip:

In Teamcenter, you can view validation results in Validation Manager. Also, you can incorporate validation requirements in workflow processes using Workflow Designer and view the processes in Workflow Viewer.

Create an NX design requirement

1. Select the node where you want to create the design requirement at the next lower level, and do one of the following:
 - Choose **File**→**New**→**Requirement**.
 - Click the **Create a new requirement**  button on the toolbar.

Note:

You can also create a design requirement from the Quick Create bar in the **Requirements** view.

- In the **Requirement Type** dialog box of the New Requirement wizard, select **Design Requirement** and click **Next**.
- In the **Requirement Information** dialog box, enter the following information for the new requirement.

Field name	Definition	Valid values
Description	Description of the requirement. Text typed into this box becomes the initial body of the requirement.	Alphanumeric (1–240 characters)
Name	Name of the requirement (required).	Alphanumeric (1–128 characters)
Requirement ID	Requirement ID (required). To fill in the Requirement ID box automatically, leave it empty or click the Assign button. Systems Engineering inserts the next identifier in the sequence and the current revision. The prefix REQ- precedes the identifier.	Alphanumeric (1–128 characters)
Note: The default prefix is set by the Default Requirement ID Rule naming rule.		
Unit of Measure	Unit of measure for the requirement.	Not valid for a design requirement.

- In the **Design Requirement Revision** dialog box, enter a revision for the requirement and click **Next**. You can leave the **Revision** box blank or click the **Assign** button to have Systems Engineering automatically create the next revision number.
- In the **Define Variables** dialog box, click the **+** button to create a new variable.
- Enter the variable information. You can continue to click the **+** button and enter the variable information until you have added all of the variables that you need for the design requirement.

7. Click **Finish**.

The new requirement is created and occupies the last position at the selected level.

You can click **Close** to close the New Requirement wizard, or repeat steps 2 through 7 to create another requirement.

8. (Optional) Edit the content of the requirement specification in either the **MS Word** view or in a separate Word window.

Create an NX validation requirement

Note:

You can create validation requirements only in Systems Engineering.

1. Select the node where you want to create the validation requirement at the next lower level, and then do one of the following:

- Choose **File**→**New**→**Requirement**.
- Click the **Create a new requirement**  button on the toolbar.

Note:

You can also create a validation requirement from the Quick Create bar in the **Requirements** view.

2. In the **Requirement Type** dialog box of the New Requirement wizard, select **Validation Requirement** and click **Next**.
3. In the **Requirement Information** dialog box, enter the following information for the new requirement.

Field name	Definition	Valid values
Description	Description of the requirement. Text typed into this box becomes the initial body of the requirement.	Alphanumeric (1–240 characters)
Name	Name of the requirement (required).	Alphanumeric (1–128 characters)
Requirement ID	Requirement ID (required). To fill in the Requirement ID box automatically, leave it empty or click the Assign button. Systems	Alphanumeric (1–128 characters)

Field name	Definition	Valid values
	Engineering inserts the next identifier in the sequence and the current revision. The prefix REQ- precedes the identifier. Note: The default prefix is set by the Default Requirement ID Rule naming rule.	
Unit of Measure	Unit of measure for the requirement.	Not valid for a design requirement.

4. In the **Define business object create information** dialog box, enter the following information for the new requirement.

Field name	Definition	Valid values
Requirement Formula	Requirement formula.	Alphanumeric (1–240 characters)
Revision	Requirement revision (required). To fill in the Revision box automatically, leave it empty or click the Assign button. Systems Engineering inserts the current revision.	Alphanumeric (1–128 characters)
Severity	Severity level of the requirement (required). The severity specifies the degree to which a validation failure creates a problem.	0/ Error 1/ Warning 2/ Information

5. In the **Define Variables** dialog box, click the **+** button to create a new variable.

Note:

For variables in validation requirements, certain Teamcenter preferences set the default variable type, measurement type, and unit type.

If you have questions about variables, consult your Teamcenter administrator.

6. Enter the variable information. You can continue to click the **+** button and enter the variable information until you have added all of the variables that you need for the validation requirement.
7. Click **Finish** to complete the new requirement and close the wizard.

The new requirement occupies the last position at the selected level.

You can click **Close** to close the New Requirement wizard, or repeat steps 2 through 7 to create another requirement.

8. (Optional) Edit the content of the requirement specification in either the **MS Word** view or in a separate Word window.

Writing formulas for validation conditions

A validation condition formula specifies the parameters for a validation requirement. The formula relates to the expression stored in a given NX part.

- In NX, you can run a requirement check on the part for validation against a selected requirement condition formula.
- The requirement check references the requirement condition formula and maps it to the part expression.

You can use arithmetic functions, operators, and symbols as well as numbers and variables.

For example, the following requirement condition formula limits part dimensions:

$$l + 2*w + 0.9*h < 200[\text{mm}]$$

In this example, the variables for the formula are defined in millimeters as **l** (length), **w** (width), and **h** (height).

You enter condition formulas in the New Requirement wizard.

Defining variables for validation condition formulas

In validation condition formulas, variables refer to expression definitions by name. Variables can refer to definitions such as type, measure, unit, and value.

You enter variables and their details in the **Define Variables** step of the New Requirement wizard.

For example, a requirement condition formula that limits part dimensions is expressed as.

$$l + 2*w + 0.9*h < 200[\text{mm}]$$

In this example, you enter the variables as:

l = length in millimeters

w = width in millimeters

h = height in millimeters

Note:

Certain Teamcenter preferences control default settings for the variables.

If you have questions about variables, consult your Teamcenter administrator.

Editing design and validation requirements

The **Requirement Expression View** displays parameters for the design or validation requirement selected in the **Requirements** view.

Scope: REQ-004053/A;1-val req

View and edit the corresponding properties of Validation Requirement Revision

Severity:
Information

Requirement Formula:
 $l + 2*w + 0.9*h < 200[mm]$
Syntax check

Requirement Expressions:

Target	Name	Type	Measure	Unit	Description	Value
☒	h	Double				
☒	w	Double	Constant	Constant		
☒	l	Double	Constant	Constant		

You can change the requirement's severity level, formula, and expressions. To change an expression variable, double-click the cell to open it for editing.

Note:

The **Requirement Expression View** displays content only when you select a design requirement or a validation requirement.

3. Editing requirements and structures with Teamcenter live Word

How you can use live Word

With the live Word capability of Client for Office, you can use Microsoft Office Word to edit requirements and modify their structures in Requirements Manager. Live Word documents contain property data for requirements and requirement specifications, which you export using options in the Teamcenter **Export to Word** dialog box (**Tools**→**Export**→**Objects to Word**).

Note:

Live Word is available only in the four-tier Teamcenter configuration. With the live Word capability of Client for Office, you can use Microsoft Office Word to edit requirements and modify their structures in Requirements Manager if you have installed and configured Client for Office.

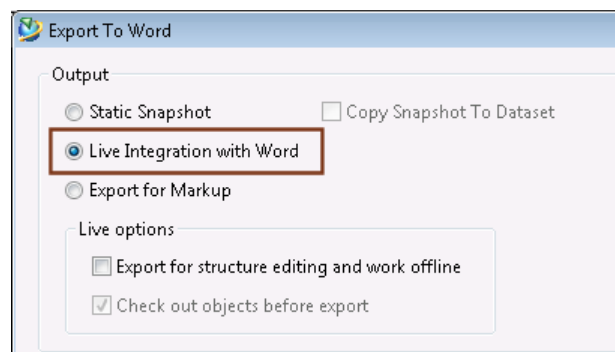
Caution:

- If the **TC_Enable_Implicit_CO** preference is set to **True**, you do not need to explicitly check out requirements before you edit or export them. Also, you do not need to check them in. However, if the **TC_Enable_Implicit_CO** preference is set to **False**, you must explicitly check out requirements before you edit or export them. You must also check them in.
- The associated object template must contain the **{%object_name}** property keyword. Otherwise, you cannot edit requirements or structures in live Word.
- Properties with dynamic or cascading lists of values (LOVs) cannot be modified with live Word.

You can make live Word changes in two ways:

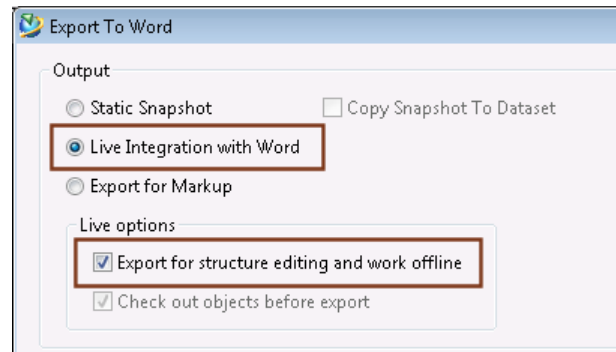
- Interactively in real time, connected to Teamcenter while the rich client is running

To obtain an interactive live Word document, you select **Live Integration with Word**.



- Offline and temporarily disconnected from Teamcenter, accumulating changes for later synchronization with the structure

To obtain an offline live Word document, you select **Export for structure editing and work offline** after selecting **Live Integration with Word**.



The live Word **Requirements** pane is available in interactive and offline documents. This pane contains an expandable tree that shows the object hierarchy of the exported structure. Also, it gives you features for working in the structure and navigating the document content

Note:

If a requirement specification contains child requirement specifications, the nested requirement specifications are also exported to live Word. You can edit the child specifications like any other object in the requirement structure.

Note:

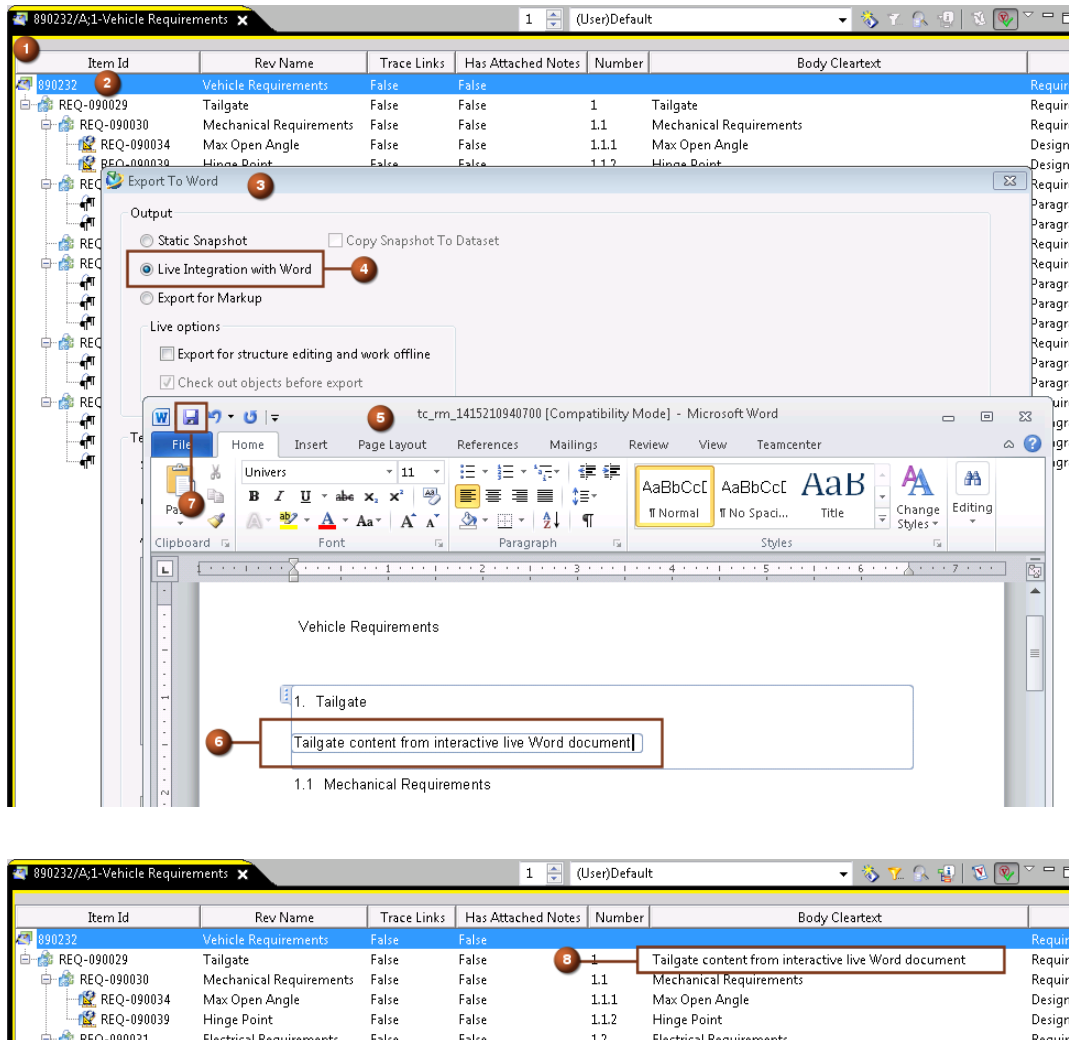
When the **fnd0Isnumbered** property set to **false**, the output of live and static export is different. Static export combines the paragraphs if the **fnd0Isnumbered** property set to **false**. However, in case of live export, since Microsoft Word is connected with Teamcenter, the numbering is maintained. The **fnd0Isnumbered** property is set to **false** for objects that are imported as separate objects because of keyword in it. When performing a static export of such objects, the object name is not exported and the body text of all such objects appears under the parent requirement as body text. This is to ensure that the imported document looks the same as the exported document.

Note:

After you open a Word document using the **Live Integration with Word** option, and you cancel the checkout by clicking **Cancel checkout**, the object is not checked in if it is an **Item** object, not an **ItemRevision** object. The **Cancel checkout** option is available only for **ItemRevision** objects, not for **Item** objects.

Interactive live Word documents

In an interactive live Word document, you work online with Teamcenter in real time. For example, you can add, modify, or delete content for specific requirements. When you save the live Word changes, the document closes and the changes are applied in the structure.



1	Requirements Manager Requirements view	5	Interactive live Word document
2	Requirement specification selected	6	Requirement content added in live Word
3	Export To Word dialog box (Tools→Export→Objects To Word)	7	Word Save button
4	Live Integration with Word option	8	Live Word document closed and changes applied to requirement in structure

Caution:

Properties with dynamic or cascading lists of values (LOVs) cannot be created or modified with live Word.

Note:

The **Word_Show_Display_Name** preference determines the display names shown by default for the objects in the **Requirements** pane. Display names are configured using the **DisplayName** property in the Business Modeler IDE.

Offline live Word documents

An offline live Word document is:

- Generated when you select **Export for structure editing and work offline** in the Teamcenter **Export to Word** dialog box.

This option becomes available when you select **Live Integration with Word**.

- Saved outside of Teamcenter, for example, on your desktop or on a shared drive.
- Temporarily disconnected from Teamcenter.

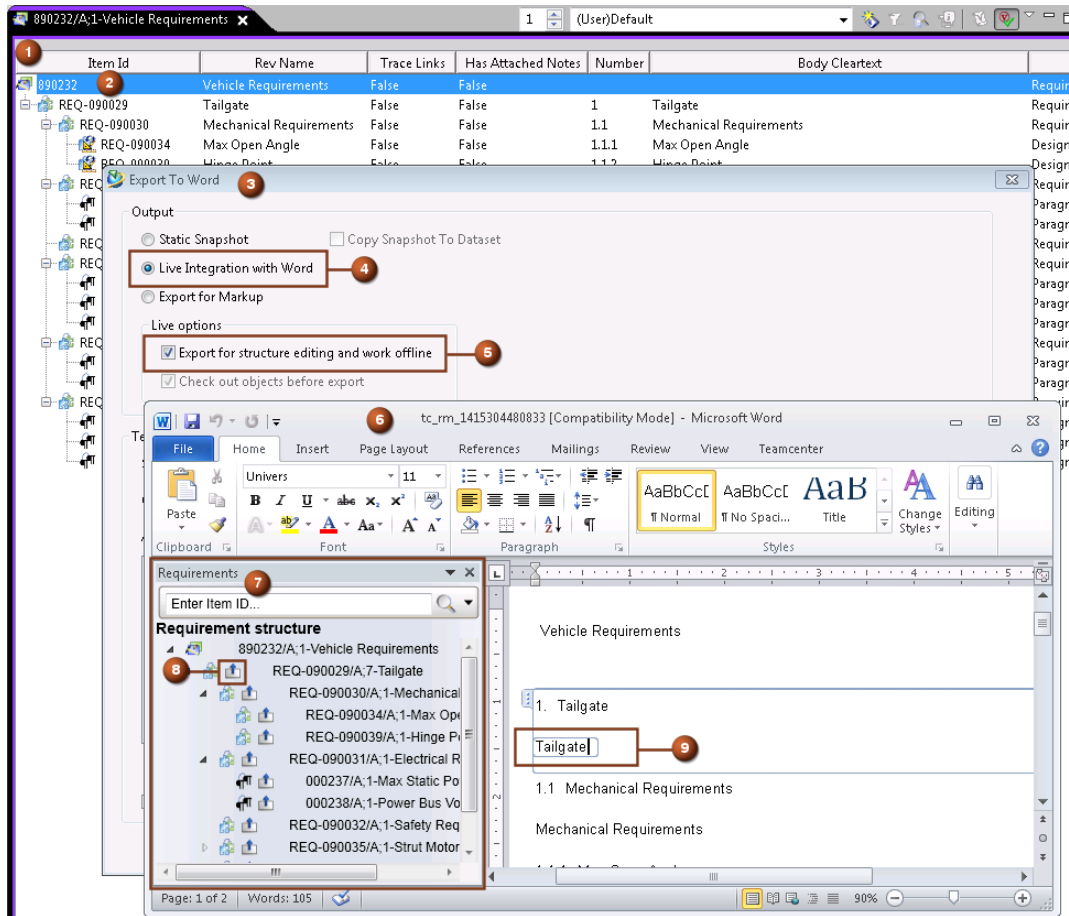
You can use an offline document to accumulate changes, until later you synchronize the changes with the structure. For example, on a business flight, you can edit the offline data on your laptop computer. When you return to your office, you can connect to your Teamcenter server and apply the changes.

Caution:

- Before you open an offline document to accumulate changes, ensure that no other live Word document is connected to Teamcenter. If another document is connected, the offline document you open shares that connection automatically and the current content is open for synchronization with Teamcenter.
- Properties with dynamic or cascading lists of values (LOVs) cannot be created or modified with live Word.

Live Word displays the **Requirements** pane automatically in offline documents. This pane contains an expandable tree, which shows the object hierarchy of the exported structure. You can click a requirement in the pane to navigate to its content in the document.

By default, the exported objects are checked out automatically before the document is generated. This default behavior can be changed with the **REQ_checkout_objects_before_export** preference, which controls the **Check out objects before export** option in the **Export to Word** dialog box.



1	Requirements Manager Requirements view	6	Offline live Word document
2	Requirement specification selected	7	Live Word Requirements pane (structure expanded)
3	Teamcenter Export To Word dialog box (Tools→Export→Objects To Word)	8	Checked out symbol
4	Live Integration with Word option	9	Requirement content
5	Export for structure editing and work offline option		

Note:

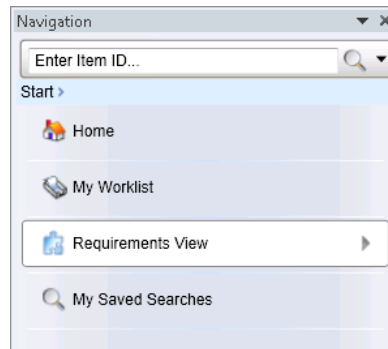
The **Word_Show_Display_Name** preference determines the display names shown by default for the objects in the **Requirements** pane. Display names are configured using the **DisplayName** property in the Business Modeler IDE.

Open the Requirements pane in an interactive live Word document

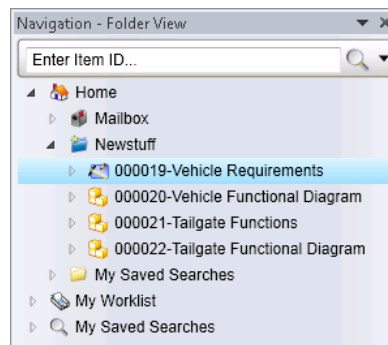
In a live Word document, the **Requirements** pane contains an expandable tree that shows the object hierarchy of an exported structure. From the **Requirements** pane, you can access features for editing requirements and paragraphs and modifying the structure. You can select the object in the **Requirements** pane to find the main content control for a requirement or paragraph.

The **Requirements** pane opens automatically in *offline* live Word documents.

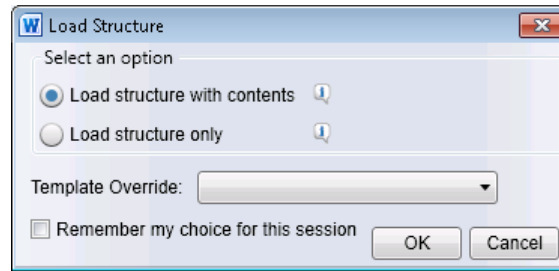
1. On the **Teamcenter** tab, click the **Navigate** button, and choose **Browse** or **Folder View**.
2. Do one of the following:
 - In the **Navigation** browse pane:
 - a. Click **Requirements View** to display your recently opened specifications.



- b. Right-click the specification that you want to load in the **Requirements** pane, and then choose **Open**.
- In the **Folder View**, right-click the specification that you want to load in the **Requirements** pane, and then choose **Open**.



3. In the **Load Structure** dialog box, select one of the following options:



Load structure with contents

Opens a new Word window, with the structure in the **Requirements** pane and all requirement content in the document.

Each requirement's level and position in the structure matches the related content's heading number and location in the document. You can select a requirement in the structure to navigate to its content in the document.

Tip:

In the **Template Override** list, you can select the object template to apply to the content. This selection overrides the object template selected in exporting the structure.

Load structure only

Opens a new Word window with only the structure in the **Requirements** pane. The content pane is empty.

You can right-click a requirement in the structure and choose **Open** to display the content in a new Word window.

Tip:

To bypass the **Load Structure** dialog box for opening subsequent structures, select **Remember my choice for this session**. With this setting, the option you first select is valid until you close all interactive live Word documents.

Note:

The **Word_Show_Display_Name** preference determines the display names to be shown by default for the objects in the **Requirements** pane. Display names are configured using the **DisplayName** property in the Business Modeler IDE.

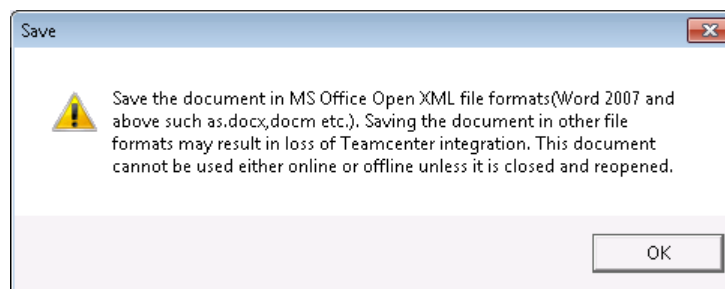
Save a live Word document for offline editing

Caution:

Live Word supports only Microsoft Office Open XML file types, such as **.docx** and **.docm**. With any other file type, such as **.doc**, the document may lose its live capability and become permanently static.

1. On the Office ribbon, click the **File** tab and choose **Save As**.

A message warns you to save the document in an MS Office Open XML file format such as **.docx** and **.docm**.



2. Click **OK** to dismiss the message.

The Word **Save As** dialog box opens. You can select the location for the offline document, for example, a folder on your local file system or a shared drive.

Caution:

Ensure that the selection in the **Save as type** list is a file type of the Microsoft Office Open XML format, such as **Word Document (*.docx)** or **Word Macro-Enabled Document (*.docm)**.

3. Click **Save** to store the document and close the dialog box.

After you close the document, it retains its live Word capability, although it is temporarily disconnected from Teamcenter. The next time you open the document, you can do either of the following:

- Accumulate changes pending connection to Teamcenter at a later time.
- Connect to Teamcenter and synchronize changes with the corresponding structure.

Accumulate live Word changes offline

In an offline live Word document, you can edit the exported requirements and modify the structure, saving the document offline until you are ready to commit the pending changes. At that time, you reconnect the document and synchronize it with Teamcenter.

Caution:

- Before you open an offline document to accumulate changes, ensure that no other live Word document is connected to Teamcenter. If another document is connected, the offline document you open shares that connection automatically and the current content is open for synchronization with Teamcenter.
- Properties with dynamic or cascading lists of values (LOVs) cannot be created or modified with live Word.

Note:

By default, the exported objects are checked out automatically before the document is generated. This default behavior can be changed with the **REQ_checkout_objects_before_export** preference, which controls the **Check out objects before export** option in the **Export to Word** dialog box.

1. From your local file system, open the offline live Word document.

A message states that you have opened a Teamcenter document and asks if you want to connect to Teamcenter.

2. Click **No**.

A message states that the document is not connected to Teamcenter and will remain in offline mode.

3. Click **OK** to dismiss the message and begin making changes.

In the content controls, you can add, change, copy, move, or delete content using the related Word features. You can also enter or modify content outside of the content controls, for example, boilerplate text and logos or informal comments.

Use Word's document storage features to save your changes.

Tip:

Asterisks (*) in the **Requirements** pane indicate which objects have changes that are not committed to Teamcenter.

4. When you complete your changes, close the document.

Note:

If you navigated to any content control, even if you made no changes, Word displays a message asking if you want to save the changes. The message appears whether you saved changes previously.

You can click **Yes** to open the **Save As** dialog box. Or, click **No** to dismiss the message and close the document.

Your changes are saved offline pending later synchronization with Teamcenter.

Editing requirements and paragraphs in Live Word

When you export requirements and paragraphs to Word, you select a specification template and an object template. The object template determines which properties are included in the output document. Teamcenter extracts the property values to Word content controls in the document window.

You use the standard Word editing features to add, modify, copy, move, or delete content in the content controls. When you save the changes, the document closes and the changes are applied to the objects in Teamcenter.

Caution:

Properties with dynamic or cascading lists of values (LOVs) cannot be created or modified with live Word.

Tip:

To find the main content control for a requirement or paragraph, select the object in the **Requirements** pane. Client for Office highlights the content control in the document window.

With the default object template applied, the Word output document contains the following Teamcenter property values for each requirement and paragraph object:

Note:

When using Teamcenter live Word to export requirements, formatting capabilities in Word are disabled for plain text entities such as the object name. Formatting capabilities are enabled for rich text entities such as body text.

- The **Name** property value is the object name.
- This value is formatted in a Word heading style with a numbered outline level.

- The heading occupies a nested content control at the top of the main content control for the object.
- If you edit the heading, the **Requirements** pane displays a red asterisk (*) to the right of the object.

The asterisk indicates that changes are pending for the object.

- The **Body Text** property value is the rich text content of the requirement or paragraph object.

Note:

If you are working in a live Word output file, it may include a URL link to body text. In this case, click the link to access and review or edit the requirement content.

- This value is formatted in a Word paragraph style.
- Each Word paragraph occupies a nested content control within the main content control for the object.
- If you place the cursor within the nested content control, the **Requirements** pane displays a red asterisk (*) to the right of the object.

The asterisk indicates that changes are pending for the object. However, the asterisk appears whether or not you edit the **Body Text** property value.

Caution:

When copying from one content control to another, the change is not applied in Teamcenter if the source selection includes the content control itself. You can avoid this condition by doing either of the following:

- In the source content control, ensure that the control border is not highlighted.
- In the target content control, select **Keep Text Only** under **Paste Options**.

This action removes the source content control and inserts the remaining clipboard contents.

Caution:

- Edits made outside of a content control are not applied to objects in Teamcenter. Such edits become static content in the document.
- A new object is not created if you copy an object's main content control in the Word document window. The content becomes static where it is pasted in the document.

Modifying requirement structures in live Word

Building requirement specifications in Word

A requirement specification is a container for requirements and paragraphs. These objects can be structured in hierarchy of parent, child, and sibling relationships.

With Client for Office, you can use Microsoft Office Word to build requirement structures in Systems Engineering by performing any of the following:

- Build a structure manually in the **Requirements** pane and the document window.
- Build a structure automatically by importing a Word document as a new requirement specification.

The following table shows the options for manipulating requirement structures and the supported document modes.

Options for manipulating requirement structures	Document mode
Creating new requirement specifications.	Online and connected to Teamcenter.
Creating and copying requirements and paragraph objects.	Offline or online
Moving requirements and paragraph objects by cutting and pasting.	Offline or online
Promoting and demoting requirements and paragraph objects to higher and lower levels.	Offline or online
Removing requirements and paragraph objects from structures.	Offline or online
Rearranging requirements and paragraph objects within a level.	Offline or online
Creating trace links between objects.	Online and connected to Teamcenter.
Editing object properties.	Offline or online
Initiating workflow processes for selected objects.	Online and connected to Teamcenter.

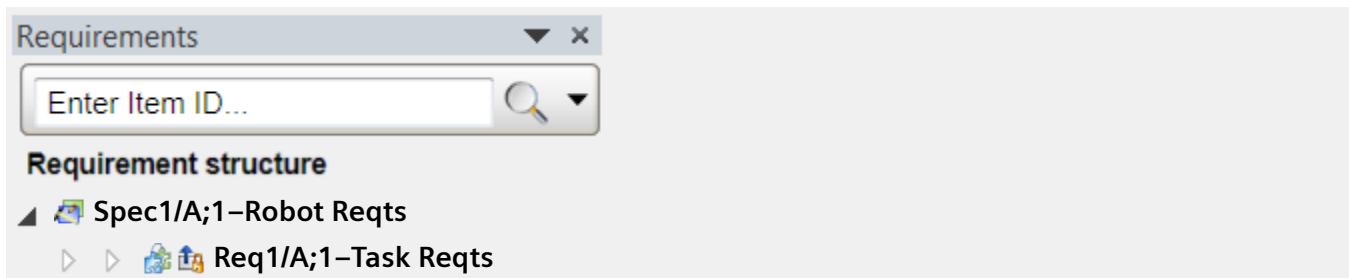
You cannot change requirement objects in the structure that are locked for edit.

The requirement object in the exported Word document is locked for edit for the following conditions:

- You do *not* have write access to the object.
- The object is at a released state and is *not* modifiable.
- The object is checked out by another user.

The locked symbol  displays on the symbol next to the object indicating the object is locked for edit.

As the following figure shows, in the **Requirements** pane in Word, the symbol  displays next to the **Req1/A;1–Task Reqts** requirement indicating it is locked for edit because another user has it checked out.



Create a requirement specification

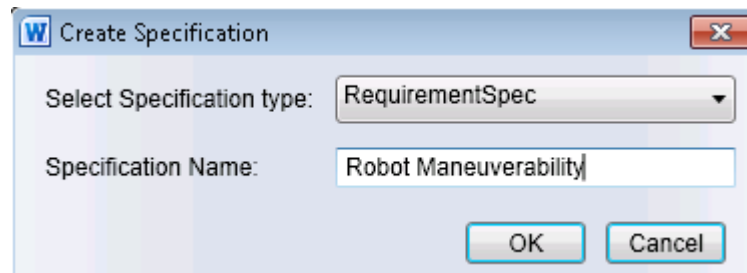
Caution:

To create a requirement specification in an offline live Word document, you must synchronize the document with Teamcenter.

1. Open a new Microsoft Office Word document.
2. On the **Teamcenter** tab, click the **Specifications** button  and choose **New**.
3. Log on to Teamcenter.

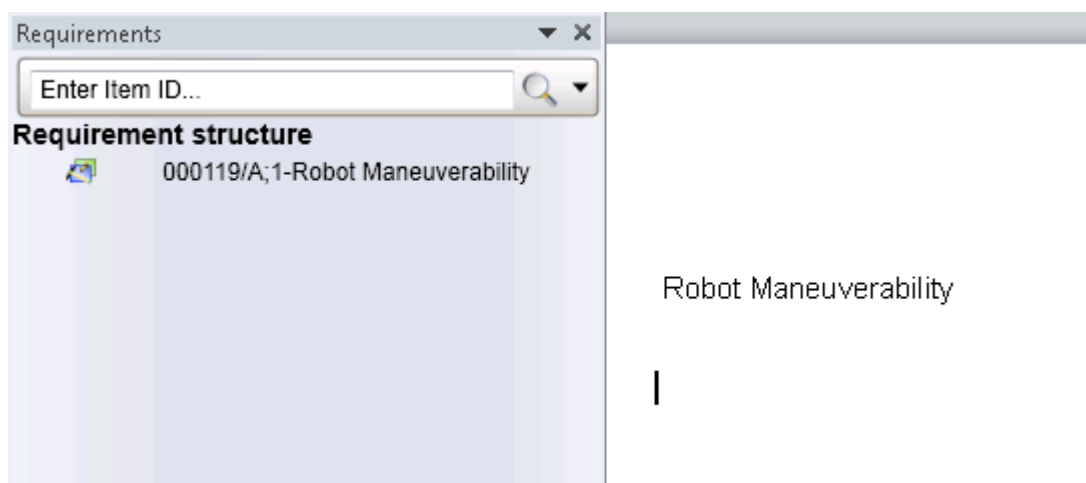
The **Create Specification** dialog box displays.

4. In the **Select Specification type** box, select **RequirementSpec**.
5. In the **Specification Name** box, enter a specification name.



6. Click **OK**.

The **Requirements** pane opens in the Microsoft Office Word document and displays the new requirement specification object you created.



Create a requirement or paragraph object

Create a requirement or paragraph from the Requirements pane

Warning:

Do not use the **New** button on the **Teamcenter** tab of the Office ribbon. The **New** button does not support requirement objects or their subtypes.

- Assign the object's subtype and set its structure level by doing one of the following:
- To accept the default subtype, right-click the intended parent object and choose **New**.

Note:

- If the parent is the peak requirement specification, the default subtype is **Requirement**.

- If the parent is a requirement or paragraph object, the default subtype is the same as the parent's subtype.

The new object displays below the parent in the **Requirements** pane and in the document.

To change the name, you can right-click the object and choose **Rename**.

Note:

The **Rename** command is available only for new objects.

- To assign a different subtype:
 1. Right-click the intended parent object and choose **New....**

Note:

The parent can be the peak requirement specification or a lower level requirement or paragraph object.

The **Create New** dialog box displays.

2. Select the subtype in the **Select Type** list, and type the object name in the **Name** box.
3. Click **OK**.

The new object displays below the parent in the **Requirements** pane and in the document.

To change the name, you can right-click the object and choose **Rename**.

Create a requirement or paragraph by copying an existing object

1. In the **Requirements** pane, right-click the source object and choose **Copy**.

The object is marked for copying.

Note:

Child objects are not copied with the source object.

2. Right-click the target parent object and choose **Paste as New**.

The new object appears below the parent and the source object marking is removed.

Note:

A new object is not created if you copy an object's content control in the Word document window. The copy becomes static where it is pasted and is not added to the **Body Text** property of the target object.

Promote or demote an object

Note:

- You cannot promote or demote an object by moving its content control in the document window.
- You cannot promote or demote an object that is the only one at its current level.
- Children of a promoted or demoted object remain with the parent at its new level.

- To change the structure level of a requirement or paragraph object, do one of the following in the **Requirements** pane:
 - For the next higher level, right-click the object and choose **Promote**.
 - For the next lower level, right-click the object and choose **Demote**.

The object occupies the last position at the new level in the **Requirements** pane. In the document window, the object's main content control moves to the corresponding location.

Move an object within a level

Note:

- You cannot move an object by moving its content control in the document window.
- You cannot move an object that is the only one at its current level.
- Children of a moved object remain with the parent in its new position.

- In the **Requirements** pane, do one of the following:
 - To move an object up by one position, right-click the object and choose **Move Up**.
 - To move an object down by one position, right-click the object and choose **Move Down**.

The object occupies the next available position at the same level in the **Requirements** pane. In the document window, the object's main content control moves to the corresponding position.

Remove a requirement or paragraph object from a structure

Caution:

This procedure also removes all lower level objects below the requirement or paragraph.

Note:

- This procedure does not delete the object and its subordinates from Teamcenter but only removes them from the structure.
- The object is not removed from the structure if you delete the related content from the Word document window.

- In the **Requirements** pane, right-click the requirement or paragraph and choose **Mark for Deletion**.

A red strike-through marks the following:

- The selected object in the **Requirements** pane
- The related content in the Word document window
- All lower level objects in the **Requirements** pane and their content in the document window

Note:

- The marked objects are removed from the structure when you commit changes to Teamcenter.
- Until you commit the changes, you can undo the strike-through by right-clicking the object and choosing **Unmark for Deletion**.

Moving and copying requirements between structures

Manipulating requirement objects between structures

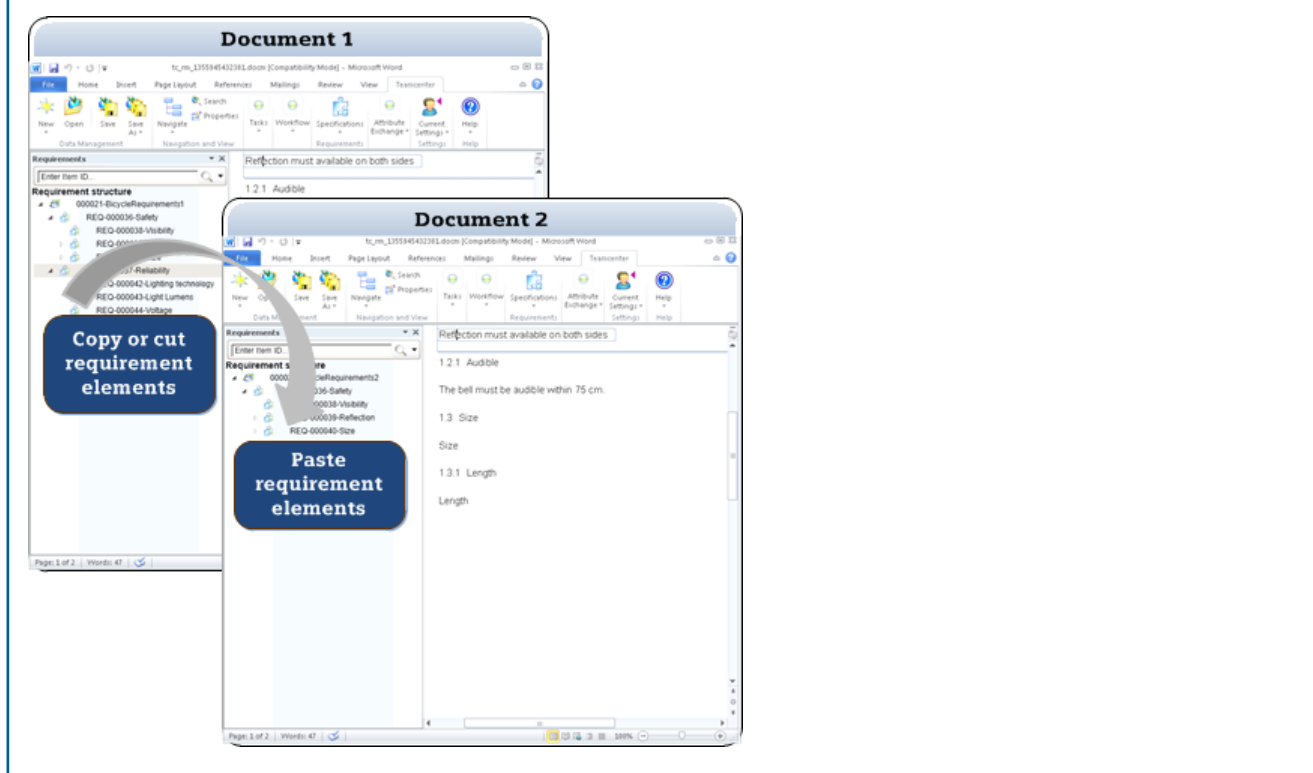
You can manipulate requirement objects between requirement structures by cutting or copying, and then pasting the objects from one document to another, or within the same document.

Note:

Requirement objects include requirement specifications, requirements, or paragraphs, and nested requirement specifications, requirements, or paragraphs.

Example:

As shown in the following figure, you can copy or cut and then paste requirement objects from the **Requirements** pane between the Word documents. You can also copy or cut and then paste within the same document.



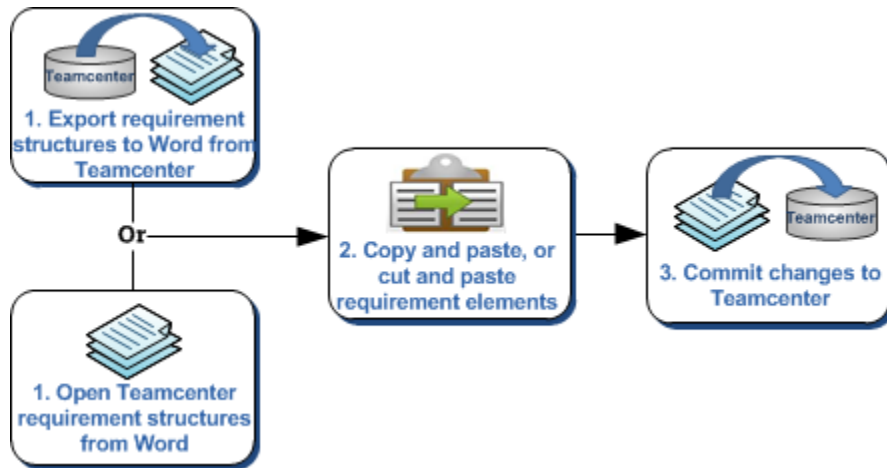
- When you perform the *copy and paste* operation, you copy a requirement object from a source requirement structure to a destination requirement structure. The source structure remains unchanged.
- When you perform the *cut and paste* operation, you remove a requirement object from a source requirement structure and move it to a destination requirement structure.
- The contents of the object copied or cut from the source structure are also copied to the destination structure.

Note:

You *cannot* perform the copy and paste operation or the cut and paste operation on the *root* requirement specification in the source structure.

Process for manipulating requirement objects between structures

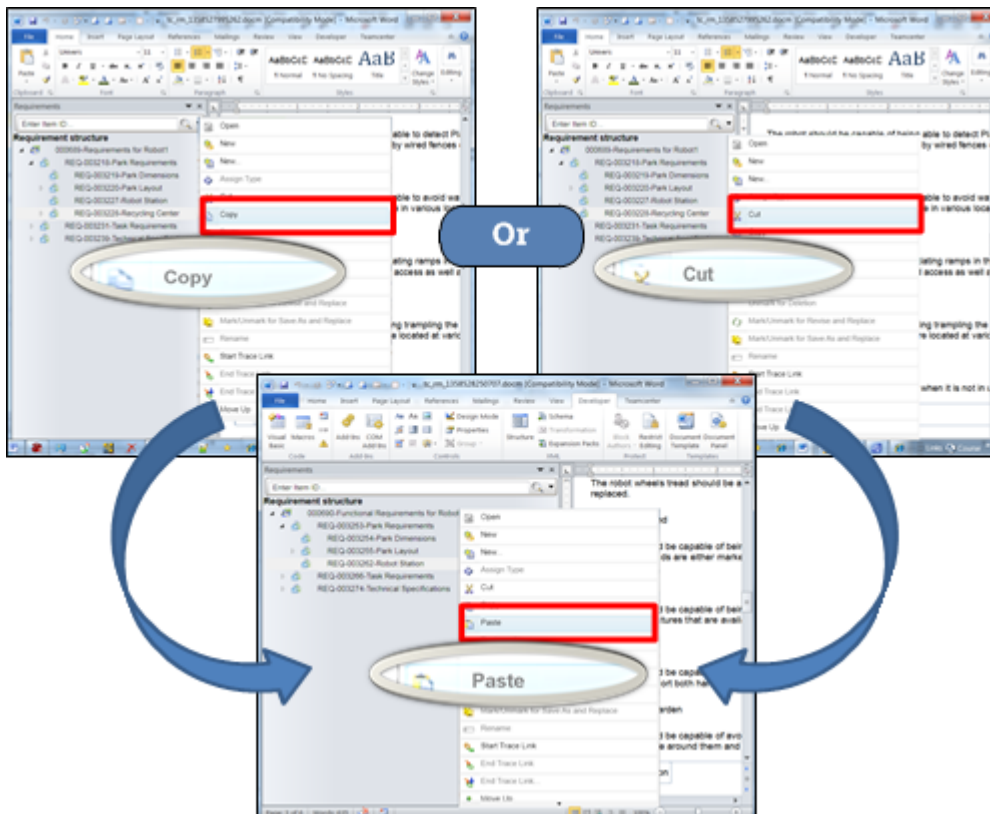
Moving or copying requirement objects between structures is part of a three-step process:



1. Open the requirements structures in live Word using one of the following methods:
 - Open an offline Word document that contains a requirements structure exported from Teamcenter.
 - Open a Teamcenter requirement structure from the Teamcenter ribbon in Word.
2. From the **Requirements** pane in the Word document, copy or cut a requirement object, such as a nested requirement specification, requirement, or paragraph, from the structure. Then, paste the requirement object to the desired location in the destination requirement structure. The destination structure can be in another Word document or within the same Word document.

Note:

You *cannot* perform the copy and paste operation or the cut and paste operation on the *top-level* requirement specification object in the requirements structure.



3. Commit the changes to Teamcenter.

- If you are working online and are connected to Teamcenter, the changes to the requirements structures are made to Teamcenter when you save the document.
- If you are working offline and save the document to the disk, when you reopen the document, you have the option to synchronize and commit the requirement structure changes to Teamcenter.

When you commit the changes to Teamcenter for the *destination* structure in Word, the copied or moved objects from the source structure are copied by *reference* to the destination structure in Teamcenter.

Limitations for cut or copy and paste between structures

You *cannot* cut or copy and paste a requirement object between structures if:

- The requirement object you are cutting is the *top-level* requirement specification of the source document.

Note:

This limitation does not apply if you are copying rather than cutting.

- The object template used when exporting the requirement structure from Teamcenter to Word is different for the source and destination documents.
- One of the documents is a document generated from a preview operation.
- The source and destination structures are different types of structures. Both structures must be either precise or imprecise.
- The target location where you want to paste the requirement object in the destination structure has any of the following conditions:
 - The BOMview revision (BVR) of the target object is released.
 - The BVR of the target object is checked out or locked by the access control list (ACL).
 - The target object is released and does *not* have a BVR.
 - The target object is locked by the ACL and does *not* have a BVR.

Note:

If you are cutting and pasting requirement objects between or within documents and the paste operation fails for the destination structure, the source structure is *not* affected and the selected requirement object is not removed from the source structure.

Impact of cut or copy and paste on structures

The following occurs when you cut or copy a requirement object from a source structure and paste it to a destination structure:

- The contents of the cut or copied objects are copied to the destination structure.
- The properties of the cut or copied objects are copied to the destination structure.
- If the destination structure includes indented levels, the level heading of the pasted requirement object is adjusted to conform to the destination structure headings.
- The requirement objects that are cut from the source document are displayed with a red strike-through indicating they have been removed from the structure.

- If you are cutting a requirement object from a source structure and pasting the same requirement object multiple times to a destination structure, duplicate occurrences are created in the destination structure in Word.

Note:

Only the properties and content of the *first* occurrence can be edited. If you want to edit the content and properties of the second occurrence, you must **Mark for Deletion** the first occurrence.

When you save the document in Word, the structural changes are committed to Teamcenter.

How to commit structure changes to Teamcenter

You can cut or copy and paste requirement objects between structures in online or offline modes. How you commit requirement structure changes to Teamcenter depends on which mode you are working in and whether you used cut or copy.

- **Online mode**

If you are connected to Teamcenter, the document is an *online* document and you are working in online mode.

- When you use cut and paste, commit the changes to Teamcenter by saving the updated source and destination documents.
- When you use copy and paste, commit the changes to Teamcenter by saving the destination document.

Note:

There are no changes to the source documents when you copy requirement objects.

- **Offline mode**

If a requirement structure was exported to live Word, and the Word document saved to disk, the document is considered *offline*. If you are working in the local copy of the Word document and are not connected to Teamcenter, you are working in offline mode.

- When you use cut and paste, commit the requirement structure changes to Teamcenter by closing and reopening the updated source and destination documents. After you have reopened the documents, you have the option of synchronizing and committing the requirement structure changes to Teamcenter.
- When you use copy and paste, commit the requirement structure changes to Teamcenter by closing and reopening the updated destination document. After you have reopened the document,

you have the option of synchronizing and committing the requirement structure changes to Teamcenter.

Note:

There are no changes to the source documents when you copy requirement objects.

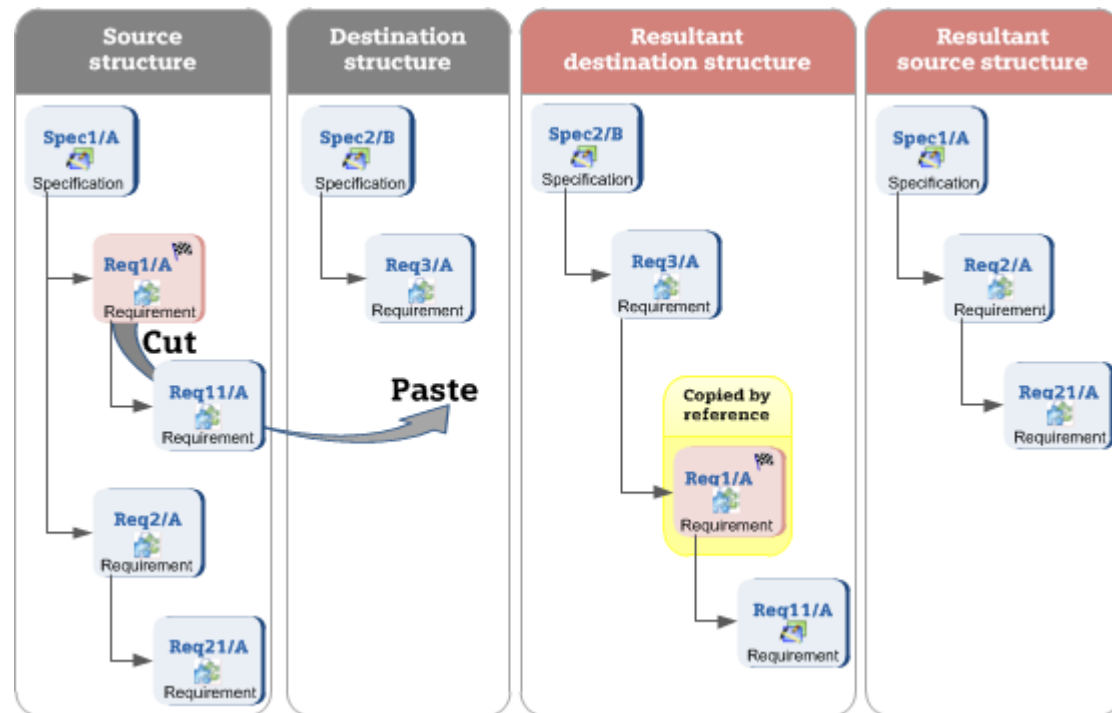
Results of cut and paste between structures

When you save the source and destination documents in Word, the structural changes are committed to Teamcenter.

- When you commit the changes to Teamcenter for the *destination* structure in Word, the object that you cut from the source document and pasted to the destination document is:
 - Added to the destination document in Word.
 - Added to the destination structure in Teamcenter as a *reference* copy.
- When you commit the changes to Teamcenter for the *source* structure in Word, the object that you cut from the structure is:
 - Removed from the source document in Word.
 - Removed from the source structure in Teamcenter.

Example:

As shown in the following example, the **Req1/A** requirement is cut and pasted from the **Spec1/A** specification in the *source* structure to the **Spec2/B** specification under the **Req3/A** requirement in the *destination* structure. When you save the Word document and commit the changes to Teamcenter, **Req1/A** is copied by *reference* to **Spec2/B** under **Req3/A** in the *resultant destination* structure. The **Req1/A** and the **Req11/A** child requirements are removed from the *resultant source* structure.



Because this example shows the source and destination structures as being *precise* structures in Teamcenter, **Req3/A** in the destination structure references **Req1/A**. If the source and destination structures were *imprecise* structures in Teamcenter, **Req3** in the destination structure would reference **Req1**.

Results of copy and paste between structures

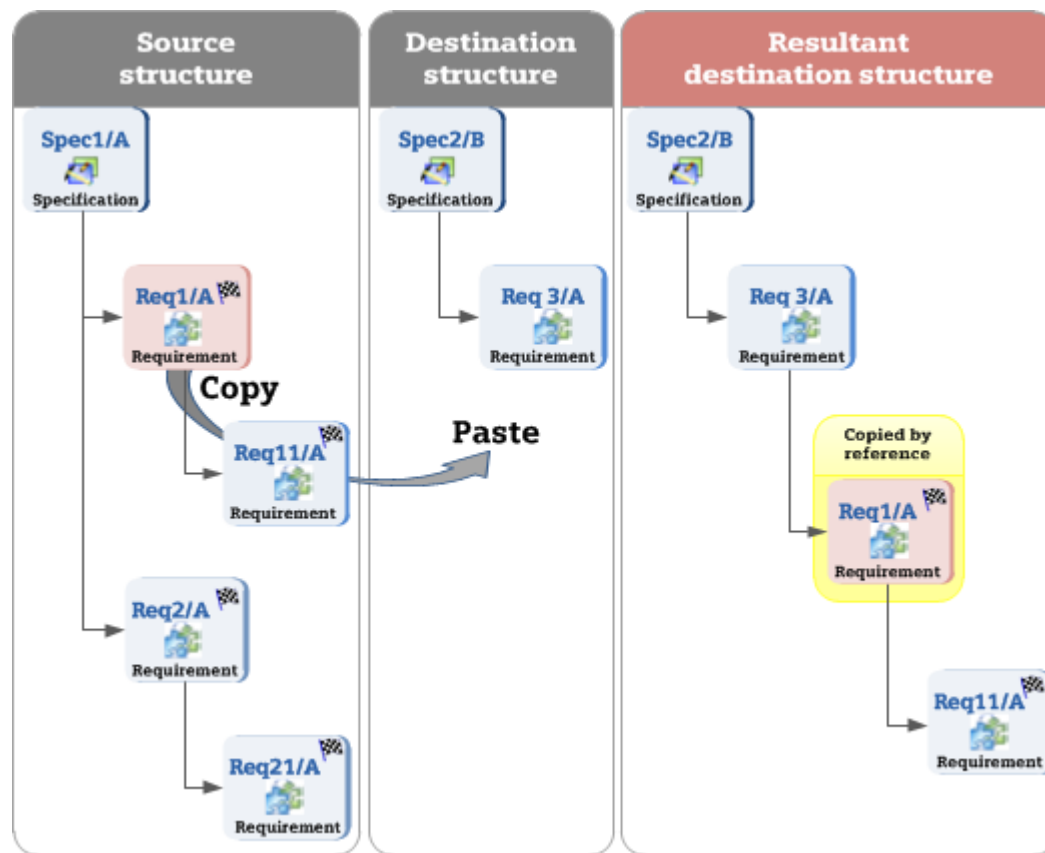
When you commit the changes to Teamcenter for the *destination* structure in Word, the object that you copied from the source document is:

- Copied as a new object to the destination structure in Word.
- Copied as a *reference* copy to the destination structure in Teamcenter.

The source document in Word and the source structure in Teamcenter remain unchanged.

Example:

As shown in the following example, the **Req1/A** requirement is copied and pasted from the **Spec1/A** specification in the *source* structure to the **Spec2/B** specification under the **Req3/A** requirement in the *destination* structure. When the changes are committed to Teamcenter, **Req1/A** is copied by *reference* to **Spec2/B** under **Req3/A** in the *resultant destination* structure.



Because this example shows the source and destination structures as being *precise* structures in Teamcenter, **Req3/A** in the destination structure references **Req1/A**. If the source and destination structures were *imprecise* structures in Teamcenter, **Req3** in the destination structure would reference **Req1**.

Cut and paste requirement objects between structures

1. Open the requirements structures in live Word using one of the following methods:
 - Open an offline Word document containing a requirement structure previously exported from Teamcenter Systems Engineering.
 - Open and load a Teamcenter requirements structure from the Teamcenter ribbon in Word.

The requirements structure appears in the **Requirements** pane in Word.

2. In the **Requirements** pane in Word, in the source structure, right-click the requirement object that you want to remove, and choose **Cut**.
3. In the **Requirements** pane in Word, in the destination structure, right-click the target object in the location where you want to add the requirement object to the structure, and choose **Paste**.

Note:

The destination structure can be within the same Word document or in a different Word document.

The requirement object that you cut from the source structure is added to the destination structure, as a child of the target object.

4. (Optional) Repeat steps 2 and 3 if you want to create multiple occurrences of the same requirement object in the destination structure.
5. Commit the requirement structure changes to Teamcenter.
 - If you are working online and connected to Teamcenter, perform the following steps for the documents containing the source and destination structures:

Step 1: Click the **Save** button  or choose **File → Save** in Word to save the changes in Teamcenter.

Step 2: In the **Save** dialog box, if you checked out the objects before the export, select **Save and Check-in** to check in the objects.

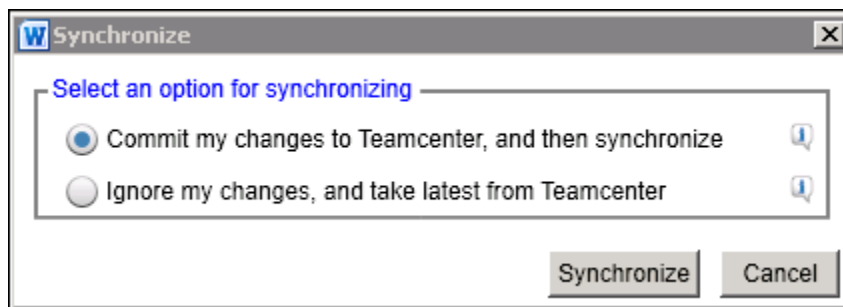
Step 3: Click **OK**.

The document is saved and the changes are made in Teamcenter.

- If you are working in an offline document and not connected to Teamcenter, perform the following steps for the documents containing the source and destination structures:

Step 1: Close the Word document and reopen it.

Step 2: In the **Synchronize** dialog box, select the **Commit my changes to Teamcenter, and then synchronize** option and then click **Synchronize**.



Step 3: In the **Save** dialog box, if you checked out the objects before the export, select **Save and Check-in** to check in the objects.

Step 4: Click **OK**.

Step 5: Complete the **Teamcenter Login** dialog box and then click **Login** to log on to Teamcenter.

Step 6: Click **OK**.

The document is saved and the changes are made in Teamcenter.

Copy and paste requirement objects between structures

1. Open the requirements structures in live Word using one of the following methods:
 - Open an offline Word document containing a requirement structure previously exported from Teamcenter Systems Engineering.
 - Open and load a Teamcenter requirements structure from the Teamcenter ribbon in Word.

The requirements structure appears in the **Requirements** pane in Word.

2. In the **Requirements** pane in Word, in the source structure, right-click the requirement object that you want to copy, and choose **Copy**.
3. In the **Requirements** pane in Word, in the destination structure, right-click the target object in the location where you want to add the requirement object to the structure, and choose **Paste**.

Note:

The destination structure can be within the same Word document or in a different Word document.

The requirement object that you copied from the source structure is added to the destination structure, as a child of the target object.

4. (Optional) Repeat step 3 if you want to create multiple occurrences of the same copied requirement object.
5. Commit the requirement structure changes to Teamcenter.
 - If you are working online and connected to Teamcenter, perform the following steps for the documents containing the destination structure:

Step 1: Click the **Save** button  or choose **File→Save** in Word to save the changes in Teamcenter.

Step 2: In the **Save** dialog box, if you checked out the objects before the export, select **Save and Check-in** to check in the objects.

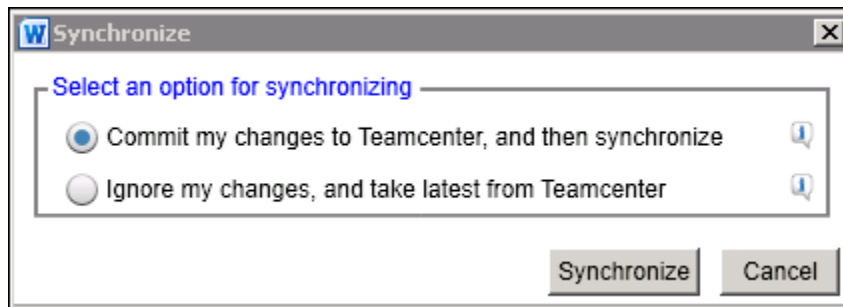
Step 3: Click **OK**.

The document is saved and the changes are made in Teamcenter.

- If you are working in an offline document and not connected to Teamcenter, perform the following steps for the document containing the destination structure:

Step 1: Close the Word document and reopen it.

Step 2: In the **Synchronize** dialog box, select the **Commit my changes to Teamcenter, and then synchronize** option and then click **Synchronize**.



Step 3: In the **Save** dialog box, if you checked out the objects before the export, select **Save and Check-in** to check in the objects.

Step 4: Click **OK**.

Step 5: Complete the **Teamcenter Login** dialog box and then click **Login** to log on to Teamcenter.

Step 6: Click **OK**.

The document is saved and the changes are made in Teamcenter.

Revising released objects in a requirements structure

Overview of revising released objects

In Word, when you open a requirement specification structure, the object hierarchy for the structure appears in the **Requirements** pane. The **Requirements** pane enables you to navigate and change the requirement structure. From the **Requirements** pane, you can create and edit a new revision of a released requirement object by marking it for *revise*. Requirement objects includes requirement specifications, requirements, or paragraphs.

The requirement objects must be released and *not* modifiable before you can mark them for *revise*.

For requirement structures displayed in the **Requirements** pane, you can perform a **Revise** operation that allows you to:

- Create a new revision of an existing released requirement object in Word.
- Make changes to the new revision of the object.
- Commit those changes to Teamcenter by saving the Word document.

The requirements structure can be an *imprecise* or *precise* structure. *Imprecise* structures are dynamic. An imprecise structure has links (occurrences) to item objects. *Precise* structures are fixed structures. A precise assembly has links (occurrences) to item revision objects.

Depending on the conditions met, when you perform a **Revise** operation on a selected released object in the structure, one of the following occurs:

- Only the single selected object is marked for *revise*.
- All of the parent nodes to the top of the structure, including the selected object, are marked for *revise* until an open BOMview revision (BVR) object is found. This only occurs for *precise* requirements structures. Additional conditions must be met for this behavior to occur.

Note:

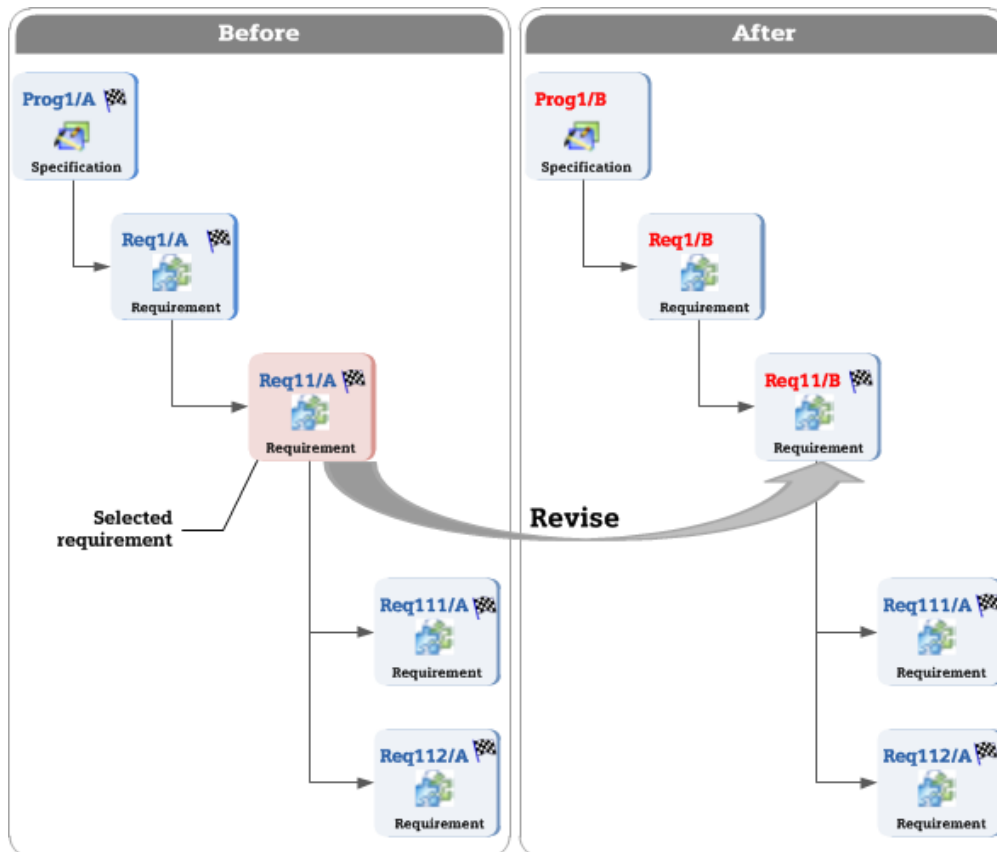
All objects below the object marked for revision are copied by reference to the revised structure.

Example of revising released objects

You create and release a *precise* requirement structure named **Prog1** for an existing product program that contains a set of requirements. The program has now evolved, and you want to revise and modify the set of requirements for the **Req11** requirement.

Because the object item revision and the BOMview revision (BVR) of its parent node are released and not modifiable, when you perform a **Revise** operation on the **Req11** object, it creates a new revision for all of the parent nodes until an open BVR is found in the structure.

The following figure shows the results after revising to the top of the **Prog1** requirement structure in Word from the **Req11** requirement. New revisions, in this case revision **B**, are created for the **Prog1** requirement specification, **Req1** requirement, and the **Req11** requirement. You can now make changes to the new revision **B** of the requirement objects.

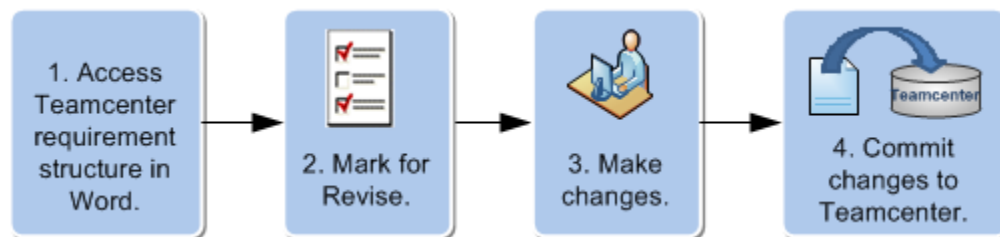


Note:

All objects below the object marked for **Revise** are copied by reference to the revised structure and remain at revision **A**.

Process for revising released requirement objects

The process for revising released requirement objects includes four steps:



1. Open the requirements structure in live Word.

- Display the **Requirements** pane in Word using one of the following methods:

- Open an offline Word document containing a requirement structure previously exported from Teamcenter Systems Engineering.
- Open and load a Teamcenter requirements structure from the Teamcenter ribbon in Word.

2. Mark for Revise.

- In the requirements structure in Word, select the requirement specification, requirement, or paragraph object for which you want to create a new revision.

Note:

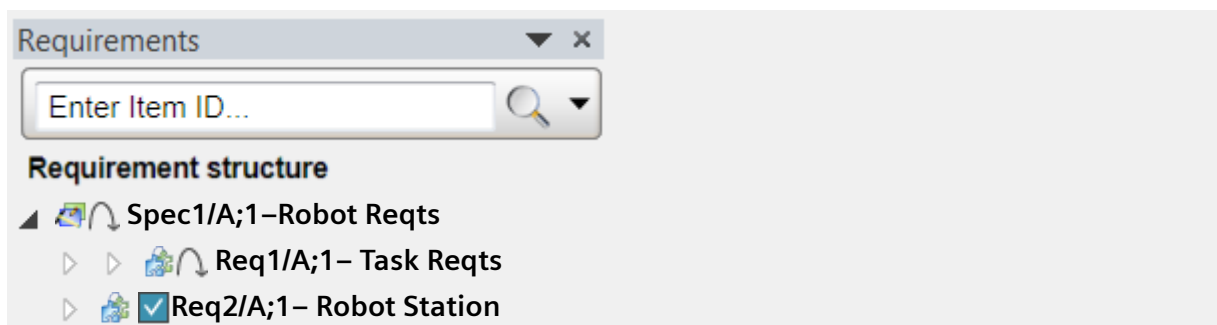
The item revision of the object must be locked by being at a *released* state and *not* modifiable before you can mark it for *revise*. The **Revise** command is *not* available for item revision objects that are *open*.

An item revision object is considered *open* if it is:

- *Not* at a released state.
- At a released state, but still modifiable.
- *Not* checked out by another user.
- *Not* locked by the access control list (ACL).

- Right-click the selected requirement object and choose **Mark/Unmark for Revise and Replace**.

As the following figure shows, when you mark an object for *revise* in Word, the revise symbol  displays next to the object in the **Requirements** pane.



3. Make changes.

You can now edit object properties for the new revision of the requirement object in Word.

4. (Optional) If you want to cancel your edits, you must unmark the object for revise *before* saving the Word document.

Note:

- You cannot unmark for *revise* if the object contains a child node that is marked for *save as*, *revise*, or *delete*.
- The unmark does not traverse the structure; only the selected object is unmarked.

- a. To unmark an object marked for revision, right-click the object in the **Requirements** pane in Word and choose **Mark/UnMark for Revise and Replace**.
- b. In the **Unmark Selected Node** dialog box, click **Yes**.

The revise symbol  is removed from the selected object. Any edits are canceled and a new revision object is not created.

5. **Commit the changes to Teamcenter.**

When you save the document in Word, the changes are saved to Teamcenter and the following occurs:

- A new revision of the object is created with a unique ID in accordance with the deep copy rules.

For more information about deep copy rules, see the Business Modeler IDE documentation.

- All changes made to the new revision, including any property and body text changes, are saved in Teamcenter.
- If the selected object is in a *precise* requirement structure, the following occurs:
 - In Teamcenter, the existing item *revision* of the selected object in the structure is *replaced* by the new item *revision*.
 - In Word, the new item revision for the object appears in the requirements structure.
- If the selected object is in an *imprecise* requirement structure, the following occurs:
 - In Teamcenter, the new item revision is added to the item revisions for the requirement object, and appears in the requirements structure.
 - In Word, the new item revision is added to the item revisions for the requirement object, and appears in the requirements structure.

Note:

Because the structure is imprecise, it points to the latest item revision for the requirement object.

- The mark for *revise* symbol no longer appears for the object in the **Requirements** pane.

For more information about committing changes to Teamcenter, see [Save interactive live Word changes in Teamcenter](#).

Consideration for creating a new requirement revision using Client for Office

In Client for Office, only one *working* revision can exist for an object. If another Teamcenter user creates a new revision of the same object concurrently with you, either through another live Word session or directly in Teamcenter, you cannot save your new revision of the object. When you attempt to save the new revision, you receive an error message that your save action is stopped and cannot be completed because other users have edited this object and saved it while you were editing. In the **Modified objects warning** dialog box, the details of the object that cannot be saved are displayed. Click **Close** to close this dialog box and reopen the object for editing.

Revising a single object

You can perform a **Revise** operation on a single requirement object under *either* of the following conditions:

- The selected requirement object is in a *released* state and the requirement structure is *imprecise*.
- The selected requirement object is in a *released* state, the BOMview revision (BVR) of the parent object is *open*, and the requirement structure is *precise*.

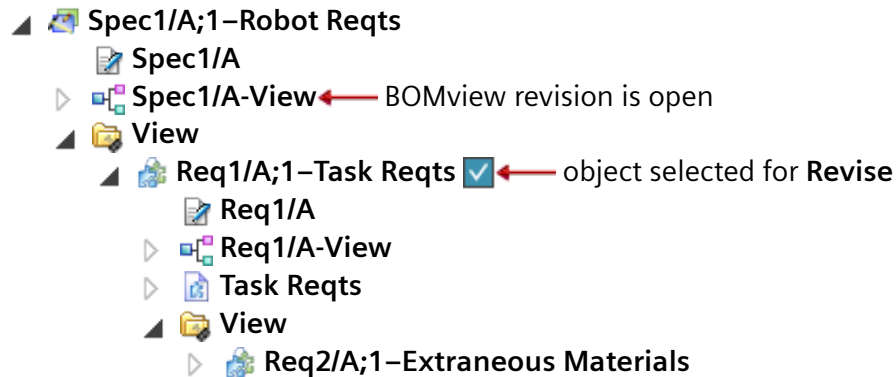
Note:

The parent object is considered open under the following conditions:

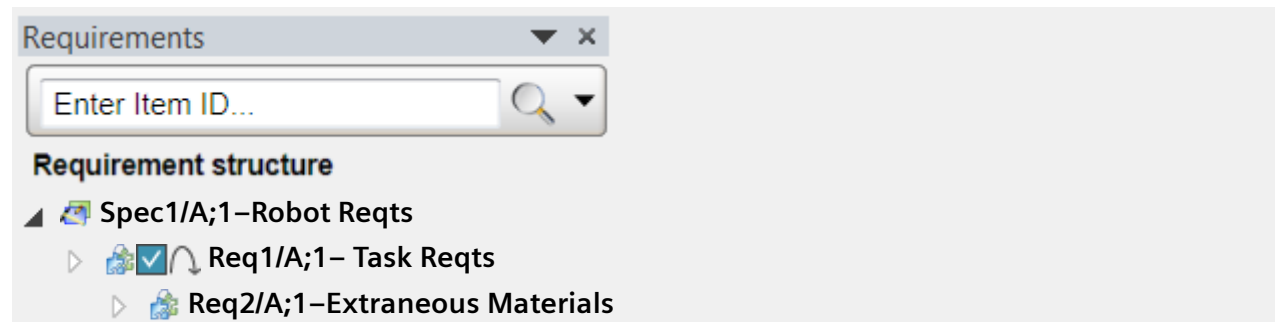
- Not in a *released* state.
- In a *released* state but still modifiable.
- Not checked out by another user.
- Not locked by the access control list (ACL).

Example:

The following expanded requirement structure in Teamcenter shows the **Req1/A;1–Task Reqts** requirement in Teamcenter is in a released state (locked) and the BOMview revision of the **Spec1/A;1–Robot Reqts** parent requirement specification is open (available for edit).



When you export this requirement structure to Word and mark for **Revise** on the **Req1/A;1–Task Reqts** requirement, only the selected **Req1/A;1–Task Reqts** requirement is marked for **Revise**. After performing the **Revise** operation in Word, the revise symbol  displays next to the **Req1/A;1–Task Reqts** requirement in the **Requirements** pane in Word.



Revising parent nodes to the top of the structure

When you perform a **Revise** operation on a selected released object, if *all* of the following conditions are met, all parent nodes to the top of the structure are also marked for **Revise**:

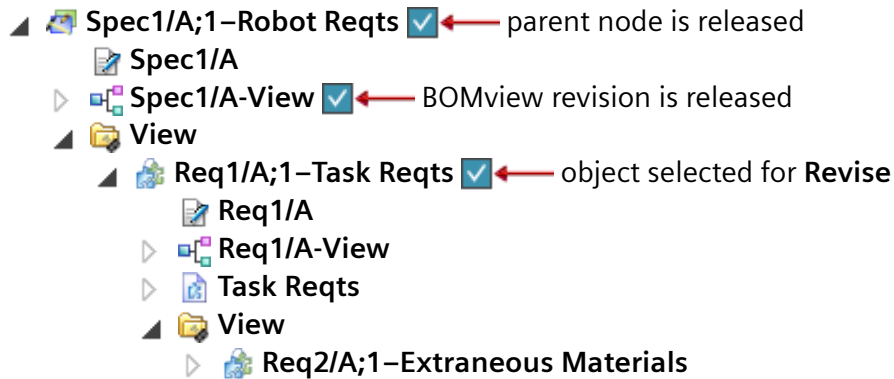
- The requirement structure is *precise*.
- The BOMview revision (BVR) of the parent object is *not* open.
- The parent node is released.

Note:

When these conditions are met, the **Revise** operation is applied to the selected node and all parent nodes to the top of the structure, until an open BOMview revision (BVR) object is found.

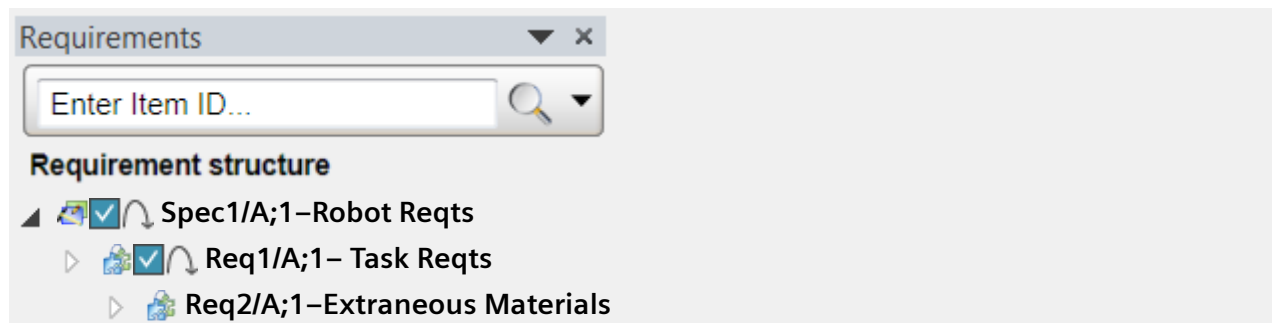
Example:

The following expanded requirement structure in Teamcenter shows the **Req1/A;1–Task Reqts** requirement, the BOMview revision of the **Spec1/A;1–Robot Reqts** parent requirement specification, and the **Spec1/A;1–Robot Reqts** parent requirement specification are all at a *released* state.

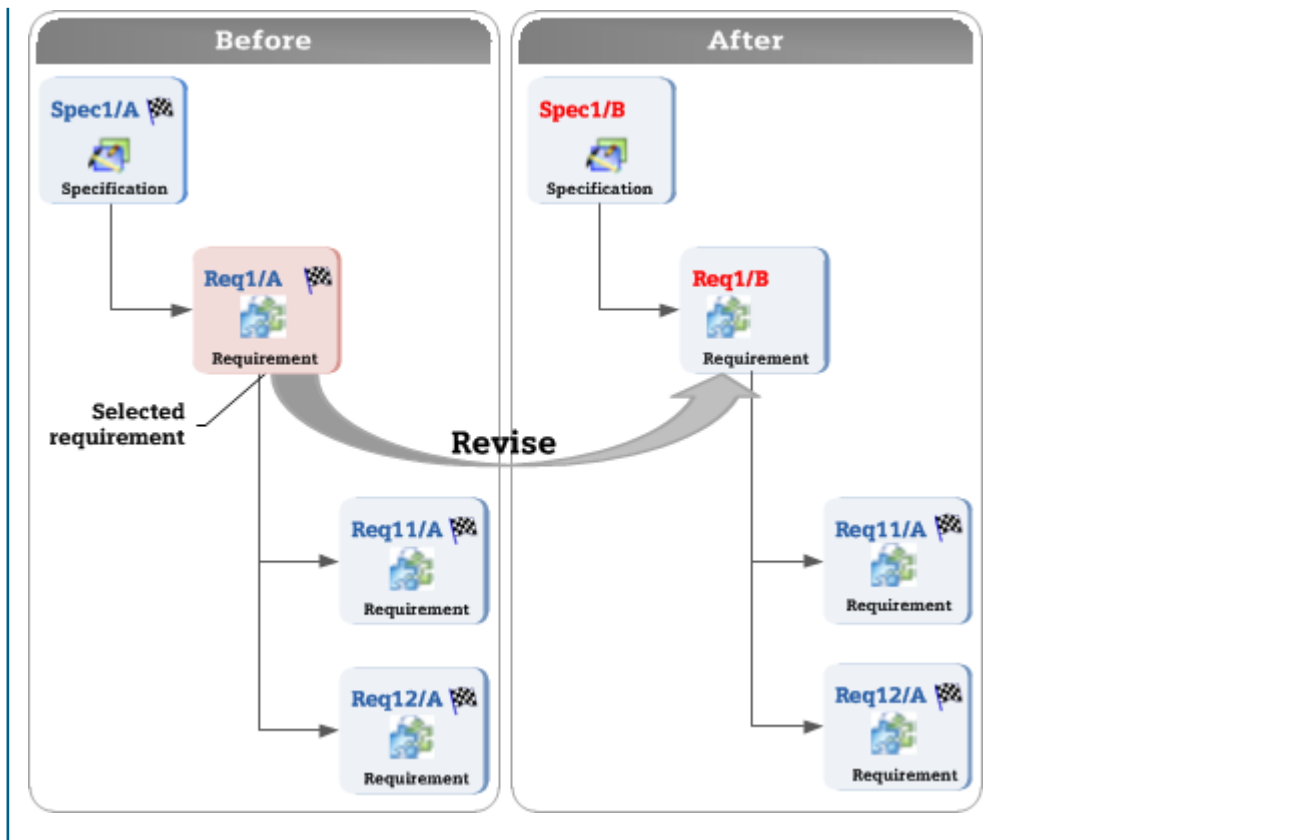


When you export the requirement structure to Word and mark the **Req1/A;1–Task Reqts** requirement object for **Revise**, the selected **Req1/A;1–Task Reqts** requirement object *and* the **Spec1/A;1–Robot Reqts** parent requirement specification object are marked for **Revise**.

The revise symbol  appears in the structure next to both the **Req1/A;1–Task Reqts** requirement object and the **Spec1/A;1–Robot Reqts** parent requirement specification object in the **Requirements** pane.



The following figure shows the structure after you save the changes to Teamcenter. New revisions **Spec1/B** and **Req1/B** are created.



Create a new revision of a released requirement

Note:

The item revision of an object must be locked by being at a *released* state and *not* modifiable before you can mark it for *revise*. The **Revise** command is *not* available for item revision objects that are *open*.

An item revision object is considered *open* if it is:

- Not at a released state.
- At a released state, but still modifiable.
- Not checked out by another user.
- Not locked by the access control list (ACL).

1. In the **Requirements** pane in Word, right-click the object you want to mark for revision in the requirement structure and choose **Mark/UnMark for Revise and Replace**.

The revise symbol  appears next to the selected object and any other objects impacted by the revise operation. The symbol indicates the object is marked for revision.

2. Edit the requirement revision object, as needed.
3. In Word, click the **Save** button  or choose **File→Save** to save the changes in Teamcenter.
4. In the **Save** dialog box, if you checked out the objects before the export, select **Save and Check-in** to check in the objects.
5. Click **OK**.

When you save the document in Word, the changes are saved to Teamcenter and the following occurs:

- A new revision of the object is created with a unique ID in accordance with the deep copy rules.
- All changes made to the new revision, including any property and body text changes, are saved in Teamcenter.
- If the selected object is in a *precise* requirement structure, the following occurs:
 - In Teamcenter, the existing item *revision* of the selected object in the structure is *replaced* by the new item *revision*.
 - In Word, the new item revision for the object appears in the requirements structure.
- If the selected object is in an *imprecise* requirement structure, the following occurs:
 - In Teamcenter, the new item revision is added to the item revisions for the requirement object, and appears in the requirements structure.
 - In Word, the new item revision is added to the item revisions for the requirement object, and appears in the requirements structure.

Note:

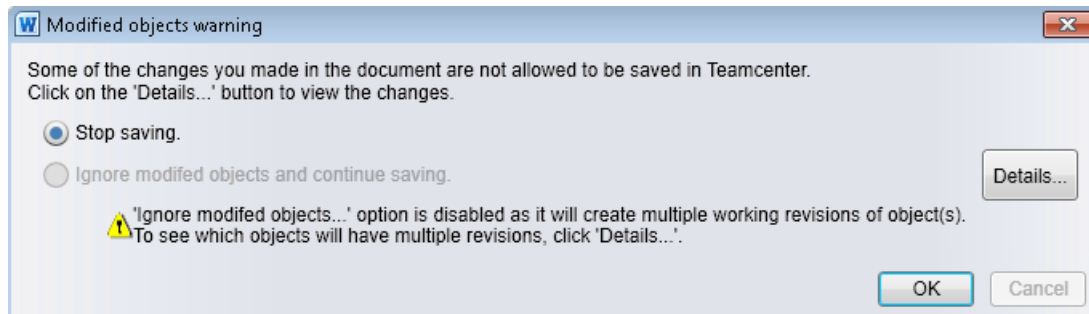
Because the structure is imprecise, it points to the latest item revision for the requirement object.

- The mark for *revise* symbol no longer appears for the object in the **Requirements** pane.

If another Teamcenter user creates a new revision of the same object as you, either through another live Word session or directly in Teamcenter, you cannot save your new revision of the object. This is because, if you perform the **Revise** operation from Client for Office, only one *working* revision can exist for an object.

When you attempt to save the new revision, you receive an error message that conflicting revisions of the object have been found and the save cannot be completed. In the **Modified objects**

warning dialog box, click the **Details** button to view a list of the objects with multiple working revisions in the requirement structure that you are trying to save.



Cancel the create a requirement revision operation

1. To unmark an object marked for revision, right-click the object in the **Requirements** pane in Word and choose **Mark/UnMark for Revise and Replace**.

Note:

You cannot unmark the object if it contains a child node that is marked for save as, revise, or delete.

2. In the **Unmark Selected Node** dialog box, click **Yes**.

The revise symbol ↺ is removed from the selected object. Any edits are canceled and a new revision object is not created.

Copying released objects in a requirements structure

Overview of copying released objects

In Word, when you open a requirement specification structure, the object hierarchy for the structure appears in the **Requirements** pane. The **Requirements** pane enables you to navigate and change the requirement structure.

In the **Requirements** pane, you can create a copy of a released requirement object by marking it for *save as*. After performing the **Save As** operation on the requirement object, you can edit the requirement information.

Requirement objects include requirement specifications, requirements, or paragraphs. The requirement objects must be released and *not* modifiable before you can mark them for **Save As**.

The **Save As** operation allows you to:

- Create new copies of existing released requirements in Word.

- Make changes to the new copy of the released requirement in Word.
- Commit those changes to Teamcenter by saving the Word document.

The requirements structure marked for *save as* can be an *imprecise* or *precise* structure. *Imprecise* structures are dynamic. An imprecise structure has links (occurrences) to item objects in the structure. *Precise* structures are fixed structures. A precise assembly has links (occurrences) to item revision objects in the structure.

Depending on the conditions met, when you perform a **Save As** operation on a selected released object in the structure, one of the following occurs:

- Only the single selected object is marked for *save as*.
- All of the parent nodes to the top of the structure, including the selected object, are marked for *save as*, until an open BOMview revision (BVR) object is found.

Note:

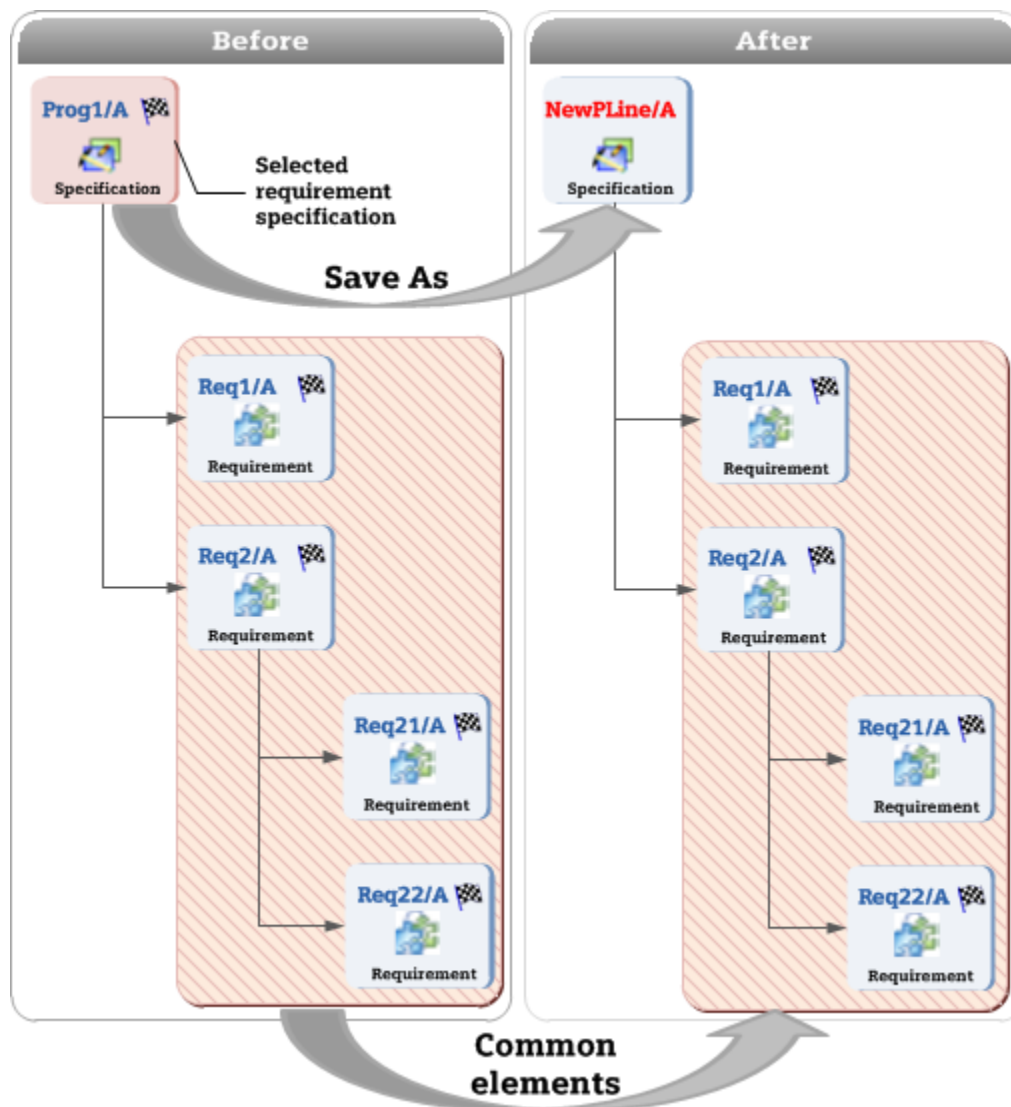
All objects below the objects marked for *save as* are copied by reference to the new structure.

Example of copying released objects

A requirement structure named **Prog1** contains the requirements for a specific product program. These requirements are used for all product lines in the program.

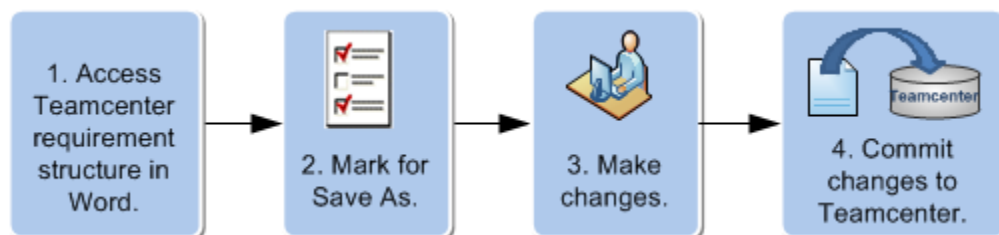
A new product line called **NewPLine** is created in the **Prog1** program that also uses this set of common requirements. Therefore, you perform a **Save As** operation on the **Prog1** requirement specification to create a new requirement specification with the same set of common requirements.

The following figure shows the results after performing a **Save As** operation in Word on the **Prog1** requirement specification to create the new **NewPLine** requirement specification. You can now edit the requirements in the **NewPLine** requirement specification structure, as needed, for the new product line.



Process for copying released requirement objects

The process for copying released requirement objects includes four steps:



1. Open the requirements structure in live Word.

- Display the **Requirements** pane in Word using one of the following methods:

- Open an offline Word document containing a requirement structure previously exported from Teamcenter Systems Engineering.
- Open and load a Teamcenter requirements structure from the Teamcenter ribbon in Word.

2. Mark for Save As.

- In the requirement structure in Word, select the requirement specification, requirement, or paragraph object for which you want to create a copy.

Note:

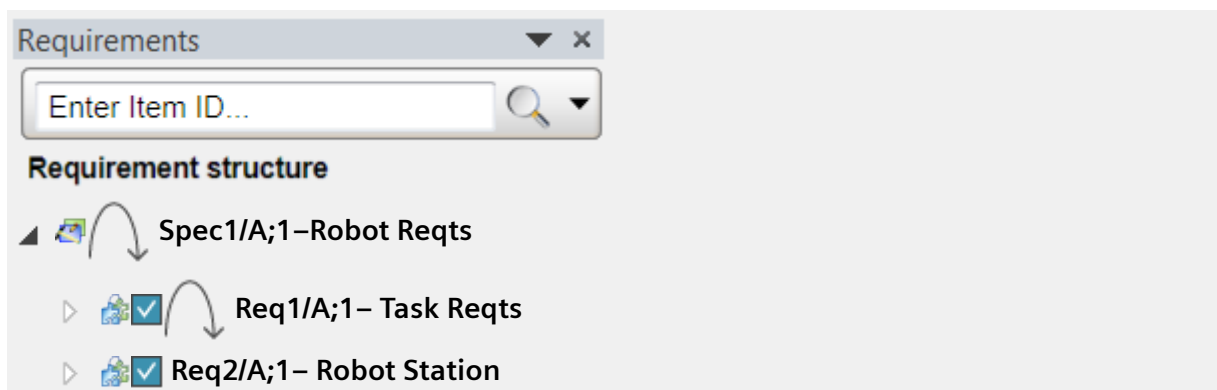
The item revision of the object must be locked by being at a *released* state and *not* modifiable before you can mark it for *save as*. The **Save As** command is *not* available for objects that are *open*.

An object is considered *open* if it is:

- *Not* at a released state.
- At a released state, but still modifiable.
- *Not* checked out by another user.
- *Not* locked by the access control list (ACL).

- Right-click the selected requirement object and choose **Mark/Unmark for Save As and Replace**.

As the following figure shows, when you mark an object for *save as* in Word, the save as symbol  displays next to the object in the **Requirements** pane.



3. Make changes.

You can now edit object properties for the new copy of the requirement object in Word.

4. (Optional) If you want to cancel your edits, you must unmark the object for save as *before* saving the Word document.

Note:

- You cannot unmark for *save as* if the object contains a child node that is marked for *revise*, *save as*, or *delete*.
- The unmark does not traverse the structure; only the selected object is unmarked.

- a. To unmark an object marked for save as, right-click the object in the **Requirements** pane in Word and choose **Mark/UnMark for Save As and Replace**.
- b. In the **Unmark Selected Node** dialog box, click **Yes**.

The save as  symbol is removed from the selected object. Any edits are canceled and no copy is created.

5. **Commit the changes to Teamcenter.**

When you save the document in Word, the changes are saved to Teamcenter and the following occurs:

- A new object is created with a unique ID in accordance with the deep copy rules configured for **Save As**.
- All changes made to the new object including any property and body text changes are saved in Teamcenter.
- All the objects under the existing object are copied by reference to the copied object.
- If the selected object is in a *precise* requirement structure, the following occurs:
 - In Teamcenter, the selected object item *revision* is *replaced* by the new item *revision*.
 - In Word, the new item revision of the object appears in the requirement structure.
- If the selected object is in an *imprecise* requirement structure, the following occurs:
 - In Teamcenter, the selected object item is *replaced* by the new object item.
 - In Word, the new object item appears in the requirement structure.

- The mark for *save as* symbol no longer appears for the object in the **Requirements** pane.

Copying a single object

You can perform a **Save As** operation to copy a single object, under the following conditions:

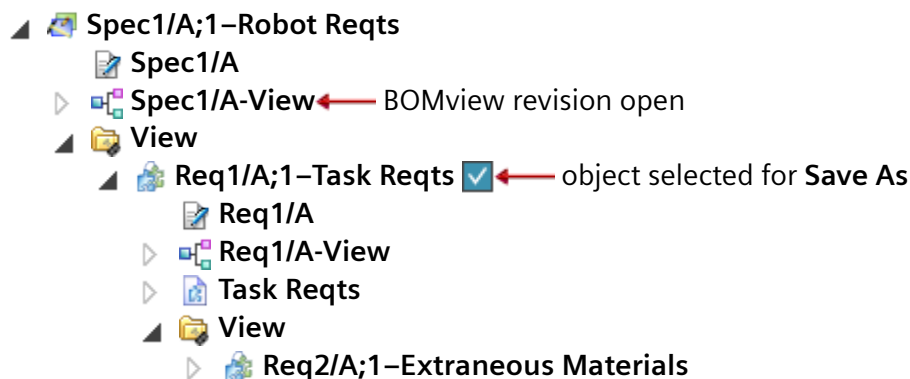
- The object you select to copy is in a *released* state.
- The BOMview revision (BVR) of the parent object is open (available for edit).

Note:

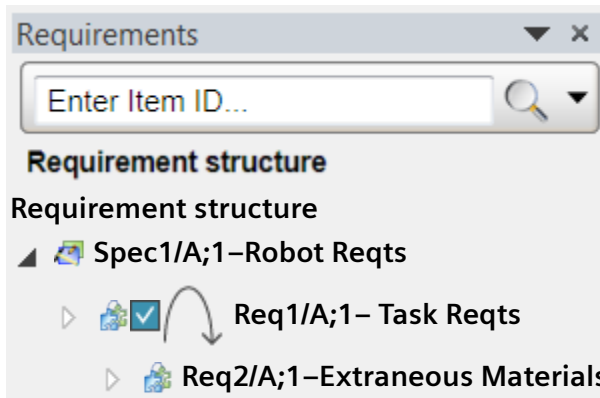
The parent object is considered open under the following conditions:

- Not in a *released* state.
- In a *released* state but still modifiable.
- Not checked out by another user.
- Not locked by the access control list (ACL).

In the following example, the expanded requirement structure in Teamcenter shows the **Req1/A;1–Task Reqts** requirement in Teamcenter is in a released state (locked) and the BOMview revision of the **Spec1/A;1–Robot Reqts** parent requirement specification is open (available for edit).



When you export this requirement structure to Word and mark for **Save As** on the **Req1/A;1–Task Reqts** requirement, only the selected **Req1/A;1–Task Reqts** requirement is marked for **Save As**. After performing the **Save As** operation in Word, the save as symbol  displays next to the **Req1/A;1–Task Reqts** requirement in the **Requirements** pane in Word.

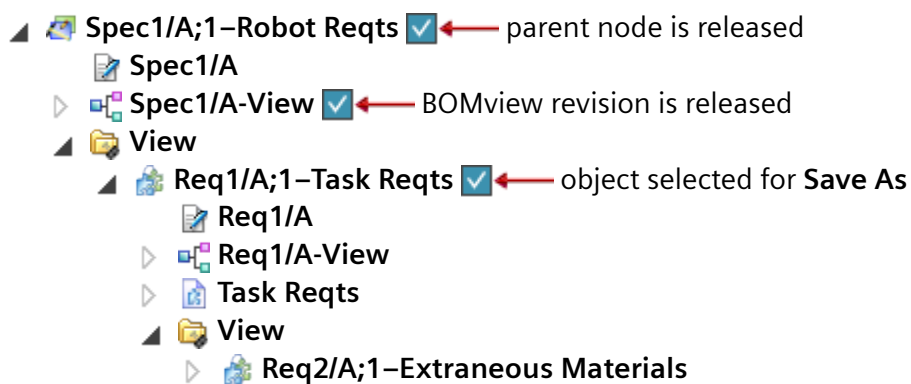


Copying parent nodes to the top of the structure

When the following conditions are met, the **Save As** operation is applied to the selected node and all parent nodes to the top of the structure, until an open BOMview revision (BVR) object is found:

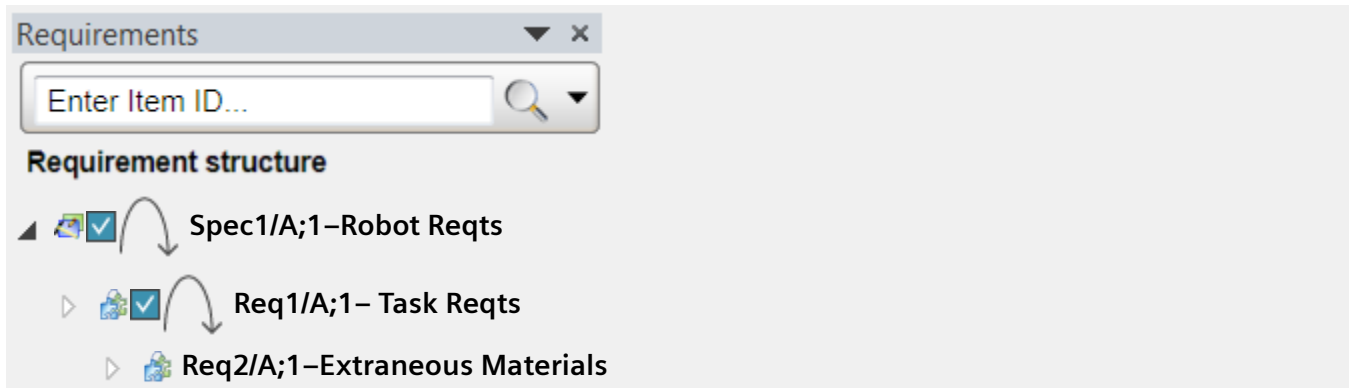
- The selected object is *released*.
- The parent object is *released*.
- The BOMview revision (BVR) of the parent object is *not* open.

In the following example, the expanded requirement structure in Teamcenter shows the **Req1/A;1-Task Reqts** requirement, the BOMview revision of the **Spec1/A;1-Robot Reqts** parent requirement specification, and the **Spec1/A;1-Robot Reqts** parent requirement specification are all at a released state.

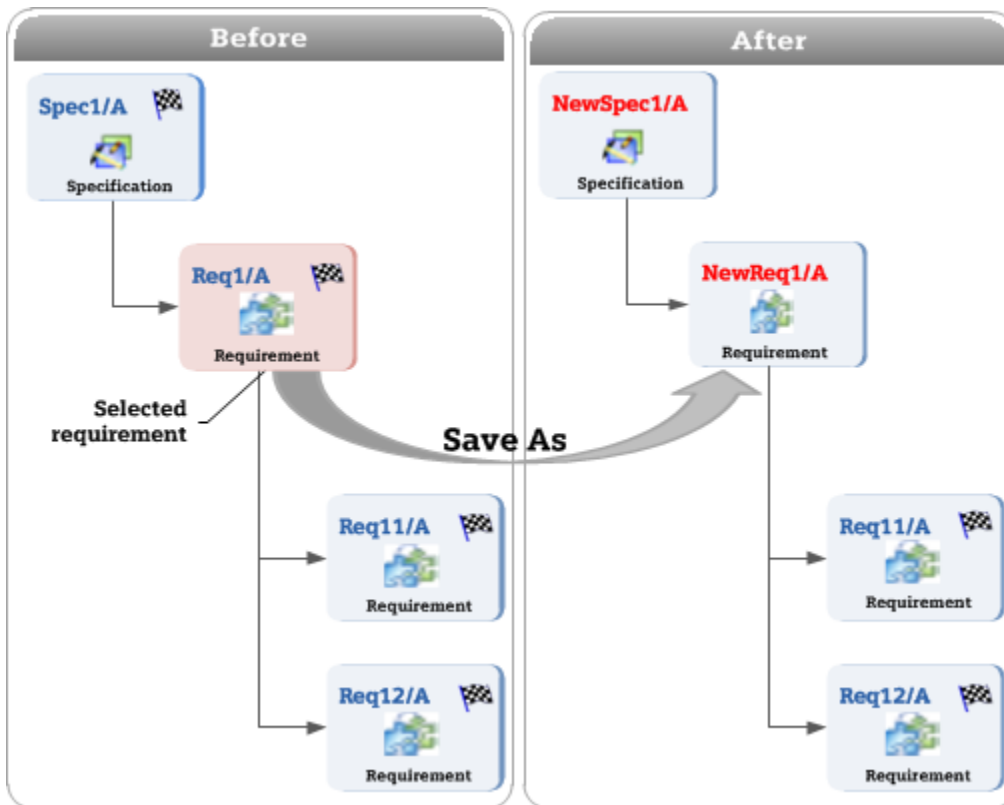


When you export the requirement structure to Word and mark for **Save As** on the **Req1/A;1-Task Reqts** requirement, the selected **Req1/A;1-Task Reqts** requirement *and* the **Spec1/A;1-Robot Reqts** parent requirement specification are marked for **Save As**.

The save as symbol  appears in the structure next to both the **Req1/A;1-Task Reqts** requirement object and the **Spec1/A;1-Robot Reqts** parent requirement specification object in the **Requirements** pane.



The following figure shows the structure after you save the changes to Teamcenter. The new requirement object, **NewReq1**, and the new parent object, requirement specification **NewSpec1**, are created.



How object copying affects traceability

When you perform a **Copy** operation on a requirement object or structure containing trace links to another requirement structure, the following behavior occurs:

- If the object containing the defining trace link and the object containing the complying trace link are both copied, a trace link is created between the two copied objects in the new structure.

The trace link is preserved and remains an *internal* trace link because it is linked to objects within the structure.

- If the object containing the defining trace link is copied, but the object containing the complying trace link is *not* copied, the trace links are created between the original complying object in the old structure, and the copied defining object in both the new and old structures.

The trace link is now *external* because it is linked to an object outside the structure.

- If the object containing the complying trace link is copied, but the object containing the defining trace link is *not* copied, the trace links are created between the original defining object in the old structure, and the copied complying object in both the new and old structures.

The trace link is now *external* because it is linked to an object outside the structure.

If you paste a reference to the object contained in the clipboard by pressing Ctrl+V, a trace link is created to the complying object in the new structure.

If you paste a duplicate of the object contained in the clipboard as a child of the current selection by pressing Ctrl+Shift+V or right-click and select **Paste as New**, a trace link is not created to the duplicate complying object in the new structure.

To create a trace link on selecting **Paste as New**, you must set the deep copy rule as follows:

Edit Deep Copy Rule
Edits a Deep Copy Rule

Project: TL3

Target Business Object: Requirement Revision [Browse...](#)

☐ Target Primary?

Operation Type: SaveAs

Property Type: ☒ Relation ☐ Reference

Relation : * FND_TraceLink [Browse...](#)

Attached Business Object: * ItemRevision [Browse...](#)

Condition: * isTrue [Browse...](#)

Action: CopyAsObject

☐ Required?

☒ Secured?

☒ Copy Properties on Relation?

[?](#) [Finish](#) [Cancel](#)

```
<TcDeepCopyRule typeName="Requirement Revision"
  relRefPropName="FND_TraceLink" propertyType="Relation"
  objectTypeName="ItemRevision" conditionName="isTrue"
  operationName="ITEM_create_from_rev" copyType="CopyAsObject"
  isTargetPrimary="false" secured="true"
  copyRelationAttributes="true" isRequired="false"
  description=""/>
```

This deep copy rule creates a trace link on the new copy of the selected complying object of the **Requirement Revision** type.

Copy a released requirement object to create a new requirement

Note:

The item revision of the selected object must be locked by being at a *released* state and *not* modifiable before you can mark it for *save as*. The **Save As** command is *not* available for objects that are *open*.

An object is considered *open* if it is:

- *Not* at a released state.
- At a released state, but still modifiable.
- *Not* checked out by another user.
- *Not* locked by the access control list (ACL).

1. In the **Requirements** pane in Word, right-click the requirement specification, requirement, or paragraph object for which you want to create a copy and choose **Mark/UnMark for Save As and Replace**.

The save as symbol  appears next to the selected object and any other objects in the structure that are impacted by the save as operation. The save as symbol indicates the object is marked for copy.

2. Edit the requirement information as needed.
3. In Word, click the **Save** button  or choose **File→Save** to save the changes in Teamcenter.
4. In the **Save** dialog box, if you checked out the objects before the export, select **Save and Check-in** to check in the objects.
5. Click **OK**.

The changes are saved and the copy is created in Teamcenter.

When you save the document in Word, the changes are saved to Teamcenter and the following occurs:

- A new object is created with a unique ID in accordance with the deep copy rules configured for **Save As**.

For more information about deep copy rules, see the Business Modeler IDE documentation.

- All changes made to the new object including any property and body text changes are saved in Teamcenter.

- All the objects under the existing object are copied by reference to the copied object.
- If the selected object is in a *precise* requirement structure, the following occurs:
 - In Teamcenter, the selected object item *revision* is *replaced* by the new item *revision*.
 - In Word, the new item revision of the object appears in the requirement structure.
- If the selected object is in an *imprecise* requirement structure, the following occurs:
 - In Teamcenter, the selected object item is *replaced* by the new object item.
 - In Word, the new object item appears in the requirement structure.
- The mark for save as symbol no longer appears for the object in the **Requirements** pane.

Cancel the copy a requirement object operation

If you want to cancel the **Save As** operation for an object, you must unmark the object for save as *before* saving any edits to the Word document.

Note:

- You cannot unmark an object if it contains a child node that is marked for save as, revise, or delete.
- The unmark does not traverse the structure; only the selected object is unmarked.

1. To unmark an object marked for save as, right-click the object in the **Requirements** pane in Word and choose **Mark/UnMark for Save As and Replace**.
2. In the **Unmark Selected Node** dialog box, click **Yes**.

The save as symbol is removed from the selected object. Any edits are canceled and no copy is created.

Create trace links between objects

Note:

Trace link features are available only if:

- The Word document is connected to Teamcenter.
- The **Tracelink Mode** check box is selected in the Teamcenter **Options** dialog box.

Choose **Edit→Options** in Teamcenter to display the **Options** dialog box.

1. In the **Requirements** pane, do one of the following:

- For a single defining object, right-click the object and choose **Start Trace Link**.

The trace link for a single defining object can target one or more complying objects.

- For multiple defining objects, hold down Ctrl or Shift while selecting the objects, and then right-click the selection and choose **Start Trace Link**.

The trace link for multiple defining objects can target only one complying object.

2. In the **Requirements** pane, right-click the complying object or multiple object selection, and then do one of the following:

- To accept the default trace link subtype and name, choose **End Trace Link**.

- To assign a custom trace link subtype and name:

- a. Choose **End Trace Link Subtype** to display the **Create Trace Link Subtype** dialog box.

All trace link subtypes are displayed.

- b. In the list on the left, select the subtype to assign to each trace link.

- c. In the **Name** box, type the name to assign to each trace link.

You can type optional additional information in the **Description** box.

A trace link symbol appears for each defining and complying object.

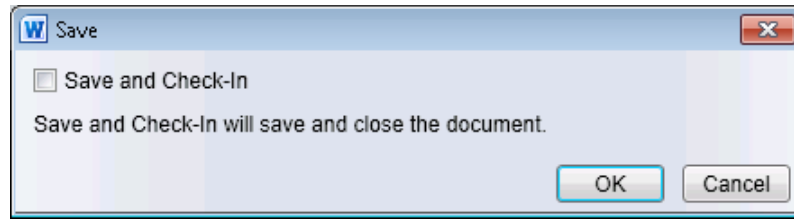
Save interactive live Word changes in Teamcenter

Note:

In the **Export to Word** dialog box, the **Check out objects before export** option explicitly checks out the exported objects before the document is generated. This option is selected and unavailable by default. Unless you changed the **REQ_checkout_objects_before_export** preference, the objects in the offline document are checked out automatically.

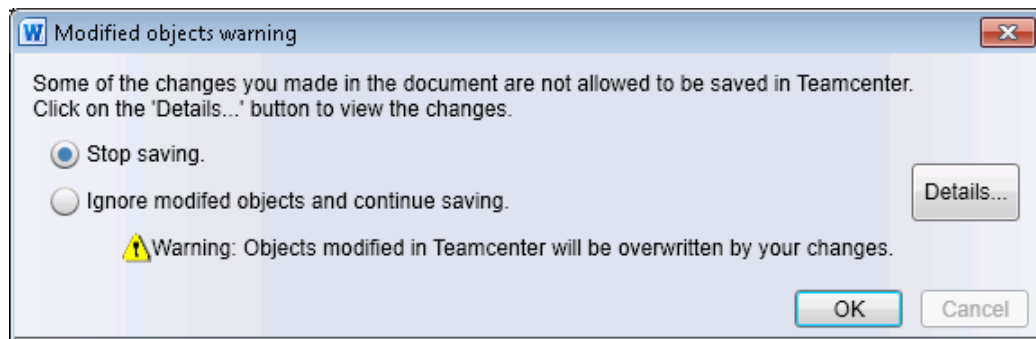
1. In the live Word document, click the Word **Save** button, or use another of the related Word features.

Live Word displays the **Save** dialog box.

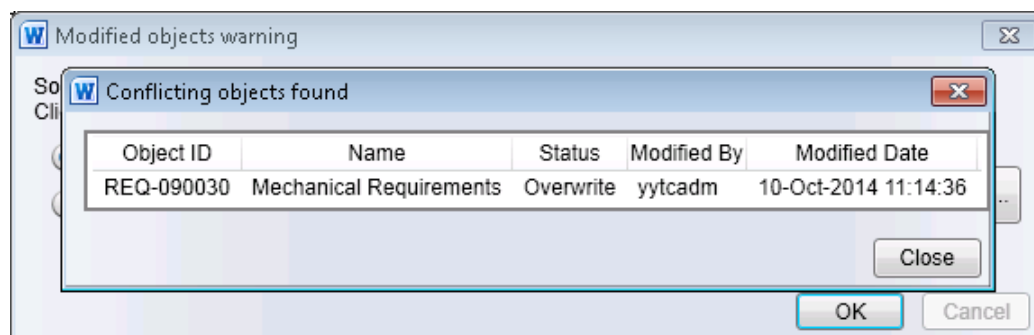


2. Select the **Save and Check-In** check box, and then click **OK**.

- Live Word displays a progress indicator if no modification conflicts exist between the document and the structure.
- The **Modified objects warning** dialog box opens if objects changed in the document were changed also in Teamcenter after the original export.



You can click **Details** to view the objects in the **Conflicting objects found** dialog box.



- a. Select one of the following options in the **Modified objects warning** dialog box:
- Stop saving** Terminates the process without applying pending changes.
 - Ignore modified objects and continue saving** Continues the process and overwrites all Teamcenter changes.

Note:

If a processing error prevents any change, Teamcenter rolls back all changes for the structure.

- b. Click **OK** to begin the process.

If modification conflicts are resolved, live Word displays a progress indicator.

The changes are applied to the corresponding objects in Teamcenter. The document closes when the process is complete.

Synchronize offline live Word changes with Teamcenter

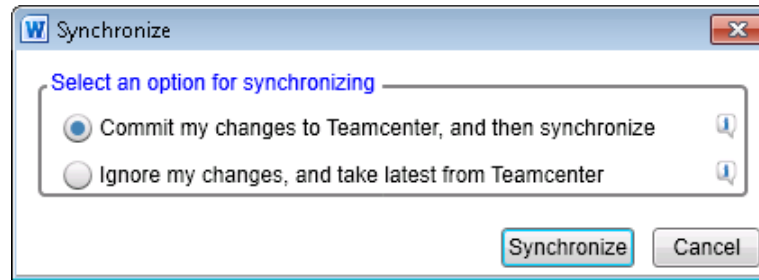
An offline document is stored outside of Teamcenter, for example, on your desktop or on a shared drive. It is temporarily disconnected from Teamcenter.

When you reconnect the document, you can commit its pending changes to overwrite the data in the corresponding structure. Or, you can nullify the pending changes and overwrite the document with the current Teamcenter data. If necessary, you can resolve conflicts in modifications between the document and the structure.

Note:

- In the **Export to Word** dialog box, the **Check out objects before export** option explicitly checks out the selected objects before the output document is generated. This option is selected and unavailable by default. Unless you changed the **REQ_checkout_objects_before_export** preference, the objects in the offline document are checked out automatically.
- The document is not available for editing until synchronization is complete.
- If live Word displays the **Teamcenter Login** dialog box during this procedure, enter your Teamcenter user name and password, and then click **Login**.

1. From your local file system, open the offline live Word document.
 - If no other live Word document is connected to Teamcenter, a message asks if you want to connect. Click **Yes** to open the connection and the **Synchronize** dialog box.
 - The **Synchronize** dialog box opens automatically if another live Word document is connected.



Note:

All open live Word documents share the same connection.

2. Do one of the following:

- To commit the pending changes in the document, select **Commit my changes to Teamcenter, and then synchronize**.

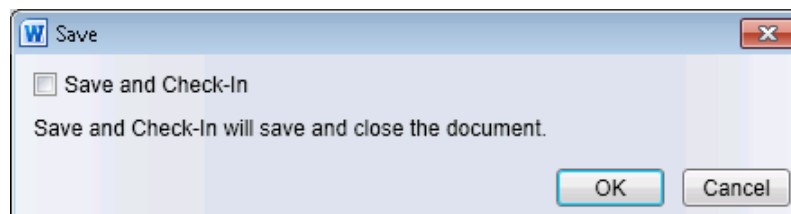
This option applies the changes to the corresponding objects in the structure. Also, it updates the document with changes to other objects since the document was last synchronized.

- To void the pending changes in the document, select **Ignore my changes, and take latest from Teamcenter**.

This option overwrites the document content with the current data in the corresponding structure.

3. Click **Synchronize**.

Live Word displays the **Save** dialog box.

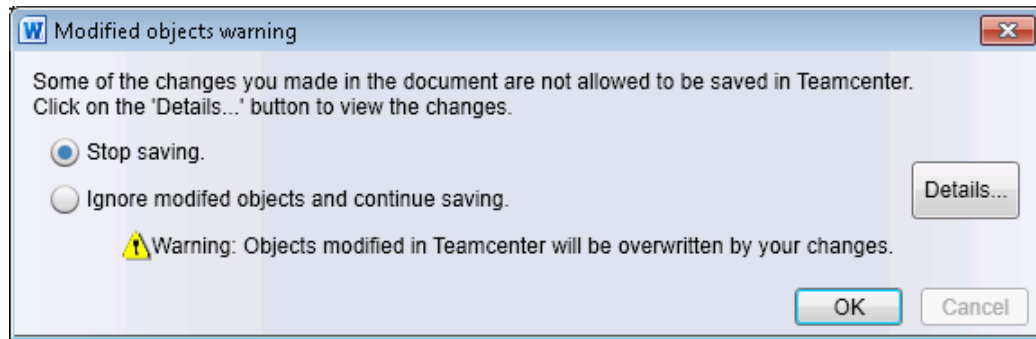


4. Select the **Save and Check-In** check box, and then click **OK**.

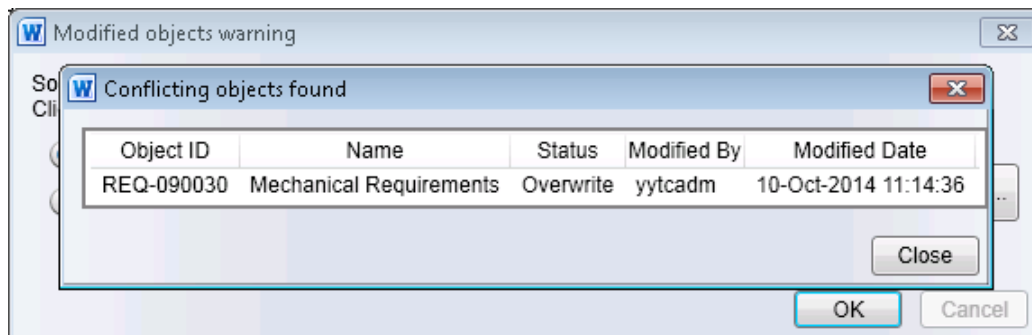
Note:

If you click **OK** with the check box cleared, the document closes and the requirements remain checked out.

- Live Word displays a progress indicator if no modification conflicts exist between the document and the structure.
- The **Modified objects warning** dialog box opens if objects changed in the document were changed also in Teamcenter after the original export.



You can click **Details** to view the objects in the **Conflicting objects found** dialog box.



- Select one of the following options in the **Modified objects warning** dialog box:
 - Stop saving** Terminates the synchronization without applying pending changes.
 - Ignore modified objects and continue saving** Continues the synchronization and overwrites all Teamcenter changes.
- Click **OK** to begin the synchronization.

The progress indicator displays informational messages at each stage of the process. The document closes when the synchronization is complete.

Accessing requirement content through URLs in live Word documents

The live Word export file for a requirement structure may include a link to the requirement content for the exported requirement revision object.

To include a link to the requirement content in the live Word export file, the export template you use when exporting the requirement revision object must include the run-time property, **fnd0_body_text_url**, on the **SpecElementRevision** business object.

- Click the body text link in the live Word export file to access and edit or view the requirement content.

Any updates you make to the requirement content, and save to the linked body text file, are also updated in Teamcenter and appear in the **Details** view.

Disconnect an offline live Word document from Teamcenter

Warning:

This procedure is permanent and cannot be reversed.

1. With the document open, choose **Specifications→Make static** on the **Teamcenter** ribbon.

A message states that you cannot again connect to Teamcenter from the document and asks if you want to continue.

2. Click **Yes**.

The live Word content becomes static and the document is permanently disconnected from Teamcenter.

Note:

The current content and object properties remain in the static document.

4. Designing functional and logical architectures

Identifying the component subsystems of a product

System *decompositions*, or *models*, separate an overall architecture into component subsystems with progressively greater degrees of detail. With *functional* and *logical* models, you can illustrate and analyze solutions that meet the requirements of the product.

Each model is captured as a structured network of subsystems.

- A functional model illustrates a set of functions that the product should deliver.

Because requirements and functions are closely related, a functional architecture is a means of exploring, expanding, and deriving requirements.

- At a given level, the functional design derives requirements at the next lower level of detail.
- The derived requirements reveal a corresponding subset of functions for that level.

The functional model ties requirements to logical solutions.

- A logical model illustrates a proposed solution for implementing the requirements and functions.
 - During this analysis, logical models for several alternative solutions can be compared.
 - Typically, the accepted solution is implemented in both hardware and software, and that implementation is modeled later in the detailed design phase.

Extending functional requirements to subsystems

As functional requirements use text paragraphs to describe a product's purposes, functional models illustrate those purposes with a hierarchy of functional blocks. Decomposition of functional models into finer detail produces parent and child models within a functional architecture. These layers of models describe the functionality of the entire system, from the major subsystems to the lowest-level subsystems.

In Teamcenter, a function object represents a purpose of the system. For each object in a functional model, the following elements can be represented:

- Interfaces (inside and outside the system)
- Inputs and outputs

- Relationships with other functions in the system

Typically, systems engineers use trace links for allocating functions to elements in downstream or upstream structures, such as:

- Other functions in the same structure or in different **Function** views.
- Requirement specifications, logical models, and physical models in Systems Engineering.

Trace links set up paths in which elements precede and succeed one another across the system's various domains. By tracing these defining and complying relationships, systems engineers can analyze change impact both upstream and downstream.

Tip:

Trace links can exist also between Systems Engineering objects and subsets in 4th Generation Design. Collaborative designers can analyze these relationships through the Systems Engineering *cross-domain tracing* feature.

In Systems Engineering, you can build functional structures by:

- Creating the structure in the **Function** view.
- Creating diagrams in Microsoft Office Visio Professional.
- Importing spreadsheets from Microsoft Excel.

Note:

A function object has a **BodyText** property, whose value is rich text content that you define. To convey information about the function, for example, its purpose in the overall architecture, you can:

- Edit the content in the **MS Word** view.
- Export the content to Microsoft Office Word.

Abstracting logical implementations

Logical models illustrate abstract physical architectures for implementing system functions. The logical model is a hierarchical structure of solution components, which are represented in Teamcenter by *logical blocks*.

Typically, systems engineers use trace links for allocating logical blocks to elements in downstream or upstream structures, such as:

- Other logical blocks in the same structure or in different **Logical Block** views.
- Requirement specifications, functional models, and physical models in Systems Engineering.

Trace links set up paths in which elements precede and succeed one another across the system's various domains. By tracing these defining and complying relationships, systems engineers can analyze change impact both upstream and downstream.

Tip:

Trace links can exist also between Systems Engineering objects and subsets in 4th Generation Design. Collaborative designers can analyze these relationships through the Systems Engineering *cross-domain tracing* feature.

To assess a solution's progress toward achieving performance goals, you can apply quantifiable parameters (such as technical performance measures, or TPMs) to logical models. In Teamcenter, TPMs are managed using *budgets*.

In Systems Engineering, you can create logical model structures by:

- Creating the structure in the **Logical Block** view.
- Creating diagrams in Microsoft Office Visio Professional.
- Importing spreadsheets from Microsoft Excel.

Note:

A logical block object has a **BodyText** property, whose value is rich text content that you define. To convey information about the logical block, for example, its purpose in the overall architecture, you can:

- Edit the content in the **MS Word** view.
- Export the content to Microsoft Office Word.

Building architectures in the Function or Logical Block view

Illustrating subsystems with object hierarchies

In the Systems Engineering **Function** view and **Logical Block** view, you can build hierarchical object structures that represent functional and logical architectures. These structures model the relationships and interactions among systems and their component subsystems.

For example, to build a functional decomposition in the **Function** view:

1. Start with a single function to represent the entire system.

The new function occupies the top level in the **Function** view.

2. At the next lower level, add functions that define the overall system functionality at a lower level.

These objects become the children of the top level object.

3. Below those functions, continue to add functions that decompose the system into finer levels of detail.

These objects become the children of the previous level.

With custom subtypes of functions and logical blocks, you can build architectures for other purposes, such as work breakdown structures and organization charts. For example, you can create subtypes to represent elements of a process, and use the subtypes to construct a process flow.

Organizing functional and logical hierarchies

Functions and logical blocks can be created at specific levels, promoted to higher levels, demoted to lower levels, and moved up or down within a level.

- For each object, the **Number** property value shows the level in the hierarchy and the position within the level.
- As the structure changes, those values are renumbered automatically for affected objects.

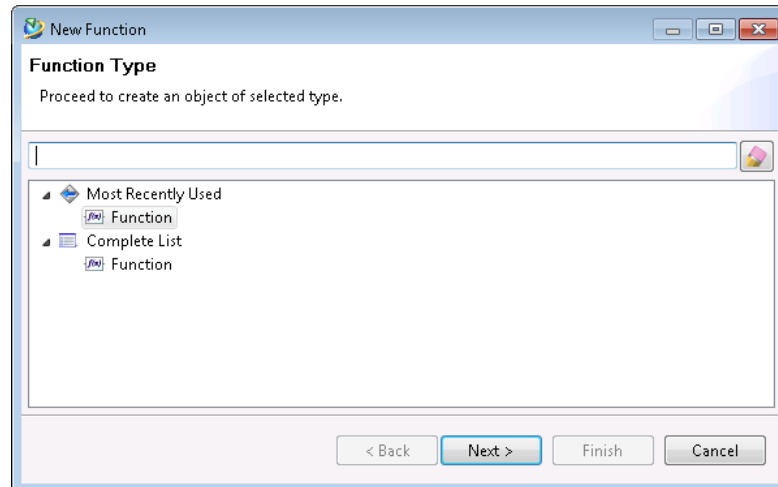
Note:

If the top-level object type is not a child of the **Specification** object type, you must modify the top-level type to enable moving objects between levels in structures.

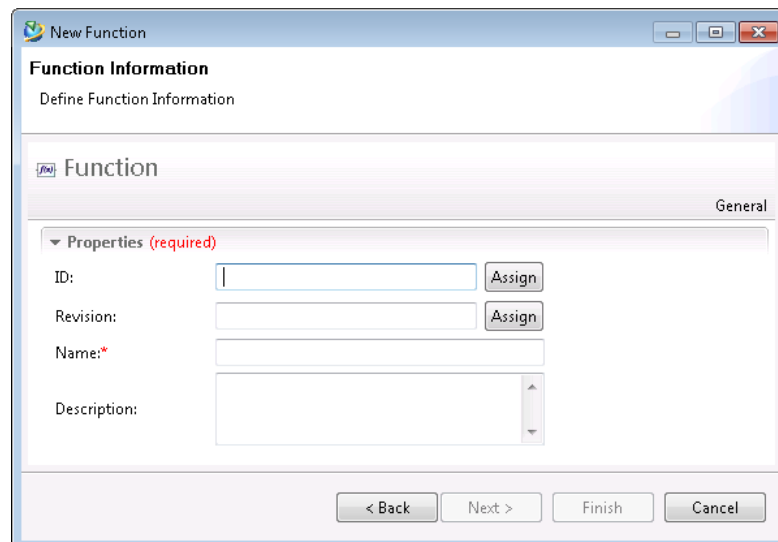
Create a function

1. In the **Function** view, choose **File→New→Blocks→Function**, or click the **Function** button  on the toolbar.

The **New Function** wizard opens.



2. Select **Function** under **Most Recently Used** or **Complete List**, and then click **Next** to advance to the **Function Information** page.



3. In the **Name** box, enter the text to appear in the **Rev Name** property column for the new function.

You can also enter the following optional data:

ID

Enter the new function's object identifier.

You can specify the identifier explicitly, or you can click **Assign** to enter the next available identifier in the automatic sequence. Or, leave the **ID** box blank.

Revision

Enter the new function's revision number.

You can specify the number explicitly, or you can click **Assign** to enter the next available number in the automatic sequence. Or, leave the **Revision** box blank.

Description

Enter plain text to convey any additional information about the function.

For example, you can describe its purpose in the architecture, or list its relationships to other structure elements. This text becomes the initial value of the function's **Rev Description** property.

After creating the function, you can edit the **BodyText** property using the rich-text features in Microsoft Office Word. You can also export the **BodyText** content to Word.

4. To create the function, click **Finish**.

Tip:

Click **Next** instead of **Finish** to open subsequent optional pages where you can:

- Define information for submitting the function to a workflow.
- Assign the function to a project.

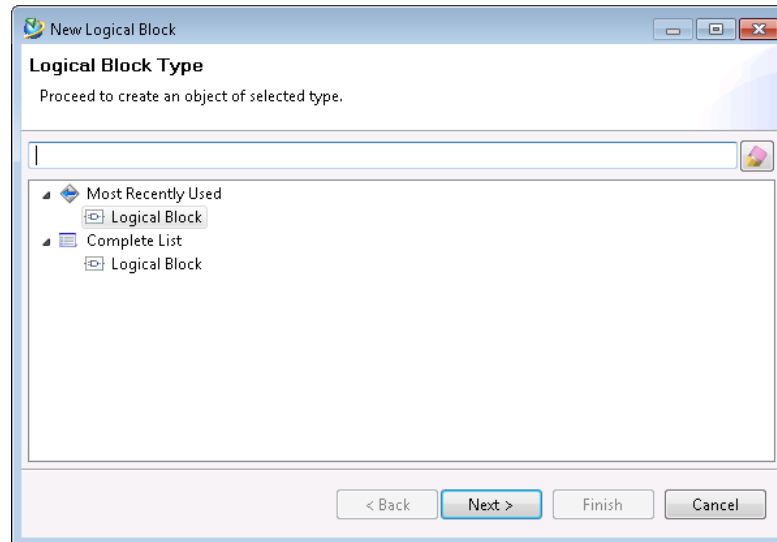
The new function is added to the **Function** view and to the **Newstuff** folder in your **Home** view.

The dialog box remains open, and you can create another function.

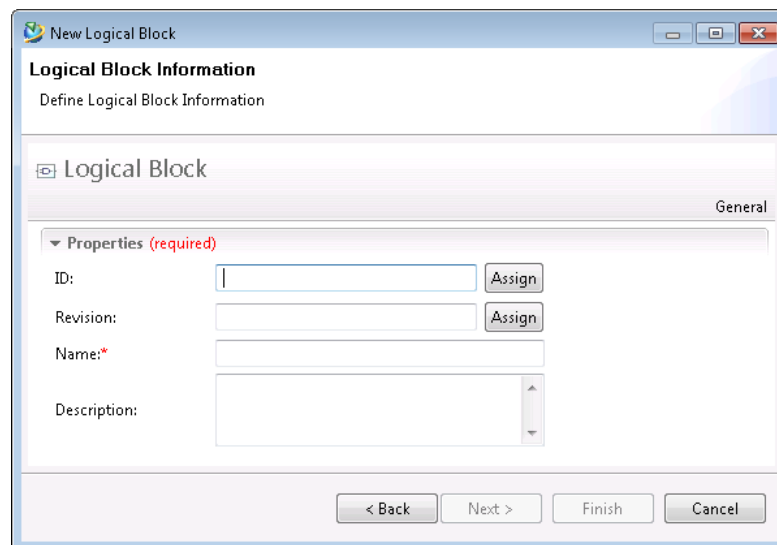
Create a logical block

1. In the **Logical Block** view, choose **File→New→Blocks→Logical Block**, or click the **Logical Block** button  on the toolbar.

The **New Logical Block** wizard opens.



2. Select **Logical Block** under **Most Recently Used** or **Complete List**, and then click **Next** to advance to the **Logical Block Information** page.



3. In the **Name** box, enter the text to appear in the **Rev Name** property column for the new logical block.

You can also enter the following optional data:

ID

Enter the object identifier for the logical block.

You can specify the identifier explicitly, or you can click **Assign** to enter the next available identifier in the automatic sequence. Or, leave the **ID** box blank.

Revision

Enter the revision number for the logical block.

You can specify the number explicitly, or you can click **Assign** to enter the next available number in the automatic sequence. Or, leave the **Revision** box blank.

Description

Enter plain text to convey any additional information about the logical block.

For example, you can describe its purpose in the architecture, or list its relationships to other structure elements. This text becomes the initial value of the logical block's **Rev Description** property.

After creating the logical block, you can edit the **BodyText** property using the rich-text features in Microsoft Office Word. You can also export the **BodyText** content to Word.

4. To create the logical block, click **Finish**.

Tip:

Click **Next** instead of **Finish** to open subsequent optional pages where you can:

- Define information for submitting the logical block to a workflow.
- Assign the logical block to a project.

The new logical block is added to the **Logical Block** view and to the **Newstuff** folder in your **Home** view.

The dialog box remains open, and you can create another logical block.

Create a trace link between functions or logical blocks

1. In the **Function** or **Logical Block** view, select the defining object and choose **Tools→Trace Link→Start Trace Link**.
2. Select the complying object, and then do one of the following:

Accept the default trace link subtype • Choose **Tools→Trace Link→End Trace Link**.

Assign a custom trace link subtype a. Choose **Tools→Trace Link→End Trace Link...** to display the **Create Trace Link** dialog box.

b. In the left pane, select the trace link subtype you want to use.

c. (Optional) Type a name for the trace link in the **Name** box.

- d. (Optional) Type a description of the trace link in the **Description** box.
- e. Do one of the following:
 - Click **Apply** to leave the dialog box open and create another trace link subtype.
 - Click **OK** to close the dialog box.

The trace link is created and the trace link symbol appears in the **Trace Link** property column for the defining and complying objects.

Note:

Depending on the relation rule for the diagram, the trace link may not appear in the diagram.

Create an interface in a structure

1. In the **Function** or **Logical Block** view, select the owning object and choose **File→New→Interface**, or click the **Interface** button  on the toolbar.

Systems Engineering displays the New Business Object wizard.

2. In the **Interface Object Type** page box, select **Network_port** and then click **Next**.
3. On the **Object Creation Information** page, type the object name in the **Name** box.

You can also do the following optional actions:

Description box	Type a description, such as the object's purpose or content.
Direction box	Select one of the following: • Input/Input specifies that the connection can carry only a consumed payload. • Output/Output specifies that the connection can carry only a generated payload. • Bi-Directional/Bi-Directional specifies that the connection can carry both a generated payload and a consumed payload.

This value is the default.

4. Click **Finish**.

The **Function** or **Logical Block** view displays the new interface (port) below the owner.

The dialog box remains open, and you can create another object of the same type.

5. Click **Cancel** to close the dialog box.

Create a connection between two interfaces

Note:

You can create multiple connections between the same two interfaces. Each connection automatically receives a unique name, which you can change after this procedure.

1. In the **Function** or **Logical Block** view, select the starting interface and choose **File→New→Connection**, or click the **Connection** button on the toolbar.

Systems Engineering displays the **New Business Object** wizard.

2. In the **Connection Object Type** list, select the connection type, and then click **Next**.
3. On the **Object Creation Information** page, type the connection name in the **Name** box.

In the **Description** box, you can type an optional description, such as the object's purpose or content.

4. Click **Finish**.

The **Function** or **Logical Block** view displays the new connection below the owner.

Note:

The connection is added to your **Newstuff** folder in My Teamcenter.

The dialog box remains open, and you can create another object of the same type.

Change interface direction in a structure

1. In the **Function** or **Logical Block** view, select the interface and choose **Set Direction**.

The **Set Port Direction** dialog box is displayed.

2. In the **Port Direction** list, select one of the following:

Input	Specifies that the interface can only consume a payload.
Output	Specifies that the interface can only generate a payload.
Bi-Directional	Specifies that the interface can both generate and consume a payload. This value is the default.

Remove elements from a functional or logical structure

- Objects removed from Systems Engineering or Requirements Manager remain in Teamcenter, for example, in My Teamcenter, if you previously saved them there. You can use the **Search** feature to recall the objects.
- When you delete items and structures from the database, they are also deleted from other Teamcenter applications. They cannot be recalled by using the **Search** feature.

In some structures, you cannot delete multiple objects; you must delete them individually.

1. In the **Function** or **Logical Block** view, select the object or objects to remove.

You can select multiple objects with the standard Windows functions.

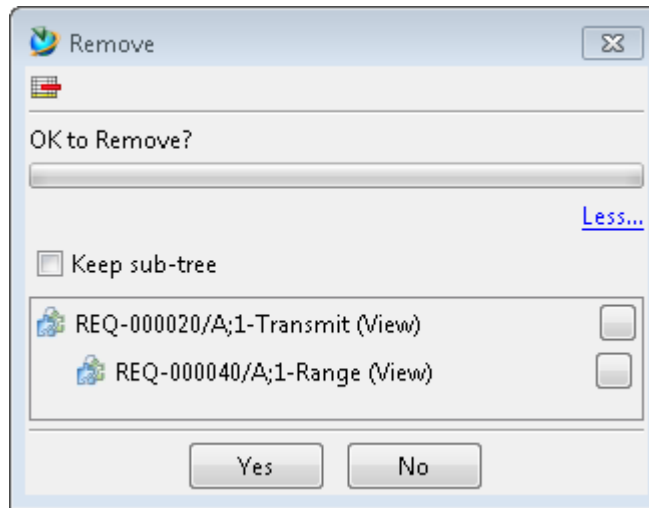
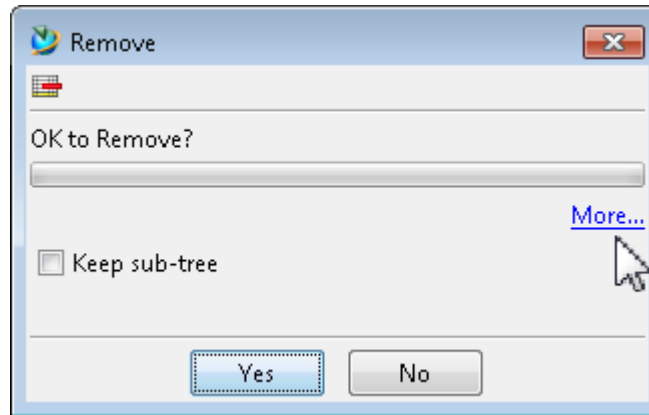
Note:

- To remove an interface that is not an occurrence, you must select the function or logical block to which the interface is attached.
- Interfaces that are occurrences can be selected and removed individually.

2. Choose **Edit→Remove**, or click the **Remove**  button.

The **Remove** dialog box opens.

Click **More...** to open a pane showing the substructures owned by the selected objects.



3. Do one of the following:

- To remove the selected objects and their substructures, click **Yes**.
- To remove only the selected objects but retain their substructures:
 - a. Select the **Keep sub-tree** check box.
 - b. Click **Yes**.

The objects are removed from the structure. Any retained substructures are promoted to the levels previously occupied by the removed objects.

Diagramming architectures in Microsoft Visio

Illustrating subsystems with diagrams

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Siemens Digital Industries Software assumes that you are familiar with Visio and with XML.

In the **Visio Diagram** window, you can create and edit diagrams that represent system architectures in the **Function** and **Logical Block** views. By expressing these structures graphically, you can show how a product's functional and logical subsystems relate to one another.

Note:

You can open multiple diagrams simultaneously, though you can work in only one at a time.

Each diagram is based on a *diagram template*, which you select when you create the diagram. The diagram template governs the diagram content through two associated files.

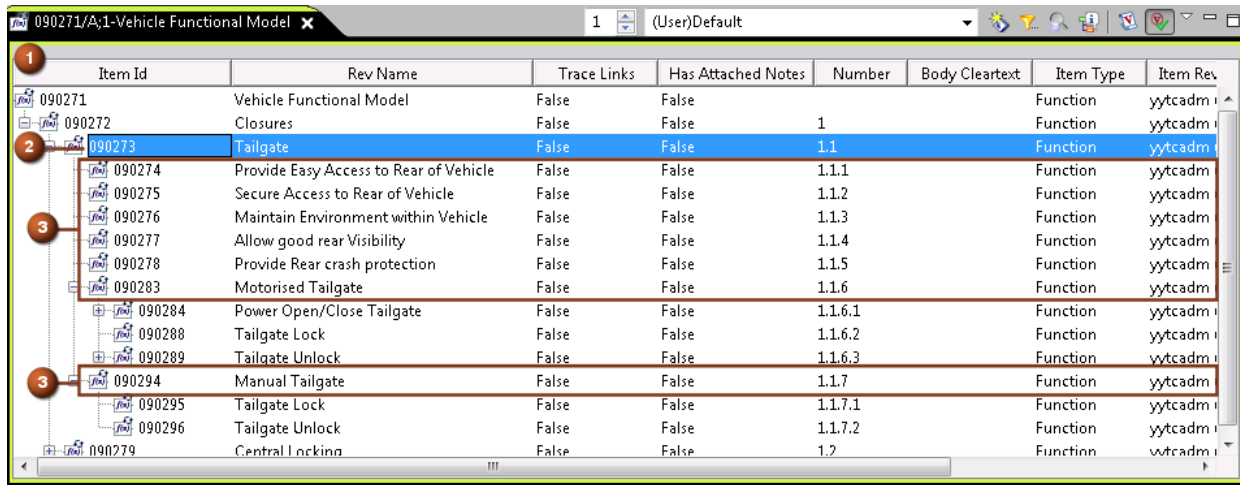
- A Teamcenter stencil (**.vsx** or **.vssx**) defines the master shapes for the diagram.
- A mapping file (**.xml**) relates the master shapes to Teamcenter object types and properties.

A subsystem diagram starts with a *diagram owner*, a selected function or logical block with descendants representing lower-level components.

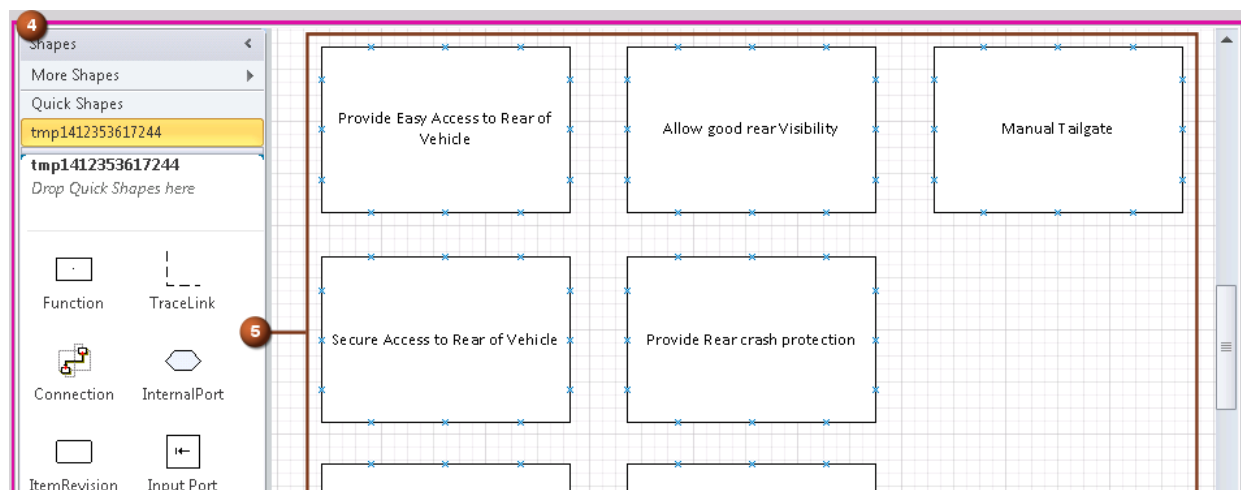
Note:

The diagram owner does not appear in the diagram.

- When you choose **File→New→Diagram**, the diagram shapes represent the diagram owner's direct children, the elements at the next lower level.



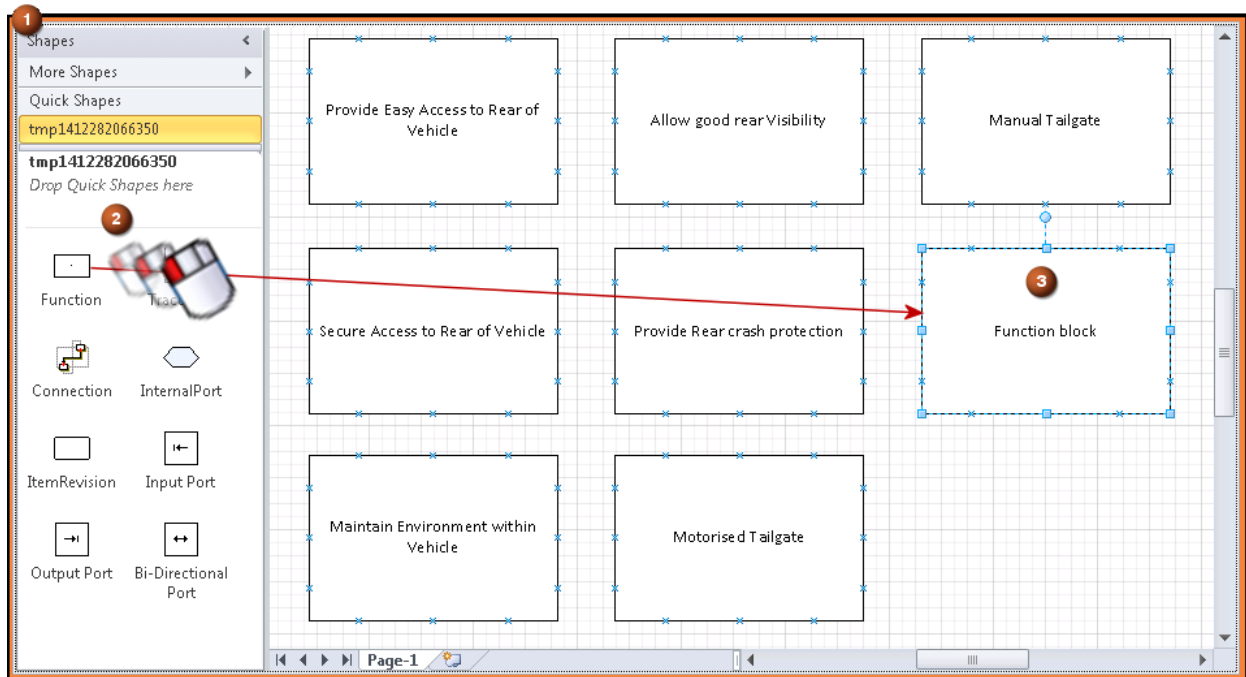
Item Id	Rev Name	Trace Links	Has Attached Notes	Number	Body Cleartext	Item Type	Item Rev
090271	Vehicle Functional Model	False	False			Function	yvtcadm
090272	Closures	False	False	1		Function	yvtcadm
090273	Tailgate	False	False	1.1		Function	yvtcadm
090274	Provide Easy Access to Rear of Vehicle	False	False	1.1.1		Function	yvtcadm
090275	Secure Access to Rear of Vehicle	False	False	1.1.2		Function	yvtcadm
090276	Maintain Environment within Vehicle	False	False	1.1.3		Function	yvtcadm
090277	Allow good rear Visibility	False	False	1.1.4		Function	yvtcadm
090278	Provide Rear crash protection	False	False	1.1.5		Function	yvtcadm
090283	Motorised Tailgate	False	False	1.1.6		Function	yvtcadm
090284	Power Open/Close Tailgate	False	False	1.1.6.1		Function	yvtcadm
090288	Tailgate Lock	False	False	1.1.6.2		Function	yvtcadm
090289	Tailgate Unlock	False	False	1.1.6.3		Function	yvtcadm
090294	Manual Tailgate	False	False	1.1.7		Function	yvtcadm
090295	Tailgate Lock	False	False	1.1.7.1		Function	yvtcadm
090296	Tailgate Unlock	False	False	1.1.7.2		Function	yvtcadm
090279	Central Locking	False	False	1.2		Function	yvtcadm



- 1 **Function** view
- 2 **Diagram** owner
- 3 **Direct** children
- 4 **Visio Diagram** view
- 5 **Function** shapes

The diagram template also synchronizes the diagram content with the diagram owner. Interaction is bidirectional between the owner's structure and the diagram. Changes that you make in one are reflected in the other.

- When you add a shape to the diagram, a corresponding element is created in the structure.



4

5

6

Item Id	Rev Name	Trace Links	Has Attached Notes	Number	Body Cleartext	Item Type	Item Revision
090271	Vehicle Functional Model	False	False			Function	yycadm (yyc)
090272	Closures	False	False	1		Function	yycadm (yyc)
090273	Tailgate	False	False	1.1		Function	yycadm (yyc)
090274	Provide Easy Access to Rear of Vehicle	False	False	1.1.1		Function	yycadm (yyc)
090275	Secure Access to Rear of Vehicle	False	False	1.1.2		Function	yycadm (yyc)
090276	Maintain Environment within Vehicle	False	False	1.1.3		Function	yycadm (yyc)
090277	Allow good rear visibility	False	False	1.1.4		Function	yycadm (yyc)
090278	Provide Rear crash protection	False	False	1.1.5		Function	yycadm (yyc)
090283	Motorised Tailgate	False	False	1.1.6		Function	yycadm (yyc)
090284	Power Open/Close Tailgate	False	False	1.1.6.1		Function	yycadm (yyc)
090288	Tailgate Lock	False	False	1.1.6.2		Function	yycadm (yyc)
090289	Tailgate Unlock	False	False	1.1.6.3		Function	yycadm (yyc)
090294	Manual Tailgate	False	False	1.1.7		Function	yycadm (yyc)
000099	Function block	False	False	1.1.8		Function	yycadm (yyc)
090279	Central Locking	False	False	1.2		Function	yycadm (yyc)

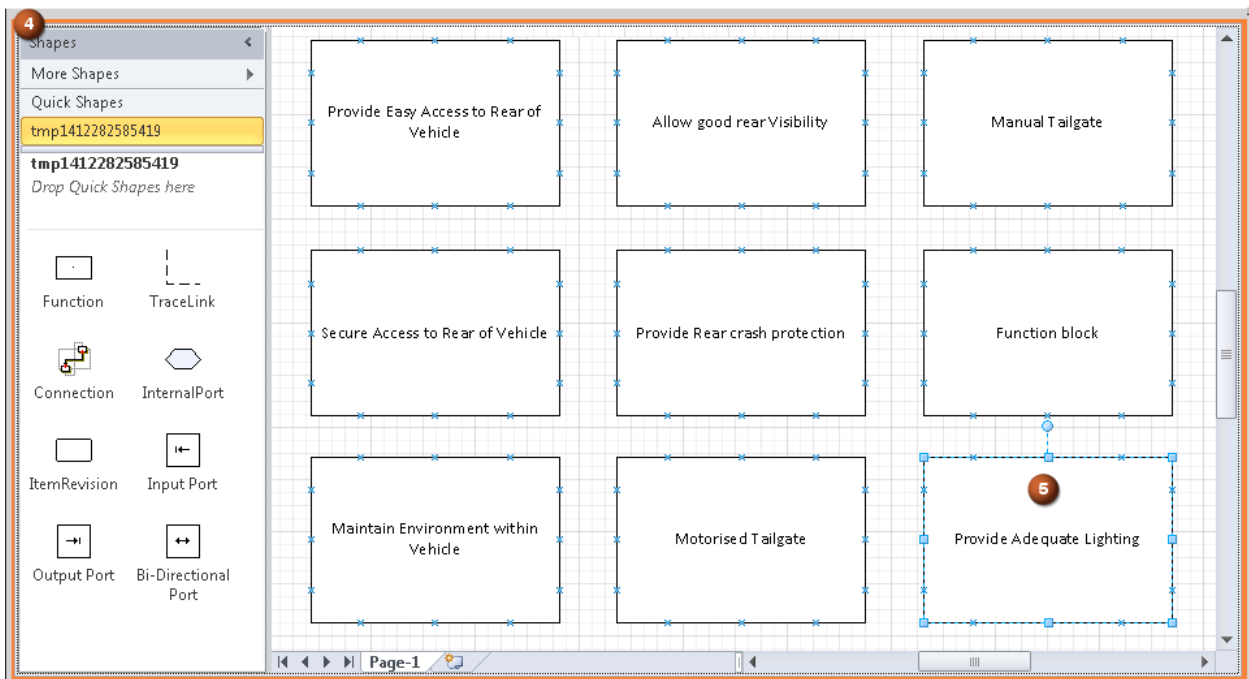
- 1 Visio Diagram view
- 2 Teamcenter stencil
- 3 New function shape
- 4 Function view
- 5 Diagram owner
- 6 Corresponding function object

Tip:

You can rename the new object by selecting the diagram shape, pressing F2 to open the text block, and typing the new name.

- When you create an element in the structure, a corresponding shape is added to the diagram.

Item Id	Rev Name	Trace Links	Has Attached Notes	Number	Body Cleartext	Item Type	Item Revision
090271	Vehicle Functional Model	False	False			Function	yvtcadm (yvt
090272	Closures	False	False	1		Function	yvtcadm (yvt
090273	Tailgate	False	False	1.1		Function	yvtcadm (yvt
090274	Provide Easy Access to Rear of Vehicle	False	False	1.1.1		Function	yvtcadm (yvt
090275	Secure Access to Rear of Vehicle	False	False	1.1.2		Function	yvtcadm (yvt
090276	Maintain Environment within Vehicle	False	False	1.1.3		Function	yvtcadm (yvt
090277	Allow good rear Visibility	False	False	1.1.4		Function	yvtcadm (yvt
090278	Provide Rear crash protection	False	False	1.1.5		Function	yvtcadm (yvt
090283	Motorised Tailgate	False	False	1.1.6		Function	yvtcadm (yvt
090294	Manual Tailgate	False	False	1.1.7		Function	yvtcadm (yvt
090295	Tailgate Lock	False	False	1.1.7.1		Function	yvtcadm (yvt
090296	Tailgate Unlock	False	False	1.1.7.2		Function	yvtcadm (yvt
000099	Function block	False	False	1.1.8		Function	yvtcadm (yvt
000100	Provide Adequate Lighting	False	False	1.1.9		Function	yvtcadm (yvt
090279	Central Locking	False	False	1.2		Function	yvtcadm (yvt



- Function view**
- Diagram owner**
- New function object**

- 4 **Visio Diagram** view
- 5 Corresponding function shape

Note:

- If the shape is not displayed, right-click the drawing pane and choose **Refresh Diagram**.
- If the diagram is closed when you create the element, the shape is added when you reopen the diagram.

To save time when viewing an object's diagrams, you can preview snapshots in the Relation Browser. Previewing these static images eliminates the need to search for and open the diagram objects themselves.

How diagram templates guide diagram content

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Siemens Digital Industries Software assumes that you are familiar with Visio and with XML.

Diagram templates supply the instructions that Systems Engineering uses to generate diagrams of functional and logical architectures. Every diagram is based on a diagram template, which determines the diagram's shapes and their behaviors.

A diagram template relates a Teamcenter *stencil* (.vsx or .vssx) to a *mapping* file (.xml). The stencil contains the master shapes that represent elements in functional or logical structures. The mapping file ties the stencil shapes to Teamcenter object types and properties. This combination sets up bidirectional interaction between the diagram and Systems Engineering.

Each diagram template designates a Teamcenter *application domain* for its diagrams. The application domain defines a context for accomplishing particular business tasks with those diagrams. The stencil shapes represent the object types and properties in the domain, and the mapping file contains domain-specific XML entries.

Systems Engineering provides the following application domains by default:

- **FunctionalModeling**, in which diagrams are fixed to the **Function** view
- **LogicalModeling**, in which diagrams are fixed to the **Logical Block** view

Note:

You cannot change a diagram's template association after the first time you save the diagram.

Diagram templates also specify *membership rules* and *relation rules* for diagrams. A membership rule lists the object types that can be represented in diagrams based on the template. A relation rule lists the General Relationship Management (GRM) relations that can exist among diagram members.

Master shapes in Teamcenter stencils

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Siemens Digital Industries Software assumes that you are familiar with Visio.

A diagram's member shapes represent elements of a structure in the **Function** view or the **Logical Block** view. Teamcenter stencils define the master shapes that represent object types in Systems Engineering.

Two default stencils are provided with the Teamcenter installation:

- **FM_Diagram_Stencil** (.vsx or .vssx) is used by diagram templates for the **FunctionalModeling** application domain (**Function** view).
- **LM_Diagram_Stencil** (.vsx or .vssx) is used by diagram templates for the **LogicalModeling** application domain (**Logical Block** view).

Each default stencil contains a *block* master shape, named **Function** or **LogicalBlock**, according to the application domain. The stencils also contain the following master shapes:

Master shape	Description
TraceLink	Using trace link shapes in a diagram, you can manage defining and complying relationships among objects in the Function and Logical Block views. With the In-Context mode enabled, you can select occurrences and item revisions as defining and complying objects. Otherwise, you can select only item revisions.
Connection	Connection shapes between other shapes join the corresponding objects in a relationship, such as a functional or logical association, a data flow, or a control flow.

Master shape	Description
	- Connections can show the names of defining and complying objects for trace links. A connection shape also can show an arrow pointing to the defining object. - When the Hide Interface Shapes option is set for a diagram template, its diagrams do not show interfaces for connection shapes.
ItemRevision and InternalPort	Caution: You cannot create objects by dragging these master shapes to the drawing pane. They are reserved for mapping when you copy a function or a logical block to a diagram with the Paste as Object option. That option inserts a shape for the copied object, without adding an occurrence in the structure.
Bi-Directional Port , Input Port , and Output Port	<i>Interfaces, or ports</i>, identify points of output and input on functions and logical blocks. You can attach connections to each type of interface shape.
Container	Containers hold other shapes inside visual boundaries for better diagram organization. Teamcenter containers show additional structure detail for a diagram owner.
	Note: Teamcenter containers are available only in diagrams for owners that are selected in the Function and Logical Block views.

You can open multiple Teamcenter stencils in the same diagram. In the **Shapes** pane, you click the **More Shapes** bar and choose **Open Stencil**. Then, you can browse to and select the stencil that you want to open.

Note:

You can also use unmapped stencils such as those provided with Visio. Native Visio shapes do not affect the corresponding structure in Systems Engineering, although they are saved in the diagram.

Adding structure detail with containers

Note:

- Microsoft Visio Professional must be installed on your computer.
- Teamcenter containers are available only in diagrams for owners that are selected in the **Function** and **Logical Block** views.

Like Visio's native containers, Teamcenter containers hold other shapes inside visual boundaries, where you can arrange related shapes in clusters. In addition, Teamcenter containers show structure levels below the immediate children of a diagram owner.

A **Container** master shape is included in each default Teamcenter stencil. The master shape is mapped to the **Function** or **LogicalBlock** object type, according to the stencil's application domain. A container in a diagram represents a function or logical block in the diagram owner's structure.

To add a container, you drop the master shape from a Teamcenter stencil onto the diagram page. In the structure, a new function or logical block is created for the empty container.

Tip:

You can add containers to other containers. Each nested container allows you to see one level deeper in the structure.

You cannot add a container to a smaller container. To add the container to the smaller container, you must reduce the size of the container and then add it to the other container.

Then, you can add shapes to the container in the diagram. You can move existing shapes inside the boundary, and you can drop new ones from the stencil into the container.

- For existing shapes that you move inside the boundary, the corresponding elements are moved below the containing object in the structure.
- For new shapes that are mapped to Teamcenter object types, corresponding elements are created below the containing object in the structure.

Warning:

When you move, copy, or delete a container, you move, copy, or delete its members also.

Containers are resized automatically to fit their contents, although you can resize them manually as well. Resizing a container over existing shapes does not add them to the container, nor does it affect the structure. Shapes become container members only when they are placed inside the boundary.

As container members, shapes remain independent of each other and of the container itself. You can work on individual members without ungrouping and regrouping the entire cluster.

Tip:

To create a connection from a container member to an external object, you must first add a connection from the member to the container's boundary. This action creates an interface on the member and an interface on the boundary.

Then, you add another connection from the boundary interface to the external object. In the structure, connections and interfaces are created for the corresponding objects.

To remove shapes from a container, you move them out of the boundary. Those shapes become the diagram owner's direct children in the structure, and their relationship to the container is deleted.

Note:

- You cannot remove a shape to which connections are attached.
- Using the related Visio features, you can lock containers to prevent shapes from being added and removed.

Container contents are updated automatically when the diagram is reopened.

Create a diagram

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Multiple **Visio Diagram** views can be open simultaneously in Visio.
- The diagram's owning object does not appear in the diagram.

1. Do one of the following:

In the **Function** or
Logical Block view

- Select the object to become the diagram owner.
- Choose **File→New→Diagram**.

You can also right-click the object and choose **Create Diagram**.

In an open parent
diagram

Do one of the following:

- For a child diagram that inherits the parent's application domain, right-click the shape for the diagram owner and choose **Open Diagram**.

- For a child diagram in an application domain other than the parent's, right-click the shape for the diagram owner and choose **Create Diagram....**

The **Create Diagram** dialog box opens.

Note:

With Visio, multiple diagrams can be open simultaneously. You are not prompted to save those diagrams before the **Create Diagram** dialog box can open.

2. Type the diagram name in the **Name** box.

You can also do the following optional actions:

ID box

- Type an object identifier for the diagram.

You can click **Assign** to enter the next available identifier automatically.

Description box

- Type a description of the diagram, such as its purpose or content.

Open check box

- Clear the check box to create the diagram object without opening it in the **Visio Diagram** view.

Tip:

Leave the **Open** check box selected to create the diagram object and open it in the **Visio Diagram** view.

Note:

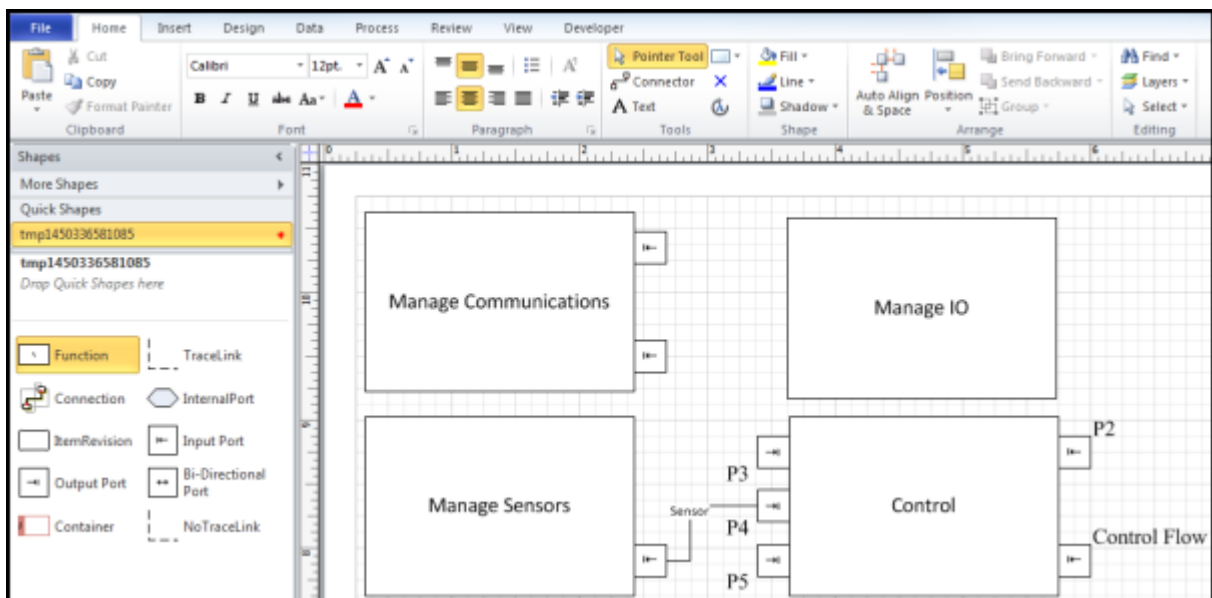
You cannot change the following:

- The **Application Domain** list shows the default application domain for the diagram owner's object type.
- The **Select Template** list shows the default diagram template for the diagram owner's object type.

You can use the **NE_default_application_domain_template** preference to change the default application domain and diagram template.

3. Click **OK** to create the diagram object in your **Newstuff** folder.

If you leave the **Open** check box selected, the diagram opens in the **Visio Diagram** view. The diagram contains a shape for each direct child of the owner.



You can edit the diagram to modify the structure in Systems Engineering.

Tip:

- The diagram is empty if a structure does not exist below the owner. You can create subordinate elements by adding shapes to the diagram from a Teamcenter stencil.

- If the diagram template uses multiple Teamcenter stencils, a window-shade button for each stencil is displayed in the leftmost pane of the **Visio Diagram** view. Click a window-shade button to see the shapes in that stencil.

Open a diagram from the Function or Logical Block view

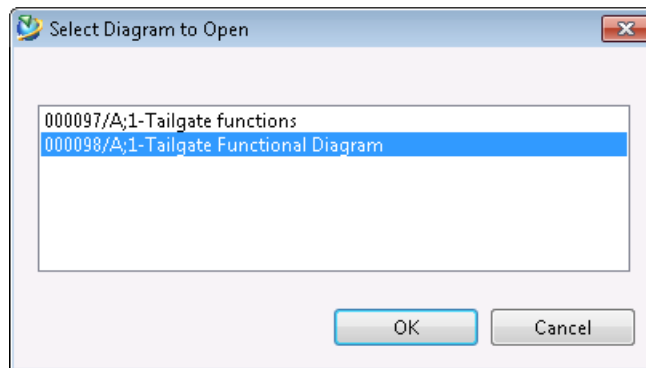
To identify which object owns the diagram you want, you can insert the **bl_rev_Fnd0Diagram_Attaches** property column in the view. Each cell value shows that object's diagram items and item revisions.

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- If another diagram is open, close it before you start this procedure. Teamcenter allows only one open diagram at a time.
- The diagram owner does not appear in the diagram.

- In the **Function** or **Logical Block** view, do one of the following:
 - Select the diagram owner and choose **Tools→Diagram→Open Diagram**.
 - Right-click the diagram owner and choose **Open Diagram**.

If the selected object owns two or more diagrams, the **Select Diagram to Open** dialog box displays the diagrams.



- Select the diagram and click **OK**.

A new view opens with the diagram owner at the top level, and the diagram opens in the **Visio Diagram** view. The contents are automatically updated with any changes in the structure since the diagram was last closed.

Note:

The **Visio Diagram** view for Visio 2010 and 2013 does not contain the Office ribbon. Those features are available through shortcut menus in the view. For example, you can select shape editing features by right-clicking a shape and choosing **Format**.

Tip:

You can also open a diagram from your **Home** view. In the **Newstuff** folder, right-click the diagram and choose **Open Diagram**.

Open a diagram from another diagram

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- The diagram owner does not appear in the diagram.

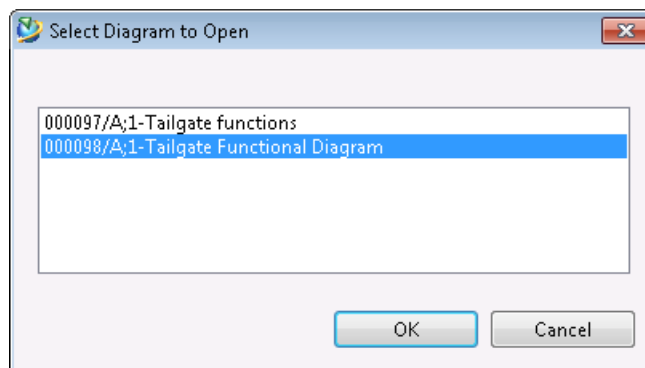
To open

A diagram for an object represented by a block shape in the current diagram

Do this

- Right-click the shape and choose **Open Diagram**.

If the selected object owns two or more diagrams, the **Select Diagram to Open** dialog box displays the diagrams.



- Select the diagram and click **OK**.

To open	Do this
	A new Visio Diagram view opens to display the selected diagram. The contents are automatically updated with changes in the structure since the diagram was last closed.
The parent diagram of the current diagram	Right-click a blank area on the diagram page and choose Open Parent Diagram. A new Visio Diagram view opens to display the selected diagram. The contents are automatically updated with changes in the structure since the diagram was last closed.

Previewing diagram snapshots

For the diagrams owned by a function or logical block, you can view static images by selecting the owner and choosing **Window→Show View→Relation Browser**.

The Relation Browser shows the static diagrams.

Navigate from a diagram shape to the underlying Teamcenter object

Note:

Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

- Right-click the shape and choose **Go to Teamcenter**.

My Teamcenter highlights the object.

Editing diagrams

How diagram edits affect Teamcenter structures

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Siemens Digital Industries Software assumes that you are familiar with Visio.

Diagrams are dynamically synchronized with structures in the **Function** view or the **Logical Block** view. Diagrams contain Visio stencils whose master shapes represent Systems Engineering objects.

Note:

In Teamcenter stencils, master shapes are related to object types through a mapping file.

You can use multiple Teamcenter stencils in the same diagram. You can also use stencils that are not mapped to Teamcenter, such as those provided with Visio.

Note:

- To access Teamcenter stencils, you use the **Shapes** button on the diagram view toolbar.
- Native Visio shapes do not affect the corresponding structure in Systems Engineering, although they are saved in the diagram.

As you add, modify, and remove shapes in a diagram, the same objects are created, modified, and removed dynamically in the structure.

To save the diagram content in Teamcenter, click the **Save** button to the right of the tab at the top of the diagram view.

Note:

In Visio, the **Save** button is always enabled.

Your changes are applied to the corresponding objects in the **Function** or **Logical Block** view.

As members of the diagram owner are added, modified, or removed in the structure, the diagram is updated with those changes when you do either of the following:

- Right-click the diagram page and choose **Refresh Diagram**.
- Open the corresponding diagram after structure changes are made.

Arranging shapes in diagrams

Note:

Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

To arrange the shapes in a diagram, you use the related features in Visio. You can place shapes anywhere on the diagram page without affecting the structure hierarchy in Systems Engineering.

Note:

- Interfaces on functions and logical blocks are moved with the owning shapes.
- You cannot move shapes from one diagram to another.

Every structure element that is represented by a shape is linked to the diagram.

- When a member is moved out of the diagram owner's structure, the member's link with the diagram is removed.
- When a diagram is moved to another owner, the diagram's link with the previous owner is removed.

The next time the diagram is updated, its previous content is deleted and all members of the new owner are added to the diagram.

Add a function or a logical block to a diagram

Note:

Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

For	Do this
A new function or logical block	1. From a Teamcenter stencil, drag the object's shape to the diagram page. The shape appears with selection handles, a rotation handle, and a default name. 2. Double-click the default name, type a new name, and then click the diagram page. In the Function or Logical Block view, a corresponding object is created as a member of the diagram owner.
An existing function or logical block	1. In the Function or Logical Block view, right-click the existing object and choose Copy. Note: The object can be a member of the diagram owner or a member of a different structure. 2. In the diagram, do one of the following:

For	Do this
	- To add the shape and create a new corresponding object, choose Paste and Create Occurrence. The shape appears in the lower left corner of the page. A new occurrence of the object is created below the diagram owner. - To add the shape without creating a new corresponding object, choose Paste as Object. The new shape appears with selection handles and a rotation handle. The shape represents the item revision of the selected object.

Add an interface to a diagram shape

Note:

Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

- From a Teamcenter stencil, drop the applicable port shape on a connection point on the block shape.

The interface is created on the corresponding element in the **Function** or **Logical Block** view.

Note:

By default, the interface is bidirectional, with capability for both input and output. You can change the interface direction to either input or output.

Change interface direction in a diagram

Note:

Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

- Right-click the interface shape and choose **Set Port Direction**.

The **Set Port Direction** dialog box is displayed.

- In the **Port Direction** list, select one of the following:

Input	Specifies that the interface can only consume a payload.
Output	Specifies that the interface can only generate a payload.
Bi-Directional	Specifies that the interface can both generate and consume a payload. This value is the default.

Add a connection between diagram shapes

Caution:

You cannot use a connection that you copy and paste from another structure. An error message is displayed if you attach such a connection to an interface (port) shape.

Note:

- Connection interface directions are defined in the **PortDirections** tag in the mapping file. By default, the starting direction is **Output** and the ending direction is **Input**. You can set the interface direction for the new connection by changing the default values.
- The diagram template and a related preference can be set for enabling connections without interfaces in diagrams. In this case, connections can be attached directly to source and target blocks.
- Multiple connections can exist between the same two blocks.

1. From a Teamcenter stencil, drag the connection shape and glue its begin point to the starting point on the source block.

Note:

If the diagram does not show interfaces, you can glue the connection directly to the block. Otherwise, the starting point must be an interface on the block.

2. Glue the connection shape's end point to the ending point on the target block.

Note:

If the diagram does not show interfaces, you can glue the connection directly to the block. Otherwise, the ending point must be an interface on the block. The ending point can reside on the same block as the starting point.

The connection's source and target context determines the result in the **Function** or **Logical Block** view.

Connection context	Result
Between two block shapes	With the PortDirections default values: • An output interface is created at the starting point on the source block. • An input interface is created at the ending point on the target block.
Between a block shape and an interface shape	A new interface is created on the source block. • If the target interface direction is Input or Output, the new interface has the opposite direction. • If the target interface direction is Bi-Directional, the new interface direction is Bi-Directional also.
Between two interface shapes	Each interface's direction is retained.
From a container boundary to an internal shape	A bidirectional interface is created at both the starting point on the container and the ending point on the internal shape.
From a container boundary to an external shape	With the PortDirections default values: • An output interface is created at the starting point on the container. • An input interface is created at the ending point on the external shape.

Note:

- Connections and their associated interfaces and diagrams are included in new structures that are created by the **Duplicate** command.
- You can release connections in a baseline structure if their parents are not locked.
- Connections can show the names of defining and complying objects for trace links. A connection shape also can show an arrow pointing to the defining object.

- When you remove a block that has connections, to avoid dangling connections, you must remove the block's connections or attach the disconnected connection to an existing block or to a newly added block.

Add a trace link between diagram shapes

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- The diagram template's trace link relation rule may override the display of trace links in the diagram.

Caution:

If you intend to assign a custom trace link subtype, ensure that it is activated for diagrams before you start this procedure.

For a trace link that

Do this

Appears immediately in the diagram, and has the default trace link subtype

1. From a Teamcenter stencil, drop the trace link shape on a glue point on the shape for the defining object.
2. Glue the end of the trace link shape to the shape for the complying object.

Appears when the diagram is refreshed, and has the default trace link subtype

1. Right-click the shape for the defining object and choose **Start Trace Link**.
2. Right-click the shape for the complying object and choose **End Trace Link**.

Note:

If the trace link relation rule permits, you can see the trace link in the diagram by right-clicking the Visio page and choosing **Refresh Diagram**.

Appears when the diagram is refreshed, and has a custom trace link subtype

Caution:

Ensure that the custom trace link subtype is activated for diagrams.

1. Right-click the shape for the defining object and choose **Start Trace Link**.

For a trace link that	Do this
	- Select the shape for the complying object and choose Tools→Trace Links→End Trace Link... The Create Trace Link dialog box is displayed. - Type the subtype name in the Name box. You can type an optional description in the Description box. - Click OK, or click Apply to keep the dialog box open and create another trace link with a custom subtype. Note: If the trace link relation rule permits, you can see the trace link in the diagram by right-clicking the Visio page and choosing Refresh Diagram.

A corresponding trace link is created in the **Trace Links** view. Next to each linked object, a trace link symbol indicates whether that object is an item revision or an occurrence.



The linked object is an item revision.



The linked object is an occurrence.

Note:

In the diagram, the trace link shape does not show whether the defining and complying objects are item revisions or occurrences.

Create a trace link from a diagram shape to a Systems Engineering object

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

The diagram template's trace link relation rule may override the display of trace links in the diagram.

- The complying Systems Engineering object can reside in the diagram owner's structure or in a different structure.

Tip:

To enable the selection of occurrences as defining and complying objects, do one of the following in Systems Engineering:

- Choose **Edit→Toggle In-Context Mode**.
- Click **Toggle In-Context Mode**  on the toolbar.

Otherwise, you can select only item revisions.

Caution:

- If you select occurrences and item revisions together as complying objects, the diagram template's trace link relation rule must allow that combination. Otherwise, a validation error is displayed.
- If you intend to assign a custom trace link subtype, ensure that it is activated for diagrams before you start this procedure.

1. In the diagram, right-click the shape for the defining object and choose **Start Trace Link**.
2. In the **Function** view or the **Logical Block** view, select the complying object, and then do one of the following:

To**Do this**

Accept the default trace link subtype

- Choose **Tools→Trace Link→End Trace Link**.

You can also right-click the object and choose **End Trace Link**. Or, click **End Trace Link**  on the toolbar.

To	Do this
	Note: If the trace link relation rule permits, you can see the trace link in the diagram by right-clicking the Visio page and choosing Refresh Diagram.
Assign a custom trace link subtype	Caution: Ensure that the custom trace link subtype is activated for diagrams. - Choose Tools→Trace Link→End Trace Link... to display the Create Trace Link dialog box. You can also click End Trace Link...  on the toolbar. - Select the subtype from the list in the Create Trace Link dialog box. - In the Name box, type a name for the trace link object. You can type an optional description in the Description box. - Click OK, or click Apply to keep the dialog box open and create another trace link subtype. Note: If the trace link relation rule permits, you can see the trace link in the diagram by right-clicking the Visio page and choosing Refresh Diagram.

A corresponding trace link is created in the **Trace Links** view. Next to each linked object, a trace link symbol indicates whether that object is an item revision or an occurrence.

 The linked object is an item revision.

 The linked object is an occurrence.

Note:

In the diagram, the trace link shape does not indicate whether the defining and complying objects are occurrences or item revisions.

Create a trace link from a Systems Engineering object to a diagram shape

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- The diagram template's trace link relation rule may override the display of trace links in the diagram.
- The complying Systems Engineering object can reside in the diagram owner's structure or in a different structure.

Tip:

To enable the selection of occurrences as defining and complying objects, do one of the following in Systems Engineering:

- Choose **Edit→Toggle In-Context Mode**.
- Click **Toggle In-Context Mode**  on the toolbar.

Otherwise, you can select only item revisions.

Caution:

- If you select occurrences and item revisions together as complying objects, the diagram template's trace link relation rule must allow that combination. Otherwise, a validation error is displayed.
- If you intend to assign a custom trace link subtype, ensure that it is activated for diagrams before you start this procedure.

1. In the **Function** view or the **Logical Block** view, select the defining object and choose **Tools→Trace Link→Start Trace Link**.

You can also right-click the object and choose **Start Trace Link**. Or, click **Start Trace Link**  on the toolbar.

2. In the diagram, do one of the following:

To	Do this
Accept the default trace link subtype	Right-click the shape for the defining object and choose End Trace Link. Note: If the trace link relation rule permits, you can see the trace link in the diagram by right-clicking the Visio page and choosing Refresh Diagram.
Assign a custom trace link subtype	Caution: Ensure that the custom trace link subtype is activated for diagrams. - Select the shape for the defining object and choose Tools→Trace Link→End Trace Link... to display the Create Trace Link dialog box. - Type the subtype name in the Name box. You can type an optional description in the Description box. - Click OK, or click Apply to keep the dialog box open and create another trace link with a custom subtype. Note: If the trace link relation rule permits, you can see the trace link in the diagram by right-clicking the Visio page and choosing Refresh Diagram.

A corresponding trace link is created in the **Trace Links** view. Next to each linked object, a trace link symbol indicates whether that object is an item revision or an occurrence.



The linked object is an item revision.



The linked object is an occurrence.

Note:

In the diagram, the trace link shape does not indicate whether the defining and complying objects are occurrences or item revisions.

Copy a block shape and create an underlying Teamcenter object

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- You cannot copy shapes from one diagram to another.

- Right-click the source shape and choose **Create new object and paste**.

The copied shape, which represents the new object, is added to the diagram.

The new object is created in your My Teamcenter **NewStuff** folder. It inherits the type and properties of the object that underlies the source shape.

The source shape's association with the diagram owner determines the result in the **Function** or **Logical Block** view.

- If the source shape represents a member of the structure, an occurrence of the new object is added to the structure.

Any structure owned by the underlying object is cloned to the occurrence.

- If the source shape does not represent a structure member, the structure is not affected.

The underlying object and its General Relation Management (GRM) relations are copied to create the new object in My Teamcenter. Deep-copy rules are used to determine whether datasets and other attachments are copied to the new object.

Copy a function or a logical block to a diagram

Note:

Microsoft Office Visio Professional must be installed on your computer. Visio Standard is not supported.

1. In My Teamcenter, select the function revision or logical block revision, and then choose **File→Copy**.

2. In the Visio drawing pane, place the cursor where you want to insert the new shape for the revision.
3. Do one of the following:

To	Do this
Add the shape and create a corresponding occurrence in the structure	Right-click the cursor location and choose Paste and Create Occurrence. The new shape is inserted at the cursor location. Under the diagram owner, an occurrence of the revision is added to the structure.
Add the shape without creating a new occurrence	Right-click the cursor location and choose Paste as Object. The new shape is inserted at the cursor location. The diagram owner's structure is not affected.

Edit Teamcenter object properties from a diagram

Note:

Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

Tip:

The text blocks of diagram shapes can be mapped to properties of the **Boolean**, **Date**, **Double**, **Integer**, and **Text** types. You can open a text block for editing by selecting the shape and pressing F2, or by double-clicking inside the shape.

You type the value directly in text block. Features for automatic selection, such as Lists of Values (LOVs), are not available. The value must be in the correct format for the property type.

1. Do one of the following:

For a single object

- Right-click the shape and choose **Properties**.

The **Properties** dialog box is displayed. Values that you cannot edit are dimmed in the dialog box.

Note:

A shape can consist of two or more subshapes that are grouped together. Each subshape can display a different property for the object.

To edit a property in a group shape:

- a. *Subselect* the individual subshape.
- b. Press F2 to open the subshape text block.

For multiple objects

- a. Press and hold Ctrl, and then click the shape for each object whose properties you want to edit.
- b. Right-click the selection and choose **Properties**.

The **Properties** dialog box is displayed. Values that you cannot edit are dimmed in the dialog box.

Note:

You can edit only those properties that are common to all selected objects.

2. Enter your changes to the property values.
3. To commit your changes, do one of the following:
 - Click **Apply** to keep the dialog box open.
 - Click **OK** to close the dialog box.

The updated values are reflected in the diagram and in the **Function** or **Logical Block** view.

Remove a diagram shape but retain the underlying Teamcenter object

Caution:

When you remove an interface shape, its connection and relation shapes are removed as well.

- Right-click the shape and choose **Remove from Diagram**.

The shape is removed from the diagram.

Note:

The corresponding occurrence remains in the **Function** or **Logical Block** view.

Delete a diagram shape and the underlying Teamcenter object

Warning:

For an interface (port) shape, the underlying interface revision stores a Graphical Design Element (GDE), a form object for Mechatronics Process Management. This procedure deletes both the interface revision and the GDE from Teamcenter.

You cannot delete the interface revision if the GDE has occurrences in two or more structures. A GDE occurrence participating in a **ConnectedTo** relation is removed from the relation, and the occurrence is deleted.

Caution:

Before you delete a connection shape, click the **Save** button on the toolbar in the **Visio Diagram** view. Otherwise, an error message is displayed.

1. Right-click the shape and choose **Delete in Teamcenter**.

A confirmation message is displayed.

2. Click **Delete**.

The **Delete Function Structure** dialog box is displayed.

3. Do one of the following:

- To delete the object and its members at all lower levels, select the **Delete sub-structure?** check box.

The object and its members are deleted from the **Function** or **Logical Block** view.

- To delete only the object and retain its members, clear the **Delete sub-structure?** check box.

The object is deleted and its members remain in the **Function** or **Logical Block** view.

Tip:

To revoke Teamcenter deletion but remove the shape, click **Remove**.

The object remains in the **Function** or **Logical Block** view.

Restore shapes to a diagram

Note:

Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

1. Right-click a blank area on the diagram page and choose **Restore Removed Members**.

The **Select Objects to Restore** dialog box lists the objects whose shapes you have removed from the diagram.

2. Do one of the following:

Restore all shapes

- Click **Restore All**.

Restore only certain shapes

- a. Select each object whose shape you want to restore.
- b. Click **Restore Selected**.

The shapes are restored to the diagram.

Measuring technical progress for design solutions

Defining budgets

As successive decomposition elicits lower levels of detail, technical measurements can help decision makers to determine whether a proposed design is achieving performance goals. Such assessments at strategic intervals provide benchmarks of progress toward a solution that meets the customer's expectations.

In Systems Engineering, quantifiable parameters (such as technical performance measures, or TPMs) can be defined through *budgets*. When budget values are applied to structure elements, the parameters can be tracked at given points in the development cycle.

For example, initial values would be expected only to fall within a tolerance band, because early designs likely would not meet performance goals. As the solution matures, values would move increasingly closer to the goals.

Budget values can be calculated and reported from the lowest level to the system level, showing the status of downstream targets to:

- Verify targets that conform to parameters.
- Identify deficient targets for recovery or correction.

Note:

Teamcenter budgets can be applied not only in solution design but throughout the lifecycle, for example, on simulation data, on physical prototypes, and on the production system.

You define budgets for parameters such as *height*, *weight*, or *power*. A budget is a pattern (like a template) on which values for individual elements are based.

Systems Engineering implements budgets through Microsoft Office Excel. For each budget, you specify:

- A *unit* of measurement for the budget values, such as *inches*, *pounds*, or *watts*.
- An *expression* for calculating downstream budget values and reporting the results upstream.
- A default *export template* for generating the budget worksheet in live Excel.

Using these definitions, you can apply budget values to entire structures (the peak elements) and to parent elements and their substructures. The **Budgets** view displays the budgets in which a selected element participates.

Reporting budget values upstream

When a budget opens in live Excel, values are processed at each level of the structure.

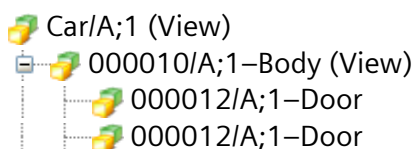
1. Values are calculated for the elements at the lowest level.
2. Those results become the values for the parent elements at the next higher level.
3. Processing continues upstream until all values are reported to the highest level at which the budget is applied.

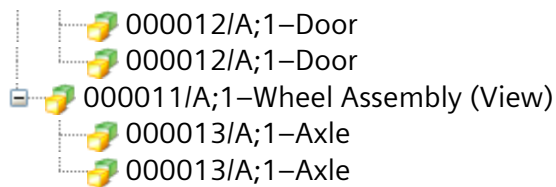
Note:

- Calculations follow the expression defined for the budget.
- Values are recalculated if they were modified since the budget was last opened.

Upstream reporting example

For an example of reporting budget values upstream, assume that a structure is organized as follows:





In this example, assume that the budget is applied at the peak level, *Car/A;1*. If the expression is **SUM**, the values are processed as follows:

A		B	
Element name		Actual Value	
1	Car/A;1	1055	This value is the sum of <i>B2</i> and <i>B7</i> .
2	000010/A;1-Body	340	This value is the sum of <i>B3</i> , <i>B4</i> , <i>B5</i> , and <i>B6</i> .
3	000012/A;1-Door	79	
4	000012/A;1-Door	79	
5	000012/A;1-Door	91	
6	000012/A;1-Door	91	
7	000011/A;1-Wheel Assembly	715	This value is the sum of <i>B8</i> and <i>B9</i> .
8	000013/A;1-Axle	420	
9	000013/A;1-Axle	295	

Create a budget definition

By default, Teamcenter places the new budget definition in the **Newstuff** folder of your **Home** view. Or, you can create a folder for your budget definitions, and then select that folder before you start this procedure.

Note:

You cannot open the budget definition, and it does not appear in the **Budgets** view. The budget definition is like a template, with predefined parameters that you use to create individual budgets on selected structures. Then, these budgets appear in the **Budgets** view. You can open and edit these budgets as necessary.

1. With the **Budgets** view active, choose **Tools**→**Budgets**→**Create Budget Definition**.

The **New Budget Definition** dialog box opens.

Note:

To display the **Budgets** view, choose **View→Show Budget**.

2. Under **Properties**, enter the following information:

Name box

- Type a unique name for the budget.

For example, use the related parameter name, such as *height*, *weight*, or *power*.

Unit box

- Type the unit of measurement for the budget values.

For example, use a unit appropriate to the related parameter, such as *inches*, *pounds*, or *watts*.

Roll-Up list

- Select the expression for calculating downstream values and reporting the results upstream.

SUM	Reports the total of the values at each level.
MIN	Reports the lowest value at each level.
MAX	Reports the highest value at each level.
AVERAGE	Reports the average value at each level.

Excel Template list

- Select the export template for generating the live Excel worksheet.

Teamcenter provides a default Excel template, **SE_BUDGET_TEMPLATE**. You can also select any custom template that is defined for your site.

Caution:

The selected template's **Excel Template Rules** property must include the values **apply_packing** and **budget_templates**.

To edit this property in Teamcenter:

- Right-click the template and choose **View→Properties** to display the **Properties** dialog box.

The template item, item revision, and dataset are located in the **RequirementsManagement Templates/ExcelTemplates** folder in your **Home** folder.

- b. Double-click the **Excel Template Rules** property.

Description box • (Optional) Type additional information to describe the budget further.

3. Click **Finish**.

The new budget definition is created in the selected folder or in your **Newstuff** folder.

Create a budget on a structure

After you create a budget definition, you can apply it at any level that represents a significant element of the design, for example, the top-level element or a lower level parent element. The budget applies to the selected element and its entire substructure.

Note:

Budgets are typically created on design elements in context, but they can be created on item revisions as well.

1. Select the structure element and choose **Tools→Budgets→Edit Budget**.

The **Edit Budget** dialog box opens.

2. In the **Budget Definition** list, select the name of a budget in which the element does not currently participate.

The **Excel Template** list displays the budget's default template. You can temporarily override the default template by selecting any other template in the list.

Caution:

The selected template's **Excel Template Rules** property must include the values **apply_packing** and **budget_templates**.

To edit this property in Teamcenter:

- a. Right-click the template and choose **View→Properties** to display the **Properties** dialog box.

The template item, item revision, and dataset are located in the **RequirementsManagement Templates/ExcelTemplates** folder in your **Home** folder.

- b. Double-click the **Excel Template Rules** property.

3. Click **OK**.

Live Excel opens the budget worksheet with empty cells.

4. Enter values in the appropriate cells of the budget worksheet, as needed.

Note:

When you click outside a cell, upstream values are calculated automatically.

You can also leave the cells empty and close the worksheet. The budget remains applied to the element and its substructure, although the values are blank in the **Budgets** view.

5. To save the budget values to Teamcenter, select the **Add-Ins** ribbon in the Excel menu bar, and then click the **Save to Teamcenter** button.
6. Exit Excel without saving the updates to the spreadsheet.

Edit budget values for a structure

Note:

You must set the **TC_Enable_Implicit_CO** preference to **true** to update budget definition values in the rich client.

For a selected structure element, the **Budgets** view displays the budgets in which the element and its substructure currently participate.

Note:

If the **Budgets** view is not displayed, choose **View→Show Budget**.

1. In the **Budgets** view, select the budget and click the **Edit** button at the bottom of the view.

The **Edit Budget** dialog box is displayed.

2. In the **Budget Definition** list, select the name of the budget.

The **Excel Template** list displays the budget's default template. You can temporarily override this template with any other in the list.

Caution:

The selected template's **Excel Template Rules** property must include the values **apply_packing** and **budget_templates**.

To edit this property in Teamcenter:

- a. Right-click the template and choose **View→Properties** to display the **Properties** dialog box.
The template item, item revision, and dataset are located in the **RequirementsManagement Templates/ExcelTemplates** folder in your **Home** folder.
- b. Double-click the **Excel Template Rules** property.

3. Click **OK**.

Live Excel opens a worksheet with the current values in the cells.

Note:

Values are recalculated if they were modified since the budget was last opened.

4. Edit values in the worksheet cells as appropriate.

Note:

When you click outside an edited cell, upstream values are recalculated automatically.

Remove a budget from a structure

This procedure removes a selected element and its entire substructure from participation in a budget. The budget itself is not deleted, and it remains with all other structures where it is applied.

Note:

If the **Budgets** view is not displayed, choose **View→Show Budget**.

1. With the **Budgets** view open, select the structure element to display its budgets.

You can select the top-level element or a subordinate element at a lower level.

2. In the **Budgets** view, select the budget and click the **Remove** button.

A confirmation message is displayed.

3. Click **Yes**.

The budget is removed for the selection and all lower level elements.

Reusing standard components from centralized libraries

Using data dictionaries

In typical functional and logical design activities, key building blocks or components of the designs are reused to save time and effort. To facilitate reuse of these building blocks or components in multiple designs, across several projects or programs, Teamcenter provides a central organizational repository for them.

These repositories are referred to as a data dictionaries. A data dictionary is modeled in the Classification perspective as a library. The Classification administrator builds a customer-specific hierarchy under one or more libraries for organizing data, for example, signals and interfaces.

Depending on access privileges, you can then:

- Create, modify, and delete data from the dictionary.
- Associate a project or program with one or more dictionaries.

Typically, the product design data is also associated with the same project or program.

- Search the dictionary for data meeting specified criteria.
- Add data from the dictionary to a functional, logical, or physical breakdown.
- Create signals in structures and associate them with interfaces and connections.

Note:

A default **Signals** library type is provided and can be used by the Classification administrator to add a signals library and data dictionary to the classification hierarchy. Your Classification administrator can also create other libraries suited to your business needs, for example, a library of functions.

Adding a signal component to a functional or logical structure

Note:

- Before you can add a signal component to a function or logical structure, the data dictionary for the signal must already exist in a **Signals** library in the **Classification** perspective.
- You must have administrator privileges to work in the **Classification Admin** perspective.

The entire process for adding a signal component to a structure, consists of the following high-level steps:

1. In the **Classification Admin** perspective, create a **Signals** library and data dictionary in the Classification hierarchy:
 - a. Create a group.
 - b. Add a class to the group.
 - c. Create a **Signals** library for the class.
 - d. Create attribute dictionary definitions for the **Signals** library.
2. In the **My Teamcenter** perspective, create a signal component and add it to the **Signals** library:
 - a. Create a signal component.
 - b. Add the signal component to the **Signals** library created previously by your Classification administrator.
3. In the **Systems Engineering** perspective, add a signal component from the library, to the structure:
 - a. Add a signal to the function or logical structure.
 - b. Associate the signal to an interface or connection in the structure.

Creating a data dictionary

Process for creating a data dictionary

Note:

- Before you can add a signal component to a function or logical structure, the signal component must already exist in a **Signals** library data dictionary, in the **Classification** perspective.
- Signals libraries and their associated data dictionaries are created in a Classification hierarchy using the **Classification Admin** perspective.
- You must have administrator privileges to work in the **Classification Admin** perspective.

The high-level process for creating a data dictionary includes the following steps:

1. Create a group.
2. Add a class to the group.
3. Create a **Signals** library for the class.

4. Create attribute dictionary definitions for the **Signals** library.

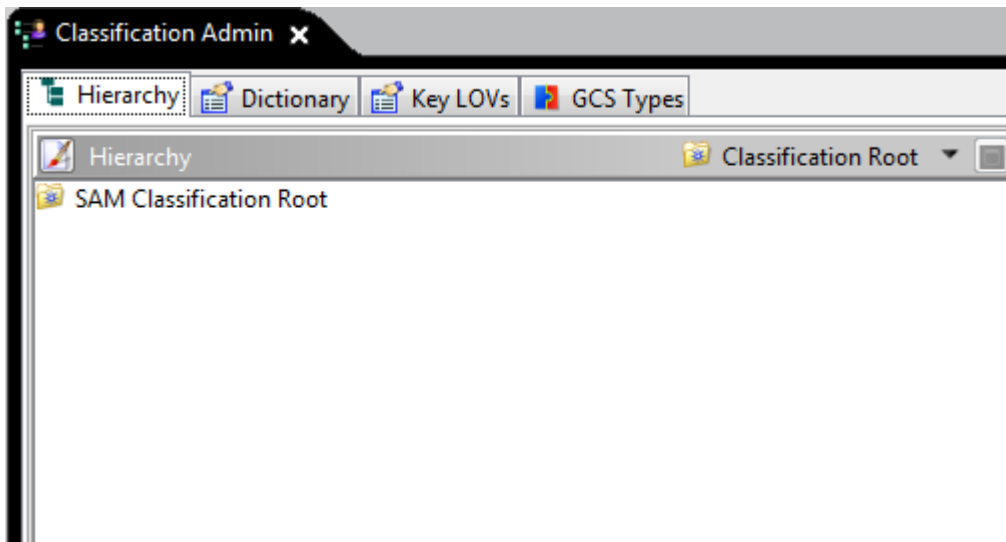
Create a group

1. Open the Classification Admin application.

Note:

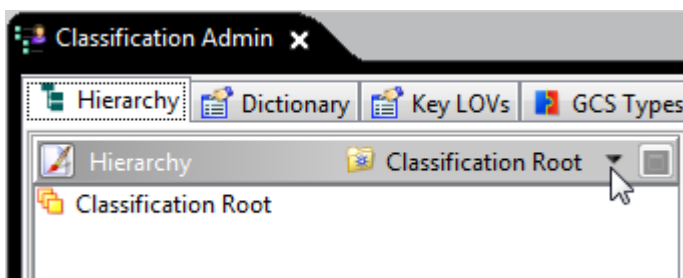
You must have administrator privileges to open Classification Admin.

The **Classification Admin** perspective opens and the **Hierarchy** pane appears.

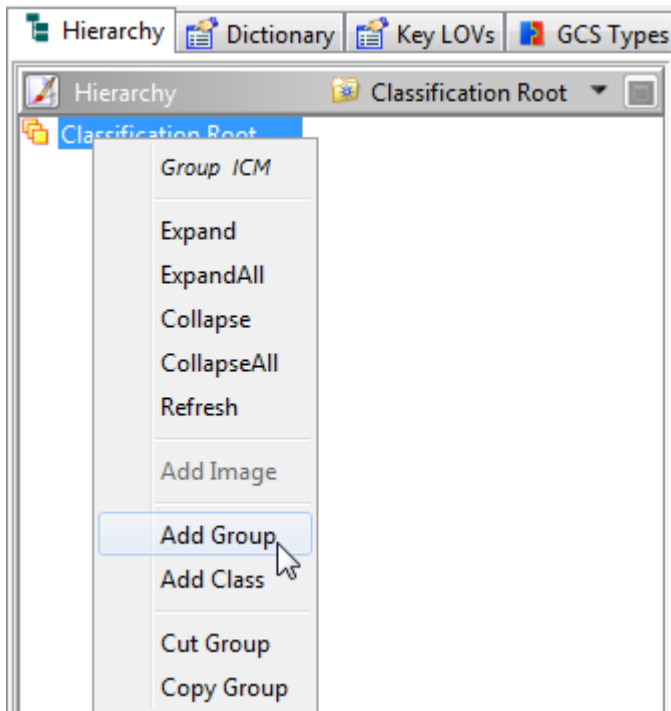


2. Choose **Classification Root**→**Classification Root**.

Classification Root appears in the hierarchy.



3. Right-click the **Classification Root** level in the **Hierarchy** pane, and select **Add Group**.

**Note:**

Only groups created in the **Classification Root (ICM)** node are visible in Classification, the perspective used when you search the data dictionary for a signal component.

4. In the **Add new Group** dialog box, type a unique identifier in the **Enter the new Group ID** box or click **Assign** to automatically assign a unique identifier.

Note:

The group identifier can be comprised of up to 31 alphanumeric characters. By default, the identifier cannot contain blank spaces or special characters (|%*:(){}[] \). You can use the dash (–) and dot (.) characters as separators in the identifier.

5. Click **OK**.
6. In the **Group** pane, type a name for the group.
7. (Optional) Complete the **Group** pane with other values, as needed.
8. In the Classification Admin toolbar, click **Save (current Instance)**.

The group is added to the **Classification Root (ICM)** hierarchy.

At this point, you can add a class to the group, for the **Signals** library.

Create a class

Note:

The process of creating a class consists of identifying the class in the database, assigning attributes to the class, assigning properties to the attributes, and saving the class definition.

1. Open the Classification Admin application.

Note:

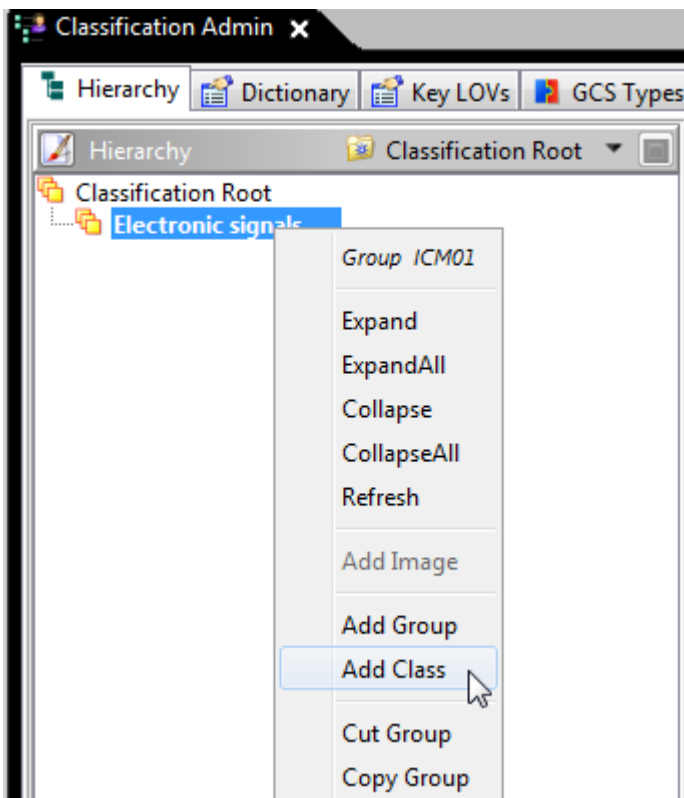
You must have administrator privileges to open Classification Admin.

2. In the **Hierarchy** pane, right-click the node in the hierarchy tree that will serve as the parent to the new class.

Note:

This may be a group or another class.

3. Choose **Add Class**.



4. In the **Add New Class** dialog box, type a class identifier or click **Assign** to automatically assign a unique identifier.

The class identifier can be comprised of up to 31 alphanumeric characters. By default, the identifier cannot contain blank spaces or special characters (|%*:(){}[] \). You can use the dash (–) and dot (.) characters as separators in the identifier.

Caution:

Once a class identifier is assigned, it cannot be modified. After the class identifier is assigned, the pane displays information about the class, including object type, class identifier, and the name of the parent group or class.

5. Click **OK**.
6. In the class definition pane, type a name for the class in the **Name** box.
7. Click **metric**, **non-metric**, or **both** (metric and non-metric) to specify the systems of measure to be applied to the attributes of the class.

The measurement system Teamcenter displays as the default can be set using the **ICS_unit_default** preference.

8. Specify whether the class is abstract or storage.

The classes are defined as abstract by default. To create a storage class, clear the **Abstract** option.

Note:

Signals libraries are created in a *storage* class but you may have additional attributes inherited from a parent *abstract* class.

You can use abstract classes to combine common attributes for use in storage classes, but you cannot store classification instances (ICOs) in them. You store instances in storage classes.

9. (Optional) Complete the class definition by typing additional values in the **Class** pane.
10. In the **Classification Administration** toolbar, click **Save current instance** to save the class definition.

The class is added to hierarchy tree, under the parent node.

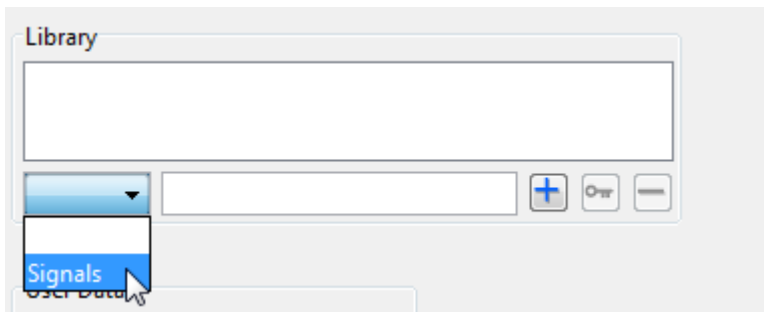
Create the Signals library for a class

1. Open the Classification Admin application.

Note:

You must have administrator privileges to open Classification Admin.

2. In the **Hierarchy** pane, navigate to the class to which you want to add the Signals library.
3. Right-click the class name and choose **Select**.
4. In the **Classification Admin** toolbar, click **Edit current Instance**.
5. In the **Class** pane, **Library** box, select **Signals** as the library type.



6. In the box next to **Signals**, type a name for the Signals library and then click **Add +**.

The name for the Signals library is added to the **Library** box.

7. In the **Classification Admin** toolbar, click **Save current Instance**.

The Signals library is created for the class.

You can now create attribute dictionary definitions for the new Signals library.

Create attribute dictionary definitions for the Signals library

1. Open Classification Admin.

Note:

You must have administrator privileges to open Classification Admin.

2. In the **Hierarchy** pane, navigate to the storage class for the Signals library to which you want to add attribute definitions.
3. Right-click the storage class and choose **Select**.

4. Click the **Dictionary** tab next to the **Hierarchy** pane.
5. In the Classification Admin toolbar, click **Create a new Instance** .
6. In the **New Attribute ID** dialog box, either type a unique identifier or click **Assign** to automatically create a unique identifier for the attribute.

Warning:

The value assigned as an attribute identifier must be either greater than 999 or less than -999. The numbers that fall within -999 and 999 are reserved for Siemens Digital Industries Software functionality.

7. Click **OK**.
8. In the **Dictionary** pane:
 - a. Type a descriptive name for the attribute in the **Name** box.
 - b. Define the units and format for the attribute.
 - c. Complete additional values for the attribute dictionary definition, as needed.
9. In the **Classification Admin** toolbar, click **Save current Instance**.

The dictionary definition is created for the Signals library in the selected storage class.

Searching the data dictionary

Open the Data Dictionary Search Dialog

You search for objects in the library using the **Data Dictionary Search Dialog**. This dialog box provides a view to the Classification application and access to the classification hierarchy where you can search for objects to add to the structure you create.

1. Select a line in the structure to which you want to add a signal or other library object.
2. Click **Add Signals from Library** .

Teamcenter displays the **Data Dictionary Search Dialog** containing the classes designated as libraries.

Find a library object

1. Search the classification hierarchy.

When searching the hierarchy, you can:

- Search for a specific ID or part of an ID.
 - Search for groups or classes.
 - Search for objects possessing specific criteria.
2. Navigate through the search results, collecting objects of interest in the selection box at the right of the pane.
 3. Select the object you want.

Searching the class hierarchy

Ways to search the class hierarchy

When you are looking for a specific library object, there are several situations you may encounter:

- You know all or part of the object ID. In this case, you can use the object ID search feature. You can narrow this search using the search by type function. This allows you to search, for example, only for library objects that classify something, or only for objects that do not classify anything.
- You do not know the ID but would like to search within a specific group or class. For this, you must know how to select a class. From within this class, you can either search all objects in the class or search for objects matching specific criteria.
- You do not know the ID but know that the library object should possess specific attribute values. Use the attribute value search to perform this search. You can narrow this search by first selecting a class, or specifying all or part of an object ID.

Note:

There are two search buttons in the **Properties** pane:

- Click the search button  to the right of the **Object ID** box to search the entire classification hierarchy. You can narrow down this search by entering a search type.
- Click the search button  at the bottom of the **Search** pane to search within a selected class.

This type of search can include any search refining features, such as object ID, attribute value, or search by type.

Additionally, there is another search button below the hierarchy tree. This is the quick search button.

Search by object ID

If you know at least a portion of the object ID of a library object, you can use the object ID search feature. An object ID search encompasses the entire classification hierarchy and cannot be limited to the contents of a particular class or subclass.

1. Click the **Search** pane located to the right of the hierarchy tree.
2. Type the exact object ID or partial object ID and wildcard characters in the **Object ID** box, located at the top of the **Search** pane.
3. (Optional) Narrow your search using the search by type feature.
4. (Optional) Click the revision rule hyperlink on the title bar to apply a revision rule for the search. If you enter a specific revision in the **Object ID** box, the search ignores the revision rule. If a revision rule is already set, it is displayed in the revision rule link.
5. Press Enter or click the **Search** button  to the right of the **Object ID** box.

The object ID search returns the objects found in alphabetical/numeric order by object ID. The first object found is displayed in the **Properties** pane. When navigating through the objects, the class and subclass to which the object belongs are highlighted in the classification tree. The total number of objects found is displayed in the **Search Results** box.

6. Navigate through the search results.
7. Select the desired library object and transfer it to Systems Engineering.
8. (Optional) Perform subsequent searches by clicking the **Search** tab, clicking **Clear**  and proceeding from step 2.

Search by type

You can search the classification hierarchy for library objects classifying specific workspace object types:

1. In the classification tree, expand the desired class and select it by either:
 - Double-clicking if it is a leaf node.
 - Right-clicking and choosing **Select** if it is not a leaf node.

The class names are highlighted in the tree.

2. Type an ID, a class ID, or attribute values.

- Click the **Drag** button . A list of workspace object types follows.

Click	To
	Search among all classification instances. This is the default search method.
	Search among all classification instances that classify a workspace object.
	Search among all classification instances that do not classify a workspace object.
	Search among all classification instances that classify an item.
	Search among all classification instances that classify an item revision.
	Search among all classification instances that classify a process.
	Search among all classification instances that classify a process revision.

Note:

The types available in this list are configurable.

- Select a type of workspace object to narrow your search. Only library objects classifying this type of workspace object are found in the search.
- (Optional) Click the revision rule hyperlink on the title bar to apply a revision rule to the search. If you search by a type that excludes revisions (for example, item or process), the search ignores the revision rule. If a revision rule is already set, it is displayed in the revision rule link.
- Click the **Search** button  at the bottom of the **Search** pane to list all matching instances within the class.

The total number of instances that match the search criteria is displayed at the bottom of the **Search** pane.

Note:

The **Search** pane displays no values upon completion of the search. You must switch to the **Properties** or **Table** pane to navigate through the results and display the values for a specific instance.

- View the search results using one of the following methods:
 - In the **Table** pane, double-click the instance to be viewed.

Its attributes are displayed on the **Properties** pane. The classification tree is updated to show the class of the selected object.

- In the **Properties** pane, use the navigation arrows at the bottom of the pane to browse forward and back through the list of items that matched your search criteria. As different objects are selected, the classification hierarchy is updated to show the class of the selected object.

Two modes are used for updating the classification hierarchy when displaying the search results:

- Click the **View in class stored** button  to display the class in which the item is stored.
- Click the **View in class searched** button  to display the class that you selected as a basis for the search.

By default, **View in class stored** is active.

As you view your search results, also use the **Viewer** pane to display any image associated with the current item. If you select multiple objects in the table and switch to the **Viewer** pane, only these objects are available in the viewer.

Selecting a class

Ways to select a class

The first step in searching the classification hierarchy is to find a class in which to search. There are three methods to select a group or class:

- Navigate through the classification hierarchy manually, clicking the groups and classes until you find your desired class.
- Use the quick search feature.
- Use the **Search Class** dialog box.

You can search for a class using any of the alias names shown in the class tooltip.

Use the quick search feature

1. In the search box located beneath the hierarchy tree in the **Data Dictionary Search Dialog**, type text corresponding to the name of the group or class that you want to locate. You can also search by ID by typing **id=xxx** in the box, where xxx is the class ID.

The search text can be the exact name or ID of the group or class you are looking for, a class alias name, or you can use character strings combined with wildcard characters.

2. Press Enter to start the search.

The hierarchy tree expands to display the first object in the hierarchy that matches the search criteria. The path of the group, class, or subclass is indicated in bold text. If multiple objects are found, the arrow buttons at the bottom of the hierarchy tree are enabled.

3. Click the left-arrow and right-arrow buttons  to display the matching objects, one at a time. This highlights the classes found in the hierarchy tree. The right-arrow button moves down the hierarchy tree, and the left-arrow key moves up the tree.

Note:

If you prefer to view a list of the results, you can display the **Search Class** dialog box by clicking the magnifying glass button located beneath the hierarchy tree.

4. Right-click the class in the hierarchy tree within which you want to search.
5. Choose **Select**.

The **Search** pane displays the attributes associated with the selected class, and the images appear in the class image window.

Note:

If the class or subclass you select is a leaf node (lowest level node) in the hierarchy, you can double-click the node to display it in the **Search** pane, rather than use the right mouse button.

Use the Search Class dialog box

1. Click the **Find Class** button  located beneath the hierarchy tree.

The **Search Class** dialog box is displayed at the bottom of your window. To move the dialog box, double-click the title bar and drag it to another location on your desktop.

2. Define the search criteria by performing the following steps:
 - a. Choose a property from the list at the upper-left corner of the dialog box. The available properties are **Class ID**, **Name**, **Attribute ID**, and **Attribute Name**.

Note:

You can use the **Name** and **Class ID** properties to search for groups and classes. When searching by attribute, the results include the class in which the attribute is defined and any subclasses in which the attribute is used. Classes that inherit the attribute are not included in the results.

- b. Type search text corresponding to the selected property.

The search text can be the exact name or ID you are looking for, or you can use character strings combined with wildcard characters.

Note:

The search box is case sensitive.

3. (Optional) If localization is enabled, select the language in which you want to search.
4. To start the search, either click the magnifying glass button located in the upper-right corner of the dialog box or press Enter.

Teamcenter displays the results of the search in the message area of the dialog box, sorted in the same order as the hierarchy tree display.

5. To display an object in the tree, double-click the entry in the results list.

The hierarchy tree expands to display the selected group, class, or subclass. The path to the object is indicated in bold text.

6. Click the left and right arrow keys beneath the class hierarchy tree  to move through the search results to locate the desired class or group.
7. Right-click the class in the hierarchy tree within which you want to search.
8. Choose **Select**.

The **Search** pane displays the attributes associated with the selected class, and the images appear in the graphics window.

Note:

If the class or subclass you select is a leaf node (lowest level node) in the hierarchy, you can double-click the node to display it in the **Search** pane, rather than use the right mouse button.

Search all objects in a selected class

1. Select the class in the hierarchy tree within which you want to search for objects.
2. (Optional) Click the revision rule hyperlink on the title bar to apply a revision rule to the search. If you search by a type that excludes revisions (for example, item or process), the search ignores the revision rule. If a revision rule is already set, it is displayed in the revision rule link.

- Determine the scope of the search by clicking the **Search Scope** button. By default, the scope is set to **Hierarchy**.

- Click **Hierarchy**  to search within the selected class and all related child classes.
- Click **Class**  to search only within the selected class.

- If you work in a multilingual environment, select the language in which to search.

Caution:

Changing the language also changes the value of the **TC_language_search** preference interactively, which affects all Teamcenter localization.

- Click one of the following at the bottom of the pane to narrow down the search.

Click	To
	Search only in metric classes.
	Search only in nonmetric classes.
	Search in both metric or nonmetric classes.

Caution:

If you are searching in a different unit of measurement than the one that you use to enter the attribute value, be sure to enter enough digits after the decimal point to avoid rounding errors.

- Click the **Search** button  at the bottom of the **Search** pane to list all search matches within the class.

The total number of objects that match the search criteria is displayed at the bottom of the **Search** and **Properties** pane.

Note:

The **Search** pane displays no values upon completion of the search. You must switch to the **Properties** or **Table** pane to navigate through the results and display the values for a specific instance.

- Navigate through the search results.
- Select the desired object and transfer it to your structure.

9. (Optional) Perform subsequent searches within the same class or subclass by clicking  in the **Search** pane and repeating the process.

Search for library objects by attribute value

1. Select the class in the hierarchy tree within which you want to search for library objects.
2. Type values, including relational operators and wildcard characters, in the boxes corresponding to the attributes by which you want to search.

You can:

- Narrow your search by specifying search criteria for multiple attributes.
- Change the displayed unit of the value by clicking it or typing the unit in the dialog box along with the value.

Teamcenter changes the unit for you automatically.

Note:

If the attribute or attribute value by which you want to search is not available for searching, one of the following reasons may apply:

- It is a reference attribute.
- It may already have a default value assigned in Classification Admin.
- It may be encrypted in the database.
- One or more of the key-LOV's values may be deprecated.

3. (Optional) Click the revision rule hyperlink on the title bar to apply a revision rule to the search. If you search by a type that excludes revisions (for example, item or process), the search ignores the revision rule. If a revision rule is already set, it is displayed in the revision rule link.

You can configure revision rules in the **Data Dictionary Search Dialog**.

4. Determine the scope of the search by clicking the **Search Scope** button. By default, the scope is set to **Hierarchy**.
 - Click **Hierarchy**  to search within the selected class and all related child classes.
 - Click **Class**  to search only within the selected class.

- If you work in a multilingual environment, select the language in which to search.

Caution:

Changing the language also changes the value of the **TC_language_search** preference interactively, which affects all Teamcenter localization.

- Click one of the following to narrow down the search.

Click	To
	Search only in metric classes.
	Search only in nonmetric classes.
	Search in both metric or nonmetric classes.

Caution:

If you are searching in a different unit of measurement than the one that you use to enter the attribute value, be sure to enter enough digits after the decimal point to avoid rounding errors.

- Click the **Search** button  at the bottom of the **Search** pane to list all search matches within the class.

The total number of objects that match the search criteria is displayed at the bottom of the **Search** and **Properties** pane.

Note:

The **Search** pane displays no values upon completion of the search. You must switch to the **Properties** or **Table** pane to navigate through the results and display the values for a specific instance.

- Navigate through the search results.
- Select the desired library object and transfer it to your structure.
- (Optional) Perform subsequent searches within the same class or subclass by clicking  in the **Search** pane and repeating the process.

Applying revision rules for searching

Configuring revision rules for searching

You can apply revision rules to searches. This narrows down the number of search results so that Teamcenter returns only the revisions you require. You can control the default behavior of the revision rules with the following preferences:

- **ICS_search_default_revision_rule_<application_name>**

Sets the default revision rule for the specified application.

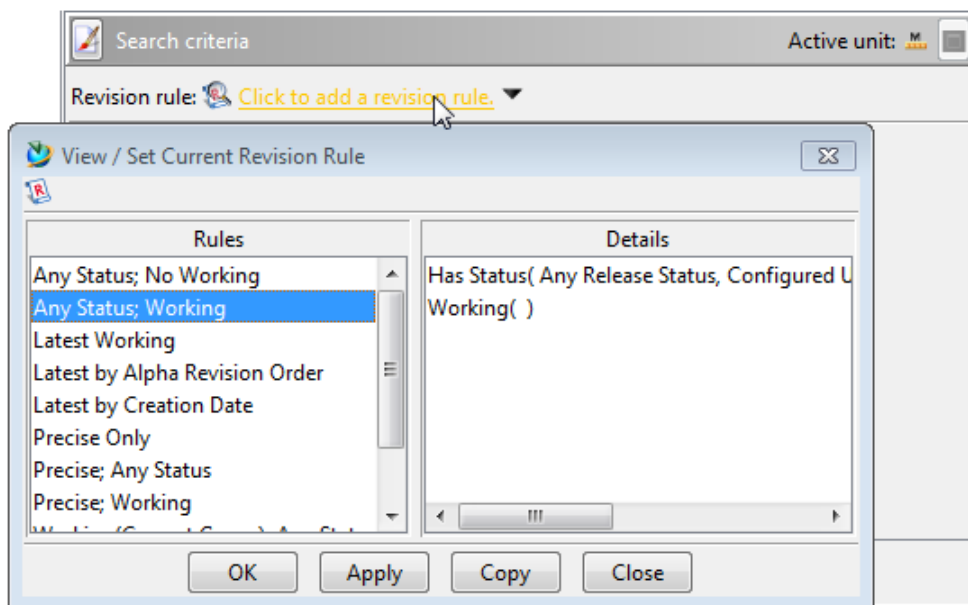
- **ICS_search_use_revision_rule**

Enables or disables revision rules for classification searches.

Apply a revision rule

1. In the **Search** pane, click the **Revision Rule** hyperlink. This link shows the name of a revision rule, if one is set, or simply states **Click to add a revision rule**.

The **View/Set Current Revision Rule** dialog box is displayed containing all existing revision rules.



2. Select the appropriate revision rule and click **OK**.
3. Click ▼ beside the revision rule hyperlink.

A list with configuration options appears.

4. Select one of the following.

Configuration option	Description
Use revision rule	Enables or disables the currently selected revision rule for the search.
Save as default	Saves the currently selected revision rule as the default rule for this application. Teamcenter saves it in the ICS_search_default_revision_rule_<application> preference. The default revision rule is directly selected and enabled for the search the next time you launch the application.
Restore default	Selects and enables the saved default revision rule. The Use Revision Rule menu command is also automatically selected.

Using relational operators

Use the following relational operators to perform Classification attribute searches.

Relational operator	Definition	For example, if you type
=	Equal to	= 3.0 in the Corner Radius attribute box of the Taper Shank End Cutter subclass, all library objects within the subclass with a corner radius equal to 3.0 are found. You can achieve the same behavior by typing a number. If no value is given after the equal sign, the system searches for all instances where no value is set for the attribute.
>	Greater than	> 3.0 in the Corner Radius attribute box of the Taper Shank End Cutter subclass, all library objects within the subclass with a corner radius greater than 3.0 are found.
<	Less than	< 3.0 in the Corner Radius attribute box of the Taper Shank End Cutter subclass, all library objects within the subclass with a corner radius less than 3.0 are found.
>=	Greater than or equal to	>= 3.0 in the Corner Radius attribute box of the Taper Shank End Cutter subclass, all library objects within the subclass with a corner radius greater than or equal to 3.0 are found.
<=	Less than or equal to	<= 3.0 in the Corner Radius attribute box of the Taper Shank End Cutter subclass, all library objects within the subclass with a corner radius less than or equal to 3.0 are found.
!=	Not equal to	!= 15.00 in the Diameter attribute box, all library objects with a diameter not equal to 15.00 are found.

Relational operator	Definition	For example, if you type
		If no value is given after the equal sign, the system searches for all instances where any value is set for the attribute.
–	Range	10.00 – 20.00 in the Diameter attribute box, all library objects with a diameter within the range of 10.00 to 20.00 (including the values 10.00 and 20.00) are found. A blank space must precede and follow the hyphen.
	OR	Walter Kennametal in the Vendor attribute box of the Taper Shank End Cutter class, all library objects within the class with a vendor of Walter or Kennametal are found. You can use the wildcard character (*) in this type of statement.
""	Exact string	"blue or green" , Teamcenter searches for the string blue or green . If you type "4 – 10" , Teamcenter searches for the string 4 – 10 , not the range 4–10.

Using wildcard characters

Classification employs the standard wildcard characters defined for your site. You can use these in any text box.

Navigating the matches

Ways to navigate the matches

After you perform a search, there are several methods to navigate the matches returned by the search. You can:

- View the results in the **Properties** pane.
- View the results in the **Table** pane.
- View the results in the **Viewer** pane.

View search results in the Properties pane

When you view library objects in the **Properties** pane, the class it belongs to is highlighted in the hierarchy. Two modes are used for updating the classification hierarchy when displaying the search results:

- Click the **View in class stored** button  to display the class in which the item is stored.

- Click the **View in class searched** button  to display the class that you selected as a basis for the search.

You can see the active unit of measurement in the attribute values title bar in the **Properties** pane using the following icons.

Icon	Description
	The library object is currently displayed and stored in a metric system of measurement.
	The library object is currently displayed and stored in a nonmetric system of measurement.
	The library object is currently displayed in metric, but was originally stored in a nonmetric measurement system.
	The library object is currently displayed in a nonmetric measurement system, but was originally stored in a metric system.

View search results in the Table pane

You can view the data of the library objects found in the search collectively using the **Table** pane.

Note:

If the measurement unit icon at the beginning of each row contains an exclamation mark, the library object is displayed in a unit system other than the one in which it was stored. For example, if the icon  is displayed, the library object was stored in a nonmetric unit system but is currently displayed in a metric unit.

- Perform one of the following actions:
 - Click ▼ to load the next page of found library objects into the table.

The next library objects are appended to those currently displayed.

Note:

By default, the number of library objects displayed on a page is set to 5. Your administrator can change the default display by modifying the **ICS_table_pagesize** preference.

- Click  to load all matches into the table.
- Use the scroll bars to move through the found library objects.

3. (Optional) Sort the table data; double-click the column header corresponding to the property to switch between ascending, descending, and natural sort order.

Teamcenter can display different units of measurement in the same column. The displayed unit depends on the optimized unit for each of the attribute values. Teamcenter sorts these columns based on the attribute values in the storage unit.

4. (Optional) Select a line in the **Tree** pane and click  to view its properties.
5. Double-click a library object in the table to view it.

The system opens the object in the **Properties** pane and highlights its class in the tree.

View search results in the Viewer pane

You can view all search results in the **Viewer** pane. You can move backward and forward in the match list using the navigation buttons.

Note:

If the object does not have an item attached, or if the item does not contain a graphic, no image appears in the **Viewer** pane.

Select the desired library objects

1. Click the **Properties**, **Table**, or **Graphics** tabs.

Note:

The **Graphics** pane is only active for a library object that has an item attached.

2. Navigate to the desired library object.
 - In the **Properties** or **Graphics** pane, use the navigation arrows.

Teamcenter displays all attributes and values for each library object.

- In the **Table** pane, do one of the following:
 - Select the library object by clicking it.
 - Double-click the library object.

Teamcenter switches to the **Properties** pane and shows all attributes and values for the selected object.

- Select multiple library objects.
- (Optional) With library objects selected in one of the panes, click **+** to add them to the selection box on the right.
 - When you finish adding objects to the selection box, select the objects in the selection box that you want to add to the structure and click **OK**.

Create a signal

- Select a folder or a containing object for the signal.
- Choose **File→New→Signal...** or click **Signal**  on the toolbar.
- Select the appropriate type of signal from the list.
The valid types are:
 - **Signal**
 - **Message**
 - **PSSignal**
- In the **New Signal** dialog box, enter the following information for the signal.

Field name	Definition	Valid Values
ID	The signal ID (required). Leave the field blank to automatically assign an ID.	Alphanumeric string (1-128 characters)
Revision	The signal revision (required). Leave the field blank to automatically assign a revision.	Alphanumeric string (1-128 characters)
Name	The signal name (required).	Alphanumeric string (1-128 characters)
Description	Description of the signal.	Alphanumeric string (1-240 characters)
Unit of Measure	The unit of measure used for the signal.	Select from the list of values.

- (Optional) Click **Next** to assign the signal to a project.
- Click **Finish**.

A new signal of the specified type is created.

Note:

To use the signal in a structure, it must first be added to a Signals library.

Add a signal to a data dictionary

1. In My Teamcenter, right-click the signal and choose **Send To→Classification**.

The Classification application displays a message asking if you want to classify the object.

2. Click **Yes** to create the ICO.
3. In the **Hierarchy** pane, select the dictionary where you want the signal classified.

Tip:

- To see a tooltip containing information about an object, place the pointer on the object in the hierarchy.
- To confirm your selection, review the related information on the **Properties** tab.

4. On the Classification toolbar, click the **Save** button.

The signal is added to the dictionary.

Tip:

- To view all of the dictionaries that use the signal:
 - a. Select the signal in My Teamcenter, and then select the **Impact Analysis** tab.
 - b. In the **Where** list, select **Used**.
 - c. In the **Display** list, select **Classified in**.
- To view other design elements that use the signal:
 - a. Select the signal in My Teamcenter, and then select the **Impact Analysis** tab.
 - b. In the **Where** list, select **Referenced**.

Add a signal to a functional or logical structure

1. In the functional or logical structure, select the top-level or sub-level structure object where you want to add the signal.

2. Click **File→New→Add Signal from Library** .
3. In the **Data Dictionary Search Dialog** dialog box, search for and select the signal object you want to add to the structure.
4. Click **OK**.

The signal component is added to the selected level of the functional or logical structure.

Associate signals with an interface or a connection

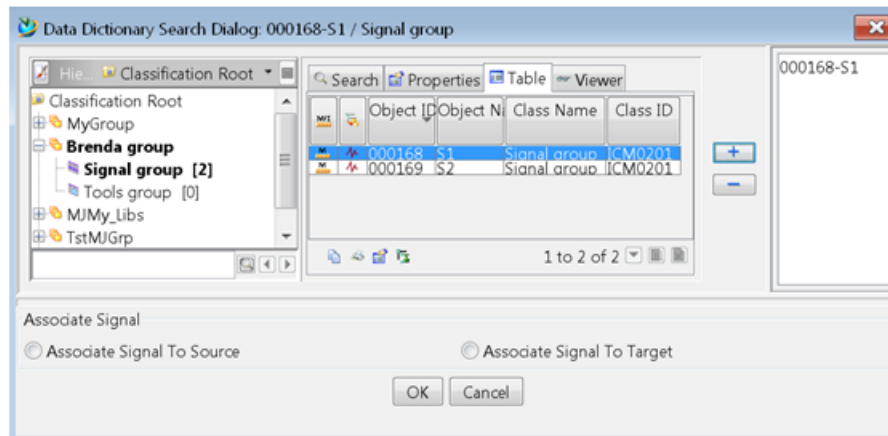
Note:

The signal must exist in the data dictionary before you can associate it to an interface or a connection.

You can associate a signal to an interface or a connection in either the **Visio Diagram** view or in the function or logical structure view.

1. Open the function or logical structure in either the **Visio Diagram** view or in the function or logical structure view.
2. If you are working with the function or logical structure view, add a signal to the function or logical structure.
3. Add an interface to the function or logical structure blocks, as needed.
4. (Optional) Add a connection between interfaces.
5. Associate a signal to the interface or connection. Depending on the view you are working in, do one of the following:
 - In the **Visio Diagram** view:
 - a. Right-click the shape for the interface or connection to which you want to associate the signal and choose **Associate Signal**.

The **Data Dictionary Search Dialog** appears.



- b. Use the **Data Dictionary Search Dialog** to find and select the signal to associate.
- c. If you are associating an interface to a signal, select either the **Associate Signal to Source** or the **Associate Signal to Target** option, as applicable.

Note:

If a connection is being associated to a signal, the associate signal options are not included on the **Data Dictionary Search Dialog**. The connection is always associated to the signal as a transmitter.

- d. Click **OK** to associate the signal to the selected interface or connection.
- In the function or logical structure view:
 - a. Select the structure line for the interface or connection to which you want to associate a signal.
 - b. Use the Ctrl key to also select the structure line for the signal to associate.
 - c. Choose **Tools→Signal Manager**, and then choose one of the following:
 - If you are associating a signal to an interface, choose either **Associate Signal To Source** or **Associate Signal To Target**, as applicable.
 - If you are associating a signal to a connection, choose **Associate Signal To Transmitter**.

The association to the signal is created.

View or navigate to signals associated with an interface or a connection

1. Open the function or logical structure in the structure view or the **Visio Diagram** view.

2. To open the **Associated Signals** view, choose **View→Show Associated Signals** or click the **Associated Signals View**  button on the application toolbar.
3. In the function or logical structure view or the **Visio Diagram** view, select the shape or the line for the interface or connection.

The signals for the selected interface or connection are listed in the **Associated Signals** view.

Note:

If signals are not listed in the **Associated Signals** view, you may need to set the view association to **None**.

4. In the **Associated Signals** view, select the signal to which you want to navigate.

The line for the selected signal is highlighted in the function or logical structure view.

Remove signals from an interface or a connection

1. Open the function or logical structure in the structure view.
2. In the function or logical structure view, select the line for the signal object from which to remove the association.
3. Choose **Tools→Signal Manager**, and then choose one of the following:
 - If you are removing a signal from an interface, choose either **Remove Signal Source** or **Remove Signal Target**, as applicable.
 - If you are removing a signal from a connection, choose **Remove Signal Transmitter**.
4. In the **Removes Signal/Process variable association** dialog box, select the signal association to remove.
5. Click **Remove**.

The association between the selected signal is removed from the interface or connection.

Set effectivity for a signal revision

Effectivity allows revisions of functions and signals to coexist in an imprecise structure. You can set the effectivity for a selected signal revision so that it can be used by a specific function revision.

Note:

The signal revision must be released through the **TCM Release Process** workflow process template in the **New Process** dialog box.

1. In Systems Engineering, select the signal revision and choose **Tools→Effectivity→Revision Effectivity**.

The **Effectivity** dialog box is displayed.

2. Select the corresponding release status, and then click **Create**.

The **Release Status Effectivity** dialog box is displayed.

3. Under **Effectivity Range**, use the calendar to select the dates in the **From Date** and **To Date** columns.
4. Click **OK**.

The **Revision Effectivity** column displays the data for the signal.

Managing connections using Connection Manager

Create a connection

You use Connection Manager to create or remove connections between a network port and one or more network connections in a functional or logical model structure. When you create the connection, the network port must be below the network connection in the structure.

1. From the navigation pane, click **Systems Engineering**.
2. Search for and open the functional or logical model where you want to create the connection.
3. In the Systems Engineering **Function** or **Logical Block** view, select a network port and one or more network connections.

The network port must be below the network connection in the structure.

4. Choose **Tools→Connection Manager→Connect**, or click **Connect**  on the toolbar.

The **Connected To** column for the network connection is set to the values of the connected network ports.

Remove a connection

1. From the navigation pane, click **Systems Engineering**.
2. Search for and open the functional or logical model where you want to remove the connection.
3. To remove *all* of the connections for a network connection, select the network connection.
4. To remove a specific network connection for a functional or logical model, select the network connection and the network port where you want to remove the connection.
5. Choose **Tools→Connection Manager→Disconnect**, or click **Disconnect**  on the toolbar.

The **Connected To** column for the network connection is set to the values of the connected network ports if any exist. The removed connections no longer display.

5. Allocating requirements and design architectures

Traceability in product design

Within Systems Engineering, you can associate requirements with design elements through trace links. You can create defining and complying trace links between objects in requirement specifications, functional models, logical models, and physical models. You can also create defining and complying trace links between Systems Engineering objects and objects in other Teamcenter applications, such as My Teamcenter, Structure Manager, and Multi-Structure Manager.

Note:

You must configure the trace links features before you can create trace links.

- A trace link establishes a directional relationship between two objects. In this relationship, one object defines a condition with which the other object must comply, that is, must partially or completely fulfill.
- A given object may define some objects and may also comply with other objects. You can view an object's defining and complying relationships in the **Trace Links** view, where you can also view the trace links themselves.

Using the **Trace Links** view, you can perform several actions including:

- Delete a trace link.
- Navigate to a linked object.
- Add a custom note to a trace link.

Custom notes are used to record ad hoc information such as decision rationale, actions, questions, or meeting minutes. You can add and edit custom notes on trace links and export custom notes on trace links to Microsoft Word.

Tip:

Defining and complying trace links can also exist between Systems Engineering objects and subsets in 4th Generation Design. Using the *cross-domain tracing* feature, collaborative designers can analyze these elements for change impact both upstream and downstream.

How trace links associate system elements

A *trace link* establishes a directional relationship between two objects. Such relationships are conveyed by the terms *defining* and *complying*.

- A defining object specifies a condition that a product or a component must fulfill.
- A complying object must partially or completely fulfill a condition specified by a defining object.

A trace link relationship creates a traceable *path*, in which one object precedes the other. For example:

- Functional requirement **A**, for target tracking, defines hardware requirement **B**, for a CPU with a certain instruction rate.
- Requirement **A** precedes requirement **B**, with requirement **B** directly *downstream* in the complying path.
- Requirement **B** succeeds requirement **A**, with requirement **A** directly *upstream* in the defining path.

A given object can assume both defining and complying relationships, continuing the path upstream and downstream. For example:

- Requirement **W**, specifying a component's weight, defines requirement **X**, specifying the use of aluminum.
- Requirement **X**, while complying with upstream requirement **W**, defines additional downstream requirements for environmental conditions that are consistent with the properties of aluminum.

An object can have any number of defining and complying trace links. These traceable relationships allow for change analysis in both directions.

For example, if the customer is considering a change to a product requirement:

- The requirement's trace links can be followed downstream, to all complying requirements that were derived from the product requirement, and eventually to all design elements that must comply.
- An engineer working on a design element that must meet detailed, low-level requirements can follow those trace links upstream, to the original customer requirements that define the design constraints.

Trace link ownership

Direct trace links are connected to the object selected in the structure view:

- A direct defining trace link starts at the object itself. The object is the trace link source.

- A direct complying trace link ends at the object itself. The object is the trace link target.

In the **Trace Links** view, the direct trace link symbol  indicates that the trace link is connected to the defining and complying objects themselves. The symbol contains a solid straight blue arrow.

Trace link inheritance

The indirect trace link concept allows all revisions of an item to implicitly inherit all trace links to the item itself. Indirect trace links are connected to the item that is associated with the item revision selected in the structure pane:

- An indirect defining trace link starts at the item. The item is the trace link source.
- An indirect complying trace link ends at the object that complies with the item. That object is the trace link target.

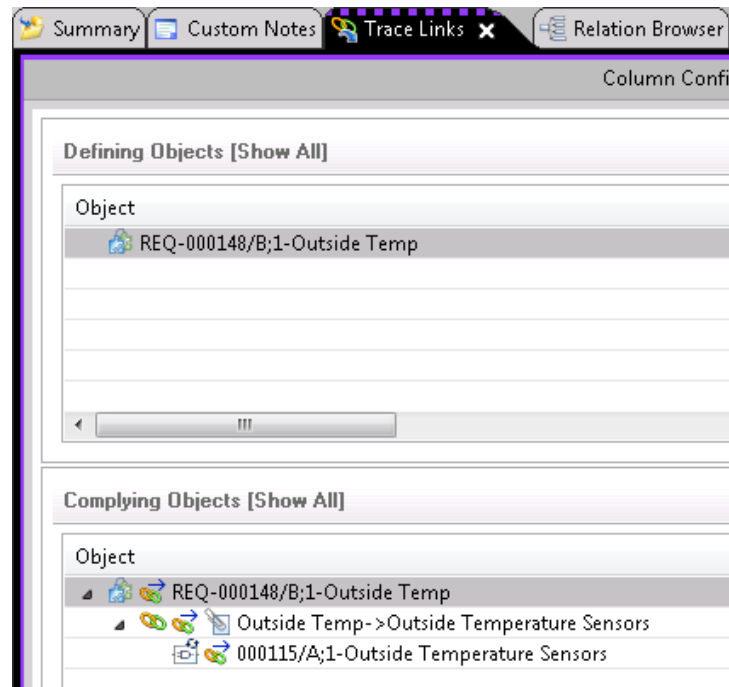
In the **Trace Links** view, the indirect trace link symbol  indicates that the trace link is connected to the item associated with the item revision selected in the structure view. The symbol contains a dashed bent blue arrow.

How new item revisions affect traceability

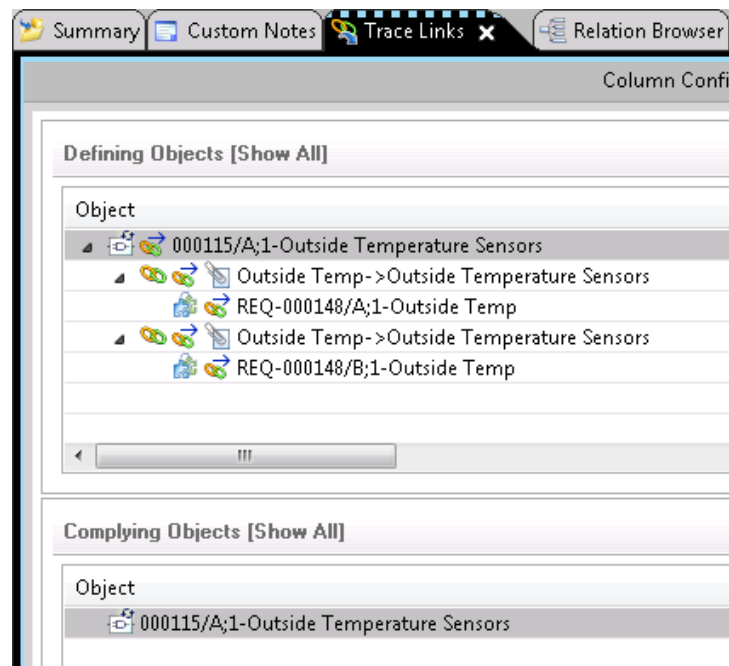
A trace link is created between a defining object, as the source or predecessor, and a complying object, as the target or successor.

- When the defining object for the trace link is revised from revision A to revision B, for example:
 - On the defining object, the trace link is carried forward to the new revision of the defining object according to the predefined deep copy rules established by your site administrator. In this example, the new revision is B.

This behavior also extends to custom notes attached to the trace link. If the trace link has a custom note, the custom note is also carried forward with the trace link to the latest revision of the defining object.

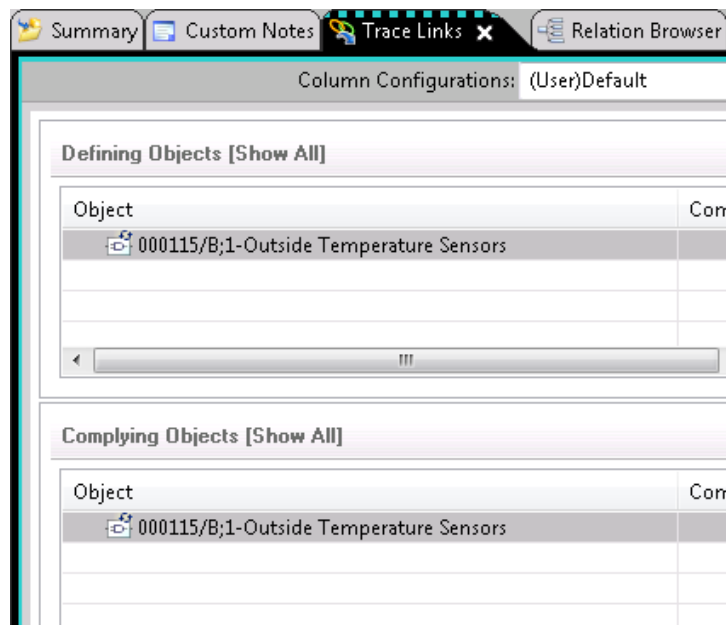


- On the complying object, the trace link is allocated to both the original revision of the defining object and the latest revision of the defining object. In this example, the trace link on the complying object extends to both the existing revision (A) and the new revision (B) of the defining object, as shown below.



- When the complying object for the trace link is revised from revision A to revision B, for example:

Trace links are retained on the original revision of the complying object (A) but are not carried forward to the new revision of the complying object (B). The following illustration shows no trace links allocated to revision B.



If you want the trace link to apply to the latest revision of the complying object, you must create a new trace link between the applicable revision of the defining object and the latest revision of the complying object.

How object modifications affect traceability

Note:

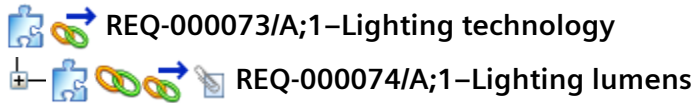
By default, trace link symbols are not displayed for objects in My Teamcenter. To determine whether trace links exist on these objects, add the **Has Tracelink** property column in the **Details** view.

For each object in the **Details** view, the **Has Tracelink** column displays one of the following values:

True The object has at least one trace link.

False The object has no trace links.

You can create custom notes for a trace link displayed in the **Trace Links** view. After you create the custom note, an attachment indicator  appears beside the trace link.



To view the custom notes, select the trace link in the **Trace Links** view with the custom note and choose **Window→Show View→Other→Systems Engineering→Custom Notes** or click **Custom Notes Manager**  on the toolbar.

When you make changes to an object on which a trace link exists, Teamcenter performs a corresponding operation related to the trace link.

- If you use the default trace link name and then change the name of the defining object, the complying object, or both, the trace link name is automatically changed.
- When a workspace or absolute occurrence object with trace links is revised, the trace links are carried forward to the new revision according to the predefined deep copy rules established by your site administrator. This behavior also extends to custom notes attached to the trace link. If you revise a workspace or absolute occurrence object with a trace link, and the trace link includes a custom note, the custom note is moved forward with the trace link.
- When a workspace object is moved, or an absolute occurrence object is moved within the same context, the trace links are moved along with the object.
- When an absolute occurrence object is moved outside of its original context, the associated trace link is deleted. It is not moved with the absolute occurrence object.
- When a workspace or absolute occurrence object with a trace link is deleted, both the objects and the corresponding trace links are deleted.

Note:

Before the deletion occurs, you are prompted to confirm the deletion.

- When a baseline is created on a structure, any trace links on objects in the structure are retained in the baseline structure. A working revision of the structure remains in the system, along with the related trace links.
- When you export entire structures or selected objects to Microsoft Office Word, you can include the content of custom notes that are attached to trace links on the objects. In the export procedure, select an object template that contains the **REQ_export_notesonlinks** transfer mode.
- When an object with trace links is transferred from one site to another, the trace links and the objects at both ends of the trace links are also transferred.
- When performing a PLM XML export operation to export a structure, you can export the trace links associated with the structure.

Note:

When using PLM XML to export a structure, use the **ConfiguredRequirementDataExport** transfer mode to include trace links in the export file. Use the **TL_import** transfer mode when using PLM XML to import the structure.

- Support for trace links is not available in Excel import.

How object copying affects traceability

When you perform a **Copy** operation on a requirement object or structure containing trace links to another requirement structure, the following behavior occurs:

- If the object containing the defining trace link and the object containing the complying trace link are both copied, a trace link is created between the two copied objects in the new structure.

The trace link is preserved and remains an *internal* trace link because it is linked to objects within the structure.

- If the object containing the defining trace link is copied, but the object containing the complying trace link is *not* copied, the trace links are created between the original complying object in the old structure, and the copied defining object in both the new and old structures.

The trace link is now *external* because it is linked to an object outside the structure.

- If the object containing the complying trace link is copied, but the object containing the defining trace link is *not* copied, the trace links are created between the original defining object in the old structure, and the copied complying object in both the new and old structures.

The trace link is now *external* because it is linked to an object outside the structure.

If you paste a reference to the object contained in the clipboard by pressing Ctrl+V, a trace link is created to the complying object in the new structure.

If you paste a duplicate of the object contained in the clipboard as a child of the current selection by pressing Ctrl+Shift+V or right-click and select **Paste as New**, a trace link is not created to the duplicate complying object in the new structure.

To create a trace link on selecting **Paste as New**, you must set the deep copy rule as follows:

Edit Deep Copy Rule
Edits a Deep Copy Rule

Project: TL3

Target Business Object: Requirement Revision [Browse...](#)

☐ Target Primary?

Operation Type: SaveAs

Property Type: ☒ Relation ☐ Reference

Relation : * FND_TraceLink [Browse...](#)

Attached Business Object: * ItemRevision [Browse...](#)

Condition: * isTrue [Browse...](#)

Action: CopyAsObject

☐ Required?

☒ Secured?

☒ Copy Properties on Relation?

[?](#) [Finish](#) [Cancel](#)

```
<TcDeepCopyRule typeName="Requirement Revision"
  relRefPropName="FND_TraceLink" propertyType="Relation"
  objectTypeName="ItemRevision" conditionName="isTrue"
  operationName="ITEM_create_from_rev" copyType="CopyAsObject"
  isTargetPrimary="false" secured="true"
  copyRelationAttributes="true" isRequired="false"
  description=""/>
```

This deep copy rule creates a trace link on the new copy of the selected complying object of the **Requirement Revision** type.

Rules for trace links

Defining trace link rules

Trace link rules guide the user to create trace links of the correct type, and in the correct direction, for specific object type pairs. The trace link rules define the:

- Object type pairs that can be selected as the defining and complying objects in a trace link.
- Valid direction of the trace link for those object type pairs.
- Trace link type used by default when a trace link is created for the object type pair.

Trace link rules can be defined as the site-level default and can be overridden by the user to create a user-defined default.

The ***_default_relation** preference can be used to define the allowed trace link direction and type used between specific object type pairs. A ***default_relation** preference is created automatically when a trace link type is set as the default during trace link creation. The preference can also be created manually by a Teamcenter administrator using the **Edit→Options** dialog box.

The order of precedence for applying the trace link rule used during trace link creation is:

1. The user-defined default for the selected object types.
2. The site (or group) defined default for the selected object types.
3. The global default trace link rule.

Following is the preference and setting for the global default trace link rule:

TraceReport_Default_TraceLink_Type with the default value **FND_TraceLink**.

Note:

The preference is modifiable by an administrator, to define a different trace link type as the global default.

Enforcing trace link rules

During trace link creation, if you attempt to create a trace link in the wrong direction based on the definition of the trace link rule, you are prompted with options to do one of the following:

- Reverse the trace link direction. The object selected as the source (defining) object in the trace link becomes the target (complying) object for the trace link instead, and visa versa.

- Proceed with creating the trace link even though it is in the opposite direction than is defined in the trace link rule for the object type pair.
- Cancel the trace link creation.

You are also given the option to set the trace link type as the user default for the object type pair selected as the defining and complying objects for the trace link. Setting the default trace link type creates a trace link rule for the object type pair.

Note:

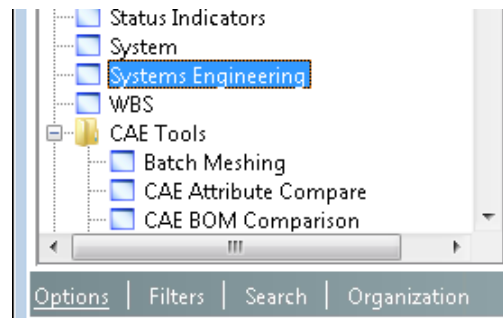
The option to set the trace link type as the default is disabled if the protection scope for the trace link rule is not set to **User**.

General Relation Management (GRM) rules defined using BMIDE control the hard rules related to creating trace links.

For example, it is possible to create a trace link by selecting a requirement as the defining object and a function block as the complying object. However, it is not possible to create a trace link by selecting a function block as the defining object and a requirement as the complying object. This results in an error. The system controls this implicitly using GRM rules.

View trace link rules for object type pairs

1. Choose **Edit→Options** from the toolbar.



2. In the **Options** dialog box, click **Search**.
3. In the **Preferences By Search** dialog box, in the **Preferences List** pane:
 - a. Type ***_default_relation** in the **Search by keywords** box.
 - b. Select **General** in the **Filter by category** box.

The preferences matching the search are listed.

4. Select the name of the *default relation* preference for the object type pair. For example, **PC4_TechSafetyCoRevision_PC4TestCaseRevision_default_relation**.

Note:

- The first object type in the preference name, **PC4_TechSafetyCoRevision**, is valid as the defining object for the trace link.
- The second object type in the preference name, **PC4TestCaseRevision**, is valid as the complying object for the trace link.

5. View the preference settings in the right side of the **Preferences By Search** dialog box:

- The default trace link type for the object type pair appears in the **Value** box.

PC4_VerifiedBy is the default trace link type for the trace link rule.

- The **Protection Scope** value is set to **User**.

Protection Scope may be set to **Site, User, Group or Role**. When the value is set to **User**, the default trace link subtype for the object type pair can be set by the user during trace link creation.

Create a trace link rule

To create a trace link rule:

- Create a trace link between the object type pair you want to use consistently as the defining and complying objects for trace links, and set a specific trace link type as the default.
- You can repeat this procedure for every object type pair for which you want to create a trace link rule.
- After a trace link rule is created for a specific object type pair, the trace link type you set as the default is used automatically when you end the trace link.

Note:

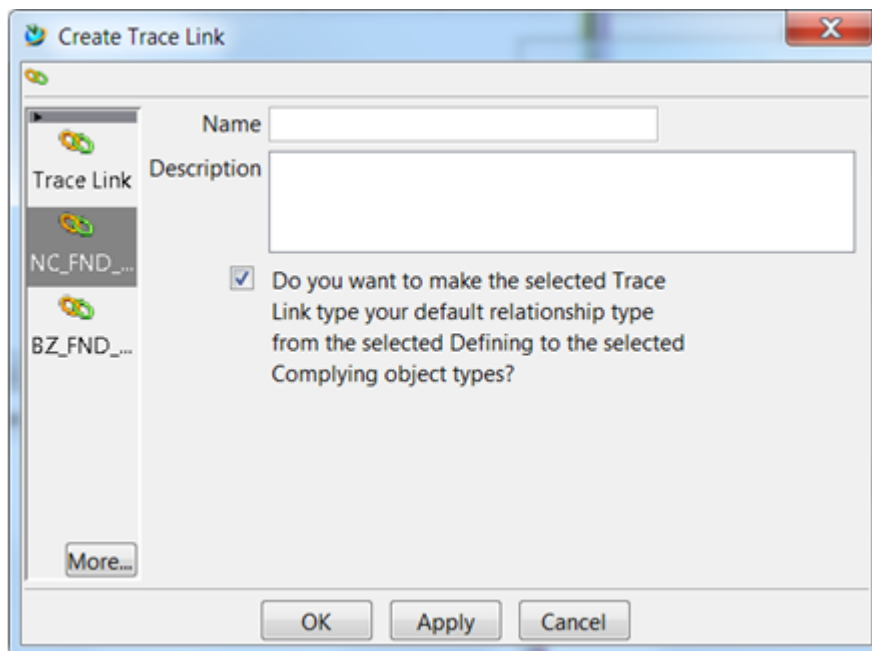
The default trace link type is also used when creating trace links in Word when editing requirement structures using the Client for Office application.

1. From the navigation pane, click **Systems Engineering**.
2. In the **Systems Engineering** perspective, select the requirement or other object type where you want to start the trace link and then do one of the following:
 - Click **Start Trace Link**  on the toolbar.

- Right-click and choose **Start Trace Link**.
 - Choose **Tools→Trace Links→Start Trace Link**.
3. Select the object where you want to end the trace link and choose **Tools→Trace Links→End Trace Link....**
 4. (Optional) In the **Create Trace Link** dialog box, enter the following information:

Field name	Definition	Valid values
Name	Name of the trace link.	Alphanumeric (1–128 characters)
Description	Description of the trace link.	Alphanumeric (1–240 characters)

5. Select the trace link type you want to set as the default.
6. Select the **Do you want to make the selected Trace Link type your default relationship type from the selected Defining to the selected Complying object types?** check box.
7. Click **OK**.



The trace link rule is created for the defining and complying object types, with the trace link type set as the default.

Note:

After creating a trace link rule by setting a default trace link type during trace link creation, you can adjust values for the trace link rule preference you just created.

Set preference values for trace link rules

After creating a trace link rule by setting a default trace link type during trace link creation, you can adjust the preferences, as needed, using the **Options** dialog box.

Click on the "Edit" button to modify the definition and update any field in order for the "Save" button to be enabled. Note that the "Description" field must not be empty. Click on the "Save" button to save the definition of the existing preference.

Name	Value Location	Protection Scope
Revision_PC4TestCaseRevision_default_rela	Site	User
Category	Environment	Type
SystemsEngineering	Disabled	String
Multiple		
Single		
Description		
Technical Safety Concept - Component Test : Verified By		
Value		
PC4_VerifiedBy		

- As the user who owns the preference, you can edit the property values for the preference. For example, if you set **SystemsEngineering** as the value for **Category**, you can filter the preferences list by the category **SystemsEngineering**. This makes it easier to find and view or edit trace link rule preferences.
- As an administrator, you can adjust other property values for the preference, or even create a new preference using the **Options** dialog box.

Caution:

Creating preferences using the **Options** dialog box requires knowledge of the Teamcenter data model. For accuracy in creating the trace link rule preference name, copy and paste the object type names for the defining and complying objects from BMIDE to the **Name** box. The preference name format is `<ItemRevision1>_<ItemRevision2>_default_relation`.

Creating trace links

Selecting defining and complying objects

When creating a trace link, you can select the defining and complying objects as follows:

- One defining object and one complying object.

The source has one defining trace link to the target, which has one complying trace link from the source.

- One defining object and multiple complying objects.

The source has one defining trace link to each target. Each target has one complying trace link from the source.

- Multiple defining objects and one complying object.

Each source has one defining trace link to the target, which has one complying trace link from each source.

Note:

- You cannot select multiple defining objects and multiple complying objects for one trace link.
- You must make multiple object selections from only one structure at a time. You cannot select an object in one structure, and then switch to another structure and select additional objects.
- To link one defining object to multiple complying objects in different structures, you must create a separate trace link to the complying objects in each structure.

Source and target objects can be:

- Elements of requirement specification structures, functional models, logical models, and physical models in Systems Engineering.

The source and target objects can reside in the same structure or in different structures.

Tip:

Defining and complying trace links can exist also between Systems Engineering objects and subsets in 4th Generation Design. Using the *cross-domain tracing* feature, collaborative designers can analyze these elements for change impact both downstream and upstream.

- Objects in other Teamcenter applications, such as My Teamcenter, Structure Manager, and Multi-Structure Manager.

Customizing trace links with subtypes

Teamcenter administrators can create custom trace link subtypes in the Business Modeler IDE. The default trace link subtype is **Trace Link**.

- Only one trace link of a given subtype is allowed between the same two objects in the same direction.
- In either direction between two objects, you can create as many trace links as the number of available trace link subtypes.

When you create a trace link, you can accept the default subtype, **Trace Link**. Or, you can assign a custom subtype, created in the Business Modeler IDE by a Teamcenter administrator.

- Custom subtypes let you create two or more trace links between the same objects in the same direction, if you assign a different subtype to each trace link.

For example, a trace link from defining requirement **A** to complying requirement **B** has the **Trace Link** subtype.

A second defining trace link, with the custom subtype **S**, can be created from requirement **A** to complying requirement **B**.

- Trace links of the same subtype are allowed in both directions between the same objects. For example:

Defining requirement **A** is linked to complying requirement **B**, with the **Trace Link** subtype assigned to the trace link.

You can create a trace link of the same subtype to link requirement **B**, as the defining object, back to requirement **A** as the complying object, completing circular trace links.

Create trace links

Trace links can be created between Teamcenter objects to support communication during the product development process. For example, creating a trace link between a requirement and a part enables the part designer to access and review the requirement content for the part they are designing.

Caution:

You cannot change a trace link after it is created. Instead, you must delete the trace link and create a new one.

To create a trace link:

1. (Optional) If you are creating a trace link on an absolute occurrence object, select the parent line in the structure and choose **Edit→Toggle In-Context Mode** or click **Toggle In-Context Mode**  on the toolbar.

Note:

- You cannot enable in-context mode for a structure line that has no children.
- Trace links created on an absolute occurrence object are related to the latest revision of the object.
- When you create a trace link on an absolute occurrence object, trace link rules are enforced on the related revision object.

2. Select the source (defining) object, or select multiple source objects by using the standard Windows functions.

Note:

If you select only one source object, you can create trace links to multiple target (complying) objects. If you select multiple source objects, you can create trace links to only one target object.

3. Choose **Tools→Trace Link→Start Trace Link** or click **Start Trace Link**  on the toolbar, or right-click the selection and choose **Start Trace Link**.
4. Select the target (complying) object, or select multiple objects by using the standard Windows functions.

Note:

If you select only one source object, you can create trace links to multiple target objects. If you select multiple source objects, you can create trace links to only one target object.

5. Specify the trace link subtype.
 - To accept the default subtype, choose **Tools→Trace Link→End Trace Link** or click **End Trace Link**  on the toolbar, or right-click the selection and choose **End Trace Link**.

The trace link is created using the default trace link type and the default trace link naming convention.

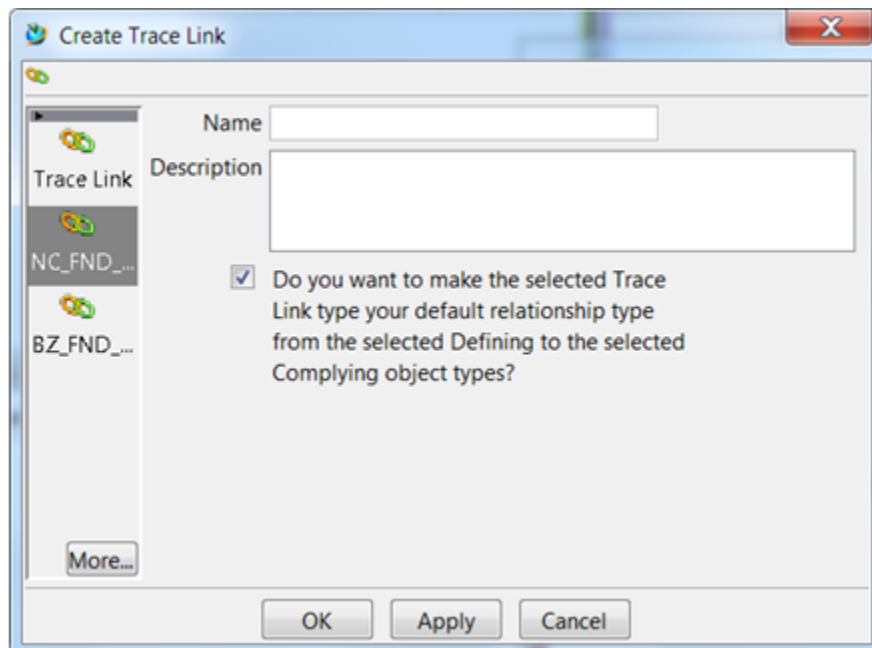
- To assign a specific trace link subtype:
 - a. Choose **Tools→Trace Link→End Trace Link....**

b. In the **Create Trace Link** dialog box:

- A. In the list of subtypes on the left, select the subtype to assign to each trace link created.
- B. (Optional) Enter the following information:

Field name	Definition	Valid values
Name	Name of the trace link.	Alphanumeric (1–128 characters)
Note: If you do not specify a name, the default naming convention is used.		
Description	Description of the trace link.	Alphanumeric (1–240 characters)

- C. (Optional) Set the subtype as the default by selecting the check box.

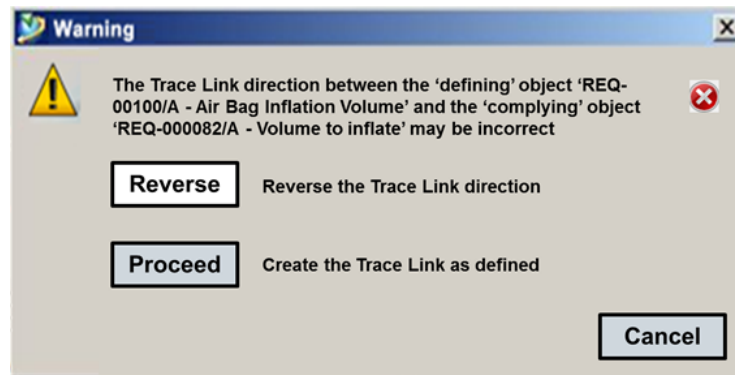


Note:

The default subtype is used automatically each time you create a trace link between the same type of objects selected as the source (defining) and target (complying) objects.

- c. Click **OK**.

If the direction of the trace link you are creating does not comply with the trace link rule, a **Warning** message box appears. For example:



Do one of the following:

- Click **Reverse**, or press Enter, to adjust the direction of the trace link. This reverses the source (defining) and target (complying) objects for the trace link.
- Click **Proceed** to create the trace link in the direction you selected for the defining and complying objects.

The trace link is created, even though it is in the opposite direction than is defined in the trace link rule for the object type pair.

- Click **Cancel** so the trace link is not created.

Note:

If you selected multiple defining or complying objects so that multiple trace links are created, a separate **Warning** message box appears for each trace link created.

Tip:

Click the help button  in the **Warning** message box for details about the trace link rule preference.

The trace link is created, and if applicable, the trace link subtype is set as the default.

A trace link symbol  is displayed for each defining and complying object. The symbol appears next to the object or in the **Trace Link** column, depending on the view. For example, the symbol appears in the **Trace Link** column in the **Requirements** view, but next to the object name in the **Trace Links** view.

Note:

- A trace link in-context symbol  appears next to the object if the trace link is created in-context.
- When you create an in context trace link to one or more packed BOM lines, the trace link can be created only on one occurrence in the selection of packed BOM lines.
- When you create an occurrence trace link on a BOM line whose **Quantity** is more than **1**, the occurrence trace link is created only on one occurrence. Subsequently when you select **Edit→Split Occurrence**, only one occurrence has and therefore displays the occurrence trace link.
- You can change the business object icon so that when an object has an associated trace link, the trace link symbol appears overlaid on the object's icon in views such as the **Requirements** view.
- Trace link symbols are not displayed for objects in My Teamcenter. To determine whether trace links exist on these objects, add the **Has Tracelink** property column in the **Details** view.

For each object in the **Details** view, the **Has Tracelink** column displays one of the following values:

True	The object has at least one trace link.
False	The object has no trace links.

You can view the defining and complying path for trace linked objects in the **Trace Links** view.

Default trace link name

When you create a trace link, the default trace link name is assigned automatically, or your can assign a specific name to the trace link.

- The default trace link name equals the name of the two trace linked objects combined, as shown below.

Complying Object	Type
 REQ-000117/A;1-Recharge Range	Requirement Revision
 Recharge Range->Voltage Low	
 REQ-000158-Voltage Low	Requirement

- If you use the default trace link name and then change the name of the defining object, the complying object, or both, the trace link name is automatically changed.

Tip:

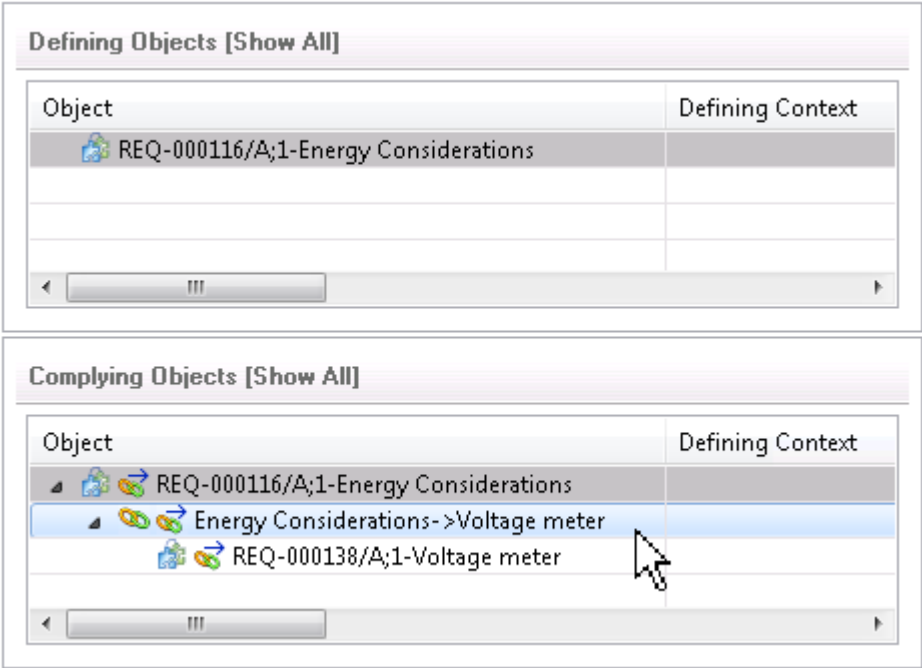
To refresh the trace link name in the **Trace Links** view:

- Select the trace link object in the view.
- Press F5, or choose **View→Refresh**.

Create a custom note on a trace link

You can add a custom note to a trace link to record information pertinent to the traceability between the linked objects.

- In Systems Engineering or Requirements Manager, open the structure that includes the trace link to which you want to add a custom note.
- In the structure, select the line with the trace link.
- Open the **Trace Links** view.
- In the **Trace Links** view, select the trace link to which you want to add the custom note.



5. Do one of the following:

- Choose **File**→**New**→**Custom Note**.
- Click **Custom Note** .
- Right-click and choose **Create Custom Note**.

6. Enter the following information.

Field name	Definition	Valid values
ID	Specifies the ID of the custom note (required).	Alphanumeric string (1–128 characters). Do one of the following: • Click Assign to automatically generate a unique ID corresponding to the naming rule. • If no naming rule is used, type a unique ID.
Revision	The revision of the custom note (required).	Alphanumeric string (1–32 characters)

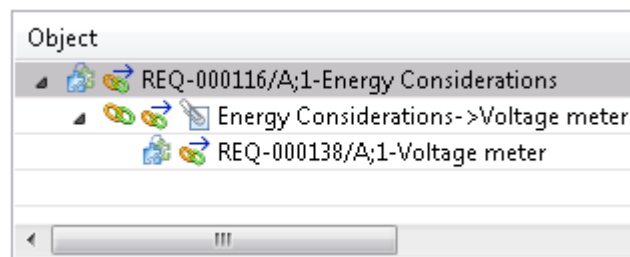
Field name	Definition	Valid values
		Click Assign to automatically assign the revision.
Name	The name of the custom note (required).	Alphanumeric string (1–128 characters)
Description	The description of the custom note (optional).	Alphanumeric string (1–240 characters)

7. (Optional) Click **Next** to define additional attribute values for the custom note.

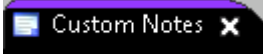
For more information about creating projects, see the *Project and Program Guide*.

8. Click **Finish** to create the custom note.

The trace link now has an attached custom note as specified by the indicator .



9. To display the custom note in the **Custom Notes** view, select the trace link with the custom note and then do one of the following:

- Click the **Custom Notes** tab .
- Click .
- Choose **Window**→**Show View**→**Other**→**Systems Engineering**→**Custom Notes**.

Copy a custom note to a trace link

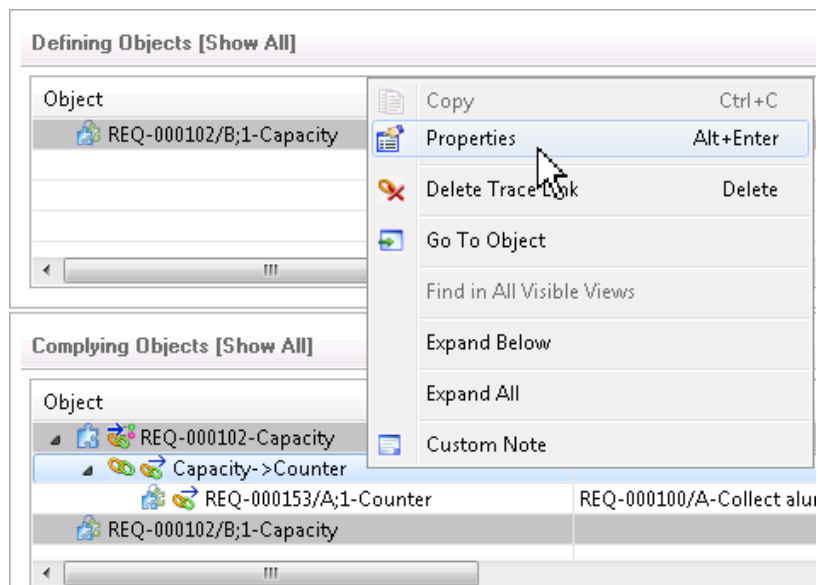
- Select the custom note you want to copy by performing one of these methods:
 - If the custom note is located in a Teamcenter folder, perform the following steps:
 - Expand the folder containing the custom note you want to copy to the trace link.
 - Right-click the custom note and choose **Copy**.

- If the custom note is attached to an object, perform the following steps:
 - a. Search and select the object that contains the custom note you want to copy to the trace link.

The custom note is highlighted in the **Custom Notes** view.

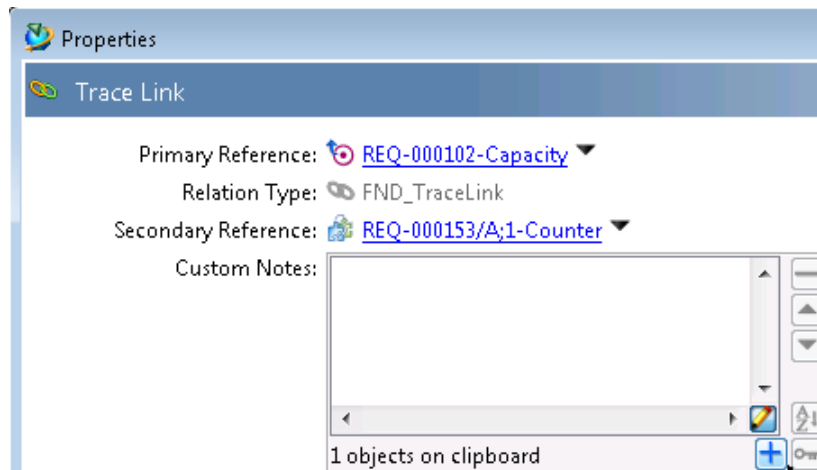
If the custom view is *not* displayed, choose **Window**→**Show View**→**Other**→**Systems Engineering**→**Custom Notes** or click **Custom Notes Manager**  on the toolbar.

- b. In the **Custom Notes** view, right-click the custom note and choose **Copy**.
2. Search for and select the object that contains the trace link where you want to copy the custom note.
 3. Display the trace link in the **Trace Links** view.
 - Right-click the object and then choose **Open with**→**Trace Links View**.
 4. In the **Trace Links** view, select the trace link where you want to copy the custom note, right-click and choose **Properties**.

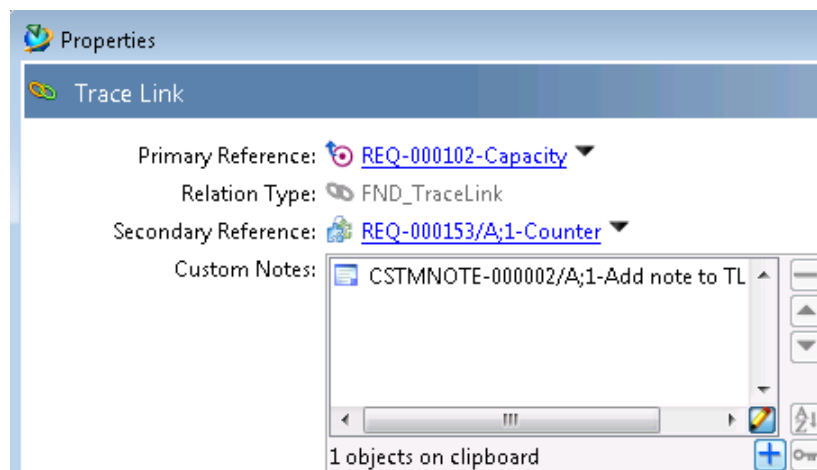


5. In the **Properties** dialog box, scroll to display the **Custom Notes** property for the trace link.
6. In the **Custom Notes** property box, click the  button in the lower right corner.

The add button  is now displayed in the lower right corner of the **Custom Notes** property box.

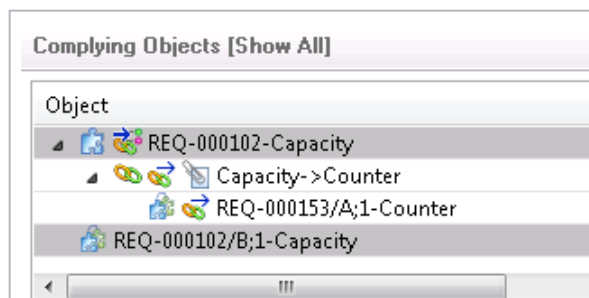


7. Click the add button **+** to add the custom note to the trace link.



8. Click **OK**.

The trace link now has an attached custom note.



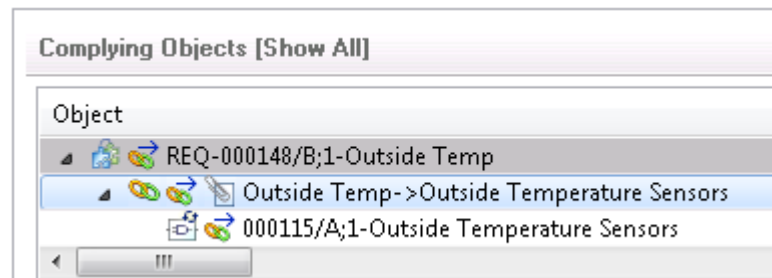
Tip:

To refresh the view so you can see the attachment symbol , press F5 or click  or choose **View→Refresh Window**.

Remove a custom note from a trace link

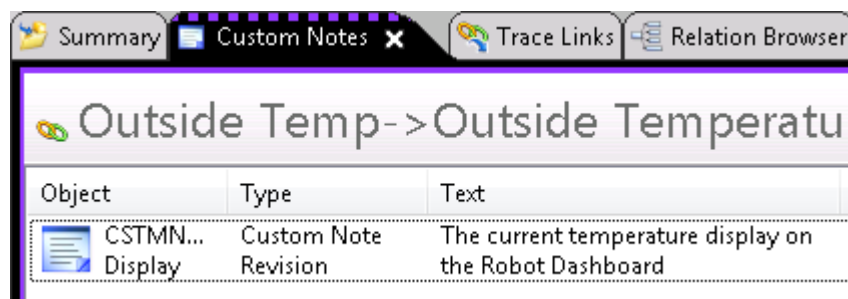
1. In Systems Engineering or Requirements Manager, open the structure that includes the trace link from which you want to remove a custom note.
2. In the structure view, select the line with the trace link.
3. Open the **Trace Links** view.

The attachment symbol  appears on any trace links with attached notes.

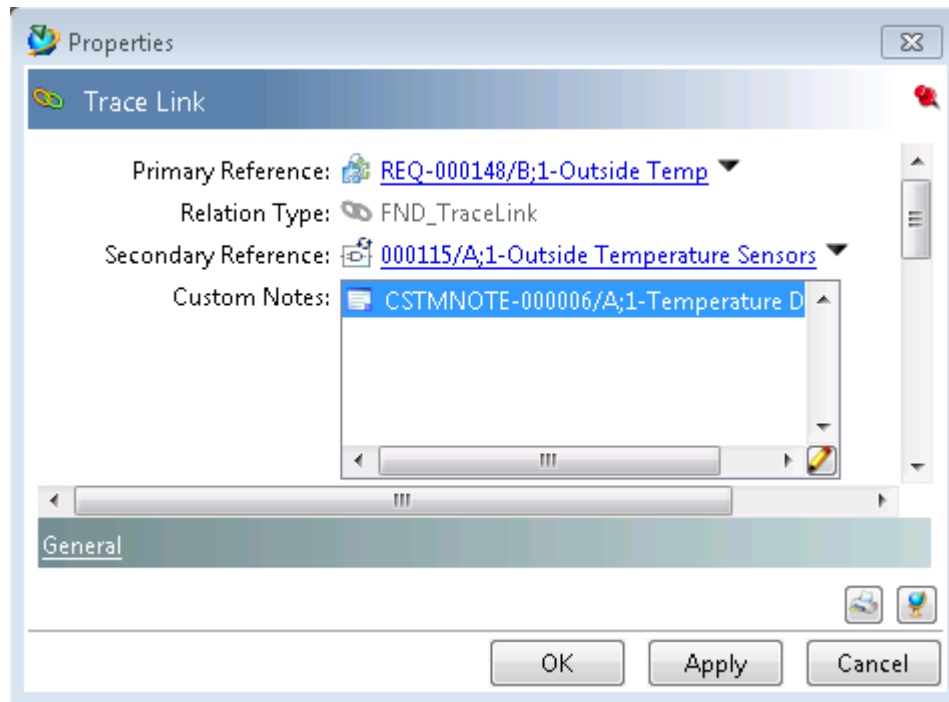


You can use the **Custom Notes** view to review the custom notes attached to the trace links.

If the **Custom Notes** view is *not* displayed, click **Custom Notes Manager**  on the toolbar.



4. In the **Trace Links** view, select the trace link object with the custom note attached, and then click **Properties** .
5. In the **Properties** dialog box, scroll to display the **Custom Notes** property for the trace link.
6. Select the custom note you want to remove.

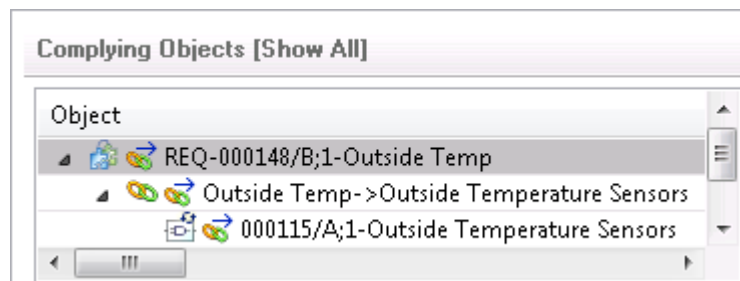


7. Click **Expand to modify**  on the lower right corner of the **Custom Notes** box.
8. Click **Remove selected objects**  to remove the custom note from the trace link.
9. Click **OK**.

The custom note is removed from the trace link.

Tip:

Select the trace link object in the **Trace Links** view and then press F5 to refresh the view display.



Delete trace links

Caution:

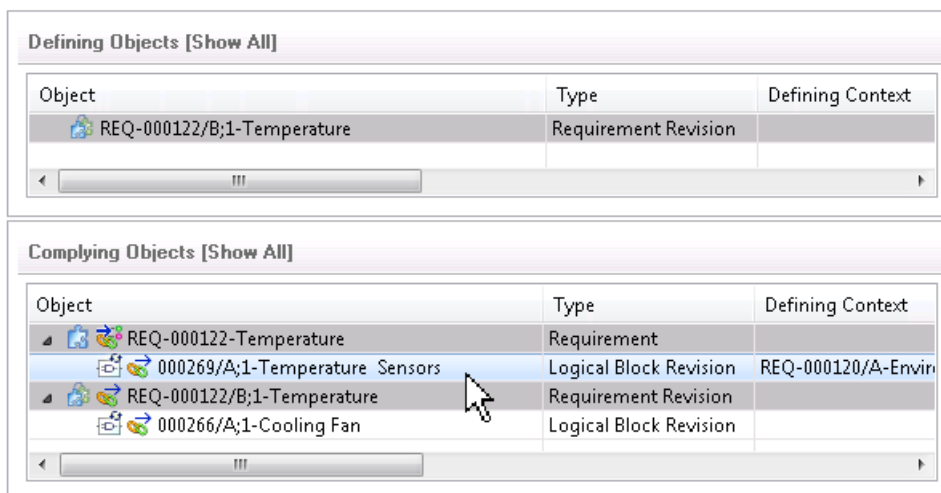
If the defining or complying object for a trace link you attempt to delete is included in a baseline structure, you may be prevented from deleting the trace link. If this is the case, an error message appears. The message is based on the value set for the **Fnd0PreventTraceLinkDelete** business object constant.

The constant setting prevents deletion of trace links to the released object revision/ it does not prevent deletion of trace links to other occurrences of the object. Therefore, you can still delete trace links on other object occurrences in the baseline structure.

In case of trace links on occurrences, if the context line BOM view revision is released, an error message is displayed according to the above values of the **Fnd0PreventTraceLinkDelete** constant.

You can use the following procedure to delete trace links from one or more occurrence or absolute occurrence objects at a time.

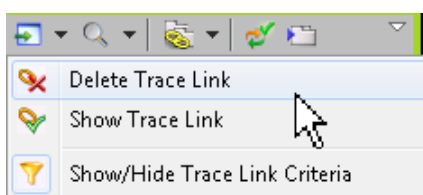
1. In the **Trace Links** view, select the trace link to delete.



Note:

For absolute occurrence objects, you must select the trace link in the correct context.

2. Click **Delete Trace Link**  on the view toolbar, or choose **Delete Trace Link** from the view menu.



3. In the **Delete** dialog box, click **Yes**.

If the trace link has custom notes, the **Delete** dialog states the trace link has custom notes.

The trace link is deleted from the latest revision of the occurrence or absolute occurrence object.

Tip:

If trace link icons are not removed from the primary view, refresh the icon display by doing one of the following:

- On the **Trace Links** view toolbar, click **Associate this 'Secondary' view to a different 'Primary' view** , and then select another primary view.
- Close the **Trace Links** view, and then open a new one.
You can select an object in the primary view and choose **View → Show Trace Links**.
Or, you can double-click the icon in the **Trace Link** column for an object.
- Press F5 to refresh the **Trace Links** view.

Viewing traceability relationships

Viewing downstream and upstream paths

There are several methods for displaying the trace link assignments that you make among and between structures:

- In the **Trace Links** view, you can view the direct and indirect predecessors or successors in an upstream or downstream path. You can also view the trace links themselves, as separate objects.
- The **Traceability Matrix** view displays the assignments between two selected structures.
- You can run an accountability check to compare two structures to ensure all lines in the source structure are consumed in the target structure.

For example, you can compare a requirement structure against a product structure, or compare a requirement structure against a manufacturing process structure.

- In the Teamcenter **Relation Browser** view, you can view the trace link hierarchies in a requirement structure.
- In the requirement's **Properties** dialog box, you can view trace link assignments if the **Calculate Defining/Complying Objects** option is selected.

Tip:

Defining and complying trace links can exist also between Systems Engineering objects and subsets in 4th Generation Design. Using the *cross-domain tracing* feature, collaborative designers can analyze these elements for change impact both downstream and upstream.

What does the Trace Links view provide?

You can display traceability information using the **Trace Links** view. You can also display traceability information using the **Traceability** view, which is used by different applications such as Manufacturing Process Planner or Structure Manager. However, working in the **Trace Links** view, you can tightly control the displayed traceability information by setting expansion levels and filtering traceability information by scope, object type, trace link type, and property.

Use the **Trace Links** view to perform the following tasks:

Note:

Some of these tasks can also be performed in the **Traceview Settings** dialog box, which appears when you open the **Trace Links** view.

- Set the traceability scope.
- Filter traceability information by configured structure, object type, trace link type, and property.
- Set traceability expansion levels.
- Set the traceability view to display defining objects, complying objects or both.
- View defining and complying relationships.
- Check requirement structures against product and process structures.
- Navigate to a linked object.
- Delete a trace link.
- Export traceability information to Excel.
- Find the selected trace link in all visible views.

Open the Trace Links view

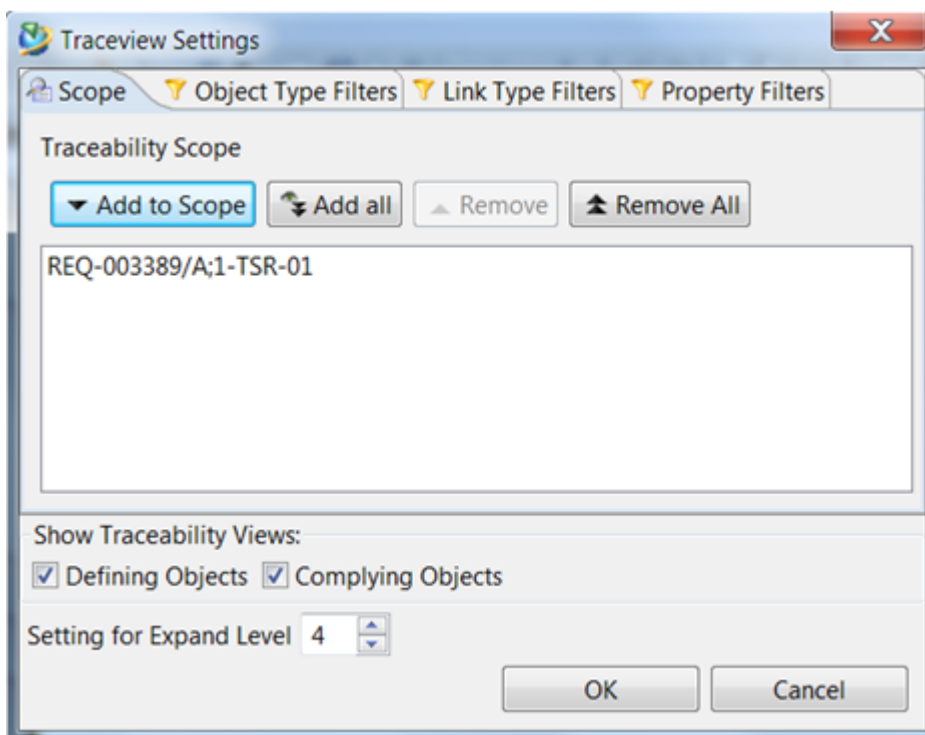
1. From the navigation pane, click **Systems Engineering**.

2. Open the requirement, logical, or functional structure that contains trace links.
3. Select the requirement object containing the trace link.
4. Do one of the following:
 - Click **Trace Links View**  on the toolbar.
 - Choose **View→Show Trace Links....**

The **Traceview Settings** dialog box opens. You can use this dialog box to set filters to limit the display of trace links associated with the selected object.

Note:

When you click **OK** in the **Traceview Settings** dialog box, the **Trace Links** view opens. The same filtering capabilities are available in the **Trace Links** view.

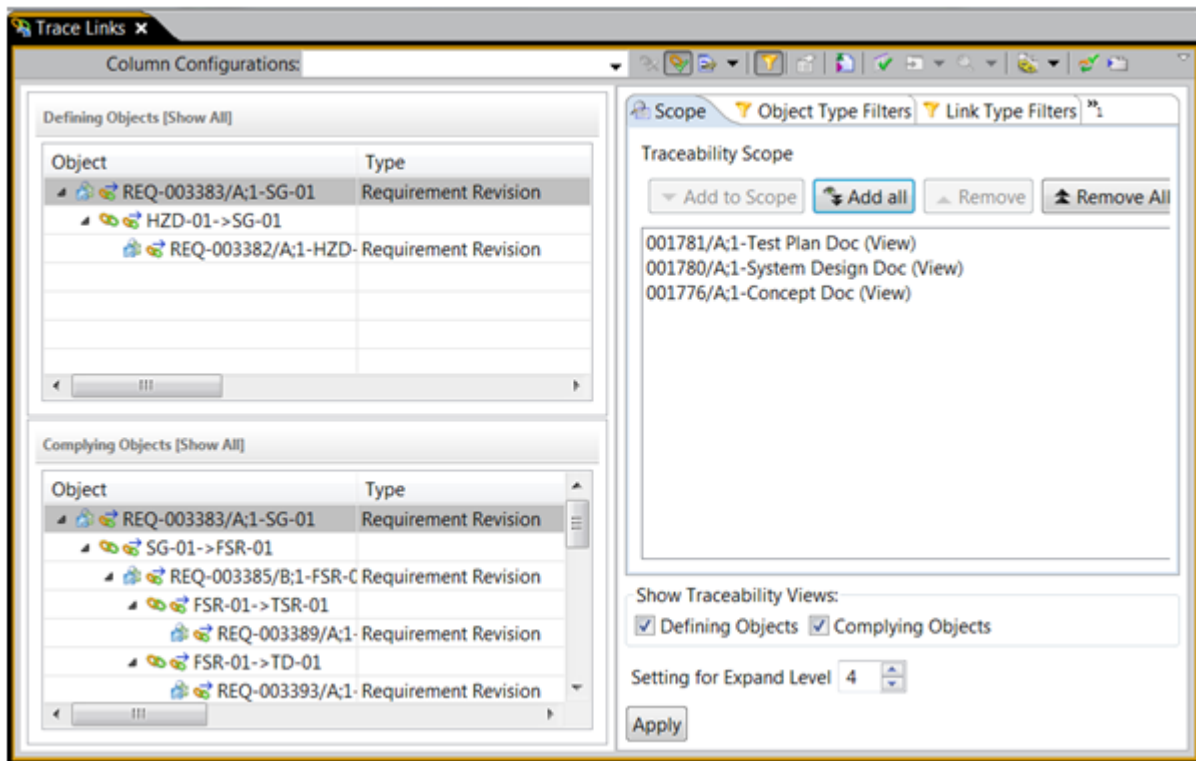


Note:

If you do not set the scope, Teamcenter queries for all trace link objects for the selected requirement object. A warning message states that no scope is set and system performance issues may occur.

5. In the **Traceview Settings** dialog box, click **OK**.

The **Trace Links** view opens.



6. To display different or additional objects containing trace links, select those objects.

Note:

To refresh the **Trace Links** view, click the **Apply** button at the bottom of the view, press the F5 key or choose **View→Refresh Window**.

Column settings in Trace Links view

You can use these preferences to set how columns appear in the Trace Links view:

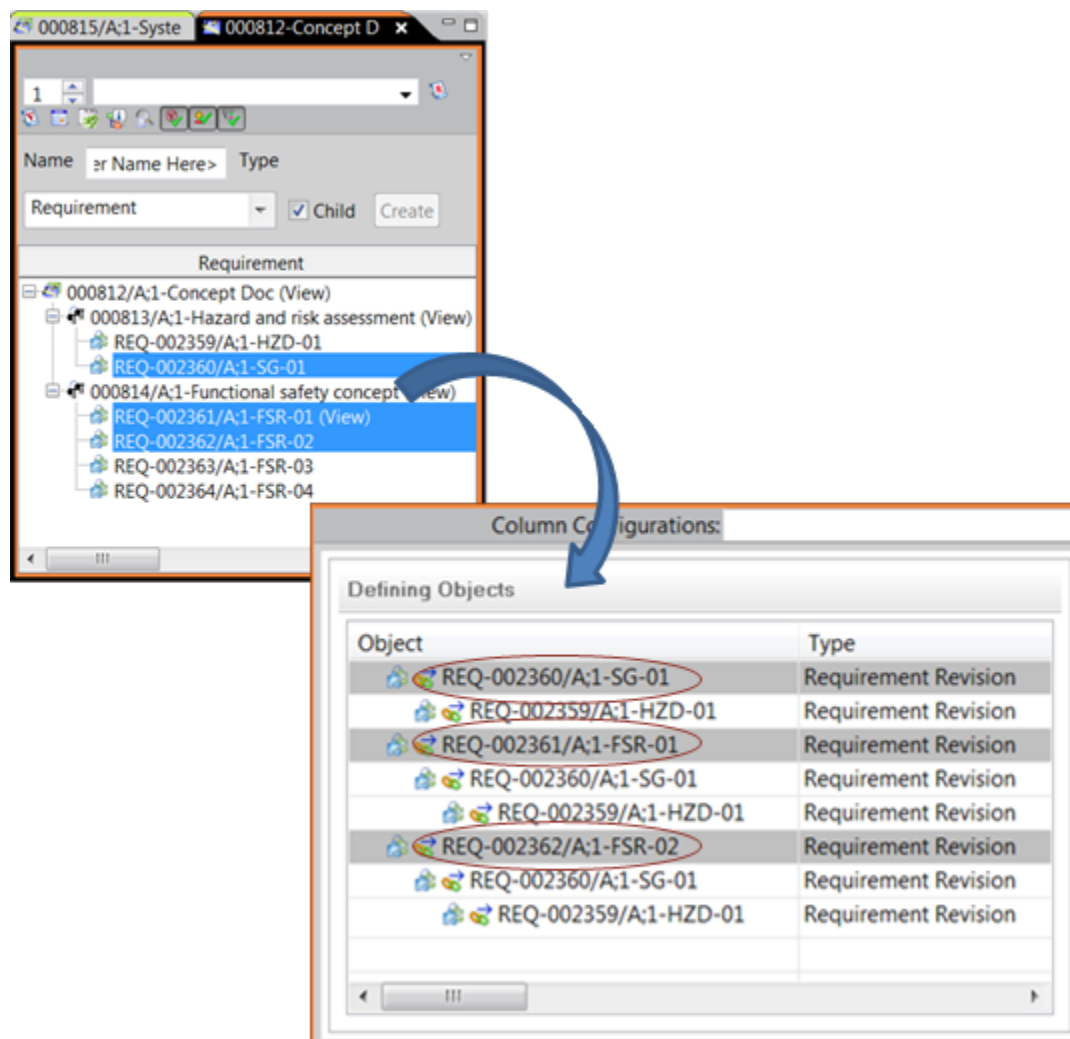
- The **TraceReportTable_ColumnPreferences** preference specifies the list of default displayed columns in the **Trace links** view.
- The **TraceReportTable_ColumnWidthPreferences** preference specifies the default column width of the displayed columns in the **Trace links** view.
- The **TraceReport_Default_Sort_Property** specifies the default property used for sorting.

How certain objects appear in the Trace Links view

Root nodes

When viewing traceability information for multiple objects in the **Trace Links** view, the root nodes are highlighted to differentiate and identify the root lines.

This example shows the complying traceability information for the **REQ-002360/A;1-SG-01**, **REQ-002361/A;1-FSR-01**, and **REQ-002362/A;1-FSR-02** requirements. The root nodes are highlighted in gray in the **Trace Links** view.

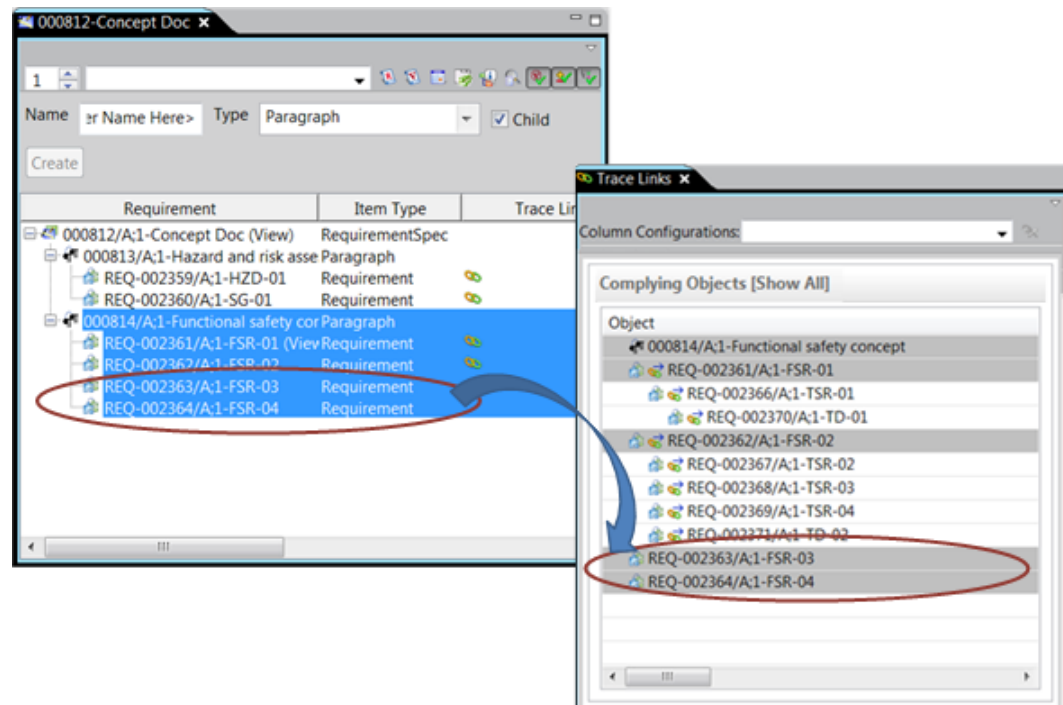


Objects with no traceability

When a requirement object has no complying or defining trace links, the object appears with no children and no trace link symbol.

Example:

This example shows the **REQ-002263/A;1-FSR-03** and **REQ-002264/A;1-FSR-04** requirements containing no defining or complying trace links.



Objects with no read access

Teamcenter supports access control on requirement information to support company access rules. If you do *not* have read access to requirement information containing trace links, the defining or complying trace link appears as **<<UNREADABLE>>** in the **Trace Links** view.

This example shows a complying trace link displayed as **<<UNREADABLE>>** for a requirement where you have no read access. The trace links for requirement information where you do have the appropriate access, display normally.

Complying Object

Object	Type
REQ-002359/A;1-HZD-01	HZD Revision
REQ-002360/A;1-SG-01	SG Revision
<<UNREADABLE>>	
REQ-002361/B;1-FSR-01	FSR Revision
REQ-002366/B;1-TSR-01	TSR Revision

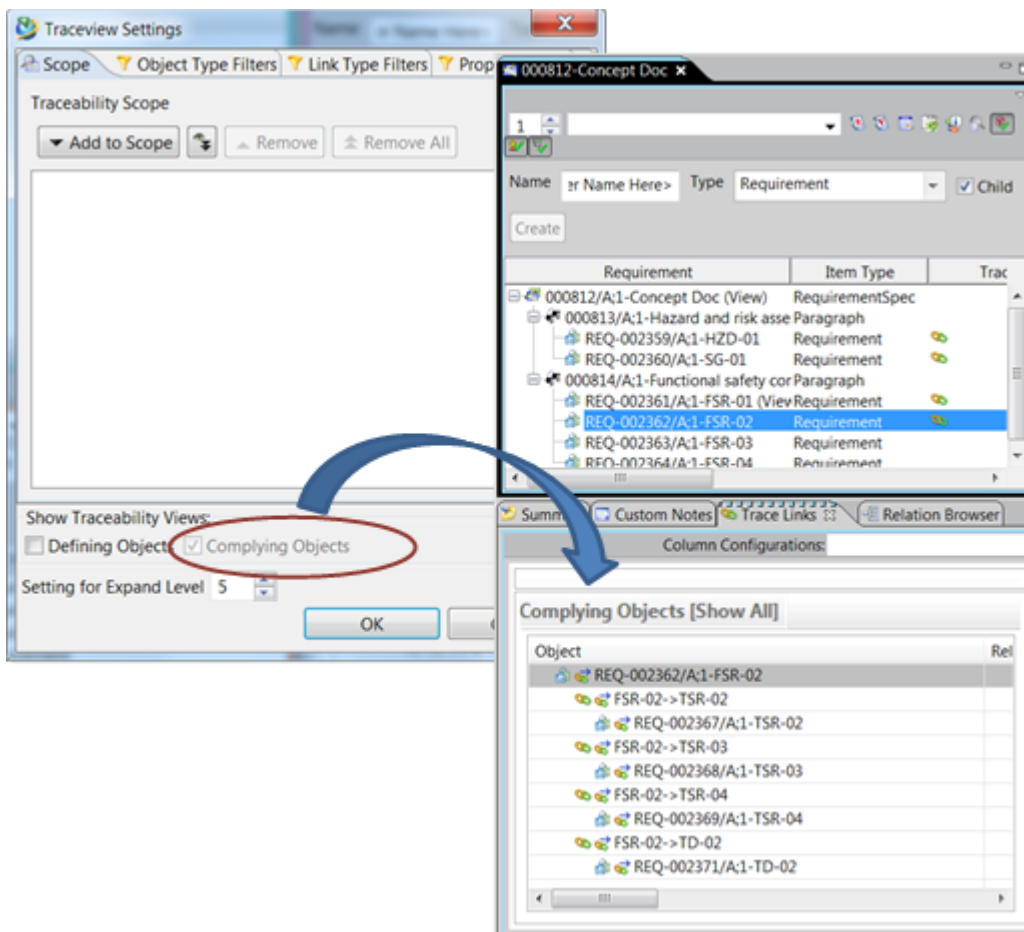
View defining and complying relationships

1. Open the **Trace Links** view for the selected requirement objects.
2. In either the **Trace Links** view or the **Traceview Settings** dialog box that opens first, perform any of the following under **Show Traceability Views**:
 - Select the **Defining Objects** check box to display the defining objects or clear the check box to hide the defining objects.
 - Select the **Complying Objects** check box to display the complying objects or clear the check box to hide the complying objects.

The resulting objects are displayed.

The *direct* trace link symbol  indicates that the trace link relates to the selected object itself.

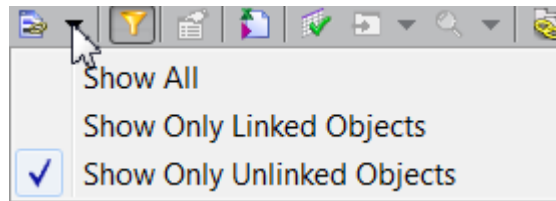
The *indirect* trace link symbol  indicates that the trace link relates to the revision or item that is associated with the selected object.



Display linked and unlinked objects in the Trace Links view

When you view the traceability information for a requirement object in the **Trace Links** view, you can view the complying and defining objects that are linked, not linked, or both.

1. Display the **Trace Links** view for the selected requirement objects.
2. From the **Orphan Report**  list, choose the objects you want to display in the view.



View the context of trace links on absolute occurrence objects

When you view the traceability information for a trace link on an absolute occurrence object in the **Trace Links** view, you can see the context in which the trace link was created.

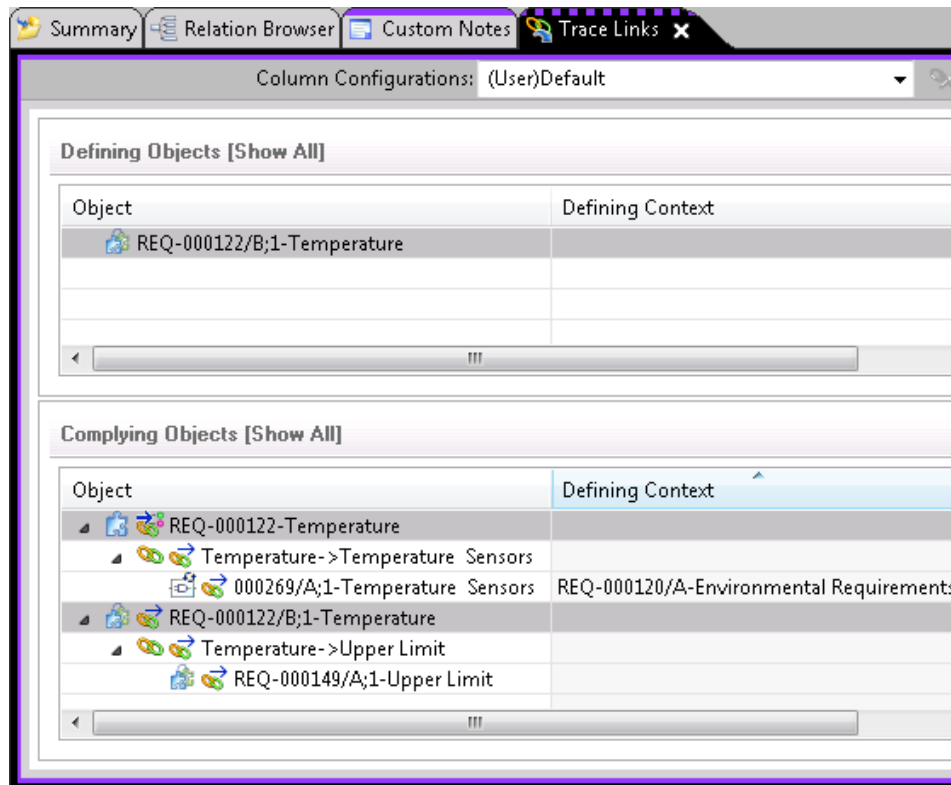
1. Select the line in the structure for the absolute occurrence object with trace links.

Tip:

The symbol for trace links created in context is .

Function #1 000096-Functional Requirements for Robot 1		
1		
Item Id	Rev Name	Trace Link
REQ-000118	Operating Environment	
REQ-000119	Operating Environment	
REQ-000122	Temperature	
REQ-000120	Environmental Requirements	
REQ-000121	Environmental Impact	
REQ-000122	Temperature	
REQ-000123	Material Requirements	

2. Open the **Trace Links** view.
3. Adjust the size of the property columns in the **Trace Links** view so that you can see the values in the **Object**, **Defining Context**, **Complying Context**, and other property columns of interest.

**Note:**

If you set the scope for the trace links in the **Traceview Settings** dialog box, only the trace links within the defined scope appear in the **Trace Links** view.

- Review the information in the **Defining Objects** and **Complying Objects** sections of the **Trace Links** view to determine which trace links on the object were created in-context and the context in which they were created.

For example, the trace link between the **REQ-000122-Temperature** requirement object and the **000269/A;1-Temperature Sensors** logical block object was created in context of the **REQ-000120/A-Environmental Requirements** substructure, as shown in the following image.

Object	Defining Context
REQ-000122-Temperature	
Temperature->Temperature Sensors	
000269/A;1-Temperature Sensors	REQ-000120/A-Environmental Requirements

Whereas the trace link between the **REQ-000122/B;1-Temperature** requirement object and the **REQ-000149/A;1-Upper Limit** requirement object was not created in-context, as shown below.

Note:

- When the trace link is to an absolute occurrence object, the object opens in its default structure application.

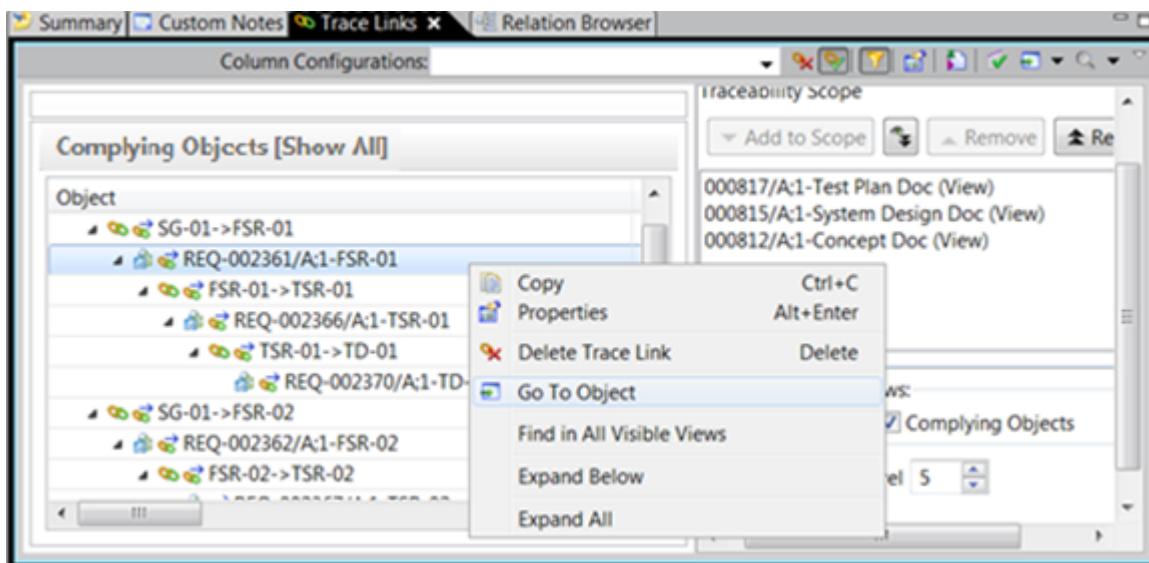
The structure is partially expanded from the absolute occurrence object to the parent context, with no other branches displayed.

- If no default structure application is identified for an absolute occurrence object, it opens in the **Requirements** view of the **Systems Engineering** perspective.

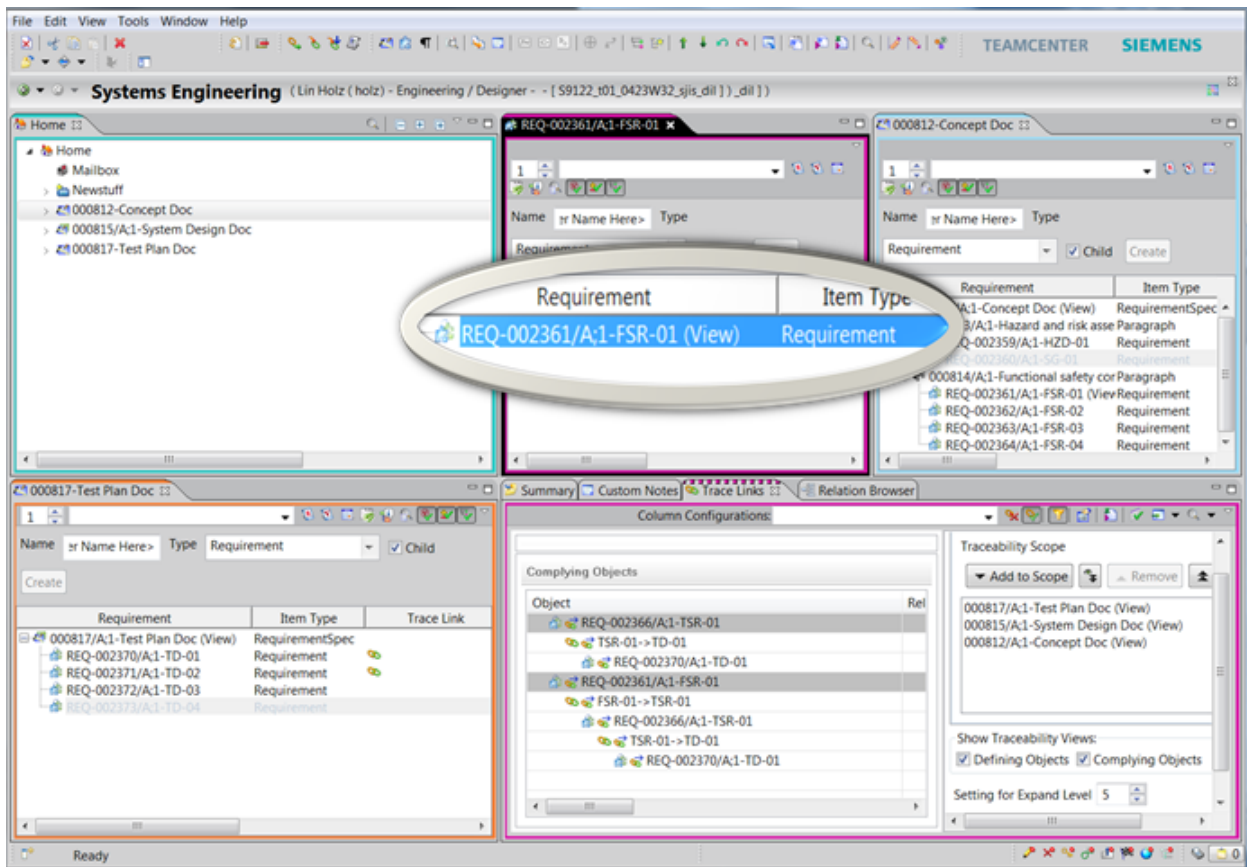
Open a parent requirement in the Trace Links view

You can select a requirement from the **Trace Links** view and the requirement opens as the root node in a new view.

- Display the **Trace Links** view for the selected objects.
- Right-click the object containing the trace link and choose **Go To Object**.



A new view is displayed with the requirement object as the root node.



Filter traceability information by configured structure or element

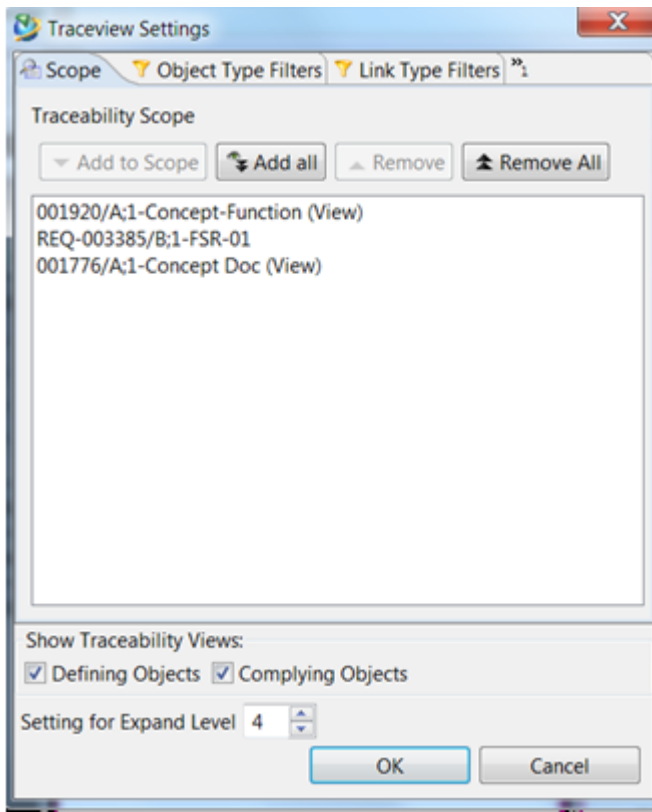
You can filter traceability information by configured structures in the **Trace Links** view. You configure the scope to search for trace links within specified open requirement structures or subsections of the structure, such as any child requirement structure or requirement object. The search scope is then limited to only displaying traceability information for the specified structures or elements for that revision. Search scope settings are preserved for any subsequent Teamcenter sessions as long as you have not cleared your Teamcenter client cache.

Note:

If you do not set the scope, Teamcenter queries for all trace link objects for the selected requirement object. A warning message states that no scope is set and system performance issues may occur.

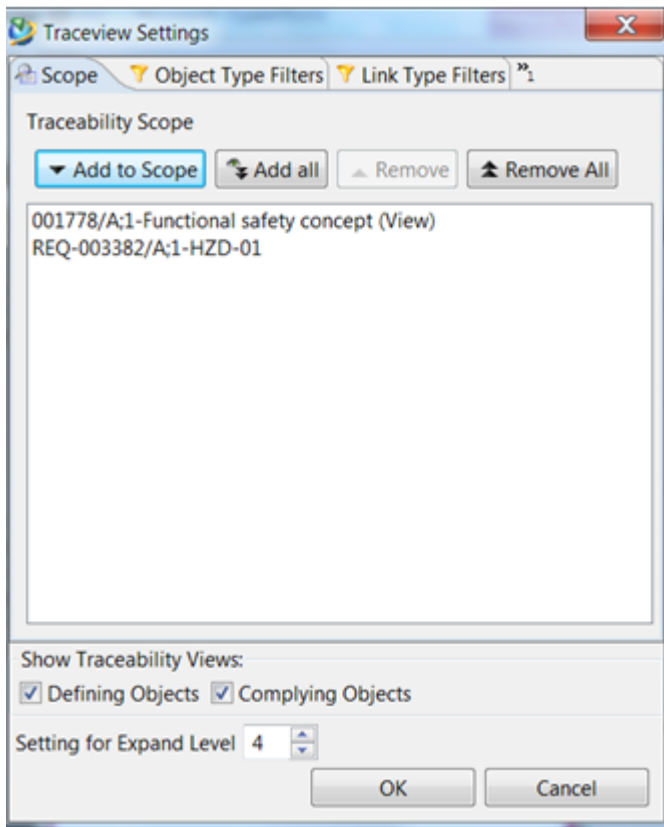
1. Display the **Trace Links** view for the selected requirement objects.
2. In either the **Trace Links** view or the **Traceview Settings** dialog box that opens first, click the **Scope** tab.
3. To add all opened structures to the scope, click **Add All**.

All opened structures are used for filtering.



4. To add child structures or requirement objects in any opened structure or a snapshot folder to the scope, select the requirement object, and click **Add to Scope**.

The selected requirement structure or requirement object is used for filtering.



5. To remove all the objects from the scope, click **Remove All**.

All the objects are removed from the scope.

6. To remove selected objects from the scope, select the objects you want to remove, and click **Remove**.

The selected objects are removed from the scope.

7. Click **OK** if you are displaying the **Traceview Settings** dialog box, or click **Apply** if you are displaying the **Trace Links** view.

Note:

To refresh the **Trace Links** view, press the F5 key or choose **View→Refresh Window**.

Filter traceability information by trace link type

You can filter traceability information by trace link type in the **Trace Links** view. The **Trace Links** view displays only the traceability information with the specified link types. You select the trace link types for filtering from the list of available trace link types. The list of available trace link types for filter selection is specified by the **TraceReport_Filter_Available_Tracelink_Types** preference. Trace link type filter settings

are preserved for any subsequent Teamcenter sessions as long as you have not cleared your Teamcenter client cache.

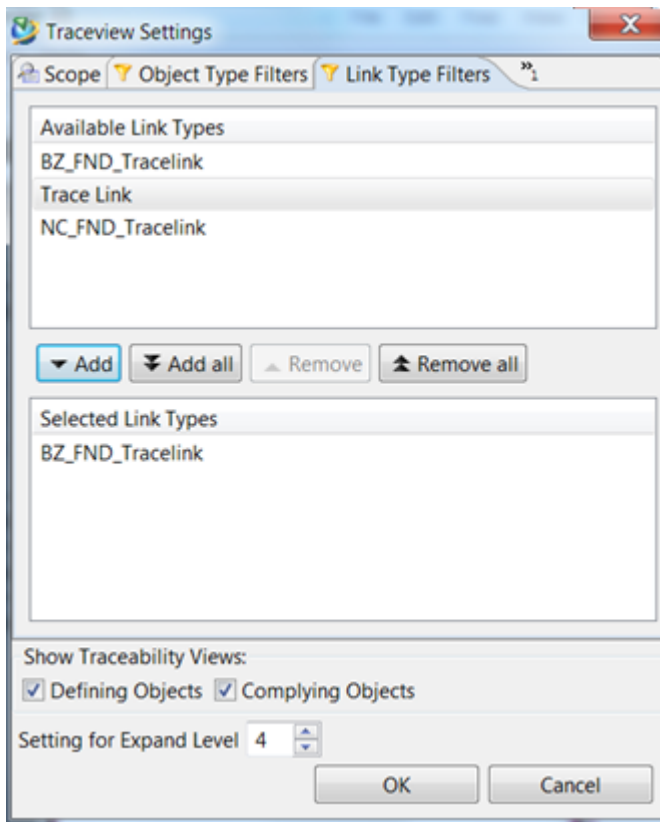
For multiple **Trace Links** views, you can use different filter values for different views. For example, one **Trace Links** view displays the traceability information for requirement objects containing **BZ_FND_Tracelink** trace link types, while another **Trace Links** view displays the traceability information for all the requirement objects containing **NC_FND_Tracelink** trace link types.

Note:

If a parent requirement object does not match the filter, none of the children are displayed for the structure, even if some of the child objects match the filter condition.

1. Display the **Trace Links** view for the selected requirement objects.
2. In either the **Trace Links** view or the **Traceview Settings** dialog box that opens first, click the **Link Type Filters** tab and perform one of the following:
 - Select the link types under the **Available Link Types** heading you want to add for filtering and click **Add**.
 - Select the link types under the **Available Link Types** heading you want to remove from filtering and click **Remove**.
 - Click **Add all** to add all the available link types for filtering.
 - Click **Remove all** to remove all the selected link types for filtering.

The link types used for filtering are displayed under the **Selected Link Types** heading.



- Click **OK** if you are displaying the **Traceview Settings** dialog box, or click **Apply** if you are displaying the **Trace Links** view.

Note:

To refresh the **Trace Links** view, press the F5 key or choose **View→Refresh Window**.

The **Trace Links** view displays defining and complying traceability information for only those requirement objects containing the selected trace link types.

Complying Object	Relation Type
 REQ-002359/A;1-HZD-01  HZD-01->SG-01  REQ-002360/A;1-SG-01  SG-01->FSR-02  REQ-002362/A;1-FSR-02	BZ_FND_Tracelink BZ_FND_Tracelink

Filter traceability information by object type

You can filter traceability information by object type in the **Trace Links** view. The **Trace Links** view displays only the traceability information for requirement information with the specified object types. You select the object types for filtering from the list of available object types. The list of available object

types for filter selection is specified by the **TraceReport_Filter_Available_Object_Types** preference. Object-type filter settings are preserved for any subsequent Teamcenter sessions as long as you have not cleared your Teamcenter client cache.

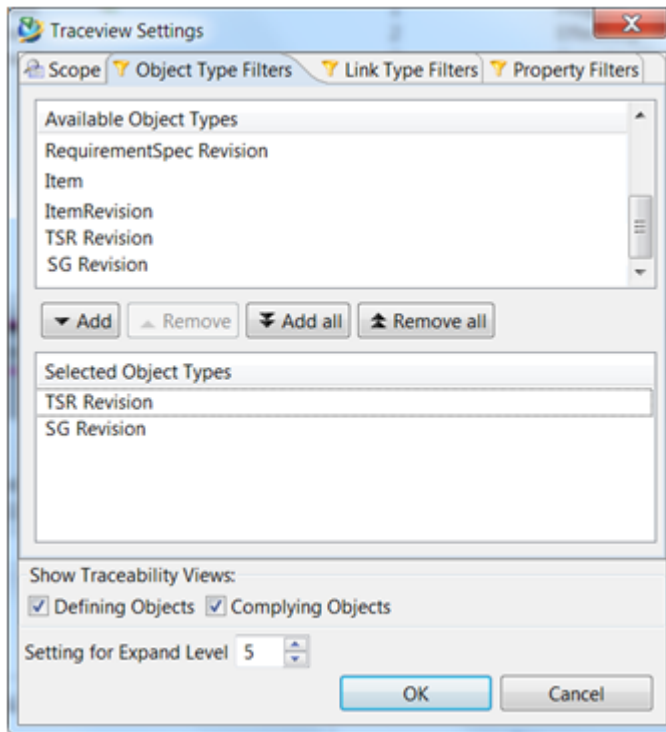
For multiple **Trace Links** views, you can use different filter values for different views. For example, one **Trace Links** view displays all the traceability information for requirement objects that are *requirement* types, while another **Trace Links** view displays all the traceability information for requirement objects that are *paragraph* types.

Note:

If a parent requirement object does not match the filter, none of the children are displayed for the structure, even if some of the child objects match the filter condition.

1. Display the **Trace Links** view for the selected requirement objects.
2. In either the **Trace Links** view or the **Traceview Settings** dialog box that opens first, click the **Object Type Filters** tab and perform one of the following:
 - Select the object types under the **Available Object Types** heading you want to add for filtering and click **Add**.
 - Select the object types under the **Available Object Types** heading you want to remove from filtering and click **Remove**.
 - Click **Add all** to add all the available object types for filtering.
 - Click **Remove all** to remove all the selected object types for filtering.

The object types used for filtering are displayed under the **Selected Link Types** heading.



- Click **OK** if you are displaying the **Traceview Settings** dialog box, or click **Apply** if you are displaying the **Trace Links** view.

Note:

To refresh the **Trace Links** view, press the F5 key or choose **View→Refresh Window**.

The **Trace Links** view displays defining and complying traceability information for only those requirement objects containing the selected object types.

Complying Object

- REQ-002360/A;1-SG-01
- REQ-002361/A;1-TSR-01
- REQ-002362/A;1-TSR-02

Type

- SG Revision
- TSR Revision
- TSR Revision

Filter traceability information by property

You can filter traceability information by any *visible* column property in the **Trace Links** view. The **Trace Links** view displays only the traceability information for requirement information with the specified properties. You can use Boolean operations to filter multiple properties. Property filter settings are preserved for any subsequent Teamcenter sessions as long as you have not cleared your Teamcenter client cache.

For multiple **Trace Links** views, you can use different filter values for different views. For example, one **Trace Links** view displays all the traceability information for requirement objects at a release status, while another **Trace Links** view displays all the traceability information for requirement objects modified at a specific date.

Note:

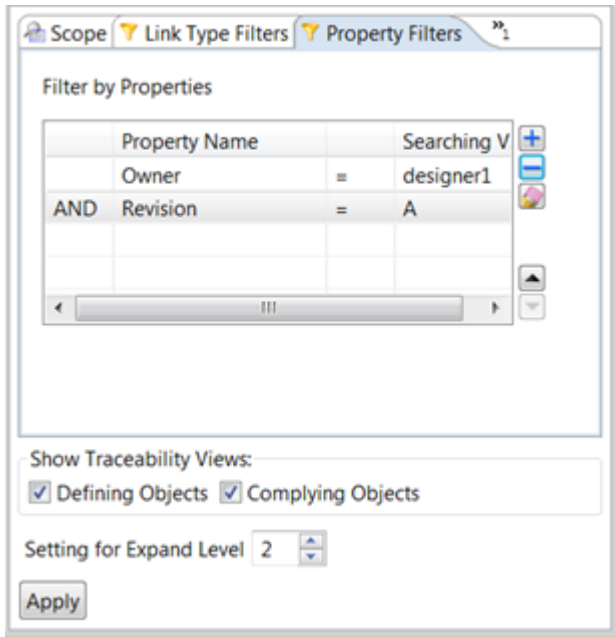
If a parent requirement object does not match the filter, none of the children are displayed for the structure, even if some of the child objects match the filter condition.

1. Display the **Trace Links** view for the selected requirement objects.
2. In either the **Trace Links** view or the **Traceview Settings** dialog box that opens first, click the **Property Filters** tab.
3. In you are in the **Traceview Settings** dialog box, select the column configuration you want to use in the **Columns Configuration** box.
4. Choose one of the following:
 - To add a property, click  to add a line to the property filter.
 - a. Click the field under the **Property Name** heading.
 - b. Click  and select the property you want to filter.

Note:

The property column must be a visible column.

- c. Click the field under the **Searching Value** heading and type the property value you want to filter.
- d. To add another property filter, click  to add a line to the property filter.
- e. Click the first field, select **AND** or **OR** to create the Boolean operation and repeat steps a-c.



- To remove a property filter, select the property filter and click .
 - To remove all property filters, click .
 - To change the order of the property filters, select the property filter and click  or .
5. Click **OK** if you are displaying the **Traceview Settings** dialog box, or click **Apply** if you are displaying the **Trace Links** view.

Note:

To refresh the **Trace Links** view, press the F5 key or choose **View→Refresh Window**.

The **Trace Links** view displays defining and complying traceability information for only those requirement objects with those specified properties.

Defining Object	Owner	Revision
 REQ-002359/A;1-HZD-01	Designer1	A
 REQ-002360/A;1-SG-01	Designer1	A
 REQ-002362/A;1-FSR-02	Designer1	A

Set expansion levels in the Trace Links view

You can set the number of expansion levels of traceability information that appear in the **Trace Links** view. You can change the default setting for the expansion level by modifying the

TraceReport_Default_Expansion_Level preference. Once you set the expansion level, that same value is used when displaying the traceability information for any requirement object and is preserved for any subsequent Teamcenter sessions as long as you have not cleared your Teamcenter client cache.

Note:

If the expansion levels exceed the value of the **FND_Tracelink_Report_Depth** preference, you receive a warning message indicating the expansion may take a long time to complete.

1. Open the **Trace Links** view for the selected requirement objects.
2. To change the expansion level, perform any of the following:
 - Right-click the node in the **Trace Links** view and choose **Expand Below** to expand the levels set to the specified expansion level below the selected node.
 - Right-click the top node in the **Trace Links** view and choose **Expand All** to expand all levels below the top node.
 - To set the expansion level, in the **Setting for Expand Level** box, click  to increase the expansion level or click  to decrease the expansion level.

The traceability information is expanded to the specified number of levels.

**Complying
Object**

-  REQ-002360/A;1-SG-01
-  REQ-002361/B;1-FSR-01
-  REQ-002361/B;1-FSR-01

Type

- SG Revision
- FSR Revision
- FSR Revision

Highlighting requirement objects from the Trace Links view

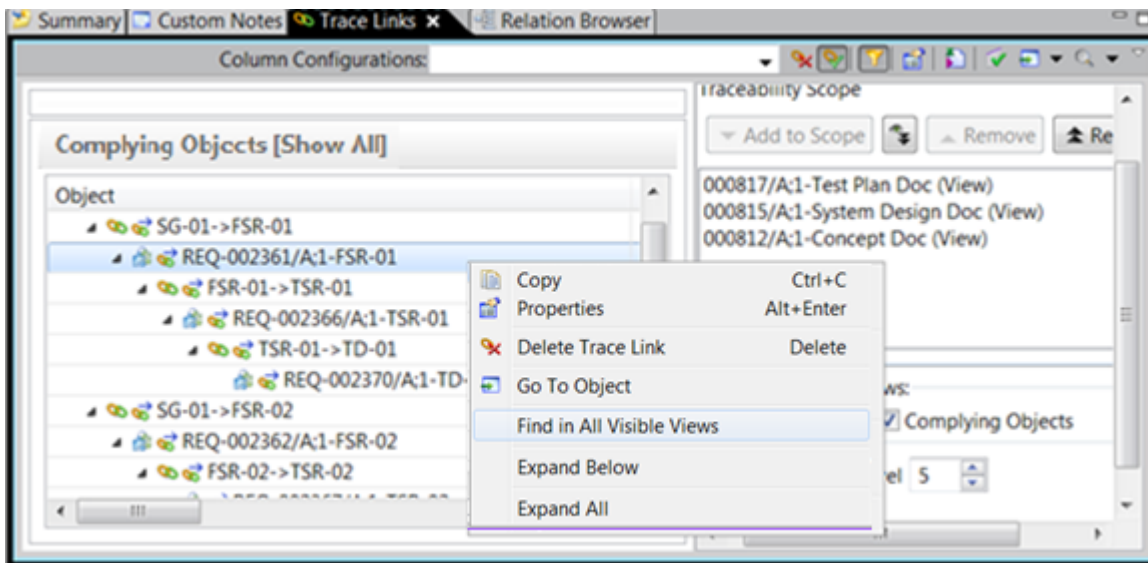
You can highlight the requirement objects in the requirement structure views by using the **Trace Links** view.

The following conditions apply:

- The first occurrence of the requirement object found in the scoped views appears highlighted.
- If the matching requirement object is hidden, the requirement object is made visible in the requirement structure and the matching requirement object appears highlighted.
- If there is no matching occurrence found in the scoped views, a warning message indicates that no matching occurrence was found in any visible views for the selected object.

1. Open the **Trace Links** view for the selected requirement objects.

- Right-click the requirement object containing the trace link and choose **Find in all Visible Views**.

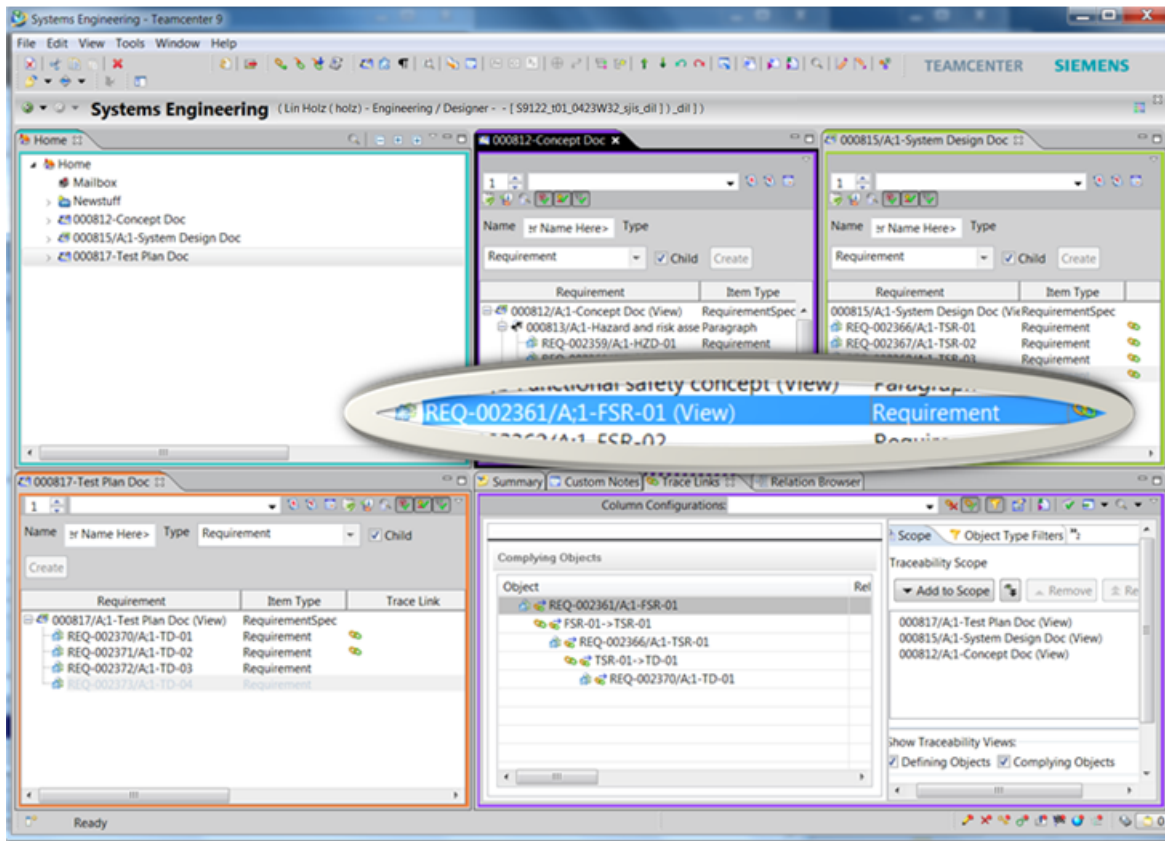


Note:

The **Find in all Visible Views** command is unavailable if:

- The selected object is the trace link itself.
- You select more than one object.

The objects are highlighted in the structure.



Set scope from custom dialog box for the Trace Links view

You can set the scope for the **Trace Links** view by adding a custom button to the **Scope** tab and using the `ITraceLinkViewSettingCriteria` interface.

The `ITraceLinkViewSettingCriteria` interface is composed of the following APIs:

- `setTraceViewScope`

```
/**
 * Set the trace view scope with input objects.
 *
 * @param scopeObjects
 * @published
 */
public void setTraceViewScope( TComponent[] scopeObjects );
```

- `getTraceViewScope`

```
/**
 * Get the trace view scope.
 */
```

```

* @return
* @published
*/
public TCComponent[] getTraceViewScope();

```

- **addToTraceViewScopeList**

```

/**
 * Add selected components to the source list.
 *
 * @param sourceObjects
 * @published
 */
public void addToTraceViewScopeList( TCComponent[] sourceObjects );

```

- **removeFromTraceViewScopeList**

```

/**
 * Remove a list of components from the source list.
 *
 * @param sourceObjects
 * @published
 */
public void removeFromTraceViewScopeList( TCComponent[]
sourceObjects );

```

Reporting traceability relationships

Generate a traceability matrix

You can generate the **Traceability Matrix** by filtering specific trace link types. You select the trace link types you want to filter in the **Show Traceability Matrix** dialog box. The filter determines the types of trace links you can *view*, *create*, or *delete* from the **Traceability Matrix**. You can also include sub types of selected trace link types for filtering.

If you do not select any trace links types for filtering, *all* trace links types are used.

1. In the **Requirement** view that contains the source structure, select the parent element and expand it to the level that you want to see in the matrix.
2. Choose **View→Show Traceability Matrix** or click **Traceability Matrix**  on the toolbar.

The **Show Traceability Matrix** dialog box displays the source element's name in the **Source** box.

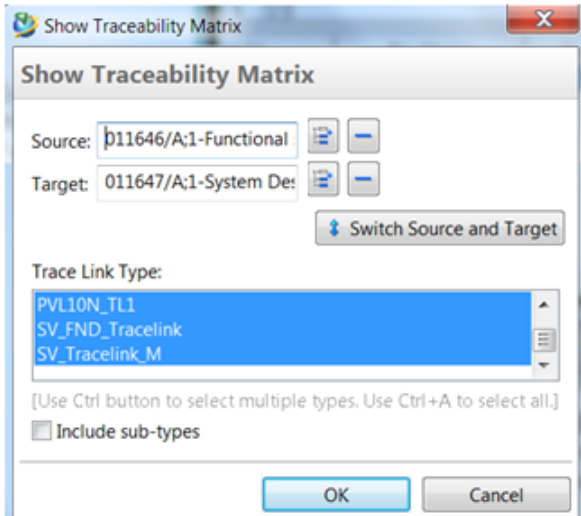
3. In the **Requirement** view that contains the target structure, select the parent element and expand it to the level that you want to see in the matrix.

Note:

The **Show Traceability Matrix** dialog box remains open while you select the target.

4. In the **Show Traceability Matrix** dialog box, click the **Set/Add current selection** button to the right of the **Target** box.

The **Target** box displays the target element name.



Tip:

You can also do the following optional actions:

- To select a different source or target structure:
 - a. Click the **Remove selected object** button to the right of the **Source** or **Target** box.
 - b. Select the new parent element in its **Requirement** view.
 - c. Click the **Set/Add current selection** to the right of the **Source** or **Target** box.
- To reverse the current source and target selections, click **Switch Source and Target**.

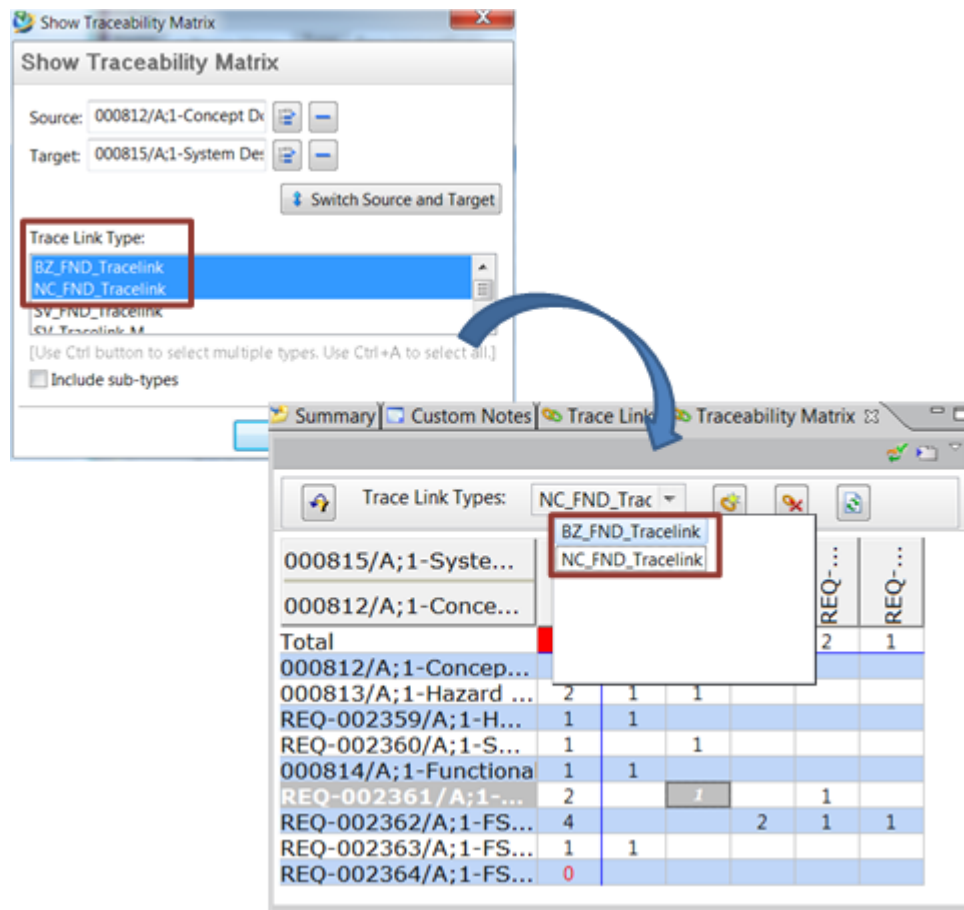
5. In the **Show Traceability Matrix** dialog box, if you want to filter the traceability matrix using trace link type filters, perform the following:

Note:

If you do not select any trace links types for filtering, *all* trace links types are used.

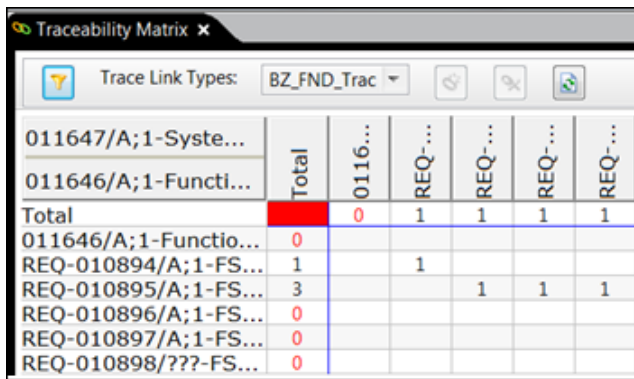
- a. Under the **Trace Link Type:** heading, select the trace link types you want to filter.
- b. Click the **Include sub-types** check box to include all selected subtypes of the selected trace links.

The list of available trace link types is populated from the trace link types you selected in the **Show Traceability Matrix** dialog box.



6. Click **OK**.

The **Traceability Matrix** view displays the allocations between the source and target structures.



Trace Link Types: BZ_FND_Trac	Total	0116...	REQ-...	REQ-...	REQ-...	REQ-...
011647/A;1-Syste...						
011646/A;1-Functi...						
Total		0	1	1	1	1
011646/A;1-Functio...	0					
REQ-010894/A;1-FS...	1		1			
REQ-010895/A;1-FS...	3			1	1	1
REQ-010896/A;1-FS...	0					
REQ-010897/A;1-FS...	0					
REQ-010898/???-FS...	0					

7. If you want to specify a trace link type from the list of available trace link types, in the **Trace Link Types** box, click ▼ and select the trace link type.
8. If you want to add or remove trace link type filters from **Show Traceability Matrix** dialog box, click .
9. To clear trace link type filters, click .
10. After modifying the trace link type filters, click  to refresh the matrix and display the changes.

Create trace links in the Traceability Matrix

You can create trace links in the **Traceability Matrix** by selecting single or multiple cells.

If the selected trace link type has visible attributes, a wizard launches enabling you to fill in the required attribute values.

1. Generate the **Traceability Matrix**.
2. In the **Traceability Matrix** dialog box, click ▼ in the **Trace Link Types** box, and select the trace link type you want to use to create the trace link.

If you have set a filter for trace link types using the **Traceview Settings** dialog box or the **Trace Links** view, you can only create the types specified by the filter.

3. Choose one of the following to create a trace link:
 - Select the cell in the traceability matrix and click **Create Trace Link** .
 - Right-click the cell in the traceability matrix and choose **Create Trace Link**.

Traceability Matrix

Trace Link Types: BZ_FND_Trac

	Total	0116...	REQ-...	REQ-...	REQ-...	REQ-...
011647/A;1-Syste...						
011646/A;1-Functi...						
Total	0	1	1	1	1	1
011646/A;1-Functi...	0					
REQ-010894/A;1-FS...	1		1			
REQ-010895/A;1-FS...	3			1	1	1
REQ-010896/A;1-...	0					
REQ-010897/A;1-FS...	0					
REQ-010898/???-FS...	0					

Create Trace Link

Note:

To select multiple cells, press the Ctrl key, or select a cell and drag the left mouse button.

The trace link is created between the cell row object and the cell column object. The *row* object is the *starting* object for the trace link and the *column* object is the *ending* object for the trace link.

The **Traceability Matrix** is refreshed and the cell count updates. The tool tip displays the trace link type.

Traceability Matrix

Trace Link Types: BZ_FND_Trac

	Total	0116...	REQ-...	REQ-...	REQ-...	REQ-...
011647/A;1-Syste...						
011646/A;1-Functi...						
Total	0	2	1	1	1	1
011646/A;1-Functi...	0					
REQ-010894/A;1-FS...	1		1			
REQ-010895/A;1-FS...	3			1	1	1
REQ-010896/A;1-...	1		1			
REQ-010897/A;1-FS...	0					
REQ-010898/???-FS...	0					

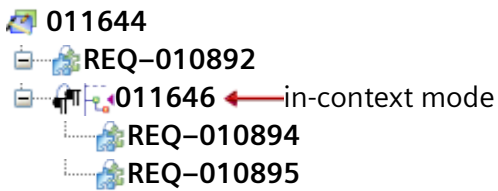
BZ_FND_Tracelink(1)

Create trace links to occurrences in the Traceability Matrix

You can create trace links to occurrences in the **Traceability Matrix** by setting the context in the source or target structures.

1. Select the requirement structure you want to set *in context* mode and choose **Edit→Toggle In-Context Mode**.

The structure is set in context mode.



2. Generate the **Traceability Matrix**.
3. In the **Traceability Matrix** dialog box, click ▼ in the **Trace Link Types** box, and select the trace link type you want to use to create the trace link.

If you have set a filter for trace link types using the **Traceview Settings** dialog box or the **Trace Links** view, you can only create the types specified by the filter.

4. Choose one of the following to create a trace link:
 - Select the cell in the traceability matrix and click **Create Trace Link** .
 - Right-click the cell in the traceability matrix and choose **Create Trace Link**.

	Total	0116...	REQ-...	REQ-...	REQ-...	REQ-...
011647/A;1-Syste...						
011646/A;1-Functi...						
Total	0	1	1	1	1	1
011646/A;1-Functi...	0					
REQ-010894/A;1-FS...	1	1				
REQ-010895/A;1-FS...	3		1	1	1	
REQ-010896/A;1-...	0					
REQ-010897/A;1-FS...	0					
REQ-010898/???-FS...	0					

Note:

To select multiple cells, press the Ctrl key, or select a cell and drag the left mouse button.

According to the set context information, an occurrence trace link is created between the cell row object and the cell column object. The *row* object is the *starting* object for the trace link and the *column* object is the *ending* object for the trace link. When you create the trace link after setting the context,

The **Traceability Matrix** is refreshed and the cell count updates. The tool tip displays the trace link type.

011647/A;1-Syste...	Total	0116...	REQ-...	REQ-...	REQ-...	REQ-...
011646/A;1-Functi...						
Total	0	2	1	1	1	1
011646/A;1-Functio...	0					
REQ-010894/A;1-FS...	1	1				
REQ-010895/A;1-FS...	3		1	1	1	
REQ-010896/A;1-...	1	1				
REQ-010897/A;1-FS...	0					
REQ-010898/???-FS...	0					

Navigate to a requirement in the Traceability Matrix

You can navigate to the requirement in the visible **Requirement** view in the **Traceability Matrix**.

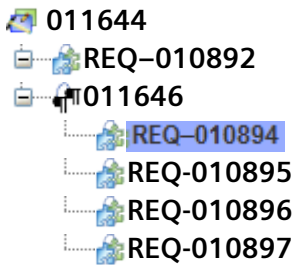
1. Generate the **Traceability Matrix**.
2. Right-click a row header or column header cell containing the requirement name and choose **Find in View**.

011647/A;1-Syste...	Total	011...	REQ-...	REQ-...	REQ-...	REQ-...
011646/A;1-Functi...						
Total	0	4	1	1	1	2
011646/A;1-Functio...	0					
REQ-010894/A;1-	1	1				
REQ-010895/A;1-f			1	1	1	
REQ-010896/A;1-f						1
REQ-010897/A;1-f						

Note:

The command is unavailable if the requirement structure is not visible or loaded in the Systems Engineering application. If you close the structure after the transaction, an error displays indicating the BOM window is currently not loaded.

In the **Requirement view**, the requirement structure expands, if needed, to display the requirement, and the requirement is highlighted.



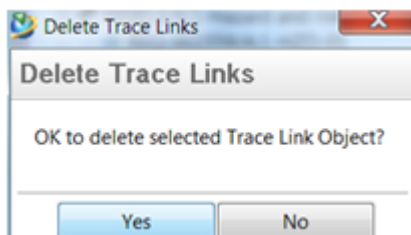
Delete trace links in the Traceability Matrix

1. Generate the **Traceability Matrix**.

Note:

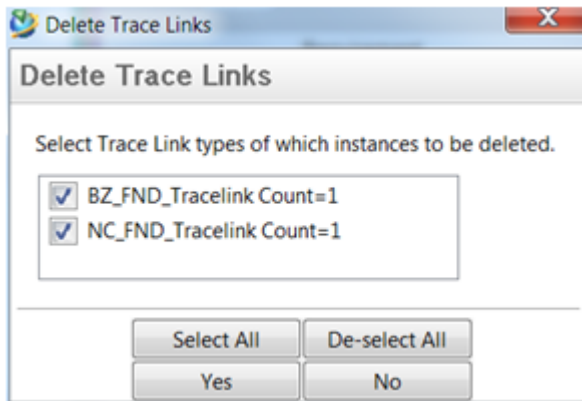
If you used trace link type filters when generating the **Traceability Matrix**, you can only delete the trace link types specified by the filter.

2. Choose one of the following to delete a trace link:
 - Select the cell in the traceability matrix that contains the trace link you want to delete and click **Delete Trace Link** .
 - Right-click the cell in the traceability matrix that contains the trace link you want to delete and choose **Delete Trace Link**.
3. If the cell has only one trace link, click **Yes** in the **Delete Trace Links** dialog box.



The **Traceability Matrix** is refreshed and the cell displays the updated count.

4. If the cell has multiple trace links, in the **Delete Trace Links** dialog box, perform any of the following:
 - Click the check box next to the trace link type you want to delete.
 - Click **Select All** to select all the trace links to be deleted.



- Click **De-select All** to clear the trace link check boxes.

Note:

When deleting trace links with custom notes, the **Delete Trace Links** confirmation dialog box also displays that one or more trace links have custom notes.

- a. Click **Yes**.

The **Traceability Matrix** is refreshed and the cell displays the updated count.

Export the Trace Links view to Excel

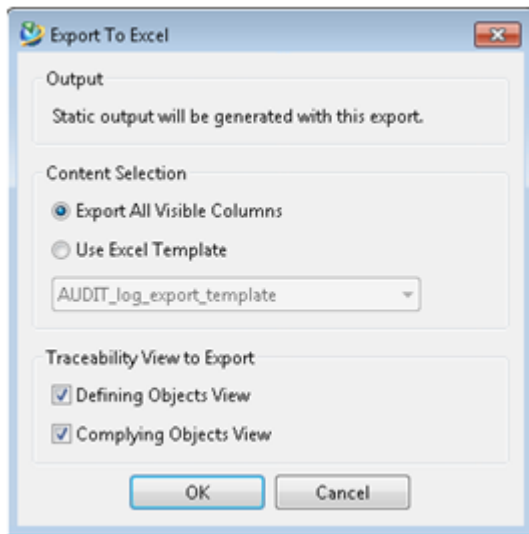
You can export traceability information to Excel from the **Trace Links** view. You export all visible columns to Excel, or optionally specify a template for the export. You can limit the export to either the defining or complying trace links. The exported Excel sheet displays the traceability hierarchy information based on the defined scope, object type, and trace link type filters.

Note:

Only static exports are supported.

1. Open the structure with trace links in a structure view.
2. Open the **Trace Links** view.
3. In the **Trace Links** view, click **Objects To Excel**  on the toolbar.
4. In the **Export to Excel** dialog box, click **Export All Visible Columns** or click **Use Excel Template** to select an export template.
5. Click the **Defining Object view** check box to display the defining objects.

- Click the **Complying Object view** check box to display the complying objects.



- Click **OK**.

Microsoft Excel is launched and the exported excel sheet is displayed.

Type	Name	Desc	ID	Owner	Find Number	Group
Requirement	FSR-02		REQ-000165	yvtcadm (yvtcadm)		dba
Revision	SG-01		REQ-000163	yvtcadm (yvtcadm)		dba
Requirement	HZD-01		REQ-000162	yvtcadm (yvtcadm)		dba
Requirement	SG-01		REQ-000163	yvtcadm (yvtcadm)		dba
Revision	HZD-01		REQ-000162	yvtcadm (yvtcadm)		dba
Requirement	HZD-01		REQ-000162	yvtcadm (yvtcadm)		dba

Export traceability information to Excel in a structure view

1. Open the structure with trace links in a structure view.

For example, send a structure to the **Structure Manager** view, the **Requirements** view, or another structure view.

2. Choose **Tools→Export→Objects To Excel**.

Teamcenter displays the **Export To Excel** dialog box.

3. Under **Output**, do one of the following:

- For a standard Excel file that is not connected to Teamcenter, click **Static Snapshot**.
- For an interactive live Excel file that is connected to Teamcenter, click **Live integration with Excel (Interactive)**.
- For a live Excel file that is not connected to Teamcenter, click **Live integration with Excel (Bulk Mode)**.

You can accumulate changes and later connect the file to Teamcenter.

- To export the data to an Excel file that also contains import processing information on a separate sheet, click **Work Offline and Import**.
- To check out objects while exporting to live Excel, select **Check out objects before export**.

Note:

The checkout applies to all objects being exported. Use this option carefully if you are exporting a large set of objects or perhaps an entire product structure.

4. Under **Output Template**:

- a. Select **Export All Visible Columns** to export all the columns in the applicable structure view.

Note:

Verify the **Trace Links** or **Trace Link** column property appears in the structure view.

- b. Select **Use Excel Template** to activate the template list.

In the list, select the template that specifies the data that you want to export.

5. Click **Copy URL**.

Note:

- **Copy URL** is unavailable if you select more than one object to export.
- **Copy URL** is unavailable if you select any of the following dialog box options:
 - **Work Offline and Import**
 - **Export All Visible Columns**
 - **Export All Objects in View**

The export file is generated and the **URL Generated** message is displayed, confirming that the URL is in your Windows Clipboard and showing the URL details.

6. Click **OK** to generate the export Excel file.

Excel opens a temporary file. You can create a permanent file by choosing **File→Save As** in Excel to display the **Save As** dialog box.

If you save a live Excel file, you can open it later in My Teamcenter to reconnect it to the database.

Note:

Values that you cannot change in Teamcenter are unavailable in the cells of the live Excel file.

Navigate to an occurrence by specifying a variant rule and perspective

Note:

This procedure is for occurrences only, not for workspace objects.

1. Open the structure with trace links in a structure view.

For example, open a requirements structure in the **Requirements** view.

2. Verify the **Show Unconfigured Variants** button on the structure view toolbar is *not* activated.

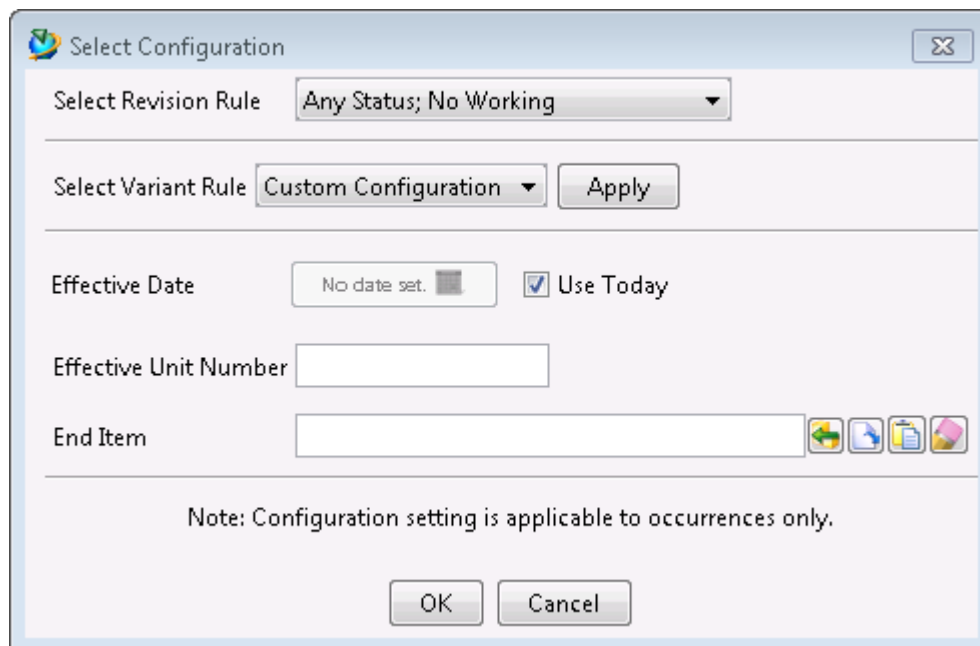
Tip:

The button is activated if it appears outlined, as shown below:



- If necessary, click the **Show Unconfigured Variants** button  to deactivate it.
- 3. In the structure view, select the lines in the structure for which you want to configure variant options and values.
- 4. In the **Trace Links** view, click **Configuration** .

The **Select Configuration** dialog box is displayed.



5. In the **Select Revision Rule** list, select the revision rule that covers the variant rule you want to use.
6. In the **Select Variant Rule** list, select the variant rule that applies to the occurrence you want to open.
7. Click **OK** to close the **Select Configuration** dialog box.
8. In the **Trace Links** view, select the occurrence to which you want to navigate and then click **Go To→Application Name**.

The occurrence opens in the perspective where the variant rule applies to the occurrence.

Note:

The perspective does not show the names of the revision rule or variant rule.

Tracing relationships among Teamcenter domains

Determining change effect on downstream and upstream targets

The Systems Engineering *cross-domain tracing* feature follows trace links from a selected object to its defining and complying objects in related Teamcenter domains. Collaborative designers can trace these relationships to determine how changes to a given object affect upstream and downstream elements in the following domains:

- Requirement specifications, functional models, and logical models in Systems Engineering
- Subsets in 4G Designer
- Physical models in Manufacturing Process Planner

Cross-domain tracing can fully traverse upstream and downstream paths faster than Systems Engineering traceability reports can. Settings in the **Options** dialog box define the depth, the direction, and the relation types for each trace.

You can open the following views to trace related elements:

- The **Requirement**, **Function**, and **Logical Block** views in Systems Engineering
- The 4GD **Content Explorer** and **3D Viewer**, both of which you can open from Systems Engineering
- The **Product** view, one of three structure views in Manufacturing Process Planner

You can select the source object in any one of the preceding views. In each of the other open views, the target objects are highlighted simultaneously.

Setting options for cross-domain tracing

In the **Options** dialog box, you can set the following cross-domain tracing options:

- The number of levels to traverse in the allocation path

By extending the trace along an entire path, you can view all of the related objects simultaneously. Otherwise, you can continue the trace by selecting a target object to highlight its related objects in the other views.

- The direction of the trace

You can trace upstream, downstream, or in both directions simultaneously.

- The types of relations to trace

You can designate trace link relation types and their subtypes.

You open the **Options** dialog box by choosing **Edit→Options**. When you select **Cross Probing** in the left pane, the dialog box displays the options in the right pane. Until you change the settings, they apply each time you run a cross-domain trace.

Caution:

These settings are effective only when you initiate a cross-domain trace by clicking the **Cross Probing Mode** button on the source application toolbar. Otherwise, the settings are ignored.

Option	Description						
Relations Depth Traversal	Sets the number of levels to traverse in the allocation path. - When the Search all levels button is selected, the trace traverses the entire path. - You can set another level by selecting the Specified depth button and typing the level number in the text box. This setting is stored in the CrossProbing_RelationDepthTraversal preference.						
Indirect Trace Link Search	Specifies whether the trace includes the implied trace links that are inherited by the item revisions of target objects. - You can include indirect trace links by selecting the ON checkbox. - When the ON checkbox is cleared, the trace follows only explicit trace links that are directly associated with target object items. This setting is stored in the Indirect_Tracelink_Report preference.						
Cross Probing Direction	Sets the trace direction. <table> <tr> <td>Find Source(s)</td><td> Specifies that the source object is the secondary (complying) object in a relation. The trace highlights related objects upstream. </td></tr> <tr> <td>Find Target(s)</td><td> Specifies that the source object is the primary (defining) object in a relation. The trace highlights related objects downstream. </td></tr> <tr> <td>Both</td><td> Specifies that the source object is both a primary and a secondary object in a relation. The trace highlights related objects downstream and upstream. </td></tr> </table> This setting is stored in the CrossProbing_RelationTraversalType preference.	Find Source(s)	Specifies that the source object is the secondary (complying) object in a relation. The trace highlights related objects upstream.	Find Target(s)	Specifies that the source object is the primary (defining) object in a relation. The trace highlights related objects downstream.	Both	Specifies that the source object is both a primary and a secondary object in a relation. The trace highlights related objects downstream and upstream.
Find Source(s)	Specifies that the source object is the secondary (complying) object in a relation. The trace highlights related objects upstream.						
Find Target(s)	Specifies that the source object is the primary (defining) object in a relation. The trace highlights related objects downstream.						
Both	Specifies that the source object is both a primary and a secondary object in a relation. The trace highlights related objects downstream and upstream.						
Allowed Relations	Lists all relation types and subtypes that you can designate for the trace. You can designate a relation by doing the following: Select the type or subtype in the Allowed Relations pane.						

Option	Description
	2. Click the > button between the two panes. The relation is added to the Selected Relations pane. Tip: You can also click the >> button to add the entire list at the same time. The list values are controlled by the CrossProbing_AllowedRelationTypes site preference.
Selected Relations	Lists the relation types and subtypes that are currently designated for the trace. You can remove a relation by doing the following: 1. Select the type or subtype in the Selected Relations pane. 2. Click the < button between the two panes. The relation is removed from the Selected Relations pane. Tip: You can also click the << button to remove all relations at the same time. The designated relations are stored in the CrossProbing_SelectedRelationTypes preference.

Open views for cross-domain tracing

Note:

Before you open source and target views, you can define the depth, the direction, and the relation types to trace.

To open a view from Systems Engineering:

1. Choose **Window→Show View→Other** to display the **Show View** dialog box.

Tip:

When you choose **Window→Show View**, you can select the view if its name appears in the list of common and recently used views.

2. Expand the category for the application, and then double-click the view name.
 - For Systems Engineering, the **Requirement**, **Function**, or **Logical Block** view
 - For 4th Generation Design, the **Content Explorer** view or the **3D Viewer**
 - For Manufacturing Process Planner, the **Product** view

To open a view from My Teamcenter:

To open this view	Do this
Requirement	In the Home view, do one of the following: • Select a requirement specification, a requirement, or a paragraph, and then choose File→Open, or double-click the object. • Right-click a requirement specification, a requirement, or a paragraph, and then choose Send To→Systems Engineering. The Requirement view opens with the object at the root level.
Function	In the Home view, do one of the following: • Select a function and choose File→Open, or double-click the function. • Right-click a function and choose Send To→Systems Engineering. The Function view opens with the function at the root level.
Logical Block	In the Home view, do one of the following: • Select a logical block and choose File→Open, or double-click the block. • Right-click a logical block and choose Send To→Systems Engineering. The Logical Block view opens with the block at the root level.
Content Explorer	• In the Home view, right-click a subset definition and choose Open With→Content Explorer. The Content Explorer view opens with the subset definition at the root level.
3D Viewer	• In the Home view, right-click a subset definition and choose Open With→3D Viewer.

To open this view	Do this
	The 3D Viewer opens with the subset definition at the root level.
Product	In the Home view, do one of the following: • Select a non R/F/L object and choose File→Open, or double-click the object. • Right-click a non R/F/L object and choose Send To→Systems Engineering. The Product view opens with the object at the root level.

Run a cross-domain trace

Note:

Before you start this procedure:

- You can define the depth, the direction, and the relation types to trace.
- Open the views that contain the source and target structures.

1. Select the source object.

Caution:

For the 4GD **Content Explorer** view, select only subset definitions in the root context.

The trace ignores any open **Content Explorer** views where the root context object is not a subset definition.

Tip:

You can select two or more source objects with the standard Windows features for multiple selection.

2. In the source application, click the **Cross Probing Mode** button  on the toolbar to initiate the trace.

Caution:

The trace is canceled if you switch to another application, for example, if you initiate the trace in Systems Engineering and then switch to My Teamcenter.

Tip:

- You can switch perspectives within the source application without canceling the trace.
For example, in Systems Engineering, you can switch from the Systems Engineering perspective to the Requirements Manager perspective.
- The trace remains effective if you close a perspective and reopen it.

In each open target view, the related objects are highlighted simultaneously. Any related objects in the source view are highlighted also.

3. (Optional) To continue tracing downstream or upstream, select a highlighted object in a target view.

Tip:

You can select two or more target objects with the standard Windows features for multiple selection.

The selection becomes the new source. Its related objects are highlighted as targets in the other open views.

4. When you are finished tracing relations, end the current trace by clicking the **Cross Probing Mode** button  again.

Checking accountability between structures

Checking requirement structures against product and process structures

Run an accountability check to compare two structures to ensure all lines in the source structure are consumed in the target structure. Comparisons can be made between a requirement structure and a product structure or between a requirement structure and manufacturing process structure. You can compare either the entire assembly or a subassembly.

For example, you can compare a product structure and requirement structure to ensure that all the requirements in the requirement structure are accounted for against the parts in the product structure for which the requirements are defined.

After running the accountability check, any anomalies found during the comparison are reported. The following anomalies are reported:

- Lines in the source structure that are not consumed in the target structure.
- Lines in the target structure that are not present in the source structure.

- Lines in either structure that are consumed more than once.

Note:

For trace links on occurrences, the accountability check compares two structures on the basis of value of the **ID In Context** property for each line. Therefore, before running an accountability check for trace links on occurrences, a value for the **ID In Context** property must exist for each line in the source and target structures being compared.

Tip:

The **Accountability Check** view does not hide lines for which a filter is applied in one of the structures. By revealing filtered lines in the view, the accountability check voids the filter settings.

To restore the filter settings to the structure, do the following in **Filter Controls**:

1. Click **Unfilter the data**  to cancel the filter operation and display all rows in the structure.
This action does not clear the filter settings.
2. Click **Filter the data**  to reapply the filter settings.

About the Accountability Check view

When you run the accountability check, Teamcenter displays the results in the **Accountability Check** view. This view lists the problematic lines in the source and target and provides a list of conflict details. Lines that are colored in the source and target structures are also colored in the view. Fully matched lines are not listed by default.

Note:

Displaying fully matched lines in large structures can clutter the view

Lines in the **Accountability Check** view behave in the same fashion as lines in the source and target structures. If you remove a line in the view, Teamcenter removes the line from the source or target. You can copy lines from the view into the clipboard, drag them to another structure, or send them to another application.

When you select a source or target line that is marked as having net effectivity or being a partial match, Teamcenter displays the conflicts in a third pane at the bottom of the view. The tabs shown in the **Partial Match** section of the view are dependent on which partial match criteria you selected in the **Partial Match** pane during the check. The table can display information about:

- Properties
- Predecessors (enterprise BOP structures only)

- Assignments (enterprise BOP structures only)
- Hierarchy (enterprise BOP structures only)

When you select a line in the source section of the **Accountability Check** view, you can click the **Equivalent Lines** tab in the target section of the view to find the lines in the target structure that match the selected source line, and vice versa. These lines are then selected in the visible structure views.

When you work through the accountability check results, resolving conflicts, the accountability check results are not updated—the lines remain highlighted in the original check colors. To update the lines, you must rerun the accountability check. You can do this using the same settings by clicking  in the **Accountability Check** view.

Note:

The **Accountability Check** view is not displayed when one or both of the structures being compared is an intermediate data capture.

Run an accountability check

Note:

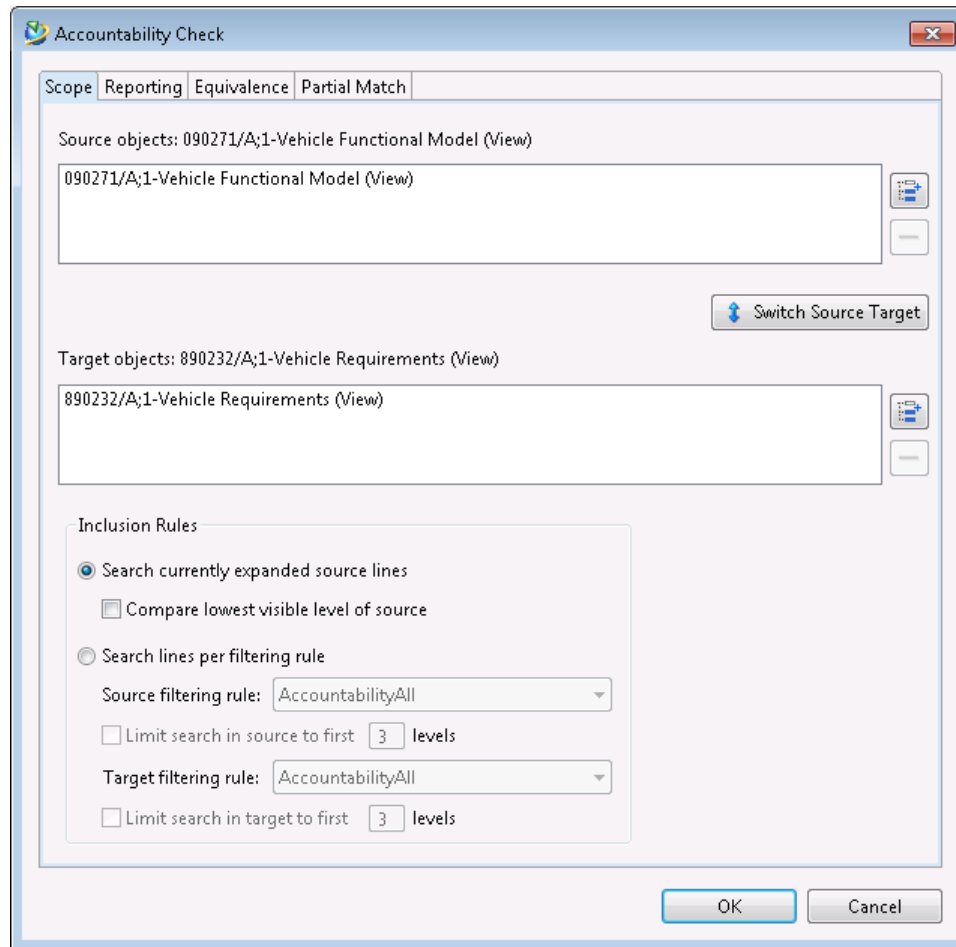
To run an advanced accountability check, you must have two linked source and target structures to compare. These are typically a source engineering bill of materials (EBOM) generated from a design BOM that is exported from a CAD application such as NX or CATIA, and a target manufacturing BOM (MBOM) created for specific manufacturing needs from the EBOM. When the EBOM is updated, the MBOM must be compared and updated as well. This is the purpose of an advanced accountability check: the two structures must be examined to ensure that they share equivalent BOM lines.

1. Open the source and target structures in two different views.
2. Select the source objects in the source pane.
3. Select the target objects in the target pane.
4. Select the source and target objects in the applicable structure views simultaneously, or select either the source or target object in the applicable structure view.
5. Choose **Tools→Compare→Accountability Check→Accountability Check**.

The **Accountability Check** dialog box opens.

- If you selected both the source and target lines in the structure views simultaneously, the selected lines appear in the **Source objects** and the **Target objects** panes, as applicable, defining the scope for the accountability check.

- If you selected either the source or target line in a structure view, the selected line appears in the **Source objects** or **Target objects** pane, whichever is applicable.



6. If you need to complete the scope setting for either the source or target:
 - a. Select the target (or source) line in the applicable structure view.
 - b. In the **Accountability Check** dialog box, click **Set/add current selection**  next to the **Target objects** (or **Source objects**) pane, as applicable.

The selected line appears in either the **Target objects** or **Source objects** pane, as applicable, completing the scope setting for the accountability check.

7. (Optional) Change the source or target object at any time by selecting the desired line in the structure and clicking the respective **Set/add current selection** . You can add multiple scope lines from the same structure or remove a scope line by selecting it and clicking .
8. Ensure that you correctly selected the source and target objects. If not, switch the source and target by clicking the **Switch Source Target** button.

9. Click the **Inclusion Rules** tab, and do one of the following:

- Select **Search currently expanded source lines** to run the accountability check on the selected lines and all visible lines below the selected lines. Additionally, you can choose to run the accountability check on all the lowest visible (expanded) lines under the selected scope line by selecting **Compare lowest visible level of source**.
- a. Select **Search lines per filtering rule** to expand the source and target structures and only consider a subset of the lines during the accountability check.
- b. Select the **Source filtering rule**. For a description of the filtering rules, move your mouse over the list. The rules along with their short descriptions are displayed.

You can specify to which level in a structure you want the check to consider by selecting **Limit search in source/target to first x levels** and typing a value for x.

Note:

For information about filtering rules, see Managing filter rules in the Tc XML and PLM XML Configuration for Data Import and Export.

Note:

When you set these options, the values take effect in all applications that support accountability checks.

10. (Optional) If you know the source and target scope lines to be different because these lines are source and target specific, select the **Do not compare selected lines** check box ☒.

In the advanced accountability check results, the selected scope lines do not appear as missing in target and missing in source.

11. Click the **Reporting** tab and set the result reporting options.
12. Click the **Equivalence** tab and set the equivalence options.
13. Click the **Partial Match** tab and specify the partial match criteria.
14. Click **OK** to run the accountability check.

Teamcenter displays the **Accountability Check** view.

Note:

The number of lines returned in the accountability check is limited by the value of the **MEAccountabilityCheckAllowableReturnedLines** preference. If the number of lines to return exceeds the preference value, a message will be displayed notifying you that there are more lines to be returned, and that an Accountability Check Excel report has been created and stored, thus saving the work.

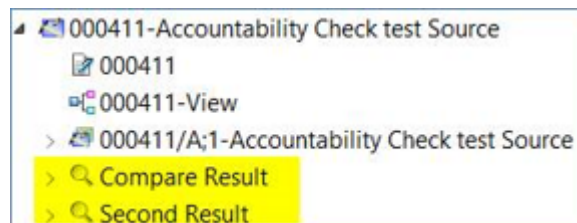
Set the result reporting options

The accountability check displays its results by highlighting the checked objects in the structure pane in a variety of colors that you can change, if desired. To help users who cannot differentiate colors well, a tool tip shows the name of the color. If the name is not available, the tooltip displays the RGB value.

1. Open the accountability check dialog box.
2. Click the **Reporting** tab.
3. Select one of the following:

- **Report in occurrence groups**

Teamcenter can store and display the result of the accountability check in an additional tab in the structure pane grouped by anomaly. This result is only computed for the source item and is stored in the database as a structure of **UsageCompareResultView** objects related to the item of the source revision. You can specify the name for the result of the check. By default, it is named **Compare Result**.



- **Report the selected check criteria**

Teamcenter offers you two methods to visualize the results of the accountability check. You can select one or both of the following:

- Select **Color the compared objects** to color the line and the matching lines according to the match results and color scheme that you set in the **Reporting** section.
- Select **Printable report** to capture the results of the accountability check in an Excel spreadsheet.

4. Check the display options you require:

Option	Description
Full Match	If the object (occurrence) in the source structure has one and only one equivalent in the target structure, both objects are set to the same background color. If you also selected the Partial Match option, a full match also means that all partial match criteria of the equivalent objects match, for example, properties or relations, if you use enterprise BOP structures. The default color is green.
Partial Match	For equivalent objects (occurrences), if one or more partial match criteria are not identical, both objects are set to the same background color. The default color is yellow.
Multiple Match	If objects have more than a single equivalent in either structure and all properties of the source objects are identical to the properties of target objects, all objects are set to the same background color. The default color is orange. Only available when you do not use net effectivity.
Multiple Partial Match	If objects have more than a single equivalent in either structure and in addition, one or more properties are not identical, all objects are set to the same background color. The default color is pink. Only available when you do not use net effectivity.
Missing Target	If occurrences of the source structure are not found in the target, the source objects are set to this color. The default color is red.
Missing Source	If occurrences of the target structure are not found in the source structure, the target objects are set to this color. The default color is light blue.

5. (Optional) Change the default color by clicking ... next to the color.
6. Set the colors by choosing a color swatch, and setting HSB (hue, saturation, brilliance) or RGB (red, green, blue) values.

Set the equivalence options

1. Open the accountability check dialog box.
2. Click the **Equivalence** tab.

3. Select the comparison options you require:

Option	Description
Equivalent Logical Designator	If you use enterprise BOP structures, the accountability check takes process logical designators into consideration as an additional comparison criteria when checking the process, operation, or partition lines for equivalence. This check box is available only when both structures are enterprise BOP structures.
Equivalent PublishLink Connection	If you use publish links, the accountability check highlights occurrences that are mapped using publish links. If a part occurrence is set to not require positioned design, it is also taken into consideration.
Equivalent TraceLink	If you use trace links, the accountability check highlights objects and occurrences allocated using trace links. This identifies objects in the source structure that are: • Linked to at least one object or occurrence in the target structure. • Not linked to any objects or occurrences in the target structure.

Note:

When checking accountability between structures, the direction of tracelinks is not considered.

Specify partial match criteria

You can specify which data to compare between equivalent lines. You can choose to compare a specific set of properties or, if you are comparing enterprise BOP structures, you can additionally compare any relation type of these lines to other lines. For example, you can compare assignment or predecessor relations.

1. Click the **Partial Match** tab.
2. On the **Properties** tab, select **Consider values of properties when searching for a partial match**.
3. Choose the properties that you want included in the accountability check by selecting them in the **Available Properties** list and clicking **+** to move them to the **Selected Properties** list.

The accountability check matches lines in the source and target that have the same properties as those you select in the **Selected Properties** list. Your administrator determines which properties are available for selection in the **AdditionalAccountabilityCheckProperties** preference. Any differences in the selected properties are identified as differences between the structures.

The values of the **DefaultAccountabilityCheckProperties** preference are ignored.

Clear display

You can reset all background colors on objects in the active window and their corresponding objects in the process window to the default colors.

- Choose **Tools→Accountability Check→Clear Accountability Check Display**.

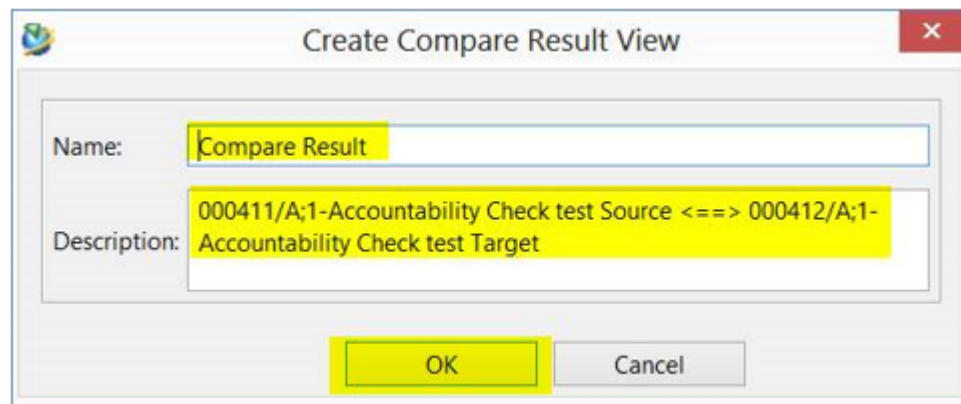
Note:

The command works only if either the source view or the target view is the active window.

Generate the results grouped by anomaly

1. To generate the report:
 - a. Open the **Accountability Check** dialog box.
 - b. On the **Reporting** tab, click **Report in occurrence group**.
 - c. Click **OK**.

The **Create Compare Result View** dialog is displayed.

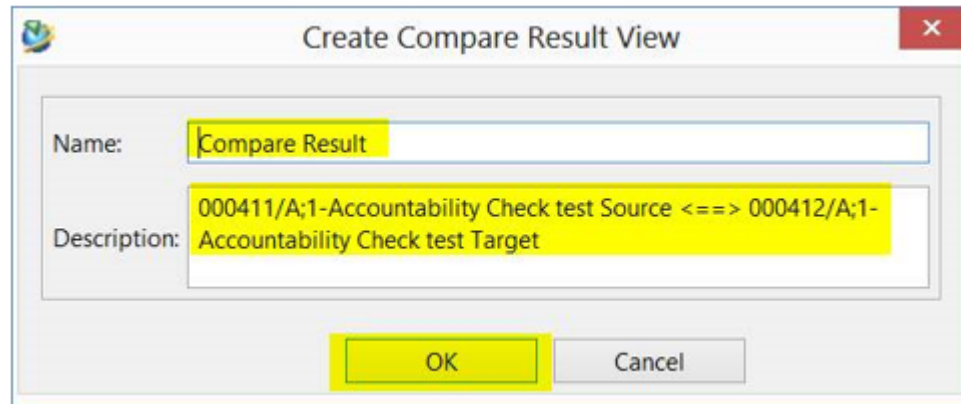


- d. Modify the system generated **Name** and **Description** as required.

The **Name** of the result view is displayed as the name of the additional tab. Multiple result views can be displayed at a time. If a view with the same name exists, the system displays a message asking if the result view should be overwritten or renamed. In this case, you must have write access to overwrite the existing view.

- e. Click **OK**.

Teamcenter creates **UsageCompareResultView** objects for each type of anomaly and displays them in the **Compare Result** tab.



Teamcenter creates occurrence groups for each type of anomaly and displays them in the **Compare Result** tab (an occurrence group tab). The names of the anomalies are fixed and are as follows:

- **Unused/Under-used**

Contains all occurrences in the source pane that do not match an occurrence in the target pane.

- **Completely Used**

Contains all occurrences in the source pane that match a corresponding occurrence in the target pane.

- **Over-used**

Contains all occurrences in the source pane that match more than one occurrence in the target pane.

- **Partial Match**

Contains all occurrences in the source pane that have mismatched properties compared to their corresponding occurrence in the target pane.

2. (Optional) Expand the subgroups to view the items contained in these groups and take appropriate action.

You must rerun the accountability check to view any changed results.

Create a printable Excel report

Teamcenter can create a printable Excel report containing the results of an accountability check. If you have write access to the target structure, Teamcenter stores the Excel file as a dataset on the top-level

item revision of the target structure. If you do not have write access, Teamcenter stores the Excel file in the **Newstuff** folder.

1. Open the **Accountability Check** dialog box.
2. Select all necessary options to create the accountability report.
3. Click the **Reporting** tab and select **Report the selected check criteria**.
4. Select **Printable report**.
5. Assign a name for the new dataset.

If a dataset of this name and type already exists in the item revision, Teamcenter opens a dialog box where you can change the name or replace the dataset with the new one. You can change this behavior in the **TC_CRF_overwrite_existing_dataset_content** preference. If this preference is set to **1** (true), Teamcenter does not remove the dataset but replaces the content.

6. Click **OK**.

Teamcenter creates a report containing:

- General information such as the date and time of the check and the user ID and name.
- Compare options.
- Context data such as the source top level item, the target top level item, incremental change, and revision and variant rules.
- Accountability check results such as the source name, target name, match status, and all the checked properties. Multiple matched objects and missing source objects are displayed in separate sections.

Teamcenter adds the Excel file to a dataset attached to the target structure.

Note:

If you use the **MEAccountabilityCheckEnableNetEffectivity** preference to report net effectivity mismatches, the Excel report also reports those mismatches.

Net Effectivity Mismatches									
Mismatched Sources			Mismatched Targets			Mismatch Details			
BOM Line	Occurrence Effectivity	Find No.	BOM Line	Occurrence Effectivity	Find No.	End Item	Units	# in Source	# in Target
000054/A.1-a1	6-UP (000049) 1-9 (00005)	10	000054/A.1-a1		10	000049-EBOM	1-5	0	1
						000050-MBOM	10-UP	0	1
									Missing in Source
000056/A.1-a3	9-15 (000050)	30	000056/A.1-a3		30	000049-EBOM	1-15, 21-UP	0	2
000056/A.1-a3	5-8 (000050)	10	000056/A.1-a3	5-8 (000050)	20	000049-EBOM	16-20	0	3
			000056/A.1-a3		30	000050-MBOM	1, 3, 16-UP	0	2
			000056/A.1-a3	5-8 (000050)	10	000050-MBOM	2, 4	0	3
			000056/A.1-a3		20	000050-MBOM	9, 11-15	1	2
			000056/A.1-a3	5-8 (000050) 16-20 (00	10	000050-MBOM	10	1	3
						000050-MBOM	6, 8	1	6
						000050-MBOM	5, 7	1	5
									Overlapping
000058/A.1-a6	01-Aug-2015 00:00 to 28-	10	000058/A.1-a6		20	000050-MBOM	1, 3, 5, 7, 9	1	0
									Missing in Target
000059/A.1-a8	6 (000050)	20	000059/A.1-a8	8 (000049)	30	000050-MBOM	6	1	0
						000049-EBOM	8	0	1
									Missing in Target
									Missing in Source

6. Recording supplemental information

What are custom notes?

Custom notes help you record information such as decision rationale, actions, questions, and meeting minutes. They usually contain ad hoc information for particular purposes, rather than established program or project parameters as in standard notes.

You can create custom notes on items and item revisions such as requirements, functions, and logical blocks. Then, you enter and maintain the note content in Microsoft Office Word, capturing issues as they arise. The rich-text content is stored in the **Body Cleartext** property of the custom note revision.

The **Custom Requirements Lists** relation associates the custom note with the item or item revision. The item revision is the primary object in the relation, and the custom note is the secondary object. When you attach an existing custom note to other items and item revisions, another **Custom Requirements Lists** relation is created between the note and each primary object.

Note:

The **Fnd0AllowMultipleRevofCustomNote** global constant must be set to **true** (the default value).

- A custom note can be related to a single object or to multiple revisions of an object.
- Custom notes can be created in the context of another object such as a requirement, a part, or a manufacturing object such as a process or operation.
- Custom notes are attached directly to an object, and not underlying occurrences.
- After custom note content has been edited in Word, it can no longer be edited by modifying the **Text** property value.

The custom note item consists of the following objects.

- ▲  **CSMTNOTE-01-Mtg min** ← custom note master
- ▲  **CSMTNOTE-01-Mtg min** ← custom note master form
- ▲  **CSMTNOTE-01/A;1-Mtg min** ← custom note revision
- ▲  **CSMTNOTE-01/A-Mtg min** ← custom note revision form
- ▲  **Mtg min** ← custom note full-text dataset

You can view additional information about the custom note by displaying the **Custom Notes** view. To display the custom notes using the **Custom Notes** view, select the custom note or the object with the custom note and choose **Window→Show View→Other→Systems Engineering→Custom Notes** or click **Custom Notes Manager**  on the toolbar.

Associating custom notes to Systems Engineering objects

You can associate custom notes to the following Teamcenter objects for Systems Engineering:

- Requirement specifications
- Requirements
- Derived requirements
- Paragraphs
- Functions
- Logical blocks
- Trace links

When you attach a custom note to a Systems Engineering object, the attachment indicator  appears in the **Notes** column.

Requirement	Notes
 000020/A;1–Robot Reqts	
  000020/A;1–Robot Reqts	

If the object containing the custom note exists in a folder, the custom notes are displayed when you expand the object.

 REQ-000001/A;1–Safety  REQ-000001/A  REQ-000001/A-View  Safety  View  CSMTNOTE-01–Mtg min  CSMTNOTE-01–Mtg min  CSMTNOTE-01/A;1–Mtg min  CSMTNOTE-01/A–Mtg min  Mtg min	

Create a custom note

You can create a custom note from Systems Engineering or Requirements Manager.

1. Choose one of these methods to create a custom note:

- Select a folder or the object where you want to add a custom note and choose **File→New→Custom Note**.
- Select a folder or the object where you want to add a custom note and click **Custom Note** .
- Create a custom note from the Quick Create bar in the **Requirements** view.

To create custom notes on trace links, the trace link must be displayed in the **Trace Links** view.

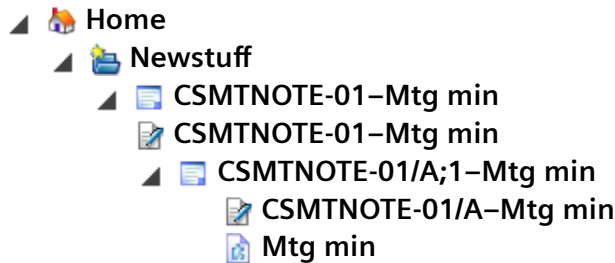
- In the **New Custom Note** dialog box, select **Custom Note** from the list and click **Next** or, alternatively, double-click **Custom Note**.
- Enter the following information.

Field name	Definition	Valid values
ID	Specifies the ID of the custom note (required).	Alphanumeric string (1–128 characters). Choose one of the following: • Leave the field blank to automatically assign a unique ID using the displayed format defined by the naming rule for custom notes. • Click the ▼ to select the format from the list and click Assign to automatically generate a unique ID. • If no naming rule is used, type a unique ID.
Revision	The revision of the custom note (required).	Alphanumeric string (1–32 characters) Leave the field blank to automatically assign the revision.
Name	The name of the custom note (required).	Alphanumeric string (1–128 characters)
Description	The description of the custom note (optional).	Alphanumeric string (1–240 characters)

- (Optional) Click **Next** to assign the custom note to a project.

- Click **Finish** to create the custom note.

If you selected a folder, the custom note is created under the folder.



If you selected an object, the attachment indicator  appears in the **Notes** column for the associated object in the **Requirements** view.

Requirement	Notes
 000020/A;1–Robot Reqts	
 000020/A;1–Robot Reqts	

Tip:

To display the custom notes using the **Custom Notes** view, select the custom note or the object with the custom note and choose **Window→Show View→Other→Systems Engineering→Custom Notes** or click **Custom Notes Manager**  on the toolbar.

Edit the contents of a custom note

Caution:

On Teamcenter servers using the ISO 8859-1 character set (**TC_CHARACTER_ENCODING_SET**), some characters of the Microsoft Office Word rich-text format cannot be translated to plain text. These characters include, for example, Word's smart quotation marks, *em* dashes, and symbols such as Ohm signs (Ω).

Such characters are replaced by question marks (?) in plain text and in the Systems Engineering interface, for example, in object names and in certain property values. To avoid this problem, do one of the following:

- Before you enter content for an object, configure the **MS Word** view to display plain text content.
- Run the Teamcenter server on Windows, using a database that is compatible with CP1252 encoding.

- Use UTF-8 encoding for the Teamcenter server and database.

1. Configure the **MS Word** view to set up the word file type associations and templates.
2. From the navigation tab, click **Systems Engineering**.
3. Display the custom note revision by choosing one of the following methods:
 - Select the object with the custom note and choose **Window→Show View→Other→Systems Engineering→Custom Notes** or click **Custom Notes Manager**  on the toolbar to display the custom note in the **Custom Notes** view.
 - Right-click the custom note or the object containing the custom note and choose **Open with→Details** to display the attached custom note revisions in the **Details** view.
 - If the custom note exists in a folder, expand the custom note to display the custom note revisions.
 - If the object containing the custom note exists in a folder, expand the object to display the custom note revisions.
4. Select the custom note revision you want to edit the contents for and click **MS Word View**  on the toolbar.
5. From the **File Download** dialog box, click **Open**.

The custom note text opens in a Word window within Teamcenter.

6. In the Word window, type your new or changed custom note text.
7. Click **Save**  in the upper right corner of the Word window to save the text.

The **Save** button becomes unavailable after the save is complete.

Attach a custom note to an item or item revision

You can associate a custom note to an object by attaching it to an object in **Systems Engineering** or **Requirements Manager**.

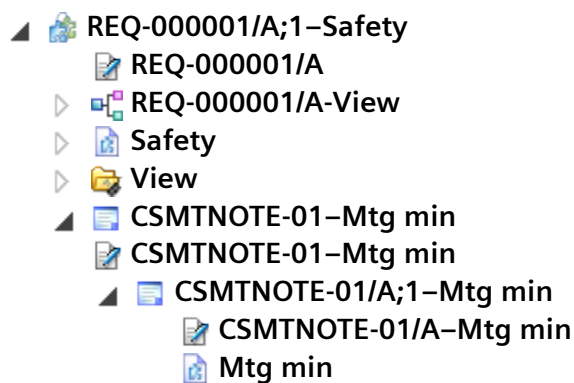
1. Right-click the custom note or custom note revision, and then click **Copy**.
2. Select the item or item revision of the object you want to associate to the custom note and choose **Edit→Attach Requirements/Notes→Custom Note**.

The custom note revision is associated with the selected object (item or item revision) through the **Custom Requirements Lists (FndOListsCustomNotes)** relation.

The **Notes** column displays the attachment symbol  for the object.

Requirement	Notes
 000020/A;1–Robot Reqts	
 000020/A;1–Robot Reqts	

If the object associated with the custom note exists in a folder, the custom note appears when you expand the object.



If the note is not visible under the item or item revision, you must specify **Custom Requirements Lists** as a shown relation for the item and item revision.

1. Choose **Edit→Options**.
2. Expand the **Options** tree and expand the **General** folder.
3. Click the **Item** or **Item Revision** node.
4. In the **General** options tab, choose **Custom Requirements Lists** from the **Available Relations** list.
5. Click  to move it to the **Shown Relations** list.
6. Click **OK**.
7. Choose **View→Refresh** or **View→Refresh Window** to refresh the display.

Create a new revision of a custom note

1. Select the custom note revision for which you want to create a new revision and choose **File→Revise**.

The **Revise** dialog box appears.

2. (Optional) In the **Define the basic information for the new item revision** pane, type the name of the new revision.
3. (Optional) Type a description of the revision and choose a unit of measure for the revision.
4. Click **Next**.

The **Define additional item revision information** pane is displayed.

5. (Optional) Enter values for the note revision attributes.
6. Click **Next**.

The **Define attached objects** pane displays the source revision and attachments and the destination revision and attachments. By default, all objects are copied forward to the new (destination) revision.

7. (Optional) Click the **Copy as object option set** button to the right of each object in the revision tree and select the desired copy option.



Don't Copy

Allows you to selectively copy objects from one revision to another. Objects that are designated as **Don't Copy** appear with a line through them in the **Destination** tree.



Copy as Object

Creates a new object of the same type and relation to the parent object. Objects created by this method are totally independent of the parent object. Therefore, any changes to the copied object are not reflected in the original object.

Copied objects appear in bold in the **Destination** tree and can be renamed.



Copy as Reference

Copies the object as a reference to the original object. All changes to the reference copy affect the original object. The names of attachments that are copied as references cannot be modified.

8. Click **Next**.

The **Select projects or programs that you want to assign new object to** pane is displayed.

9. (Optional) Assign the new revision to a different project or program.

10. Click **Next**.

The **Select open option and alternate id display option** pane is displayed.

11. (Optional) Set the open, display, or checkout option for the new revision.

12. Click **Finish**.

The new revision is created as a copy of the previous revision object. The new revision is independent of the source revision. Modifications to the new revision are not reflected in the source revision, which allows you to edit the note text for each revision.

Delete a custom note

Note:

This procedure removes the custom note object from the Teamcenter database.

Note:

You must be in the My Teamcenter application to delete the note. If you are in the Systems Engineering or Manufacturing Process Planner application, cut the note and send it to the My Teamcenter application.

To delete a note, you must remove all references to it. To do so, cut the custom note from the attached revision object and click **Delete**.

You cannot delete the note from the **Custom Notes** view in the Systems Engineering application. However, you can remove it from the **Custom notes** view by using the **Cut** command.

1. Select the custom note or custom note revision object you want to delete.
1. Choose **Edit→Delete** or click the **Delete** button on the toolbar or press the Delete key.

Teamcenter displays a request for confirmation.

2. Click **Yes**.

Note:

The standard rules for deleting an item or item revision apply to deleting a custom note or custom note revision.

7. Capturing data for downstream disciplines

Exporting product data

By exporting requirement specifications and design architectures, you can capture your product data for analysis and distribution to downstream disciplines. From the structure views in Requirements Manager and Systems Engineering, you can export property values for requirements, functions, and logical blocks, including their traceability relationships and levels of subordination.

You can select an entire structure or only certain portions, and then export that data to the following applications or formats:

- Microsoft Office Word and Excel

An export template defines which property values are exported and how that data is formatted in the output file. Teamcenter provides Word and Excel export templates by default. Your Teamcenter administrator can modify the default templates and create custom templates for specific purposes at your site. You select a template each time you export data.

Tip:

You can export data to a Word or Excel file that is synchronized with Teamcenter in real time. Directly in this file, you can modify a structure or its elements interactively.

- Teamcenter internal email

You can send object data from Systems Engineering to other Teamcenter users, and to SMTP addresses for recipients who are not Teamcenter users. The export process attaches the Word or Excel output file to an email message. The data consists of property values specified in an export template that you select.

- PLM XML format

Your selected transfer mode determines what data is exported. You can also track your PLM XML exports using Application Interface mode.

- Teamcenter application interface

Application interface objects store parameters required to share data in PLM XML format, allowing you to repeatedly share data with other sites without defining the parameters each time you export the data.

- Teamcenter format and STEP format

When you export to these formats, you also export all internal data associated with the selected objects. For example, exported items can have relations such as item masters, item revisions, requirements, specifications, and reference objects.

- Remote sites through Global Services

Note:

You can export product structure data also from Structure Manager, from Multi-Structure Manager, and from the My Teamcenter **Details** view and navigation pane.

Exporting data to Microsoft Office Word

Exporting data to Microsoft Office Word

For analysis or distribution, requirements can be exported to Word, thus capturing both their content and their organization within the requirement specification structure. For each requirement structure element that you specify, Teamcenter extracts the content from the database to a temporary Word document.

The Word export process uses two templates, a *specification template* and an *object template*.

- A specification template controls the output document format, including page layout. This template defines the Word styles that are applied to the exported data. It also designates which object template is referenced by the export process.
- An object template controls the content, property format, and the properties whose values are exported for the selected objects.

Note:

To further control exported content, you can add closure rules and transfer modes to Word templates. Closure rules are not allowed in the body text of requirements.

You can specify a rule in the specification template to define which objects are exported.

You can view and print the output document and send it by e-mail and fax. Also, you can save the document, for example, on a local or shared drive, where the content can be edited independently of Teamcenter.

The saved document can be imported to create requirement specification structures automatically, complete with parent, child, and sibling requirements.

Setting the structure context for export to Word

The data that is exported to Microsoft Office Word is determined by the object template that you select during the export procedure. However, the data's layout in the output document depends on the way you select Teamcenter objects to start the procedure.

Object selection	Output document layout
Requirement specification object, peak level	Data for all objects at all levels, including all direct children and lower level descendants of each parent object. - For the requirement specification object: - The object name is extracted to a Word paragraph that is formatted in the Heading 1 style. - The object content is placed under the heading. - For each child and descendant: - The object name is formatted in a Word heading style that matches the object's level in the structure. - The object content is placed under the heading.
One object, at any level below the peak	Data for the selected object and all of its direct children and lower level descendants, if any. - For the selected object: - The object name is extracted to a Word paragraph that is formatted in the Heading 1 style. - The object content is placed under the heading. - For each direct child, if any: - The object name is formatted in Word's Heading 2 style. - The object content is placed under the heading. - For each lower level descendant, if any: - The object name is formatted in the heading style that matches the level in the structure.

Object selection	Output document layout
Multiple objects, at the same level or at different levels below the peak	- The object content is placed under the heading. Data for the selected objects only, not including direct children and lower level descendants. For each object: - The object name is formatted in Word's Heading 1 style. - The object content is placed under the heading. - Headings appear in the order in which the objects are displayed in the structure.

Override the default object template for frequent Word exports

Each Word export operation uses an object template to extract property values for the selected objects. You can override the current object template to set the default template for frequent exports of an object type.

1. In the **Home** view, choose **Tools→Microsoft Office Templates→Override Word templates**.

The **Override Word templates** dialog box opens.

2. In the **Specification Template** list, select the specification template that is associated with the object template you want to override.
3. In the **Override Object Template for** list, select the object type.
4. Under **Available Object Templates**, select the new object template and click **»**.

The right pane displays your selections. If necessary, you can click **«** to reverse the selections.

5. Click **Save** to store the template override settings for subsequent exports.

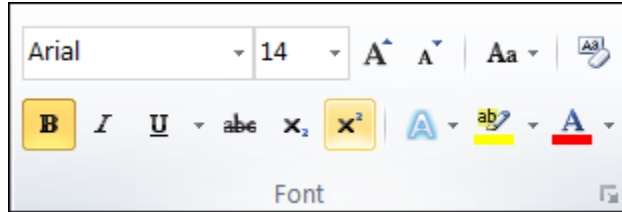
Creating new objects in Word using object templates that contain formatting

When you use an object template that contains specific formatting and styles for exporting requirement structures to Word for offline or online editing, any new objects created in the requirement structure in Word use the same formatting and styles specified by the object template.

Only *inline* styles are preserved; inline styles are defined using Word format options or table menus and do *not* include predefined styles.

For example, inline styling includes:

- Using styles and formats for text that you apply using the Microsoft Word ribbon as the following figure shows:



- Using tables that you create using the **Insert Table** command located in the Microsoft Word ribbon.

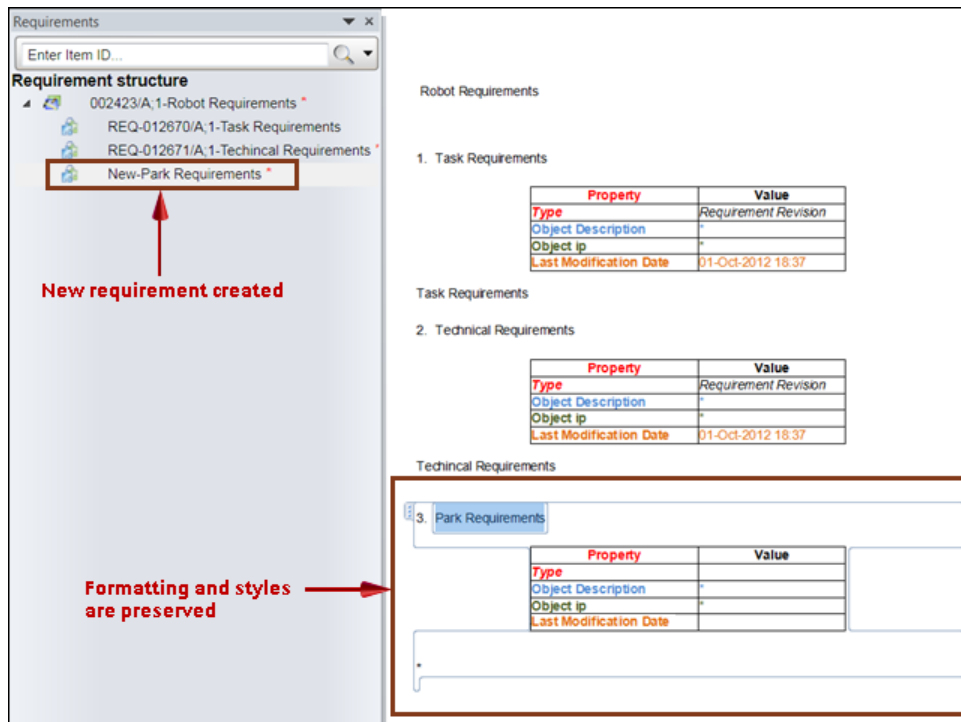
In the following example, you create a new object template for requirement revisions that uses the following format and layout:

1. {%object_name}

Property	Value
Type	<u>{%object_type}</u>
Object Description	<u>{%object_desc}</u>
Object ip	<u>{%ip_classification}</u>
Last Modification Date	<u>{%last_mod_date}</u>

{%body_text}

When you export the requirement structure to Word and override the default template to use the new object template for requirement revisions, the requirements are displayed in Word using the object template format. As the following figure shows, when the new requirement called **Park Requirements** is added, the formatting and styles are preserved.



Capturing a fixed data record in Word

In the **Export to Word** dialog box, the **Static Snapshot** output mode generates a Word document with the standard Microsoft capability. Select this mode when you want to capture a fixed record of the data as of a given time.

For example, you can send the document by e-mail and fax to customers, suppliers, and colleagues.

- The output document contains static data for the objects and properties defined in the specification template and object template that you select.

The templates also determine the document layout and formatting styles.

- The data is not synchronized with Teamcenter.

Changes are not applied in the underlying structure.

The **Copy Snapshot to Dataset** option is available if you select a single object to start the export process. Select this option to generate a Word dataset containing a fixed copy of the output. The dataset is associated with the selected object by a **Static Snapshot** relation. You can copy the dataset's URL to the Teamcenter clipboard by right-clicking the dataset and choosing **Copy**. You can then paste the URL into an e-mail message, for example, and send the message to recipients such as your manager or colleagues. By clicking the URL, recipients can open the dataset in read-only mode.

Note:

- If the snapshot dataset is not visible under the item revision, you must specify **Static Snapshot** as a shown relation for the item revision.
- The content is not synchronized with Teamcenter and changes are not applied in the structure.
- The dataset is not copied forward to new revisions of the selected object.
- If a snapshot dataset exists for the object, it is replaced by a new dataset containing the output of this export process.

Teamcenter stores the previous dataset versions until the version limit is reached.

- To find previous dataset versions with the Teamcenter **Search** feature, you can set the **QRY_dataset_display_option** preference value to **1**.
- You can set the dataset version limit with the **AE_dataset_default_keep_limit** preference.

Display static snapshot datasets

When you use the **Static Snapshot** mode and the **Copy Snapshot to Dataset** option to export data to Word, a Word dataset that contains a fixed copy of the output is generated. For the snapshot dataset to be visible under the item revision, **Static Snapshot** must be selected as a shown relation for the item revision.

1. Choose **Edit**→**Options**.
2. Expand the **Options** tree and expand the **General** folder.
3. Select the **Item Revision** node.
4. On the **General** options tab, choose **Static Snapshot** from the **Available Relations** list.
5. Click **+** to move it to the **Shown Relations** list.
6. Click **Apply**.
7. (Optional) To display snapshots in a pseudofolder:
 - a. On the **Related Object** tab, choose **Static Snapshot** from the **Available Relations** list.
 - b. Click **+** to move it to the **Shown Relations** list.
 - c. Click **Apply**.

8. Click **OK**.

Changing property values interactively with live Word

In the **Export to Word** dialog box, the **Live integration with Word** output mode generates a Word document with Teamcenter live Word capability. Select this mode when you want to change Teamcenter property values directly from Word.

- The output document contains live data for the objects and properties defined in the specification template and object template that you select.

The templates also determine the document layout and formatting styles.

Caution:

- In the object template, 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.

Note:

The **Live integration with Word** output mode ignores landscape orientation in specification templates. The page layout is portrait orientation in all live Word documents.

- The data is synchronized with Teamcenter; you can edit property values interactively.

Changes are applied in the structure when you save and close the document.

This mode contains the following options under **Live Options**.

- **Export for structure editing and work offline**

Generates a live Word document that you can store outside of Teamcenter, for example, on a local or shared drive.

- While offline, the document retains its live capability although changes are not applied to the structure in real time.
- Changes are accumulated pending later connection to the Teamcenter server.
- After establishing connection, you can commit the changes to Teamcenter, or you can cancel the changes and refresh the document with the current structure data.

- **Check out objects before export**

Explicitly checks out the selected objects from Teamcenter before the live Word document is generated.

You reserve access to the objects until you check them in, either when you commit changes to the structure or cancel the changes.

Note:

The **Check out objects before export** option is unavailable by default. You can enable the option by setting a preference in the **Options** dialog box.

Note:

There is no integration between live Office and Client for Office add-ins.

When you use Systems Engineering within the rich client, the live Office add-in disables the Microsoft Office save option and displays its own save button. When saving a requirement dataset in the embedded Microsoft Word window, use the live Office save button, not the save button in the Client for Office add-ins tab.

If you attempt to save a requirement dataset using the Client for Office menu, Client for Office does not recognize the Teamcenter object context and does not know where the dataset originates in Teamcenter.

Sharing comments from review participants

In the **Export to Word** dialog box, the **Export for Markup** output mode generates a Word document containing Teamcenter comments entered by participants in a document review.

You can export markups to:

- Static documents with fixed content.
- Documents with Teamcenter live Word capability.

Note:

Siemens Digital Industries Software does not support markups when different Microsoft Office versions are used. For example, if a markup comment is created in Microsoft Office 2010, it does not open in Microsoft Office 2013.

Export data to Word

Note:

You must have at least **Read** access to all Word export templates for your project.

Tip:

If you anticipate frequent exports of an object type, you can override the default object template for that type before you start this procedure.

1. Set the structure context by doing one of the following:

- To export data for all objects at all levels, select only the requirement specification at the peak level.
- To export data for one object at any level, select only the object.

Data is also exported for all direct children and lower level descendants of the selected object.

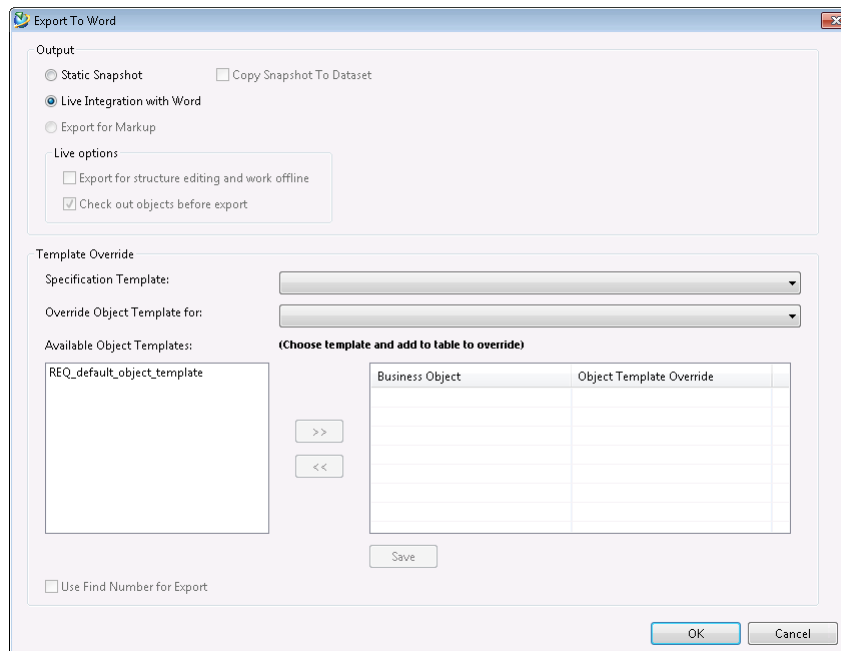
- To export data for two or more objects, use the standard Windows functions for multiple selection.

You can select multiple objects at the same level or at different levels. A multiple selection cannot include the requirement specification at the top level.

Note:

If properties are specified in the specification template's header and footer, the properties of the first selected object are substituted for the template properties in the output document.

2. To open the **Export to Word** dialog box, choose **Tools→Export→Objects To Word** or click **Export objects to Word** on the toolbar.



3. Under **Output**, select one of the following modes:

- **Static Snapshot** generates a standard Word document without Teamcenter live Word capability.

Note:

The **Copy Snapshot to Dataset** check box is available if you select a single object to start the export process. Select this check box to generate an additional Word document containing a static record of the output. The static document is associated with the selected object by a **Fnd0ExportContent** relation.

- **Live integration with Word** generates a Word document with Teamcenter live Word capability.

Caution:

This mode requires an object template that contains the **{%object_name}** property keyword. Otherwise, you cannot edit structure data in live Word.

The **Live integration with Word** output mode ignores landscape orientation in specification templates. The page layout is portrait orientation in all live Word documents.

- **Export for Markup** generates a Word document containing the markups for the selected requirement specification or requirement revision.

Note:

Siemens Digital Industries Software does not support markups when different Microsoft Office versions are used. For example, if a markup comment is created in Microsoft Office 2010, it does not open in Microsoft Office 2013.

- Under **Live options**, you can also select the following.
 - **Export for structure editing and work offline** generates a live Word document that you can store outside of Teamcenter, for example, on a local or shared drive.

In the offline document, you can:

- Accumulate structure edits while disconnected from Teamcenter.
- Commit pending edits to the structure by connecting to the Teamcenter server.
- **Check out objects before export** explicitly checks out the selected objects before the live Word document is generated.

You reserve access to the objects until you check them in, either when you commit changes to the structure or cancel the changes.

Note:

By default, the **Check out objects before export** ☒ check box is selected and unavailable. To enable the check box, you must change the value of the **REQ_checkout_objects_before_export** preference.

4. (Optional) To use Word export templates other than the defaults, do the following under **Template Override**:

- To use a different specification template than the one originally assigned to the object, select it from the **Specification Template** list.
- To use a different object template, do one of the following:

Caution:

If you select the **Live integration with Word** output mode, you must select an object template that contains the **{%object_name}** property keyword. Otherwise, you cannot edit structure data in live Word.

- For a particular business object in the selected context:
 - a. Select the business object from the **Override Object Template for** list.

- b. Select the template to use in the **Available Object Templates** pane, and then click **»**.
 - For all business objects in the selected context:
 - a. Leave the value blank in the **Override Object Template for** list.
 - b. Select the template to use in the **Available Object Templates** pane, and then click **»**.
 - To remove a business object/template override combination from the list, click **«**.
 - To store template override settings for subsequent exports, click **Save**.
5. Click **OK** to generate the output document in a new Word window.

Generate in	Output document
Static Snapshot mode	No interactive capability with Teamcenter and cannot be used to edit property values in the requirement structure. • The initial document is read-only. • You can create a permanent document by choosing File→Save As to display the Save As dialog box. Note: If you select the Copy Snapshot to Dataset check box, the output is copied to an additional Word document containing a static record of the output. If the snapshot dataset is not visible under the item revision, you must specify Static Snapshot as a shown relation for the item revision.
Live Integration with Word mode	Synchronized with the Teamcenter rich client. You can change property values in the requirement structure by editing corresponding values in the live Word document.
Export for Markup mode	You can add, delete, save, and edit your own markups in the document using the Markup Manager in Word. You cannot change other users' markups.

Exporting data to Microsoft Office Excel

Exporting data to Microsoft Office Excel

A Microsoft Office Excel export file contains the values of Teamcenter properties that are specified in a selected Excel template. The output file can be generated in different modes, depending on your purpose.

- Excel export templates control both the format and the content of exported Teamcenter data.

An export template specifies the properties whose values are exported for the selected objects. It also defines the output format of the exported data.

Note:

To further control exported content, you can add closure rules and transfer modes to Excel templates. Closure rules are not allowed in the body text of requirements.

- Exports of Teamcenter data are either static (unchangeable) or live (modifiable).
- Using live Excel, updates to data in the exported Excel spreadsheet result in real-time updates to Teamcenter.
- Live Excel export includes a bulk mode feature. You may want to use this feature to enhance Excel export performance in certain situations.

Note:

Microsoft Excel 2010, or a later version, is required to use live Excel.

Capturing a fixed data record in Excel

In the **Export to Excel** dialog box, the **Static Snapshot** output mode extracts the property data to a standard Excel file that is not synchronized with or connected to Teamcenter. Select this mode when you want to capture a fixed record of a structure as of a given time.

For example, you can:

- Export the current content of structure elements and save the file for comparison with subsequent revisions.
- Export structure elements hierarchically to show ongoing relationships within the structure.

You can save the export file and share it with others outside of Teamcenter, for example, customers, suppliers, and colleagues. You can send the file by e-mail and by fax.

Editing properties interactively in Excel

In the **Export to Excel** dialog box, the **Live integration with Excel (Interactive)** output mode extracts the property data to an Excel file with Teamcenter live Excel capability. Select this mode when you want to change Teamcenter property values directly from Excel.

Interactive mode allows you to use live Excel:

- While the Teamcenter rich client is running.
 - The live Excel file is connected to the Teamcenter server and the data is synchronized with Teamcenter.
 - Changes in live Excel are applied in real time to the corresponding Teamcenter objects.
- *Offline* from the Teamcenter rich client, by choosing **File→Save As** in the output file to store it outside of Teamcenter, for example, on your local desktop or on a shared drive.
 - The offline file is unconnected to the Teamcenter server but retains its live capability.
 - You can accumulate pending changes in the offline file by choosing **File→Save** in Excel.
 - Later, you can commit pending changes to Teamcenter by connecting the offline file to the Teamcenter server.

Tip:

If you select the **Check out objects before export** option, you can explicitly check out the selected objects from Teamcenter before the live Excel file is generated. You reserve access to the objects until you check them in, either when you commit changes to the structure or cancel the changes.

The **Check out objects before export** option is cleared by default. You can configure the object checkout option by setting the **Show_Checkout_option** preference in the **Options** dialog box.

Live Excel is included in Teamcenter Extensions for Microsoft Office and is available only for the four-tier Teamcenter rich client.

Processing Excel edits in bulk

In the **Export to Excel** dialog box, the **Live integration with Excel (Bulk Mode)** output mode reduces overhead in communicating with the Teamcenter server, for example, in committing a large number of changes to Teamcenter. Select this mode if you want to make multiple changes in the live Excel file and commit all changes to Teamcenter at the same time.

- To commit the changes, click **Save to Teamcenter** on the **Add-Ins** tab in Excel.

- To undo changes, select the cells containing the changes and click **Refresh** on the **Add-Ins** tab in Excel.

Tip:

If you select the **Check out objects before export** option, you can explicitly check out the selected objects from Teamcenter before the live Excel file is generated. You reserve access to the objects until you check them in, either when you commit changes to the structure or cancel the changes.

The **Check out objects before export** option is cleared by default. You can configure the object checkout option by setting the **Show_Checkout_option** preference in the **Options** dialog box.

Live Excel is included in Teamcenter Extensions for Microsoft Office and is available only for the four-tier Teamcenter rich client.

Preparing structure changes offline in Excel

The **Work Offline and Import** output mode extracts object property data to an Excel file that is preconfigured for subsequent importing to Teamcenter through Teamcenter Client for Microsoft Office. Select this mode when you want to combine structure data and import processing instructions in a single input file.

The import process uses this input to create or modify the corresponding structure in Teamcenter. The preconfigured file saves time and effort in setting up the input.

Caution:

- The output file cannot be used for a subsequent import if the selected structure exceeds Excel's eight-level limit. In this file, all objects are exported to the same level and Excel's **Outline** features are disabled.
- Before you start the export process, ensure that you do not select objects based on types other than **Item** and **Item Revision**. Otherwise, the subsequent import fails without notification.

Working around Excel's eight-level limitation

Excel's outline hierarchy is limited to eight levels. When the selected Teamcenter hierarchy exceeds that limit, all objects are exported to an unstructured file. Excel's **Outline** features are disabled in this file.

Data cannot be imported if the **Outline** features are disabled on Excel's **Data** tab.

To work around this limitation, you can:

- Export the structure to a standard Excel file by selecting **Static Snapshot** in the **Export to Excel** dialog box.

- Edit the static output file to match the object and relationship hierarchy in the structure.
- Initiate the import process from the edited static file.

Note:

Associate the disabled output file's control sheet with the edited static file.

Export data to Excel

Note:

- If the selected structure exceeds Excel's eight-level limit, it is exported to an unstructured file. Excel's **Outline** features are disabled and all objects occupy the same level.

To reference the Teamcenter hierarchy in the unstructured file, you can set the **REQ_ExportToExcel_add_level_column** preference to insert the **Level** column. This column takes the leftmost position, where each cell shows an object's structure level in Teamcenter.

Unstructured files from the **Work Offline and Import** option cannot be used for a subsequent import.

- The sort order in the Teamcenter structure does not affect the object sequence in the output file. Each object is exported according to its position in the structure hierarchy. In Systems Engineering, an object's **Number** property determines the row in which that object is placed in the output file.
- You must have at least **Read** access to all Excel export templates for your project.

1. Set the structure context by doing one of the following:

- To export all objects at all levels, select only the top-level object.
- To export a single object below the top level, select only that object.

Note:

The selection includes child objects at lower levels.

- To export two or more objects below the top level, use the standard Windows functions for multiple selection.

You can select multiple objects at the same level or at different levels. The selection cannot include the object at the top level.

Note:

The selection does not include child objects at lower levels.

- To export structure search results to Excel and include the top-level object, click any object in the search results table and press **Ctrl-A** to select all of the objects.

The output spreadsheet for structure search results does not contain data for the top-level object if you select only that object.

- When you select a folder in Teamcenter and export the folder to Excel:
 - All the contents of the folder are exported to Excel.
 - If the template contains a transfer mode, additional secondary objects are exported as a result of the execution of the transfer mode.

Note:

As a result of the execution of the transfer mode, secondary objects may already be exported as contents of the folder. This may result in duplicate objects in Excel.

For example, suppose you select a **Parent** folder in My Teamcenter and export to Excel using a custom template and transfer mode containing a conditional clause to skip any folders named **Child3**.

The folder structure is as follows:

```

Parent
    | ---Child1
    |   | ---Child3
    | ---Child2
    |   | ---Child3
    | ---Child3
    |   | ---Child4
  
```

You create a transfer mode named **Folder2** with the following closure rule:

```

CLASS.Folder:CLASS.Folder:PROPERTY.contents:PROCESS+TRAVERSE:PRIMARY
.object_name!="Child3"
  
```

When you export the **Parent** folder, all its children are exported. Excel export uses the transfer mode only to collect additional secondary objects. In this example, the **Child1**, **Child2**, and **Child3** folders just below the **Parent** are secondary objects. However, when the transfer mode is traversed, only the secondary **Child3** folder is skipped, but other secondary folders, namely

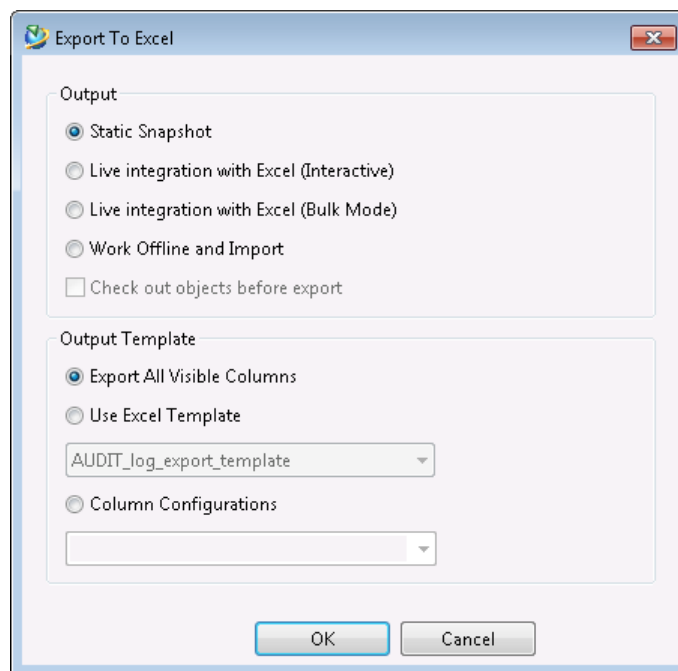
Child1 and **Child2** are exported, resulting in duplicate **Child1** and **Child2** folders. In such cases, duplication of objects during export is unavoidable.

Note:

PLM XML export and Excel export are completely different and the way they process objects using transfer modes is different, so the PLM XML output cannot be compared with the output of the Excel export. For examples of using transfer modes with Excel export, see [Using transfer modes with Excel export](#).

2. Choose **Tools→Export→Objects To Excel**.

The **Export To Excel** dialog box opens.



3. Under **Output**, select one of the following modes:
 - **Static Snapshot** exports the data to an Excel file with the standard Microsoft capability.
 - **Live integration with Excel (Interactive)** exports the data to an Excel file with Teamcenter live Excel capability.
 - **Live integration with Excel (Bulk Mode)** exports the data to an Excel file with Teamcenter live Excel capability, but does not commit the changes to Teamcenter until you click **Save to Teamcenter**.
 - **Work Offline and Import** exports the data to an Excel file that also contains import processing information on a separate sheet.

Caution:

Only items, item revisions, and their subtypes can be imported from offline Excel files. Before you start the export process, ensure that you do not select objects based on types other than **Item** and **Item Revision**. Otherwise, attempts to import such offline data fail without notification.

Note:

Export using work offline mode exports properties such as **item_id**, **object_name** and the real type name, so that the data can be imported correctly to Teamcenter. These properties are exported even if they are not specifically selected.

Only numeric and string properties are supported for import from Excel. Read-only, reference, and relation properties should be marked as **Ignored** in the Excel control file sheet. Run-time properties also are not supported for work offline.

- If you select **Live integration with Excel (Interactive)** or **Live integration with Excel (Bulk Mode)**, you can also select the **Check out objects before export** option.

The selected objects are explicitly checked out from Teamcenter before the live Excel document is generated. You reserve access to the objects until you check them in, either when you commit changes to the structure or cancel the changes.

Caution:

The checkout applies to all objects being exported. Use this option carefully if you are exporting a large set of objects or an entire specification.

Note:

The **Check out objects before export** option is cleared by default. You can configure the object checkout option by setting the **Show_Checkout_option** preference in the **Options** dialog box.

4. Under **Output Template**, do one of the following:

- To export the property data in the currently displayed columns, select **Export All Visible Columns**.

Note:

When you export all visible columns to Excel, the relation column is not exported because the relation is not a property on the object.

- To export the property data specified in an Excel template:
 - a. Select **Use Excel Template** to activate the template list.
 - b. Select the applicable template from the list.

Caution:

The template list is blank if you do not have **Read** access to any one Excel export template for your project.

Excel templates also specify the output file's overall layout, including headers, footers, and formatting styles.

- To export the property data in the columns defined in a saved column configuration:
 - a. Select **Column Configurations** to activate the column configurations list.
 - b. Select the applicable configuration from the list.

Tip:

- **Work Offline and Import**
- **Check out objects before export**
- **Export All Visible Columns**
- **Column Configurations**

5. Click **OK** to generate the output file in a new Excel window.

Caution:

Do not use Excel's native sorting and filtering features in the output file. The exported objects and properties must remain aligned with their counterparts in the structure.

Note:

An informational message may state that the export process skipped some objects because the Excel template does not contain rules for those objects. To suppress this message in the future, select the **Don't show this message again** checkbox. You can restore the message by clearing your client cache.

Generate in	Output file
Static Snapshot mode	No interactive capability with Teamcenter and cannot be used to edit properties in the requirement structure. You can save the file, for example, to: - Record the data as of a given time. - Distribute the data to customers and suppliers by e-mail or fax.
Live integration with Excel (Interactive) mode	Synchronized with the Teamcenter rich client. You can change property values in the requirement structure by editing corresponding values in the output file.
Live integration with Excel (Bulk Mode) mode	Changes saved to Teamcenter rich client once you click Save to Teamcenter. You can change property values in the requirement structure by editing corresponding values in the output file.
Work Offline and Import mode	Automatically configured for subsequent importing to Teamcenter. In Teamcenter Client for Microsoft Office, you can import this data to create and modify structure elements.

Using transfer modes with Excel export

A transfer mode is created in the PLM XML Export Import Administration application and is displayed as an option when users import or export objects to or from Teamcenter applications.

The objects collected from the transfer mode are added to the list of objects to be exported and each object is then matched with the rules in the rule table to find the best matching rule (from top to bottom). If a rule matches, the properties corresponding to that rule are considered for export. If no rule matches, the object is skipped for export. The rule table in the Excel template acts as a filter to the objects being exported.

A transfer mode has a closure rule that defines a traversal algorithm to identify the additional objects required for a specific principal object. For example, the following closure rule ensures that when you export item revisions, the datasets and master forms attached to them are exported. The following Excel sheet contains the properties table on the left and the **rule table** on the right.

Siemens PLM Software

Teamcenter Excel Export

"Live" edit on this spreadsheet requires Teamcenter Extensions for Microsoft Office

Type	Object	Name	ID	Owner	Group	UserData1	Data Modified	CO
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}
{%object_type}	{%object_string}	{%object_name}	{%item_id}	{%owning_user}	{%owning_group}	{%user_data_1}	{%last_mod_date}	{%checked_out}
{%object_type}	{%object_string}	{%object_name}	{%item_id}	{%owning_user}	{%owning_group}	{%user_data_1}	{%last_mod_date}	{%checked_out}
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.								

The **ItemRevTM** transfer mode is defined as shown here:

PLM XML Export Import Administration

TransferMode

Name: ItemRevTM *

Context:

Description:

Type of Transfer: ☒ Export ☐ Import

Output Schema Format: PLMXML

Support Incremental: ☐

Closure Rule: REV_RULE

Filter Rule:

Property Set:

Revision Rule:

Action List:

List of defined tools:

List of selected tools:

The **REV_RULE** closure rule specifies that when you export item revisions, the datasets and master forms attached to them are exported. The **REV_RULE** closure rule is defined as shown here:

PLM XML Export Import Administration

ClosureRule

Traversal Rule Name: REV_RULE *

Description:

Scope of Traversal: ☒ Export ☐ Import

Output Schema Format: PLMXML

Primary O...	Primary O...	Secondary...	Secondary...	Relation Type	Related Property Or Object	Action Type	Condition...
CLASS	ItemRevision	CLASS	Form	RELATIONP2S	IMAN_master_form_rev	PROCESS+...	
CLASS	ItemRevision	CLASS	Dataset	RELATIONP2S	IMAN_specification	PROCESS+...	

The output contains item revisions, item revision master forms, and Microsoft Word datasets. The output is displayed in an Excel file as shown here:

Type	Object	Name	ID	Owner	Group	User	Data1	Data Modified	CO
ItemRevision	000023/A;1-Car	Car	000023	infodba (infodba)	dba			18-May-2012 17:51	
ItemRevision Master	000023/A	000023/A		infodba (infodba)	dba			18-May-2012 17:40	
MSWord	Dataset_Car	Dataset_Car		infodba (infodba)	dba			18-May-2012 17:46	
ItemRevision	000019/A;1-EngineA	EngineA	000019	infodba (infodba)	dba			18-May-2012 17:45	
ItemRevision Master	000019/A	000019/A		infodba (infodba)	dba			18-May-2012 17:39	
MSWord	Dataset_EngineA	Dataset_EngineA		infodba (infodba)	dba			18-May-2012 17:43	
ItemRevision	000020/A;1-ShaftA	ShaftA	000020	infodba (infodba)	dba			18-May-2012 17:46	
ItemRevision Master	000020/A	000020/A		infodba (infodba)	dba			18-May-2012 17:39	
MSWord	Dataset_ShaftA	Dataset_ShaftA		infodba (infodba)	dba			18-May-2012 17:43	
ItemRevision	000021/A;1-EngineB	EngineB	000021	infodba (infodba)	dba			18-May-2012 17:45	
ItemRevision Master	000021/A	000021/A		infodba (infodba)	dba			18-May-2012 17:40	
MSWord	Dataset_EngineB	Dataset_EngineB		infodba (infodba)	dba			18-May-2012 17:43	
ItemRevision	000022/A;1-ShaftB	ShaftB	000022	infodba (infodba)	dba			18-May-2012 17:46	
ItemRevision Master	000022/A	000022/A		infodba (infodba)	dba			18-May-2012 17:40	
MSWord	Dataset_ShaftB	Dataset_ShaftB		infodba (infodba)	dba			18-May-2012 17:44	

Excel template rule tables

Rule tables enable you to filter the data. There are four key options in a rule table with which you can do additional filtering:

- **Level**
- **Relation**
- **Type**
- **TransferMode**

	Level	Relation	Type	TransferMode
<rule>	{%L.2}			
<rule>		{%R-FND_TraceLink}		
<rule>			{%S.Item}	
<rule>				{%T-Req_default_TM}

Note:

A rule table can have different keys set separately or together depending on what information you would like to export.

Property tags ({%RealPropertyName}) should not be added to the rule table.

A rule table is not exported with data.

Processing of rules in a rule table

Rules in Excel template rule tables are processed as follows:

- From top to bottom, beginning with the topmost rule in the table.
- 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.
- If the current rule is completely blank, all criteria are matched by default. Place rules containing more key field criteria above rules with fewer key field criteria.
- 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 is displayed which states that objects that do not match any template criteria have been ignored.

Rule Table – Level Key

The Level key specifies the hierarchical level of the objects you want to export. It filters the other objects so that only the selected level is displayed in the export. For example, when **2** is the argument, objects at the second level in the structure are exported. The level key applies only when exporting data from structure editors, such as Structure Manager, Multi-Structure Manager, and Systems Engineering (Requirements Manager).

Syntax: {%L-Level}

Example: {%L-2}

	Level	Relation	Type	TransferMode
<rule>	{%L-2}			
<rule>		{%R-FND_TraceLink}		
<rule>			{%S-Item}	
<rule>				{%T-Req_default_TM}

- Exporting by level lets you select a single or multiple levels to include in your export.

- Each level must be reflected in its own rule in the rule table. The levels do not have to be contiguous.

	Level	Relation	Type	TransferMode
<rule>	{%L-0}			
<rule>	{%L-2}			

Rule Table – Relation Key

The Relation key filters data based on the relation between two objects. In the property table (between the <Start> and <End> tags) you must specify the relation property using the same tag in the form {%R-Relation Property}. An example is {%R-name} for the name of the relation. All GRM relations are supported.

Syntax: {%R-Relation-Name}

Example: {%R-TC_Attaches}

	Level	Relation	Type	TransferMode
<rule>	{%L-2}			
<rule>		{%R-FND_TraceLink}		
<rule>			{%S-Item}	
<rule>				{%T-Req_default_TM}

- Export by Relation filters data based on the relation between two objects. The relation that is returned is based on the primary object in the selection. Only the relation on the primary object is returned.
- The returned Relation properties are not displayed in the results unless you add relation property references in the property table in the form {%R-RelationProperty}.

Group	Relation Name	Relation Desc	Relation Type Name		Level	Relation
{%owning_group}	{%R-name}	{%R-description}	{%R-relation_type}	<rule>		{%R-FND_TraceLink}

- This filtering option can be used with other filtering options in the rule table. In this example, we demonstrate the setup for filtering by Level and Relation. This returns the trace links only on Level 1 objects.

	Level	Relation	Type	TransferMode
<rule>	{%L-1}	{%R-FND_TraceLink}		

Rule Table – Type Key

The Type key specifies the business object subtype to filter in the exported Excel template. For example, when you have a structure containing documents, parts, and items and you only want to export the items, specifying {%S-Item} restricts the export only to the item subtype.

Syntax: {%S-Type-Name}

Example: {%S-Item}

	Level	Relation	Type	TransferMode
<rule>	{%L-2}			
<rule>		{%R-FND_TraceLink}		
<rule>			{%S-Item}	
<rule>				{%T-Req_default_TM}

- Exporting by Type allows you to filter one or more specific subtypes from the exported structure.
- Each subtype desired needs to be reflected in its own rule in the rule table.

	Level	Relation	Type	TransferMode
<rule>			{%S-ItemRevision}	
<rule>			{%S-PartRevision}	

- Filtering on the subtype can also be used with other filtering rules. The following example filters all Level 2 requirements only.

	Level	Relation	Type	TransferMode
<rule>	{%L-2}		{%S-RequirementRevision}	

Rule Table – TransferMode Key

The TransferMode key specifies the transfer mode to be applied during the export to Excel to get additional properties or limit them with the use of PLM XML transfer modes. An example of this is exporting an item revision with properties from the item revision master form.

Syntax: {%T-TransferMode}

Example: {%T-Req_default_TM}

	Level	Relation	Type	TransferMode
<rule>	{%L-2}			
<rule>		{%R-FND_TraceLink}		
<rule>			{%S-Item}	
<rule>				{%T-Req_default_TM}

- 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.

To filter using a transfer mode in the Excel rule table:

- You must create a transfer mode or use a pre-existing transfer mode.
- You must designate the transfer mode in the rule table. The transfer mode key needs to include the subtype in which it will be used in conjunction with the data collection.

Note:

The transfer mode should be designed to work from the selected perspective, such as from a BOMLine view (Structure Manager) or an Object View (Details Panel). It could be designed to work from both.

	Level	Relation	Type	TransferMode
<rule>			{%S-ItemRevision}	{%T-ThisColor}

Exporting objects to Teamcenter email

Exporting objects to Teamcenter email

Using the internal email facility, you can send object data from Systems Engineering to other Teamcenter users. You can also send the data to SMTP addresses for recipients who are not Teamcenter users.

Extracted through the export process, the data consists of object property values specified in a Microsoft Office export template that you select. You can export the data to Word or Excel in the static mode or the live mode.

- In the static mode, the output document itself is attached to the email message.

Recipients can save the attachment locally and open it from the message.

The content is a fixed record of the data as of a given time. Changes cannot be applied to the underlying structure in Systems Engineering.

- In the live mode, the output document's URL is inserted in the email message body.

Recipients can click the URL to open the document and edit the data.

Changes are committed to the underlying structure in Systems Engineering.

Certain Teamcenter preferences govern the internal email facility.

- **Mail_server_name**

- **Mail_server_port**

The default value is **25**.

- **Mail_OSMail_activated**

The default value is **true**.

- **Mail_internal_mail_activated**

The default value is **true**.

These preference values can be customized for your environment.

Send structure data by email

1. Set the structure context by doing one of the following:

- To send data for all objects in a structure, select the parent object.

This context includes data for all direct children and lower level descendants.

- To send data for two or more objects, use the standard Windows functions for multiple selection.

This context specifies data for the selected objects only.

2. Choose **File**→**Send via Email**, or right-click the selection and choose **Send via Email**.

The **Send via Email** dialog box is displayed.

3. To address the data, do one or both of the following:

- For Teamcenter users, click the **TO** or **CC** button.

The **Select Recipients** dialog box is displayed.

- Type search criteria in the **Search** box to search by user ID, group, or address list.
- Click one of the following buttons to execute the search: **User** , **Groups** , or **Address List** .
- Select the users, groups, or address lists that you want to receive the message, and click the **To** or **CC** buttons.

Tip:

You can remove recipients from the lists by selecting the user and clicking the **Remove** button (–). In addition, you can create a new address list by clicking the **Launch Address List** button .

- Click **OK** to close the **Select Recipients** dialog box.

- For external users, type the SMTP addresses in the **TO** or **CC** boxes.

- Type a message description in the **Subject** box.

In the **Message** box, you can enter additional text for the message body, up to 240 characters.

- In the **Requirement Templates** list, click the export template that specifies the data you want to send.
- To specify the output document format, click the **Word** button or the **Excel** button.
- To specify the export mode, select **Static** or **Send live URL**.
- Click **send** to send the message.

Access structure data in email

- Open the Teamcenter email message.
- Do one of the following:
 - If a Word or Excel document is attached to the message, save the attachment locally or open it from the message.

The content is a fixed record of the data as of a given time. You cannot save changes to the underlying structure in Systems Engineering.

- If the message body contains a URL, click the URL to open the document in live Word or live Excel.

You can edit this document and save the changes to the underlying structure in Systems Engineering.

Export objects in the PLM XML format

1. Select the objects to be exported.

You can export only a single BOM. Do not configure multiple BOM structures for export as PLM XML.

2. Choose **Tools**→**Export**→**To PLMXML**.

The system displays the **PLM XML Export** dialog box.

3. In the **Export Directory** box, select the directory in which to place the PLM XML file. Click **Browse** to locate the directory.
4. Specify the name of the export file in the **Export Filename** box.
5. Choose a transfer mode from the **Transfer Mode Name** list.
6. (Optional) Click **Select Languages** to display the **Language Selection** dialog box for the languages configured at your site.
 - The **Available Languages** box lists the languages you can select as additional languages for export.
 - The **Select Languages** box lists the preconfigured languages for the selected transfer mode.
 - To select additional languages for export, follow these steps:
 - a. Select a language in the **Available Languages** box.
 - b. Click the **Add** button .
 - The selected language is added to the **Selected Languages** box.
 - To reorder the languages in the **Selected Languages** box, select a language and use the move   buttons as required.
 - c. Click **OK** or **Apply**.

The languages selected for export are displayed in the **Languages** box.

Note:

If you change the transfer mode, the **Languages** entries remains unchanged. You can use the **Language Selection** dialog box again to add or remove more languages.

7. (Optional) Select a revision rule to configure the assembly to be exported for the selected root object.

Note:

If you select multiple root objects, no revision rule is applied, so only the selected root objects, not the structure, are exported.

8. (Optional) Select **Open PLM XML File** to view the file when the export operation is complete.
9. (Optional) Select the **Perform Export In Background** check box to perform an asynchronous export.
10. Click **OK** to export the objects in the **Object List** list and close the dialog box.

When the export completes, a dialog box displays.

If the export was successful, you can click **Yes** in the dialog box to view the log file for the export.

If there were errors during the export, the dialog box lists the errors. Click **Yes** to view information about the errors that occurred.

Note:

Exports that completed without errors in past versions may now show errors. Previously, unless the **PLMXML_log_file_content** preference was set to **detailed**, errors did not display. The detailed error information now displays in the log file by default. Setting the **PLMXML_log_file_content** preference to **detailed** produces detailed information of all objects (those successfully exported and those exported with errors).

Export objects to an application interface

Application interface objects store parameters required to share data in PLM XML format, allowing you to repeatedly share data with other sites without defining the parameters each time you export the data.

1. Select the objects to be exported.
2. Choose **Tools→Export→Objects**.

The system displays the **Export** dialog box.

- Click **AppInterface** in the left pane of the dialog box.
- Choose an application interface by clicking **Find Application Interface**  or create a new application interface by clicking **Create Application Interface**  and entering information in the following boxes:

Name	Description
Description	Describes the application interface (optional).
Import Transfer Mode Name	Specifies the transfer mode used when objects are imported into your database.
Export Transfer Mode Name	Specifies the transfer mode used when objects are exported from your database.
Target Sites	Specifies the sites to which objects will be exported when this application interface is used.
Active Incremental Change	Specifies the incremental change into which objects will be imported. The buttons to the right of the Active Incremental Change box enable you to create an incremental change object, search for a change, or select from the most recently used changes. In addition, the Information button allows you to view or modify the active change.

- (Optional) Choose **Is Partial Export**.
- (Optional) Choose **Open on Export**.
- Click **OK** to export the objects in the **Object List** and close the dialog box.

Export objects in the Teamcenter format

- Select the objects to be exported.
- Choose **Tools→Export→Objects**.

The system displays the **Export** dialog box.

- Click **Teamcenter** in the left pane of the dialog box.
- Enter the path to the directory containing the objects to be exported in the **Parent Directory** box.

(Optional) Click **Browse** to locate the directory.

5. Specify the name of the export directory in the **Export Directory** box.
6. (Optional) Type comments related to the export operation in the **Reason** box.
7. Use the buttons to the right of the **Target Sites** list to select the Multi-Site Collaboration export destinations for the objects.
8. (Optional) Set the export preferences by clicking **Export Settings** .
9. Click **OK** to export the objects in the **Object List** and close the dialog box.

Export objects in the STEP format

Note:

You can view STEP files in the stand-alone viewer only. In addition, to work with STEP files in the stand-alone viewer, you must have the optional STEP geometry license.

1. Select the objects to be exported.
2. Choose **Tools→Export→Objects**.

The system displays the **Export** dialog box.

3. Click **STEP** in the left pane of the dialog box.

The system displays the **STEP Export** dialog box.

4. Enter the path to the directory containing the objects to be exported in the **Export Directory** box.
(Optional) Click **Browse** to locate the directory.
5. Specify the name of the export file in the **Export Filename** box.
6. (Optional) Enter comments related to the export operation in the **Export** box.
7. Select an **Export Format** option: **AP 203** or **AP 214**.
8. (Optional) Select the **View Log File** option to view the log file when the export operation is complete.
9. (Optional) Set the export preferences by clicking **Export Settings** .
10. Click **OK** to export the objects in the **Object List** and close the dialog box.

Note:

The output *filename.step* file contains metadata only; CAD geometry data is exported separately along with the *filename.step* file. Both the metadata and geometry data can be imported into Teamcenter.

8. Using allocations

Understanding allocations

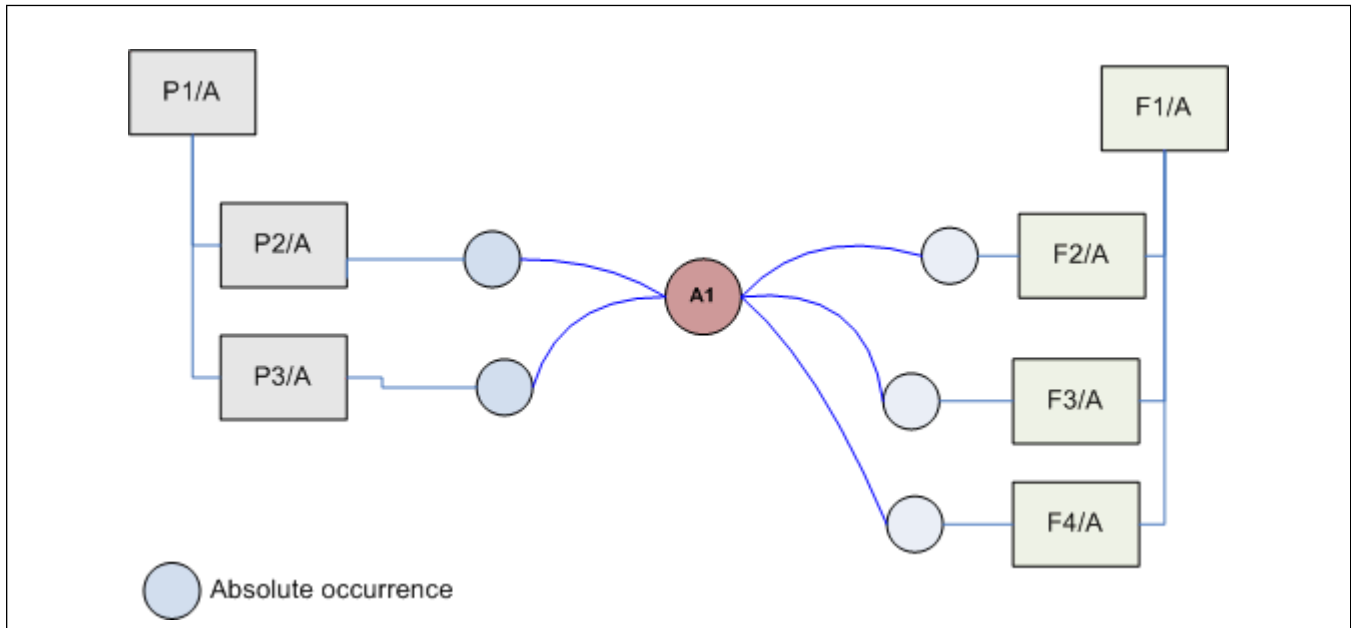
During the development process, a product evolves through various revisions of the product structure and its components. It also evolves through various *representations*. For example, during system analysis and engineering, you express the representation of the product in terms of a set of requirements the product must satisfy and a set of functions it is expected to perform. As the product design matures, you may identify other representations, including the logical structure, the assembly structure and the manufacturing structure. All these representations are related and a change to one representation may necessitate changes to other representations. For example, a change to one of the product requirements may cause the product to perform an additional function. A new physical component is added to perform this function and, in turn, a new manufacturing process is needed to assemble this component into the product. You can capture, locate, and manage these dependencies between representations with allocations.

Using allocations between multiple sources and targets

An allocation represents a directional relationship between a specific instance of an item revision in one product structure and one or more item revisions in another structure. For example, you might relate lines in the functional structure to the logical structure or the logical structure to lines in the physical structure. These relationships may span multiple configurations and revisions of both structures. They may change according to the overall configurations or the specific revisions that are configured into the structures.

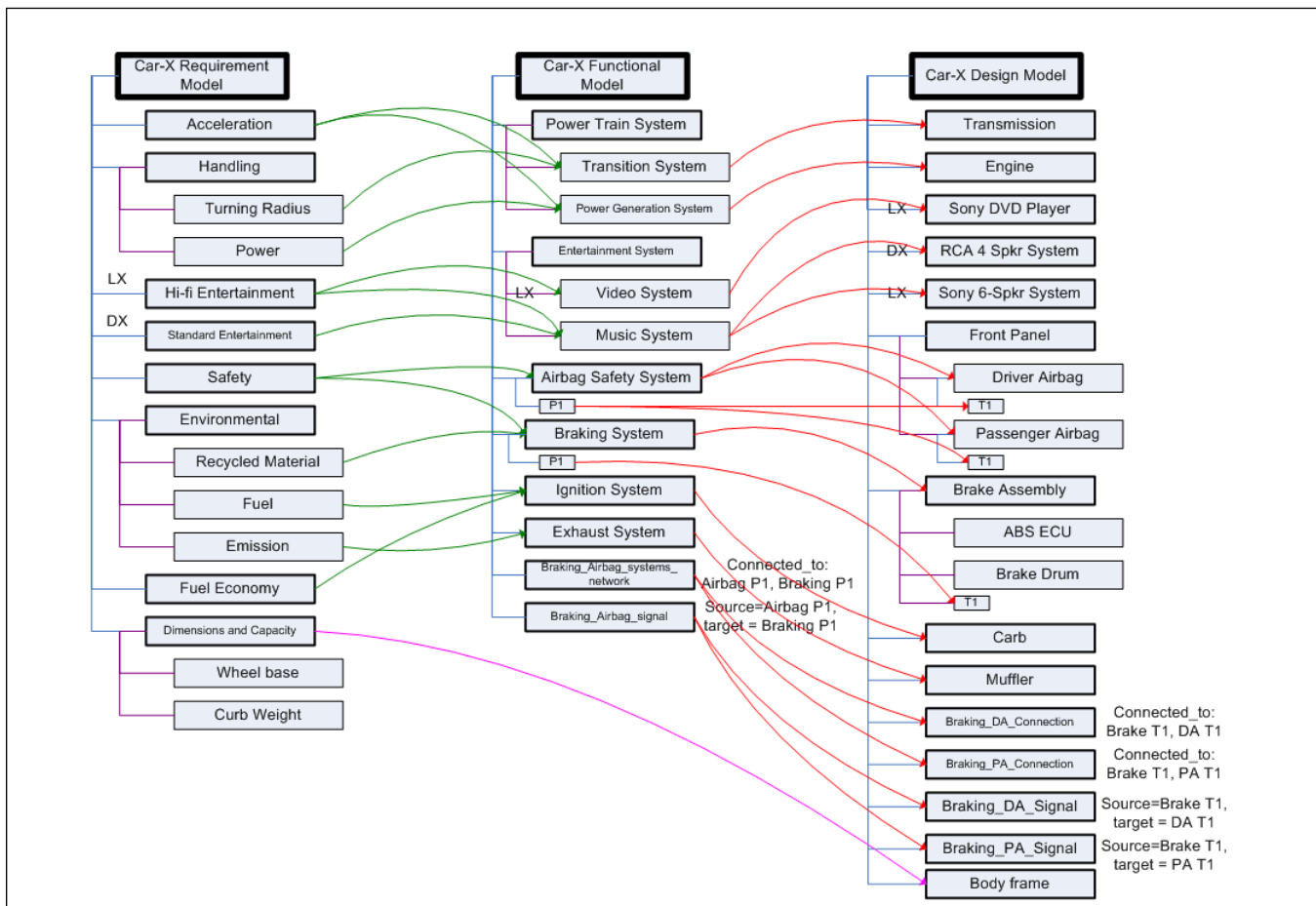
A group of allocations that map together two or more structures is referred to as an *allocation map* or allocation context. An allocation map is a Teamcenter object that can be revised separately from the structures that it maps. However, depending on your business rules, you may want to revise the allocation map at the same time as you revise the associated structures. When you revise an allocation map, you create an allocation map revision in the same way that revising an item creates an item revision. Depending on your business logic, the allocations in the allocation map may be carried forward or dropped from the new allocation map revision; this behavior is determined by preference settings.

You can use allocations to map multiple sources to multiple targets in more than one structure, if the combinations are valid according to your business rules. In the following example, allocation A1 maps two sources (**P2/A** and **P3/A**) to three targets (**F2/A**, **F3/A**, and **F4/A**).



Allocations between multiple sources and targets

You create allocations that map several views or representations of the same structure. For example, you may create three allocation maps that map the requirements, functional and assembly views of a product, as follows.



Allocations between multiple views or representations

To achieve this mapping, you would create three allocation maps in three different contexts:

- **Req-Funct-Map/A** between the requirements view and the functional view.
- **Req-Assm-Map/A** between the requirements view and the assembly view.
- **Funct-AssmMap/A** between the functional view and the assembly view.

Note that you can only work with a maximum of two open structures at any given time. Consequently, you must create these three allocation maps one at a time.

Each of these allocation maps may also be revised independently of the others. The requirements view may be managed in Teamcenter systems engineering and requirements management, while the other views are managed in core Teamcenter.

If you apply configuration rules independently to the source and target structures, you may cause inconsistencies between components configured in the two views. You can apply the same configuration criteria to the source structure, target structure and the allocations mapped between them. To achieve this, you can create a collaboration context and the corresponding configuration context in the Multi-

Structure Manager application. When you place your structures and allocation map in the collaboration context, the specified configuration context configures everything in the collaboration context identically unless you manually override it.

Allocations have a direction, that is, each allocation has a source and a target, for example, **Function 4 is allocated to ECU 1**. An allocation whose source is occurrence A and whose target is occurrence B is not the same as an allocation whose source is occurrence B and whose target is occurrence A.

You can make a *where allocated* check on a selected occurrence to view its relationships in multiple representations. Teamcenter shows the occurrences mapped to the selected occurrence as sources or targets, depending on whether you chose an *allocated from* or an *allocated to* check.

You can optionally implement user exits to determine the allocations between structures are complete and correct according to your company's business rules. For details of how to implement user exits with allocations, see the *Teamcenter Server Customization*.

Choosing allocations or trace links

Teamcenter provides two methods of linking objects in a structure, *allocations* and *trace links*. To help you choose the most appropriate method, the following table lists their advantages.

	Allocation	Trace link
Can additional attributes be added to relationship?	Yes	Yes
Relationship supported with additional data object?	Yes	No
Can additional data object have occurrences?	No	Not applicable
Can be extended in the Business Modeler IDE?	Yes	Yes
Configuration management support?	Yes (revision support for allocation maps; incremental change for allocations)	No
Multi-Site Collaboration support?	No	No
Directional?	Yes	Yes
Maps how many structures?	n (each occurrence must still be a source or target)	Two (typically)
Map $n:m$ objects?	Yes	No
Accountability check support?	No	Yes

Implementing allocations in Teamcenter

Understanding allocations in Teamcenter

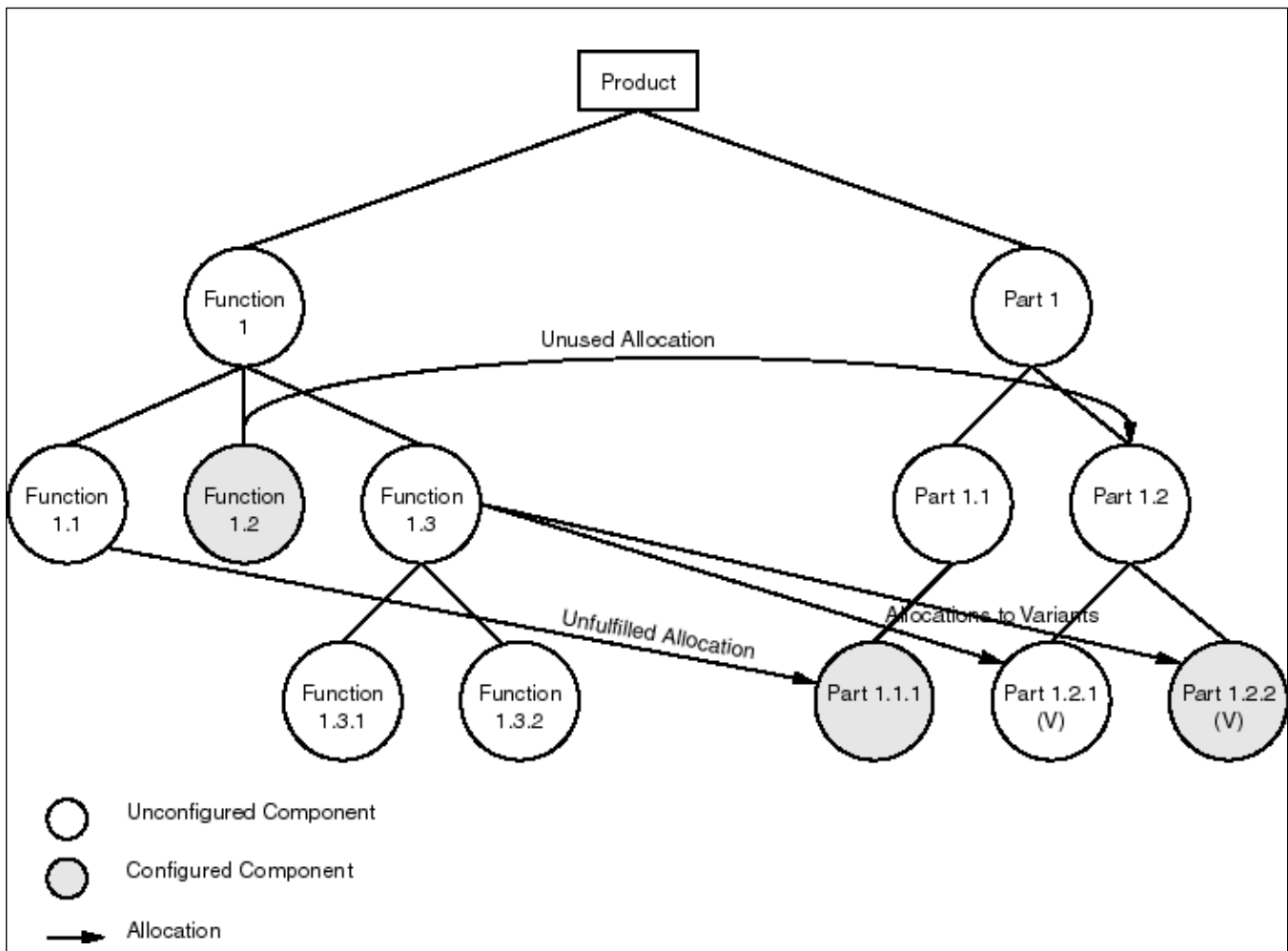
Allocations represent a directional relationship between specific instances of an item revision in one product structure and an item revision in a related structure. For example, you may relate the functional structure to the logical structure or the logical structure to the physical structure. These relationships may span multiple configurations and revisions of both structures. They may change according to the overall configurations or the specific revisions that are configured into the structures.

Making allocations between configured structures

You can use configuration management to determine those components that appear in a target structure. When those components are configured, you can determine the allocations that are possible.

You can manage configurations in Teamcenter by placing options and variants on components in the structures. You can then place allocations between the components in the various structures; these allocations have no conditions associated with them. If a source component is configured in, the associated allocations are configured in. If the target component is not configured in, the allocation is not configured in. You can create multiple allocations from one source component to various target components.

In the following figure, the components shown in white are configured in, while the components shown in gray are configured out.



Configuring allocations

In this example, allocations exist between various components, and there are three basic cases as follows:

- Allocation from function 1.1 to part 1.1.1

Part 1.1.1 is configured out, so the allocation is configured out or invalid.

- Allocation from function 1.2 to part 1.2

Function 1.2 is configured out, but part 1.2 is configured in. As the source component is configured out, the allocation is configured out as an unused allocation. (The allocation is not evaluated because Teamcenter does not need to look for allocations related to this function.)

- Allocations from function 1.3 to part 1.2.1 and to part 1.2.2

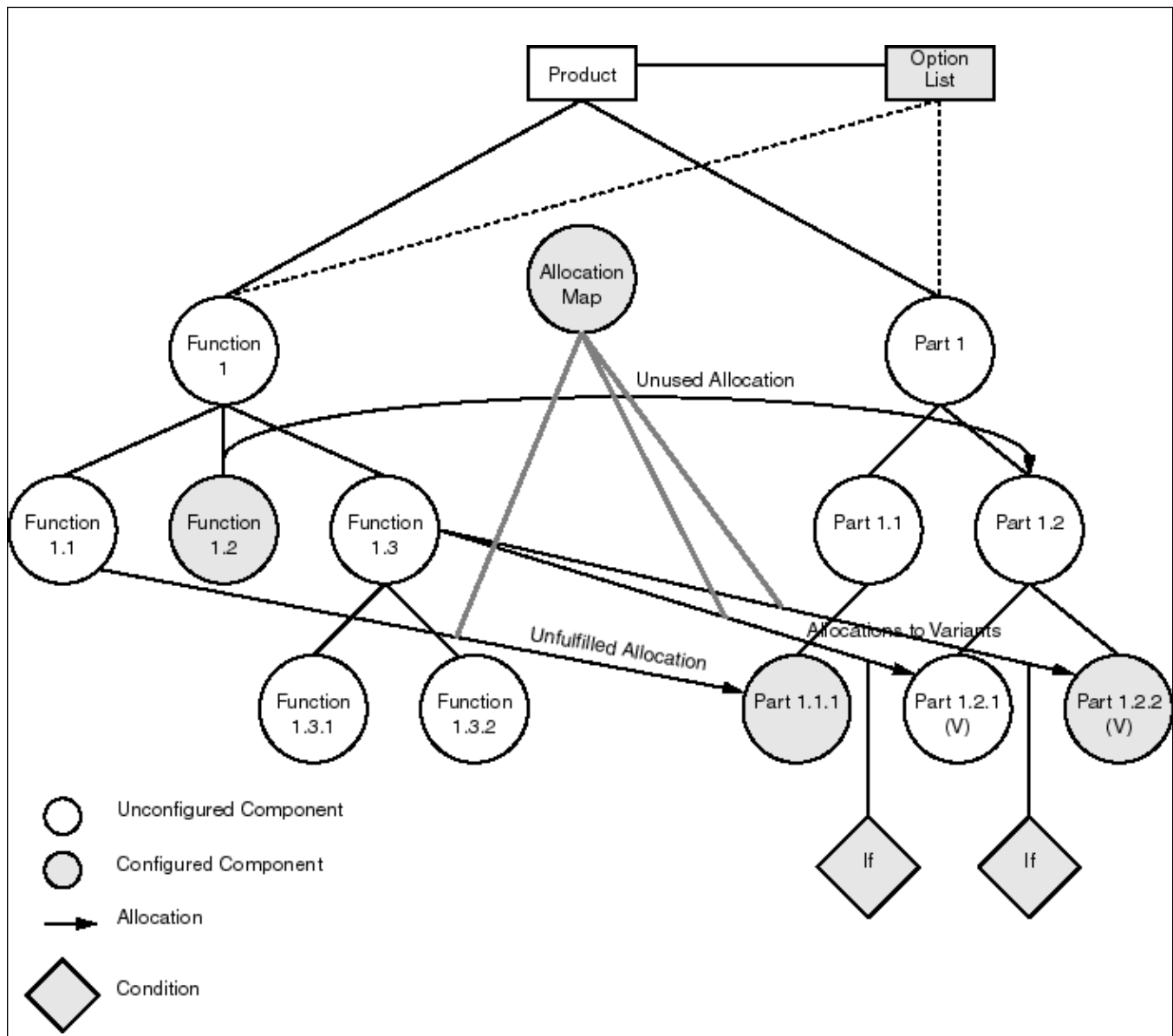
Part 1.2.1 and part 1.2.2 are variants of each other. Only one part is configured in at any time. The allocation to the part that is configured in is valid, and the allocation to the second part is configured out.

Making conditional allocations

You can also use allocations to determine the components included in the target structure. In this case, you can configure the source structure, and also the allocations. The basic configuration rule (if a source component is configured out, the allocation is ignored) is still enforced. From the list of allocations that are configured in, Teamcenter determines the relevant set of physical components. This may be achieved by a conditional relationship, in which you can define certain criteria by which a functional component can be fulfilled by one of many physical components. In this case, the list of parts is not related to a product structure, but is simply a list of parts. When the target parts are part of a product structure, Teamcenter determines the instances of a component to fulfill the functionality that is specified.

If you use conditional relationships, you can define a group of allocations with a set of options that are relevant to the product. You can define various conditions with these options. For example, you could set an option to describe the model of the car. Depending on the model, a particular allocation may be required. As the targets of these relationships are part of a product structure, you can apply configuration rules to the structure. If an allocation is unfulfilled, Teamcenter reports an error that the target structure is not configured properly because there are unfulfilled allocations.

The following figure shows how options and conditions may be used to manage conditional allocations.



Configuring conditional allocations

Creating allocations

You can create allocations at various levels of a structure, but in each case they specify an object of interest—one that performs a particular function. It is possible that every item revision in a structure could contain an allocation. For example, this may be necessary for manufacturing purposes, where all items must be consumed.

Because allocations are independent of the structures they map, you can create and manage them independent of the structures themselves. For example, a functional designer may design the functional model, whereas a mechanical engineer designs the physical model. When both structures are complete, a systems engineer may map the functional model to the physical model. The systems engineer

may trade off studies between allocating functions to different components and perform the actual assignments.

Note:

The model is the representation of a particular aspect of the product, not the CAD model. For example, the functional model of a car represents a product containing an air conditioning system, power braking, power steering, and other functionality. In Teamcenter, the model is represented by the product structure.

When you create an allocation, Teamcenter automatically assigns it a unique identifier. The identification scheme is determined by user exits set by the Teamcenter administrator.

Because you can configure structures in Teamcenter, you can create allocations between variable product structures. Allocations are valid only in a particular context and therefore all allocations between two structures are part of a group, and Teamcenter evaluates them as a unit. The allocations tie together various components and revisions of items in the structures.

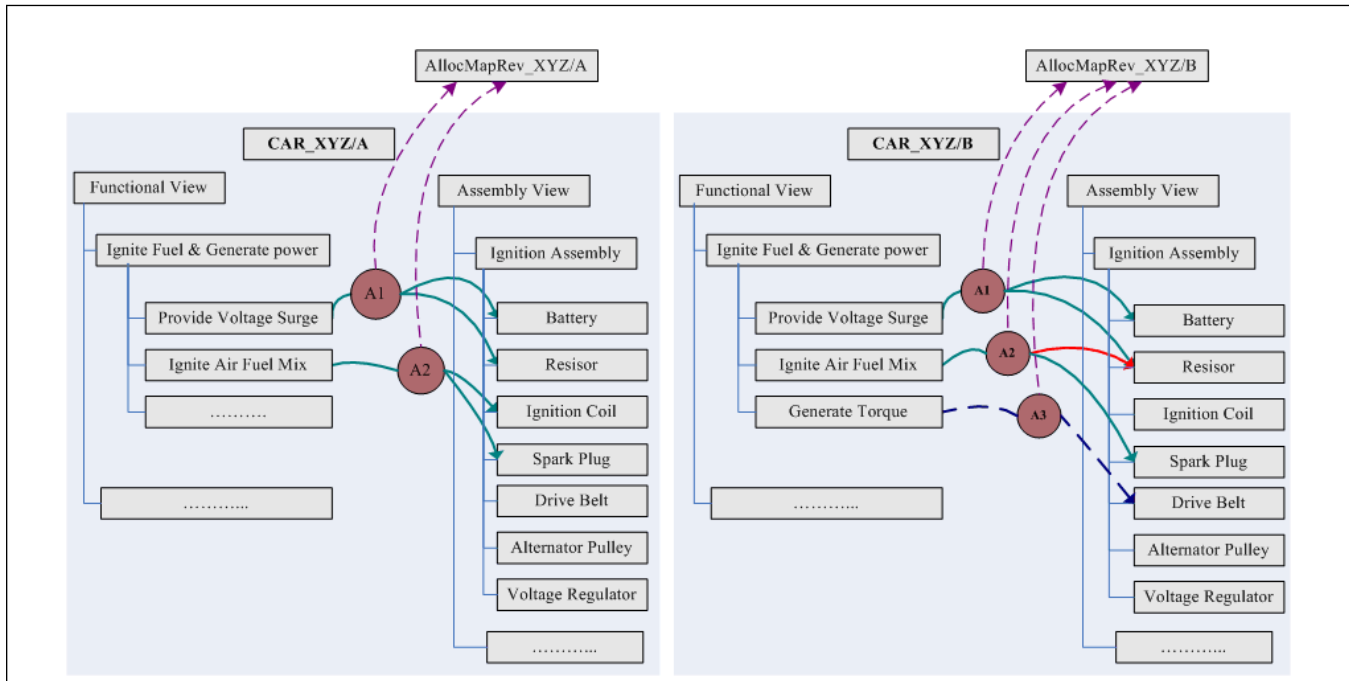
Optionally, a customizer can create subtypes of allocations and allocation maps using the Business Modeler IDE. The Teamcenter administrator can set preferences that determine the allocation subtypes that are valid in a particular allocation map type and, similarly, the allocation maps that are valid between structure type (BOM view types). The administrator also determines whether each allocation subtype is valid for:

- One source and one target.
- One source and multiple targets.
- Multiple sources and one target.
- Multiple sources and multiple targets.

Revising allocations

Allocations do not depend on a particular revision of a item. The allocation exists between the representation of the source item in the source structure and the target item in the target structure. The target item is not a specific revision because all revisions of an item are (by definition) compatible in form, fit, and function. For example, if a source item is revised from revision A to revision B, you can optionally use the same allocations between revision B and the target. To do this, the administrator can configure the system so that, when the allocation map is revised at the same time as the item, all the allocations are carried forward.

The following figure shows allocation **A2** assigned to allocation map revision **XYZ/A**. It has the function **Ignite Air Fuel Mix** in the functional view, which is allocated to the **Ignition Coil** and **Spark Plugs** components in the assembly view.



Revising allocations

When you revise allocation map **AllocMapRev_XYZ/A** to **AllocMapRev_XYZ/B**, Teamcenter optionally copies the entire set of allocations to the new revision. You can then edit the existing allocations, add new allocations or delete selected allocations.

In the example shown, allocation **A2** is edited with respect to the target structure, and is now allocated to the ignition coil and the resistor.

Similarly, revision **B** includes a new allocation between the functional component **Generate Torque** and the physical component **Drive Belt**.

Note:

By default, Teamcenter does not carry forward allocations when you revise an allocation map. To configure this behavior, your administrator sets the necessary preference.

You can delete the source or target of an allocation (or both) and Teamcenter does not delete the allocation itself. If appropriate, you can make a new assignment to replace the deleted source or target.

If appropriate, you can manage changes to allocation with incremental changes. Allocations may be added to or removed from an allocation map in the context of a specified incremental change order.

Configuring allocations

You can define configurations in which an allocation is valid. A *nonconfigurable* allocation represents a relationship between two items in two structures. The allocation is fixed and always exists if the items

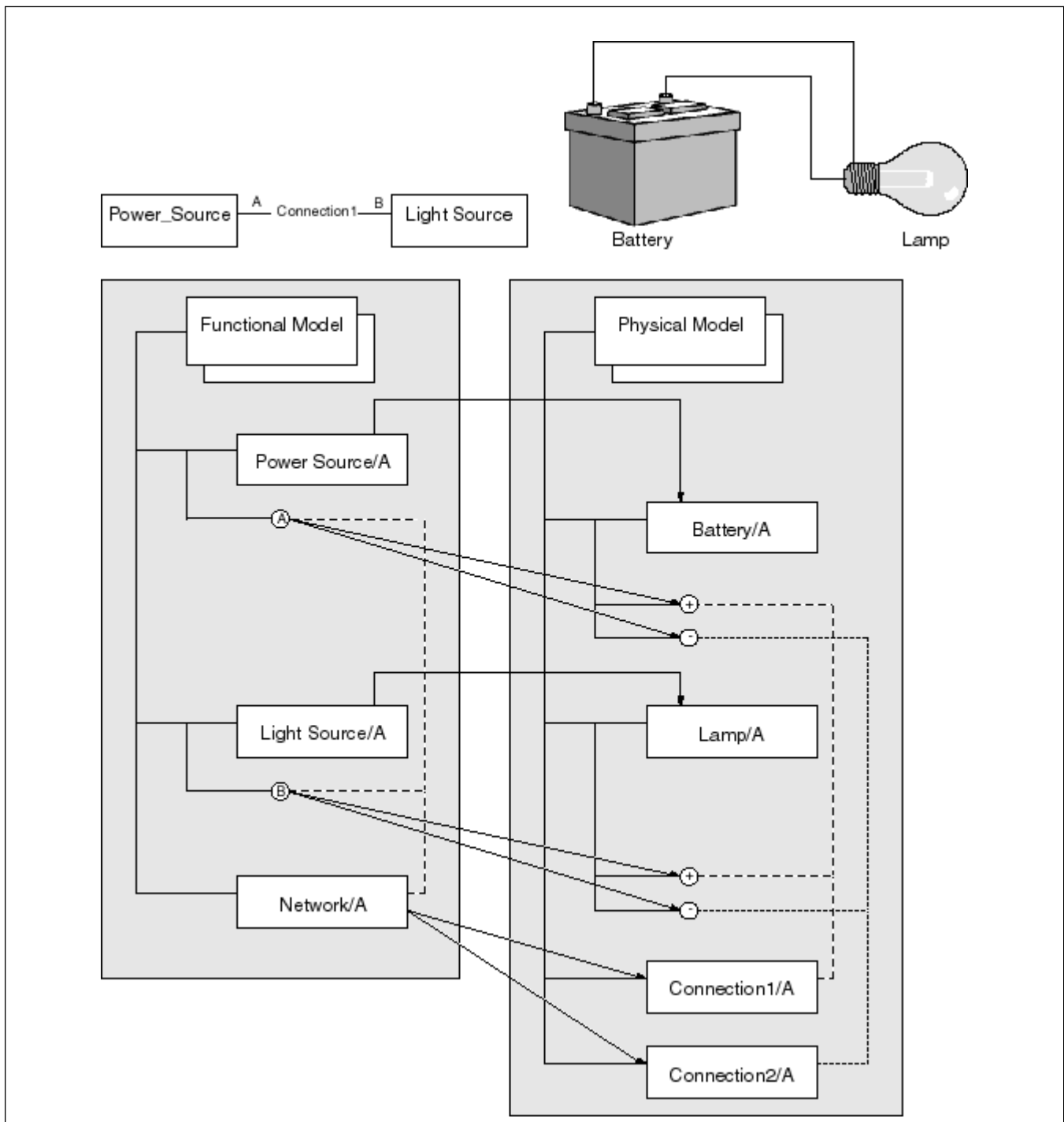
exist in the structures, for example, **Function 1 is allocated to Part 2**. However, if an allocation is configurable, you can attach conditions and configure it in a similar way to items in a structure. By associating conditions with relationships, you can configure various allocations in and out of a product. For example, this allows you to manipulate whether a function is allocated to a certain engine control unit (ECU). For example, you could put a condition on an allocation of the form **If Model==Sport and Color == Green then configure this allocation**. On the second allocation, you could use the condition **If Model == Sport and Color == Blue then configure that allocation**.

Once the option values are set, you can determine the necessary set of allocations. For each of these allocations, if the source is not configured in, the allocation can be discarded. If the destination is not configured in, the allocation can be noted as unallocated or discarded, depending on your site's business practice.

To define the context of a set of allocations, you must define an allocation map, as described previously. The allocation map ties together the two structures that are being allocated and provides the necessary context.

When you create allocations, the structures you work with are typically unconfigured products. You can then create allocations between many items in the structure; multiple allocations can originate from a source item and be allocated to several target items. If the source item is configured out, the allocation is assumed to be configured out with the source item. If the target item is missing, Teamcenter notifies an error as all allocated items must be allocated to a target.

The following figure shows how allocations can be configured in for an electromechanical product.



Configuring allocations

Releasing allocations

When can I release allocations?

You can release allocations independent of their associated structures or items. Similarly, you can release the structures or items without releasing their allocations.

Releasing an allocation

You can release allocations and, therefore, they may be given a status and possibly an effectivity. The release status may configure allocations in one of two ways:

- The release status and effectivity on the allocation configures the allocation from the revision rule.
- Each of the allocations can be associated with an incremental change order that determines effectivity. Teamcenter uses the effectivity on the incremental change order to configure the allocation.

If you add status to an allocation, you can apply various effectivities to this relationship independent of the structures themselves. This provides flexibility when applying several allocations to the structure.

Releasing an item with an allocation to a physical component

Allocations are independent of the structures they allocate. The allocations are modeled in their own environment and must form a valid set. Releasing an item on either side of the allocation has no effect on the allocation itself.

Note:

The allocated target component is identified as a property on the source structure. Because the allocation is independent of source and target release status and the allocation can be created or deleted on a released object, the allocation information (property value) on the source component can change, regardless of its release status.

Teamcenter may allocate the same function structure to multiple physical components. Consequently, the **allocated to** information on the source may differ, depending on the context of allocation.

Releasing allocation maps

Because an allocation map is a workspace object, you can release it. Similarly, you can release allocation map revisions, which is necessary because users typically work with allocation map revisions, not allocation maps. After an allocation map revision is released, you can define additional security rules that depend on the release status, allowing users to control a set of allocations as a whole. For example, you can define a rule that prevents users from modifying a set of allocations once it has been approved.

A. Configuring Systems Engineering

Configuring Systems Engineering

While Systems Engineering can be used after a standard installation, you can configure it to more closely adhere to your working preferences.

The **Requirement** view is flexible and can display datasets, documents, and trace links in a variety of ways.

Overview of Systems Engineering options

You can control the behavior and display of Systems Engineering features using options and preferences.

- Preferences are configuration variables stored in a Teamcenter database that are read when a Teamcenter session is initiated.

Besides Systems Engineering preferences, Manufacturing Process Planner preferences and Structure Manager preferences are applicable to Systems Engineering. In case you use preferences from both Structure Manager and Manufacturing Process Planner to control the same functionality, the Manufacturing Process Planner preference overrides the Structure Manager preference.

For example, if you use both the **PSE_expand_on_open** preference, which is a Structure Manager preference, and the **MEEExpandFirstLevelOnLoad** preference, which is a Manufacturing Process Planner preference, the **PSE_expand_on_open** preference is overridden by the **MEEExpandFirstLevelOnLoad** preference.

- Options are preferences presented in the rich client by category (in a tree). Options usually have predefined valid values.

To set options in the rich client, choose **Edit→Options** to display the **Options** dialog box, and then select **Systems Engineering** from the **Options** list. Use the cursor to hover over an option to display the associated preference name and information about the preference.

Option	Use option to do this
Trace Link Mode	Choose whether trace link functionality is enabled. • Selected (ON): Trace link menu and toolbar commands are available. If objects already have trace links, trace link icons appear in the navigation pane.

Option	Use option to do this
	• Not selected: Trace link menus and commands are not available. Trace links cannot be viewed or created.
Indirect Trace Link Search	Choose whether tracing includes indirect trace links. Indirect trace links are implicitly inherited by the item revisions of target objects. • Selected (ON): Traces follow both direct and indirect trace links. • Not selected: Traces follow only explicit trace links that are associated with target object items.
Left-Right Trace View	Arrange the Defining Object and Complying Object panes in the Traceability view. • Selected (ON): The panes appear horizontally. • Not selected: The panes appear vertically.
Apply Templates (for embedded editing)	Select which requirement content appears in the MS Word view .
Browse Clear Text	Select the appearance of text in the MS Word view. • Selected (ON): Text appears as plain text in the view. • Not selected: Text appears as rich text in the view. Rich-text content can include headings, body paragraphs, bulleted and numbered lists, special characters such as equations and symbols, hyperlinks, tables, and graphics.
Quick Create Panel	Choose whether to enable the Quick Create bar. • Selected (ON): The Quick Create bar is available. • Not selected: The Quick Create bar is not available.
Calculate Defining/Complying Objects	Choose whether to display Defining Objects and Complying Objects properties in the Properties dialog box for requirements objects. • Selected (ON): Defining Objects and Complying Objects appear. • Not selected: Defining Objects and Complying Objects do not appear.
Keywords to Import	Enter keywords to define the start of requirements when a requirement specification is imported. When you import requirement specifications from Microsoft Office Word, Systems Engineering uses Word outline-level heading styles

Option	Use option to do this
	to parse documents into a hierarchy of parent, child, and sibling requirements. You can further control the separation of requirements by using keywords that occur in content under the headings.
	Type the keywords, using a comma to separate multiple keywords. Special characters and partial words are not supported.

Configuring Systems Engineering naming rules

Naming Systems Engineering objects

As an administrator, you can optionally create naming rules that define the data entry format for a business object property. A naming rule consists of rule patterns and a counter. Before creating a naming rule, you should be familiar with the pattern and counter formats. After you create a naming rule, you must attach it to the business object property. You use the Business Modeler IDE to create the naming rules in a custom template that you define. You must deploy the custom template after making the changes.

You can configure the following Systems Engineering naming rules:

- **Default Requirement ID Rule**

You can optionally create a naming rule to automatically name requirements, design requirements, and validation requirements. There is already a default rule defined for the ID property of these objects, but you can change this rule or create additional rules to specify the format you desire.

- **Default Custom Note ID Rule**

You can optionally create a naming rule to automatically name custom notes. There is already a default rule defined for the ID property of the custom note object, but you can change this rule or create additional rules to specify the format you desire.

Autoname requirements

The **Default Requirement ID Rule** naming rule is used to automatically name requirements, design requirements, and validation requirements. By default, the naming rule is already attached to the following properties for the following business objects, but you can change this rule or create additional rules to specify the format you desire.

Business object	Property	What the naming rule does
Requirement	Requirement.item_id	Provides the option to automatically name the requirement.
ValidationReq	ValidationReq.item_id	Provides the option to automatically name the validation requirement.
DesignReq	DesignReq.item_id	Provides the option to automatically name the design requirement.

The following table shows the default values for the naming rule.

Pattern	Initial value	Maximum
"REQ-"nnnnnn	REQ-000001	REQ-999999

Autoname custom notes

The **Default Custom Note ID Rule** naming rule is used to automatically name custom notes. By default, the naming rule is already attached to the following property for the custom note business object, but you can change this rule or create additional rules to specify the format you want.

Business object	Property	What the naming rule does
Fnd0CustomNote	Fnd0CustomNote.item_id	Provides the option to automatically name the custom note.

The following table shows the default values for the naming rule.

Pattern	Initial value	Maximum
"CSTMNOTE-"nnnnnn	CSTMNOTE-000001	CSTMNOTE-999999

Defining allowed child item types for Systems Engineering structures

You can define the item types that are permitted as children in functional, requirement specification, requirement, paragraph, design requirement, validation requirement, and logical block structures by setting the following preferences:

- `TAllowedChildTypes_Functionality`
- `TAllowedChildTypes_RequirementSpec`
- `TAllowedChildTypes_Requirement`
- `TAllowedChildTypes_Paragraph`
- `TAllowedChildTypes_DesignReq`
- `TAllowedChildTypes_ValidationReq`
- `TAllowedChildTypes_Fnd0LogicalBlock`

Note:

You must define custom item types before importing structures into Teamcenter.

Enable moving objects between levels in structures

Typically when you build the structure of a requirement specification, you create a requirement specification object at the top level of the structure and add structure elements by creating requirement and paragraph objects at lower levels. Using the default Teamcenter data model and business objects to build this structure, you can move objects up and down between the levels in your structure.

If your top level object is a type of object that is not a child of the **Specification** business object in the data model, for example, a **Document**, then you must configure the object to allow moving the first level objects up and down in your structure. You can configure any business object that is a subtype of **Item**.

1. In the Business Modeler IDE, open the business object revision, for example, **Document Revision**.
2. Select the business object constant **Fnd0AssociatedBOMLine**.
3. Set the **Value** to **Fnd0BuildingBlockBOMLine** or any of its subtypes, such as **Fnd0RequirementBOMLine**.

Migrate Microsoft Excel templates from versions prior to Teamcenter 8.3

Starting in Teamcenter 8.3, all Microsoft Excel templates included in Teamcenter no longer contain Teamcenter macro code. If you upgraded Teamcenter from a version prior to 8.3, you must update your Excel templates.

1. In My Teamcenter, search for the following items and delete them:

- REQ_defview_template
- REQ_default_excel_template
- REQ_default_RM_template
- REQ_default_rel_export_template
- REQ_default_struct_template
- MFG_default_struct_template

2. At the server prompt, run the **add_req_templates** utility.

This utility installs the new Excel templates in the database. If the templates you deleted were customized, you must repeat the customizations in the new templates installed by the utility.

3. If you created custom Excel templates other than the ones included in Teamcenter, you must remove the Teamcenter macro code from them.
 - a. Open a custom template and press Alt-F11.
 - b. Remove any Teamcenter macro code and leave your custom macro code.
 - c. Save the macro and template.
 - d. Repeat for these steps for all custom templates.

Custom macro code is preserved in Excel sheets generated at the server.

Exclude specification types from the Import Spec wizard

1. Choose **Edit→Options** to display the **Options** dialog box.
2. At the bottom left, click **Filters** to display the **Preferences By Filters** page.
3. Under **Filters** in the **Search by preference name** box, enter **Import_Spec_Info_EXCLUDETYPES**.
4. Select the preference in the list, and then click **Edit** at the bottom of the right pane.
5. In the text box below the **Values** list, enter the type's internal name, and then click the **Add** button to the right.

The internal name is from the Business Modeler IDE.

- Click **Save**.

Note:

When upgrading to a newer Teamcenter version, the preference utility should be run to ensure that the appropriate preferences are installed in the database.

Configure keywords for importing requirement specification structures

When you import requirement specifications from Microsoft Office Word, Systems Engineering uses Word outline-level heading styles to parse documents into a hierarchy of parent, child, and sibling requirements. You can further control the separation of requirements by configuring the **REQ_import_keywords** site preference with keywords that occur in content under the headings.

Note:

The **REQ_import_keywords** user preference can also be set from the **Keywords** box in the **Import Spec** wizard.

- Choose **Edit→Options** to display the **Options** dialog box.
- Select **Systems Engineering**.
- In the **Keywords to Import** box, enter the keywords at which occurrence you want imported requirements to begin.

Note:

Use a comma to separate two keywords.

- Click **Apply** to commit your edits and keep the **Options** dialog box open, or click **OK** to apply edits and close the dialog box.

Configure object checkout for export to Microsoft Office Excel

- Choose **Edit→Options** to open the **Options** dialog box.
- At the bottom left corner, click **Filters** to display the **Preferences By Filter** pane.
- In the **Filters** box, type **show_**.
- Select **Show_Checkout_option** in the **Preferences List** table, and then click **Definition** at the top of the right pane.

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

Configure trace link features

To enable the trace links features in Systems Engineering, including the ability to create trace links, you must configure the trace link features.

1. Choose **Edit→Options** to display the **Options** dialog box.
2. Select **Systems Engineering**.
3. Select the **Trace Link Mode**, **Indirect Trace Link Search**, and **Calculate Defining/Complying objects** check boxes.
4. Click **Apply** to commit your edits and keep the **Options** dialog box open, or click **OK** to apply edits and close the dialog box.

Note:

This procedure sets the Teamcenter **Tracelink_Edit_enabled** preference to **true**. That setting also configures trace link features for other Teamcenter applications including My Teamcenter, Structure Manager, and Multi-Structure Manager.

This procedure sets the display for **Defining Objects** and **Complying Objects** properties for requirements objects only. To display the **Defining Objects** and **Complying Objects** properties for other types of business objects, set each type's **Fnd0RuntimeTraceProperty** constant to **True** in the Business Modeler IDE.

Configuring defining and complying object calculation

You can display the **Defining Objects** and **Complying Objects** properties for the following requirement objects by choosing **Edit→Options**, selecting **Systems Engineering**, and selecting the **Calculate Defining/Complying Objects** option:

- **SpecElement**
- **SpecElement Revision**
- **RequirementSpec**
- **RequirementSpec Revision**

- **Functionality**
- **FunctionalityRevision**

To display the **Defining Objects** and **Complying Objects** properties for other types of business objects, set each type's **Fnd0RuntimeTraceProperty** constant to **True** in the Business Modeler IDE.

Configuring display of the Summary view and Properties dialog box

You can control the display of the **Summary** view and the **Properties** dialog box for the **Specification**, **Specification Revision**, **Requirement**, **Requirement Revision**, **Diagram**, and **Diagram Revision** objects using style sheets. By default, Teamcenter includes the following properties:

- **Name**
- **Description**
- **Owner**
- **Group Id**
- **Last Modifying User**
- **Checked-Out**
- **Checked-Out By**

The following actions may also be configured for each of the objects:

- **Submit For Review**
- **End Tracelink**
- **Start Tracelink**
- **Generate Report**
- **Thumbnail** is configured for **Diagram Revision** objects only.

Configure baseline features

The following preferences apply to the baseline features:

- **Baseline_create_snapshot_folder**

To create a **Snapshot** folder for a baseline and attach the folder to the baseline revision, set this preference value to **1**.

Note:

The Teamcenter administrator sets this preference.

- **ItemRevision_shown_relations**

The **Snapshot** value is added to the default values for this preference. The **Snapshot** value ensures that objects attached to item revisions with a **Snapshot** relation are displayed in Systems Engineering.

Setting up a custom template for requirements

Process for setting up a custom template

To set up a custom template for new requirements, complete the following procedures:

1. Add the **FullText** subclass for datasets.
2. Create the requirement template.
3. Associate the template revision with Microsoft Office Word.
4. Release the template revision.
5. Apply the template revision to a business object.

Note:

You perform this procedure in the Business Modeler IDE.

Add the FullText subclass for datasets

This procedure adds the **FullText** subclass to the **TYPE_DISPLAY_RULES_list_types_of_subclasses** preference. The preference values comprise a list of the subclasses that can be assigned to datasets.

Microsoft Office Word is the default tool for the **FullText** subclass. Once you add the subclass, you can associate Word with each requirement template that you create.

1. Choose **Edit→Options** to open the **Options** dialog box.
2. At the bottom left of the dialog box, click **Filters**.

The **Preferences By Filters** page is displayed.

3. Under **Filters** in the **Search by preference name** box, type **TYPE_**.

The preferences list displays each preference whose name contains **TYPE_**.

4. In the list, select the **TYPE_DISPLAY_RULES_list_types_of_subclasses** preference.
5. Add the **FullText** subclass to the preference value.
 - a. At the bottom of the right pane, click the **Edit** button to activate the **Values** list.
 - b. In the box below the **Values** list, type **FullText**.
 - c. Click the **+** button to the right of the box, and then click **Save**.
 - d. Close the **Options** dialog box, and then exit Teamcenter.
6. Restart Teamcenter, and then choose **Edit→Options** to open the **Options** dialog box.
7. In the left pane under **Options**, select **Dataset**.
8. In the right pane, select **FullText** from the **Dataset Type** list, and then click **OK**.

Create a requirement template

1. Choose **File→New→Item** to open the New Item wizard.
2. Type **DMTemplate** in the text box, or select it under **Most Recently Used** or **Complete List**, and then click **Next**.
3. On the **DMTemplate** page, enter the template name in the **Name** box.

You can also do the following optional actions:

- In the **Description** box, enter text for the default content of new requirements.

Note:

If you leave the box empty, the initial content of each new requirement is the requirement name.

- In the **ID** box, enter an identifier for the template, or click **Assign** for Teamcenter to assign the identifier automatically.
- In the **Revision ID** box, enter an identifier for the template's initial revision.

- Select a value from the **Unit of Measure** list.
- Select the **Open On Create** check box to open a tabbed view containing the new template revision.

4. To create the template, click **Finish**.

Tip:

Click **Next** instead of **Finish** to open subsequent pages where you can enter optional information for:

- The template item master
- The template item revision
- The template item revision master
- The workflow release process

The template is added to your **Home** view.

If you selected the **Open On Create** check box on the **DMTemplate** page, the template revision is displayed in a new tabbed view.

Note:

To continue setting up a custom template, complete the steps in [Associate a requirement template revision with Microsoft Office Word](#).

Associate a requirement template revision with Microsoft Office Word

After creating a requirement template, complete this procedure to associate the template revision with default Word content for new requirements.

Note:

Before you start this procedure, ensure that the **FullText** subclass is available for new datasets.

1. In the **Home** view, expand the structure below the template and select the template revision.
2. In the **Summary** view, click the **Attachments** tab, and then click the **Add New** button under **Files**.

The **New Dataset** dialog box opens.

3. In the left pane, click the **More** button to open a list of dataset types.

4. Scroll down to the options for Microsoft Office, and then select **MSWord** or **MSWordX**.

The **Name** box and the **Tool Used** and **Relation** lists are filled in automatically.

Tip:

Leave the **Open On Create** check box selected to automatically open the new dataset in Word for entering default content.

5. Click **OK** to create the dataset and close the dialog box.

The dataset is added to the structure below the template revision.

In the **Summary** view, the **Attachments** tab shows the dataset in the object pane under **Files**.

6. Open the dataset in Word, and then enter the default content for new requirements that are based on the template.

Note:

If you leave the **Open On Create** check box selected in the **New Dataset** dialog box, the dataset opens automatically.

7. Save the default content in Word, and then close the dataset.

Note:

To continue setting up a custom template, complete the steps in [Release a requirement template revision](#).

Release a requirement template revision

When the template revision is associated with Word, the revision's development is complete. This procedure sets the workflow status to **TCM Released**, indicating that the revision is mature.

You release the attached Word dataset with the template revision. Then, the revision and dataset are available in the Business Modeler IDE, where you apply them to the requirement business object that you select.

1. In the **Home** view, expand the structure below the template revision.
2. Select the template revision, hold down the Ctrl key, and select the Word dataset.
3. Choose **File**→**New**→**Workflow Process** to open the **New Process** dialog box.

Tip:

- The **Process Name** box contains the revision name automatically.
You can accept the default name or enter a different name for this process.
- The **Attachments** tab shows the revision with its structure collapsed.
You can expand the revision to view its attachments.

4. In the **Process Template** list, select **TCM Release Process**.

Note:

Your action is not required for the **Add Status** task on the **Process Template** tab. The system performs the task, which assigns the **TCM Released** status.

5. Click **OK** to initiate the workflow process.

Note:

To continue setting up a custom template, complete the steps in **Apply a requirement template revision to a business object**.

Apply a requirement template revision to a business object

With the **TCM Released** workflow status assigned to the template revision and Word dataset, create an *item revision definition configuration* (IRDC) in the Business Modeler IDE. This IRDC applies the revision and dataset to the requirement business object that you select.

1. In the Business Modeler IDE, open the **Extensions** and **Document Management** folders, right-click the **IRDC** folder, and choose **New IRDC**.

The New IRDC wizard opens.

2. On the **IRDC Base Criteria** page, create the IRDC object.
 - a. In the **Name** box, type the name of the new IRDC.
 - b. In the **Description** box, type a description of the new IRDC.
 - c. To the right of the **Applies to Business Object** box, click the **Browse** button and select the item revision business object to which the IRDC applies.

Note:

You can select only children of the **ItemRevision** business object.

- d. To the right of the **Condition** box, click the **Browse** button and select the condition that determines when the new IRDC applies.

Note:

- If the IRDC applies to all objects of a type, use the **isTrue** condition.
- You can also create your own condition.

- e. To the right of the **Create Template** box, click the **Browse** button and select your new custom requirement template.

You can also do the following optional actions on the **IRDC Base Criteria** page:

- To the right of the **Specification Export Template** box, click the **Browse** button and select the template to be used when the item revision specification document is exported.
- To the right of the **Object Export Template** box, click the **Browse** button and select the template to be used when the item revision is exported.

3. Click **Next** to open the **IRDC Dataset Criteria** page.
4. To the right of the **Source Dataset** table, click the **Add** button to open the **Select Source Dataset Type** dialog box.
5. Specify the source dataset subclass and its relation to the template revision.
 - a. To the right of the **Source Dataset** box, click the **Browse** button and select **FullText**.
 - b. To the right of the **Item Revision Relation** box, click the **Browse** button and select **IMAN_specification**.
 - c. Click **Finish** to close the **Select Source Dataset Type** dialog box.
6. On the **IRDC Dataset Criteria** page, click **Finish** to create the IRDC object and close the New IRDC wizard.

Tip:

Click **Next** instead of **Finish** to open subsequent pages where you can enter optional information:

- Naming rules for the dataset
- System actions that occur when the template revision is checked in
- Deep-copy rules for the template revision
- Markup rules for the dataset

7. To save the changes to the data model, choose **BMIDE→Save Data Model**, or click the **Save Data Model** button on the main toolbar.
8. Deploy the template.

B. Customizing Word export templates

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.

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 components

Note:

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
- Transfer modes

- Object export structure

The title page of a specification template can be a combination of static and dynamic text. Title pages often include:

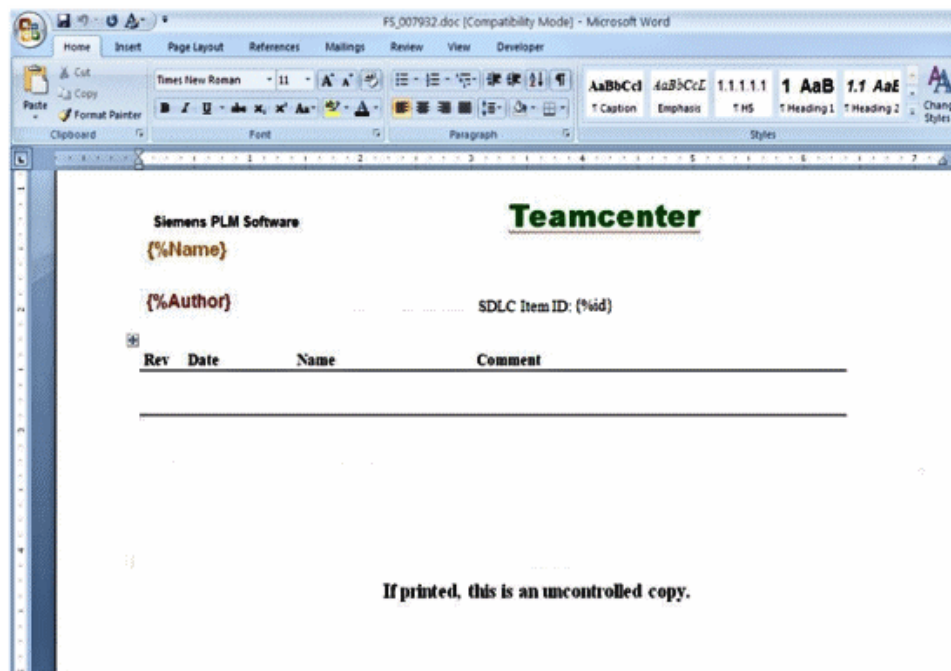
- Document title derived from a Teamcenter property value identified by a keyword.
- Copyright and trademark information.
- Description and owner name.

For example, if you want to include the description and owner name of the selected object, say a requirement, then include **{%object_desc}** and **{%owning_user}** in the appropriate locations in template. **object_desc** and **owning_user** are persistent properties of the **requirement** object.

Note:

To view the object properties that you can use in a template, you can look up the object's properties in BMIDE. You can use only persistent properties in templates.

Out-of-the-box, Siemens Digital Industries Software does not support export of contextual information using property names.



Tables of contents are added to the specification template using the **Table of Contents** options in Microsoft Word.

You can specify the number of heading levels and choose a format for the table of contents.

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 Microsoft Word.

You can also add footnotes and endnotes to the specification template.

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:

- 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 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.

The body content of the specification template contains free-form text and information extracted from the Teamcenter database using keywords. 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.

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.

Example: Static structure in a specification template

```

Company Name
{%Name}

Requirement Specification
{%Author}

Table of Contents
1. Introduction
2. Market Challenges
3. Architecture Overview

{%Rule:Selection.Export.Static}|

```

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}

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

Company Name

{%object_name}

Requirement Specification

{%owning_user}

Table of contents

1. Introduction

2. Market challenges

3. Architecture overview

{%Rule:Selection.Export.Static}

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}`

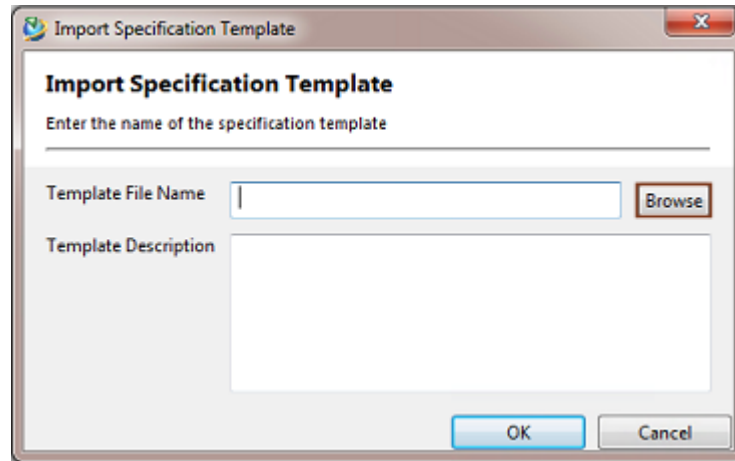
9. Save the document, and close Microsoft Office Word.

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**.

- 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.

- Type a name for the new item in the **Name** box.

You can type an optional description in the **Description** box.

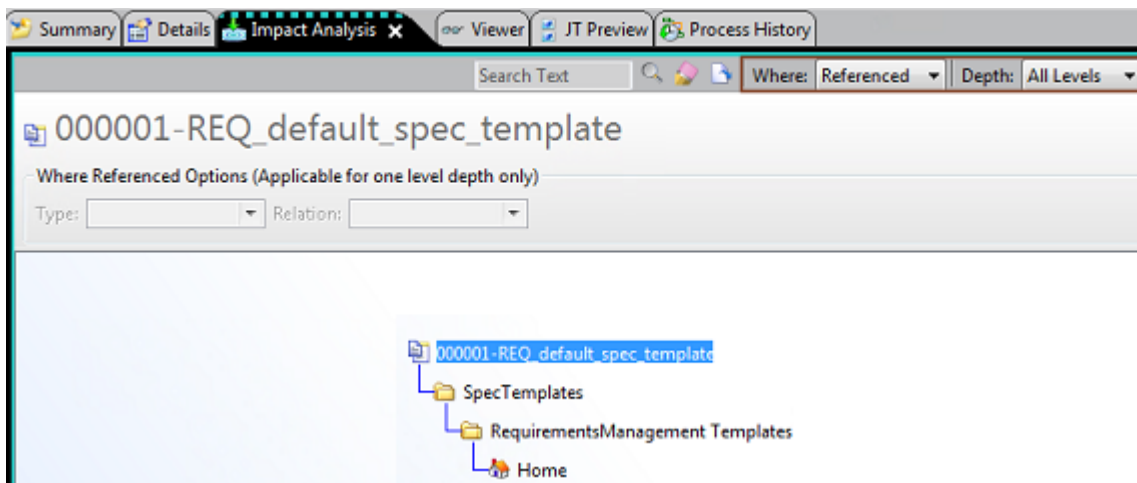
- (Optional) Select the **Open on Create** check box to open the item after it is created.
- Click **OK**.

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.

- (Optional) Rename the **Full Text** dataset.

Delete a specification template

- Ensure that the **Spec Template** item is not referenced by other objects in the Teamcenter database by performing a where-referenced search.



- In the **Home** view, select the **Spec Template** item.
- Choose **Edit→Delete**.
- 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.

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}

Object export template document with rules

{%object_name}

Property	Value
Type	{%object_type}
Object Description	{%object_desc}
last_mod_date	{%last_mod_date}

{%body_text}

{%RULE:SELECTION.EXPORT.TM.My_Transfer_Mode}

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.	<code>{%Rule:Selection.Export.TM.My-Transfer-Mode}</code>

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.Reg_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.

Applying Microsoft Office Word templates to exported objects

Caution:

Users must have at least **Read** access to all Word export templates for their projects. A user without such access to one template cannot select any template in the **Export to Word** dialog box.

The overall style for the generated document comes from the specification template:

- Rules and keywords in the specification template are applied to the first object of the structure.
- The first object of the structure uses the specification template managed by item revision definition configuration (IRDC), unless the user chooses another template at run time.
- When there is no IRDC-associated template, the default is used.

When you select a specification for export in Systems Engineering:

- All the objects under specification are exported.
- Rules specified in the object template are applied to each exported object.
- Property keywords in the object template are replaced with corresponding attribute values of the specification object.

When you select a single specification element such as a requirement or paragraph for export in Systems Engineering:

- All the objects subsumed under the selected specification element are exported.
- Rules specified in the object template are applied to each exported object.
- Property keywords in the object template are replaced with corresponding attribute values of the specification element object.

When you select more than one specification element objects for export in Systems Engineering:

- Only the selected objects are exported. Subsidiary elements under each selected object are not exported.
- Rules specified in object template are applied to each exported object.
- Property keywords in the object template are replaced with the corresponding attribute values of the first specification element object.

When you select a single object for export in My Teamcenter:

- The object template, if managed by an IRDC object, is used for export unless you choose otherwise.
- When there is no IRDC-associated template, the default is used.
- Property keywords in the object template are replaced with corresponding attribute values of the specification element object.

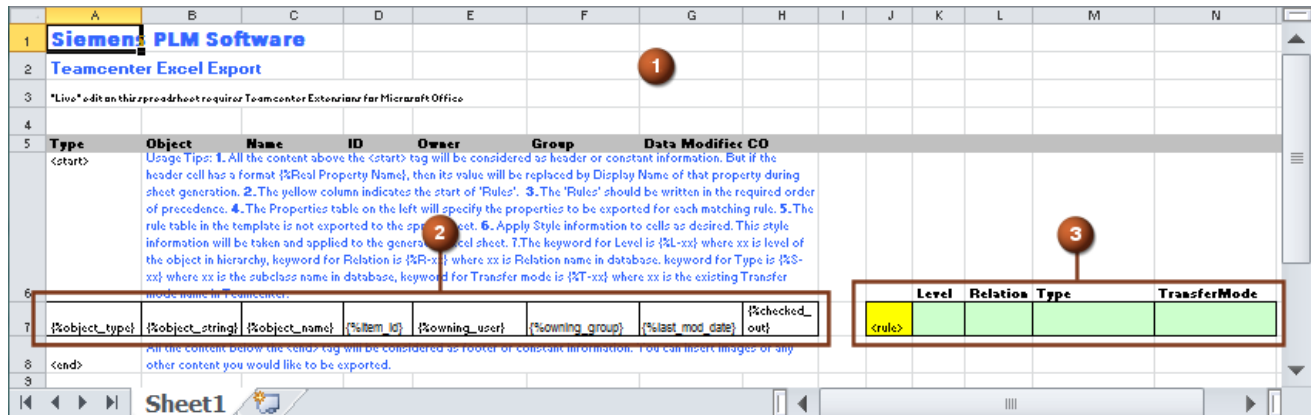
When you select multiple objects for export in My Teamcenter:

- Only the selected objects are exported.
- The default object template is used for export unless you choose otherwise.
- Rules specified in object template are applied to each exported object.
- Property keywords in the object template are replaced with corresponding attribute values of the top specification element object.

C. Customizing Excel export templates

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.

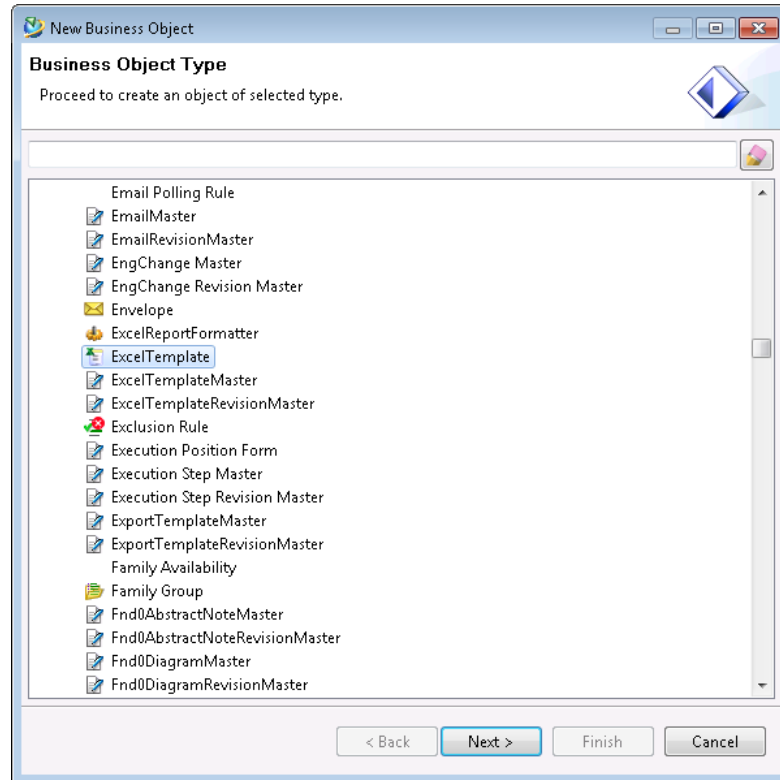
The layout, formatting, and styles are applied to the exported data in the output files. You can also add static text, such as corporate boilerplate and column headings, to appear in the output files.

Note:

Siemens Digital Industries Software does not recommend applying formatting, conditional or otherwise, to empty cells.

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.

New Business Object

Object Create Information
Define business object create information

ExcelTemplate General

Description:

ID:

Name:*

Unit of Measure:

Relation: ☒ Open On Create

< Back Next > Finish Cancel

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 ExcelIX** 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 ExcelIX** dataset by default.

8. (Optional) To update the contents of the specification export template document, double-click the **MS ExcelIX** dataset.
9. (Optional) Modify the contents of the template document, save the Excel file, and close Excel.

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.

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.

Copy an Excel template

1. In the **Home** view, click the **ExcelTemplate** 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 **ExcelTemplate** item and **ExcelTemplateRevision** are created. The **MSEExcelIX** dataset is copied as a new object with the same name as the original dataset.

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>			

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.

Key field	Syntax	Example	Description
Transfer Mode	<code>{%T-Transfer-Mode-Name}</code>	<code>{%T-Req_default_TM}</code>	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.

					Level	Relation	Type	TransferMode
<code>{%object_type}</code>	<code>{%object_name}</code>	<code>{%item_id}</code>	<code>{%object_desc}</code>	<code>{%R-name}</code>	<code><rule></code>	<code>{%R-FND_TraceLink}</code>		

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.

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.

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.

LOV value	Description
	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 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).

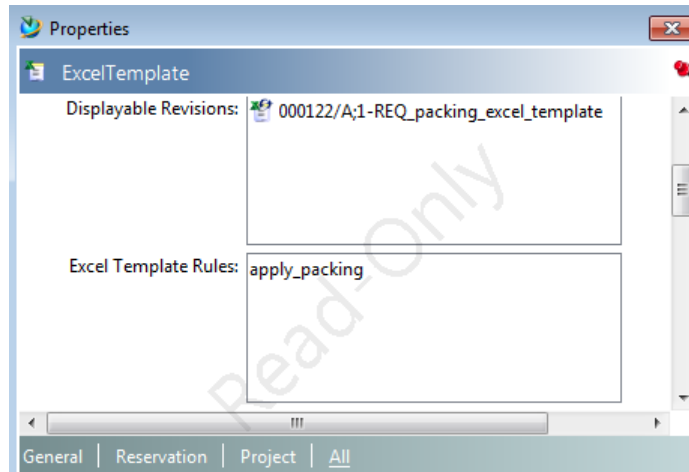
Note:

You can set multiple LOV values for the same property.

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

Insert macros in an Excel template

1. Verify that the custom template has the Teamcenter macro by using the ALT and F11 keys in Microsoft Office Excel.

If the template does not have the Teamcenter macro, copy the Teamcenter macro from any out-of-the-box Excel template.

2. Add your custom macro code such that the Teamcenter macro is not affected or the methods are not overridden to cause different behavior.
3. Save the macro and the template.

Users can use the Excel template for both static and live sheets.

Teamcenter preserves the macro in the generated sheet for static and live export to Excel.

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.

Delete an Excel template

1. In My Teamcenter, perform a where-referenced search to ensure that the **ExcelTemplate** item is not referenced by other objects in the Teamcenter database.
2. In the **Home** view, select the **ExcelTemplate** item.
3. Choose **Edit→Delete**.
4. Click **Yes** in the **Delete** dialog box.

The **ExcelTemplate** item, **ExcelTemplateRevision** item revision, and **MSExcels** dataset are deleted from the database.

D. Customizing Visio diagram templates

Diagram template components

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Siemens Digital Industries Software assumes that you are familiar with Visio and with XML.

Systems Engineering refers to a user-selected *diagram template* in generating each diagram. For diagrams based on a given diagram template, the template:

- Controls the diagram contents through a *stencil* file (**.vsx** in Visio 2010; **.vssx** in Visio 2013 and later versions) and a *mapping* file (**.xml**).

Stencil files and mapping files are stored in Teamcenter datasets that are associated with the diagram template revision object (of the **VisioDiagramTemplateRevision** object type).

- Designates a Teamcenter *application domain* and supplies domain-specific context information.
- Specifies rules for the diagram's member objects and rules for General Relationship Management (GRM) relations among the members.

Every diagram template has a flag that makes it available or unavailable. If a template is available, users can select it from a list when creating a diagram.

Note:

A diagram's template association cannot be changed after the first time the diagram is saved.

Default Teamcenter stencils

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Siemens Digital Industries Software assumes that you are familiar with Visio.

Two default stencil files are provided in the **TC_ROOT/install/systemsengineering** directory.

- **FM_Diagram_Stencil.vsx** contains master shapes for the **FunctionalModeling** application domain (the **Function** view).
- **LM_Diagram_Stencil.vsx** contains master shapes for the **LogicalModeling** application domain (the **Logical Block** view).

The default stencils define master shapes that are mapped to Teamcenter object types. Each stencil is associated with the default diagram template for its designated domain.

Note:

The diagram template associates the stencil with a mapping file (**.xml**). The mapping file's XML entries relate the stencil's master shapes to Teamcenter object types and their properties. You can then apply the shapes in diagrams based on the diagram template.

You can customize the default stencils for site-specific purposes, for example, to define new master shapes or to modify their behavior. Also, you can create stencils with your own master shapes, and then map the master shapes to Teamcenter object types that you specify. You can draw master shapes with the Visio drawing tools, use master shapes from different Visio stencils, or paste objects from other applications.

Default mapping files

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Siemens Digital Industries Software assumes that you are familiar with Visio and with XML.

A diagram template associates a stencil file with a mapping file in the template's designated application domain. The stencil file (**.vsx** or **.vssx**) defines the master shapes that represent Teamcenter objects in diagrams. The mapping file (**.xml**) relates Teamcenter object types and properties to the master shapes.

Using an XML editor, such as Microsoft WordPad or NotePad, you can:

- Create new mapping files, pairing each one with a stencil file through a diagram template.
- Modify existing mapping files, including the default files.

Note:

Two default mapping files are provided in the **TC_ROOT\install\systemseengineering** directory.

- **FM_Diagram_prop_map.xml** is used with diagram templates designated for the **FunctionalModeling** domain (**Function** view).
- **LM_Diagram_prop_map.xml** is used with diagram templates designated for the **LogicalModeling** domain (**Logical** view).

Mapping files use specific XML entries for the following purposes:

- Declaring master shape names
- Mapping master shapes to Teamcenter object types
- Differentiating among master shapes that are mapped to the same object type
- Designating a required property and default value for Teamcenter objects created by adding diagram shapes
- Mapping master shape text blocks to Teamcenter properties
- Assigning interface names automatically
- Setting interface direction for new connections
- Identifying custom master shapes, such as blocks, interfaces, connections, and relations

Designating application domains for diagram templates

Each diagram template designates a Teamcenter application domain for the diagrams based on that template. By default, two domains are available, both for the Systems Engineering application.

- Diagrams based on templates designating the **FunctionalModeling** domain are fixed to the **Function** view.
- Diagrams based on templates designating the **LogicalModeling** domain are fixed to the **Logical Block** view.

Note:

In the Business Modeler IDE, you can add custom application domains for diagrams by adding the domain names to the **Fnd0TcApplication** list of values (LOV).

An application domain defines a context for accomplishing particular business tasks. For example, the **Functional Modeling** domain gives direction for the tools that you use in the **Function** view.

Diagram content is synchronized according to the underlying application domain.

- Domain-specific context information is stored with the diagram object when the content is saved in the diagram view.
- When the diagram is opened, the context information is passed to the diagram owner to initiate the diagram view.

If the diagram information does not match the diagram owner's current structure, the diagram view content is updated to reflect the structure changes.

Caution:

The related application must be running while the diagram is open.

Setting diagram template membership rules

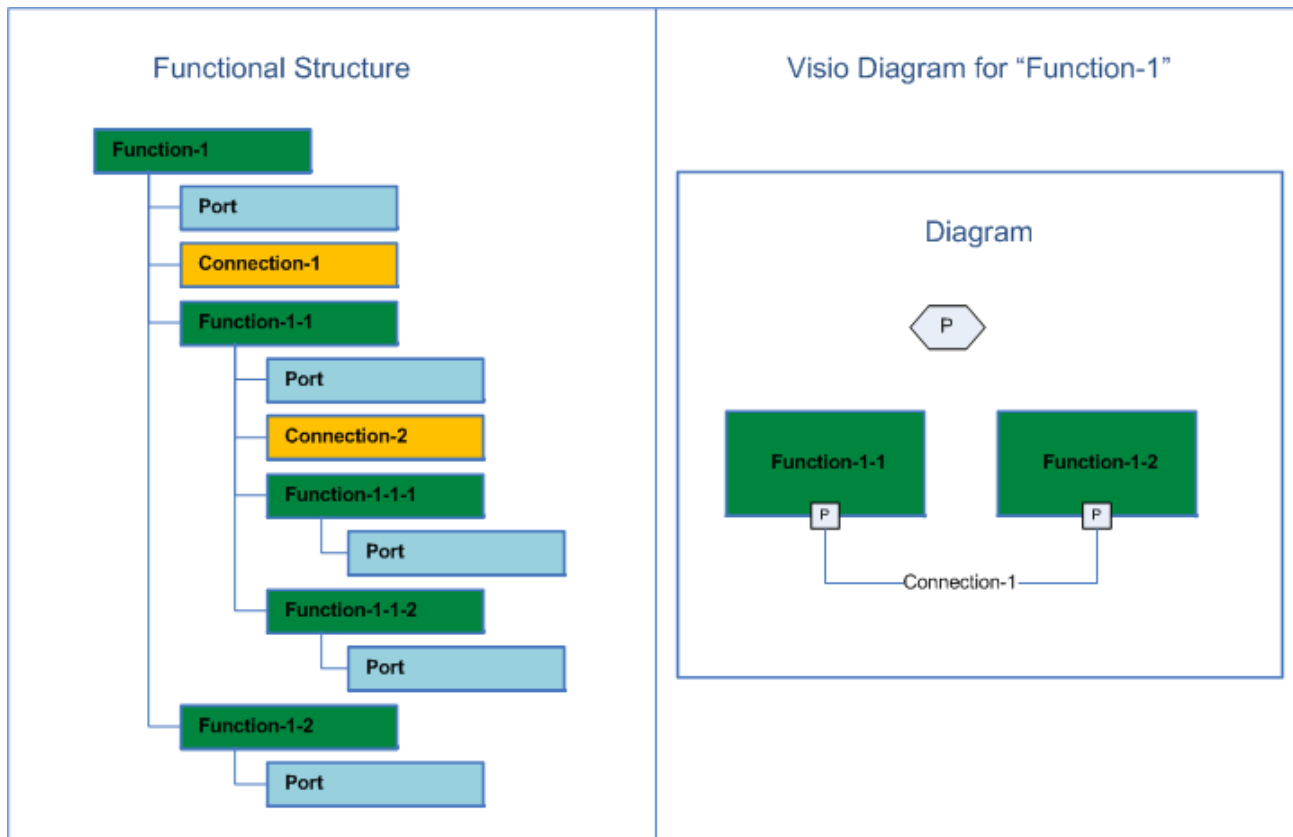
Membership rules are Teamcenter transfer modes that use closure rules. For each diagram template, a membership rule lists the object types that can be represented in diagrams based on the template.

You select the member object types from the **Membership Rule** list in the **New Diagram Template** or **Edit Diagram Template** dialog box.

Note:

Membership Rule choices can vary by application domain, depending on custom configurations at your site. Some domains may have built-in membership rules and may not support rule selection.

The following example shows how the **FMDiagramDataExport** transfer mode (provided by Systems Engineering) is used in deciding the content of a diagram.



Setting diagram template relation rules

General Relationship Management (GRM) relations among Teamcenter objects appear as connector shapes in Visio diagrams. For each diagram template, a *relation rule* lists the GRM relations that can be represented in diagrams based on the template.

You select the relations from the **Relation Rule** list in the **New Diagram Template** or **Edit Diagram Template** dialog box.

Note:

Relation Rule choices can vary by application domain, depending on custom configurations at your site. Some domains may have built-in relation rules and may not support rule selection.

The relations are available in the template's diagrams when you do the following:

- Create shapes for the relations in a Visio stencil that is associated with the template.
- Map the relation shapes from the stencil to the corresponding relations in Teamcenter.

Then, you can work from a diagram to create those relations between objects in Systems Engineering:

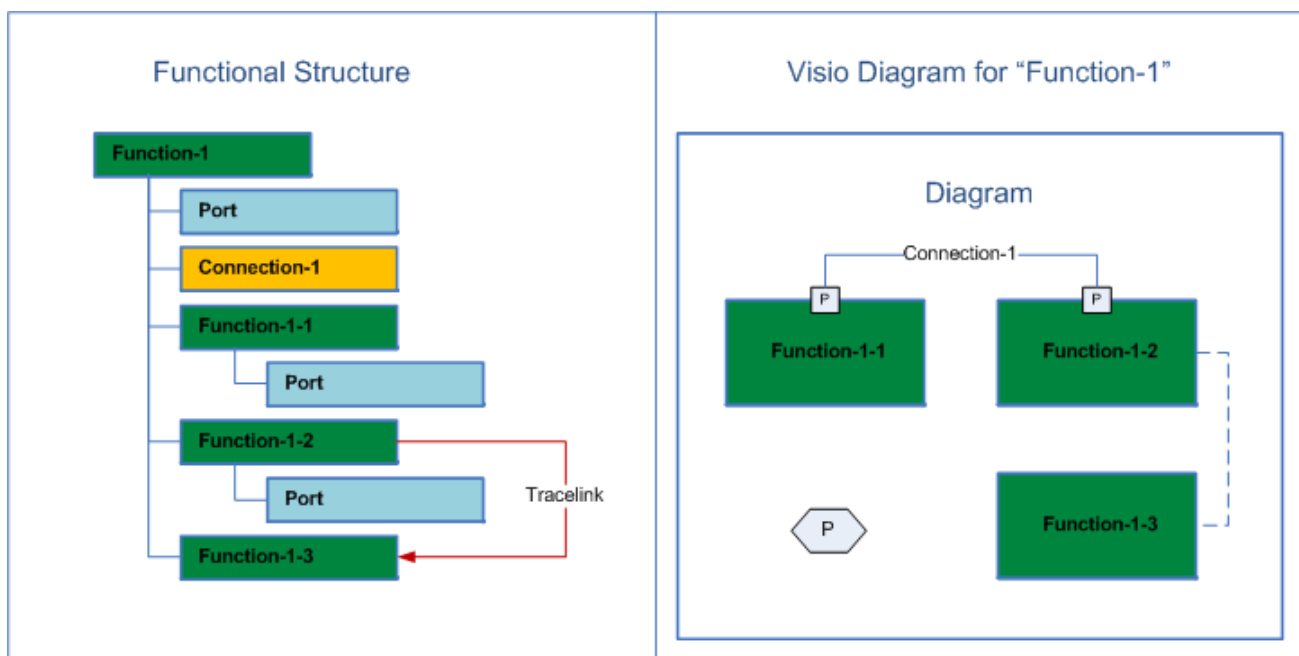
- From the stencil, drop a relation shape on the diagram page.
- Glue the relation shape ends to the connection points on the shapes that represent the related objects.

A corresponding relation is created in the structure.

The same relations are available when you use the template to create a diagram for a selected structure in Systems Engineering. For example, assume the following:

- For the diagram template, the membership rule specifies that functions, interfaces (ports), and connections appear in the diagram.
- The template's relation rule includes a Teamcenter relation called **Tracelink**.
- The stencil contains a master shape defined as a dotted line, which is mapped to the **Tracelink** relation.
- In the **Function** view, you select **Function-1** as the root object.

When you choose **File→Create Diagram**, the new diagram displays the shapes that represent the **Function-1** structure elements, including the trace link between **Function-1-2** and **Function-1-3**.



Activating trace link subtypes for diagrams

Custom trace link subtypes are defined in the Business Modeler IDE. When a custom subtype is deployed, it is available for assignment to trace links between structure elements in Teamcenter.

To use the trace link subtype in diagrams of structures, you must relate it to a diagram template by doing the following:

- Add the subtype’s *internal* name to the **NE_relation_rule_grm_list** preference value.

Caution:

Use the subtype’s class definition name from the Business Modeler IDE, not the name displayed in the Teamcenter rich client.

- Map a trace link master shape to the custom trace link subtype.
- In the diagram template, associate the stencil with the mapping file and select the custom trace link relation in the **Relation Rule** list.

Note:

You can do both of these actions in the **New Diagram Template** dialog box or the **Edit Diagram Template** dialog box.

In the mapping file, declare the master shape in the **DiagramShapeMap** entry, with a **typeMap** tag to map it to the subtype. For example:

```
<DiagramShapeMap DiagramShapeName="trace_link_master_shape">
  <typeMaps>
    <typeMap BaseFoundationClass="FND_TraceLink" SubClassName="" />
    <typeMap BaseFoundationClass="FND_TraceLink"
      SubClassName="trace_link_subtype"
      default="Boolean"/>
  </typeMaps>
</DiagramShapeMap>
```

Parameter	Value
DiagramShapeName	Replace <i>trace_link_master_shape</i> with the exact name of the master shape in the stencil.
SubClassName	Replace <i>trace_link_subtype</i> with the class definition name from the Business Modeler IDE.
	Caution: Do not use the name displayed in the Teamcenter rich client.
default	(Optional) Replace <i>Boolean</i> with true to designate the subtype as the default object type for the trace link master shape.

Parameter	Value
	Caution: If the master shape has another typeMap tag with true as the default value, change that value to false or delete that default parameter entirely.
	Note: You can use additional typeMaps entries to define more mapping details for the master shape.

Setting the diagram template availability flag

An *availability* flag indicates whether the diagram template is available for users to select in the **Create Diagram** dialog box. You set this flag through the **Available** check box in the **New Diagram Template** or **Edit Diagram Template** dialog box.

- When the check box is cleared, the dialog box does not display the diagram template name in the **Select Template** list.

Note:

The check box is cleared by default.

- When the check box is selected, the **Select Template** list displays the diagram template name.

For example:

- When you create the diagram template, you can leave the check box cleared while you create new diagram types that use the template.

Then, you can edit the template and make it available for new diagrams by selecting the check box.

- You can clear the check box for an obsolete diagram template that is no longer used for new diagrams.

Enable connections without interfaces in diagrams

For Visio diagrams, you can specify that connection shapes are glued directly to functions and logical blocks. With this setting, diagrams:

- Do not require interfaces for creating connections.

- Do not display interfaces that exist in the corresponding structure.

Note:

Diagram templates must include the **Hide Interface Shapes** option.

1. Choose **Edit→Options** to display the **Options** dialog box.
2. Use the **Search** feature to display the **Connected_ToRules** preference in the **Preferences List** pane.
3. Under **Current Values**, select **Network:Functionality**.
4. Do one of the following:
 - Click **Apply** to commit your edits and keep the **Options** dialog box open.
 - Click **OK** to apply your edits and close the dialog box.

Hiding interfaces in diagrams

In the **New Diagram Template** dialog box, the **Hide Interface Shapes** option indicates that interfaces are not displayed in diagrams based on the template.

When the diagram template is selected in the **Create Diagram** dialog box, the new diagram does not display shapes for interfaces that exist in the corresponding structure. Connection shapes are glued directly to the function or logical block shapes.

Warning:

- The **Hide Interface Shapes** option cannot be changed after the diagram template is created.
- A diagram without interface shapes cannot be changed to display interfaces.

Create a diagram template

Note:

Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.

1. Choose **Tools→Diagram→Create Template** to open the **New Diagram Template** dialog box.
2. In the **Select Template Type** dialog box, accept the default setting of **Visio Template**.
3. Enter an identifier, name, and description for the template.

4. Select the application domain for the template's diagrams.

Note:

You cannot change the application domain after the template is created.

You can also do the following optional actions:

- Select the **Available** check box to make the template available for creating new diagrams.
- Select the **Hide Interface Shapes** check box to specify that the template's diagrams do not display interface shapes for connections that exist in the corresponding structures.

Connection without interfaces must be enabled by setting the **Connected_ToRules** preference to **true**.

Caution:

- The **Hide Interface Shapes** setting cannot be changed after the diagram template is created.
- A diagram without interfaces cannot be changed to display interfaces.

5. Select a membership rule.

When an object is selected in a diagram, the membership rule determines the set of related objects that may appear. Membership rule choices vary with each application domain.

6. Select general relation management (GRM) rules to be used as relation rules.

When objects are selected in a diagram, the relation rule determines which relations among the objects appear as available connector shapes. The relation rule choices vary with each application domain.

7. Select a stencil file containing shapes to associate with the diagram template.

You can select two or more stencils with the standard Windows features for multiple selections.

Note:

The default stencil file depends on the diagram template's application domain.

- For the **Function** view, the default stencil file is **FM_Diagram_Stencil.vsx**.
- For the **Logical Block** view, the default stencil file is **LM_Diagram_Stencil.vsx**.

8. Select a mapping file that maps the selected stencil file to Teamcenter object types and properties.

Note:

The default mapping file depends on the diagram template's application domain.

- For the **Function** view, the default mapping file is **FM_Diagram_prop_map.xml**.
- For the **Logical Block** view, the default mapping file is **LM_Diagram_prop_map.xml**.

9. Click **OK** to create the diagram template object and close the **New Diagram Template** dialog box.

Note:

By default, new diagram templates are stored in the **NewStuff** folder.

Edit a diagram template

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- You cannot change the application domain for an existing diagram template.

1. Select the diagram template, and then choose **Tools→Diagram→Edit Template** to open the **Edit Diagram Template** dialog box.
2. Do any or all of the following:
 - Edit the name or description for the template.
 - Select the **Available** check box to make the template available for creating new diagrams.
 - Select the **Hide Interface Shapes** check box to specify that the template's diagrams do not display interface shapes for connections that exist in the corresponding structures.

Caution:

- Connections without interfaces must be enabled by setting the **Connected_ToRules** preference to **true**.
- The **Hide Interface Shapes** setting cannot be changed after the diagram template is created.

- A diagram without interfaces cannot be changed to display interfaces.

- In the **Membership Rule** list, select a different rule for the object types that can be represented in the template's diagrams.

Membership rule choices can vary with each application domain.

- In the **Relation Rule** list, select different General Relation Management (GRM) relations for the connector shapes that can appear in the template's diagrams.

Relation rule choices can vary with each application domain.

- In the **Select Template/Stencils** list, select a different stencil file to associate with the diagram template.

You can select two or more stencils with the standard Windows features for multiple selections.

Note:

The default stencil file depends on the diagram template's application domain.

- For the **Function** view, the default stencil file is **FM_Diagram_Stencil.vsx**.
- For the **Logical Block** view, the default stencil file is **LM_Diagram_Stencil.vsx**.

- In the **Select Mapping File** list, select a mapping file with different Teamcenter objects and properties for the stencil's master shapes.

3. Click **OK** to commit your changes and close the **Edit Diagram Template** dialog box.

Defining master shapes

Define a block master shape

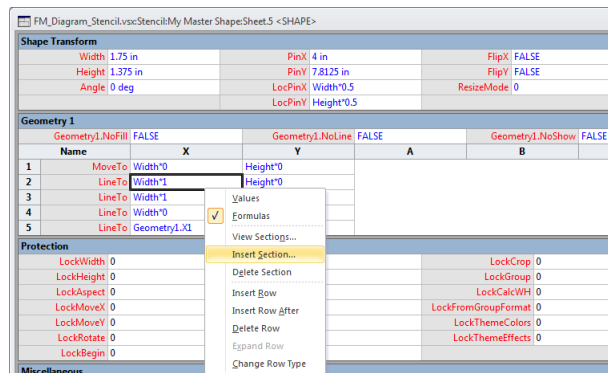
1. In the Teamcenter stencil file (**.vsx** or **.vssx**), open the stencil that contains the master shape.
2. Right-click the master shape and choose **Edit Master**→**Edit Master Shape**.

If the menu command is disabled, the stencil may be in read-only mode.

3. Right-click the master shape in the **Edit Master** window and choose **Show ShapeSheet**.

You can also choose **Show ShapeSheet** from the **Developer** tab.

4. Right-click anywhere in the shape sheet and choose **Insert Section**.

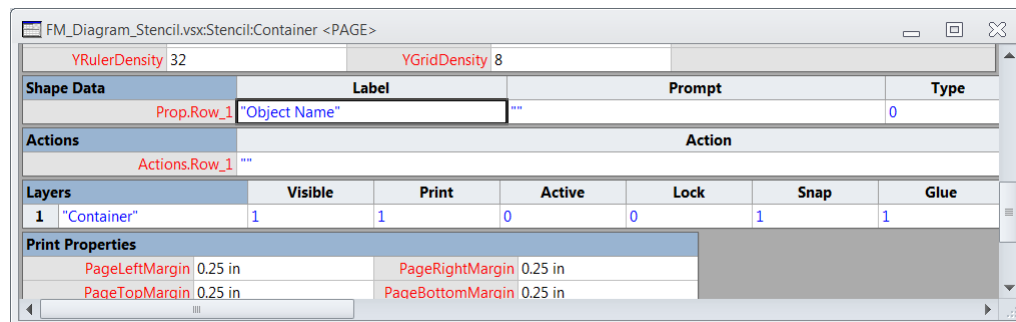


5. In the **Insert Section** dialog box, select the **Shape Data** and **Actions** check boxes.

If you want to add connection points, also select the **Connection points** check box.

Click **OK** to close the dialog box.

6. In the **Shape Data** section of the shape sheet, type **Object Name** in the first box under the **Label** box. Do not change any of the rest of the boxes.



7. Modify the **Actions** section of the shape sheet to create the shortcut menu commands.

Right-click and insert rows in this section and edit the various boxes to match the following table.

Actions	Action	Menu	Tag name	Button face
Actions.Properties	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=Properties")	"Properties"	""	""
Actions.BeginTraceLink	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=BeginTraceLink")	"Start Trace Link"	""	""
Actions.EndTraceLink	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=EndTraceLink")	"End Trace Link"	""	""

Actions	Action	Menu	Tag name	Button face
Actions. EndTraceLinkAsSubtype	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=EndTraceLinkAsSubtype")	"End Trace Link..."	""	""
Actions. RemoveFromDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=RemoveFromDiagram")	"Remove from Diagram"	""	""
Actions. GotoTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=GoToTeamcenter")	"Go to Teamcenter"	""	""
Actions. OpenDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=OpenDiagram")	"Open Diagram"	""	""
Actions. DeleteInTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=DeleteInTeamcenter")	"Delete in Teamcenter"	""	""

The section should look like this.

11	Width*0.5	Height*0	0 in	0 in	0			
12	Width*0.75	Height*0	0 in	-1 in	0			
Actions	Action					Menu	TagName	ButtonFace
Actions.Properties	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=Properties")					"Properties"	""	""
Actions.BeginTraceLink	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=BeginTraceLink")					"Begin Trace Link"	""	""
Actions.EndTraceLink	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=EndTraceLink")					"End Trace Link"	""	""
Actions.EndTraceLinkAsSubtype	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=EndTraceLinkAsSubtype")					"End Trace Link..."	""	""
Actions.RemoveFromDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=RemoveFromDiagram")					"Remove from Diagram"	""	""
Actions.GoToTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=GoToTeamcenter")					"Go to Teamcenter"	""	""
Actions.OpenDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=OpenDiagram")					"Open Diagram"	""	""
Actions.DeleteInTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=DeleteInTeamcenter")					"Delete in Teamcenter"	""	""
Geometry 1								

8. (Optional) Define the connection points in the **Connection Points** section of the master shape sheet.

You can define as many connection points as you want.

Right-click and insert rows in this section and edit the various boxes so it is similar to the following table.

Connection points	X	Y	DirX / A	DirY / B	Type / C
1	Width*1	Height*0.25	-1 in	0 in	0
2	Width*1	Height*0.5	0 in	1 in	0
3	Width*1	Height*0.75	1 in	0 in	0
4	Width*0	Height*0.75	0 in	1 in	0
5	Width*0	Height*0.5	0 in	1 in	0
6	Width*0	Height*0.25	-1 in	0 in	0
7	Width*0.5	Height*1	0 in	-1 in	0

Connection points	X	Y	DirX / A	DirY / B	Type / C
8	Width*0.25	Height*1	0 in	0 in	0
9	Width*0.75	Height*1	0 in	0 in	0
10	Width*0.25	Height*0	0 in	0 in	0
11	Width*0.5	Height*0	0 in	0 in	0
12	Width*0.75	Height*0	0 in	-1 in	0

Prop.Row.1 "Object Name"		""		0		""	
Connection Points	X	Y	DirX / A	DirY / B	Type / C	D	
1	Width*1	Height*0.25	-1 in	0 in	0		
2	Width*1	Height*0.5	0 in	1 in	0		
3	Width*1	Height*0.75	1 in	0 in	0		
4	Width*0	Height*0.75	0 in	1 in	0		
5	Width*0	Height*0.5	0 in	1 in	0		
6	Width*0	Height*0.25	-1 in	0 in	0		
7	Width*0.5	Height*1	0 in	-1 in	0		
8	Width*0.25	Height*1	0 in	0 in	0		
9	Width*0.75	Height*1	0 in	0 in	0		
10	Width*0.25	Height*0	0 in	0 in	0		
11	Width*0.5	Height*0	0 in	0 in	0		
12	Width*0.75	Height*0	0 in	-1 in	0		

- Close the shape sheet and **Edit Master** window to save and update the master shape.
- In the related mapping file (.xml), enter a **BlockShape** tag for the master shape.

The **BlockShapes** section identifies which shapes are defined as blocks, such as function or logical blocks. You can enter multiple **BlockShape** tags, each tag relating a specific block to a specific shape. Enter the **Default="true"** parameter for the default block shape.

```
<BlockShapes>
  <BlockShape Name="Function" Default="true"/>
</BlockShapes>
```

Define an interface master shape

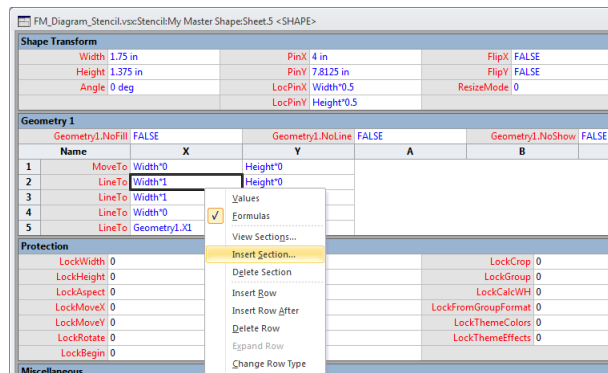
Note:

You can define separate master shapes for input interfaces and output interfaces in the same stencil.

- In the Teamcenter stencil file (.vsx or .vssx), open the stencil that contains the master shape.
- Right-click the master shape and choose **Edit Master**→**Edit Master Shape**.
If the menu command is disabled, the stencil may be in read-only mode.
- Right-click the master shape in the **Edit Master** window and choose **Show ShapeSheet**.

You can also choose **Show ShapeSheet** from the **Developer** tab.

4. Right-click anywhere in the shape sheet and choose **Insert Section**.



5. In the **Insert Section** dialog box, select the **Shape Data**, **Connection points**, and **Actions** check boxes. Click **OK**.
6. In the **Shape Data** section of the shape sheet, type **Object Name** in the first box under the **Label** box. Do not change any of the rest of the boxes.

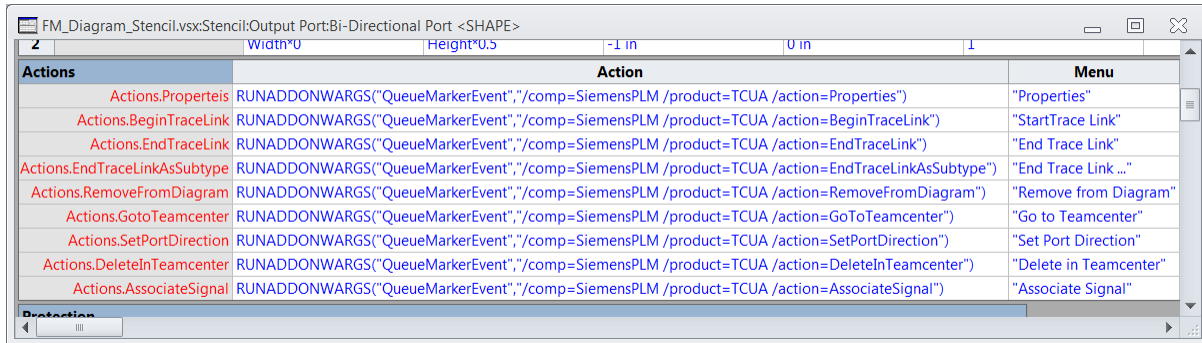
User-defined Cells	Value	Prompt	Type	Format
Shape Data	Prop.Row.1 "Object Name"		0	0

7. Modify the **Actions** section of the shape sheet to create the shortcut menu commands.

Right-click and insert rows in this section and edit the various boxes to match the following table.

Actions	Action	Menu
Action.Properties	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=Properties")	"Properties"
Actions.BeginTraceLink	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=BeginTraceLink")	"Start Trace Link"
Actions.EndTraceLink	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=EndTraceLink")	"End Trace Link"
Actions.EndTraceLinkAsSubtype	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=EndTraceLinkAsSubtype")	"End Trace Link..."
Actions.RemoveFromDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=RemoveFromDiagram")	"Remove from Diagram"
Actions.GotoTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=GoToTeamcenter")	"Go to Teamcenter"
Actions.SetPortDirection	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=SetPortDirection")	"Set Port Direction"

Actions	Action	Menu
Actions.DeleteInTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=DeleteInTeamcenter")	"Delete in Teamcenter"
Actions.AssociateSignal	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=AssociateSignal")	"Associate Signal"



8. Define the connection points in the **Connection Points** section of the master shape sheet.

You can add as many connection points as you want.

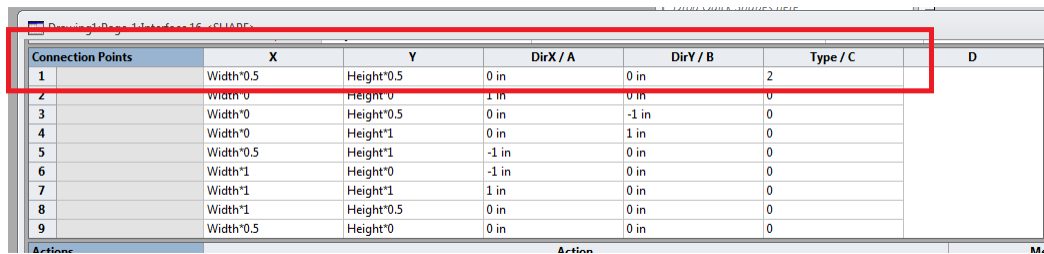
Right-click and insert rows in this section and edit the various boxes so it is similar to the following table.

Caution:

You must define connection point 1 as shown in the table and the red box in the figure to create a connection point that can be attached to any noninterface shape, such as a function.

Connection points	X	Y	DirX / A	DirY / B	Type / C
1	Width*0.5	Height*0.5	0 in	0 in	2
2	Width*0	Height*0	1 in	0 in	0
3	Width*0	Height*0.5	0 in	-1 in	0
4	Width*0	Height*1	0 in	1 in	0
5	Width*0.5	Height*1	-1 in	0 in	0
6	Width*1	Height*0	-1 in	0 in	0
7	Width*1	Height*1	1 in	0 in	0
8	Width*1	Height*0.5	0 in	0 in	0
9	Width*0.5	Height*0	0 in	0 in	0

The section should look like this.



Connection Points	X	Y	DirX / A	DirY / B	Type / C	D
1	Width*0.5	Height*0.5	0 in	0 in	2	
2	Width*0	Height*0	1 in	0 in	0	
3	Width*0	Height*0.5	0 in	-1 in	0	
4	Width*0	Height*1	0 in	1 in	0	
5	Width*0.5	Height*1	-1 in	0 in	0	
6	Width*1	Height*0	-1 in	0 in	0	
7	Width*1	Height*1	1 in	0 in	0	
8	Width*1	Height*0.5	0 in	0 in	0	
9	Width*0.5	Height*0	0 in	0 in	0	

9. Close the shape sheet and **Edit Master** window to save and update the master shape.
10. In the related mapping file, enter a **PortShape** tag for each interface master shape.

The **PortShapes** section identifies which shapes are defined as interfaces. Each **PortShape** tag relates a specific interface to a specific shape.

```
<PortShapes>
  <PortShape Name="Interface" Default="true"/>
</PortShapes>
```

Note:

- Inward connection point [0]
Attracts endpoints of 1D shapes and outward or inward/outward connection points of other shapes. It is appropriate for block and function shapes.
- Outward connection point [1]
Attracts inward and inward/outward connection points of other shapes.
- Inward/outward connection point [2]:
Behaves like an inward connection point for outward connection points and like an outward connection point when being attached to shapes with inward connection points.

Define a connection master shape

Note:

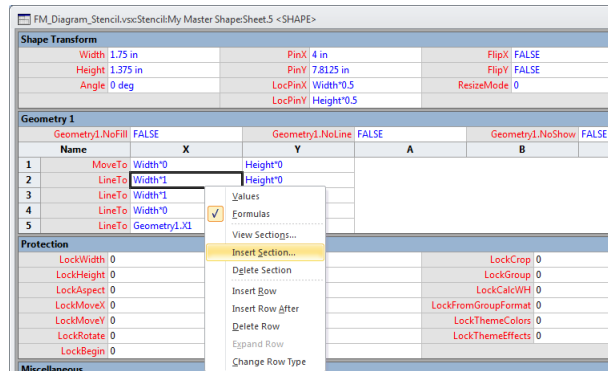
- Connection master shapes and relation master shapes are mapped to connection objects in Teamcenter. Connection shapes differ in that they do not use relation rules.
- By selecting the **Hide Interface Shapes** option when you create the diagram template, you can use diagrams that do not show interfaces for connection shapes.

1. In the Teamcenter stencil file (**.vsx** or **.vssx**), open the stencil that contains the master shape.

- Right-click the master shape and choose **Edit Master→Edit Master Shape**.

If the menu command is disabled, the stencil may be in read-only mode.

- Right-click the master shape in the **Edit Master** window and choose **Show ShapeSheet**.
- Right-click anywhere in the shape sheet and choose **Insert Section**.



- In the **Insert Section** dialog box, select the **ShapeData** and **Actions** check boxes, and then click **OK**.
- In the **Shape Data** section of the shape sheet, type **Object Name** in the first box under the **Label** box.

Do not change any of the rest of the boxes.

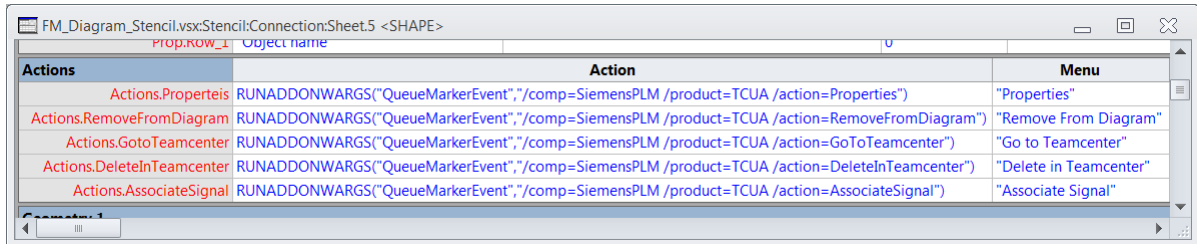
User-defined Cells					
	Value	Prompt			
Shape Data		Label	Prompt	Type	Format
	Prop.Row.1	"Object Name"		0	0

- Modify the **Actions** section of the shape sheet to create the shortcut menu commands.

Right-click and insert rows in this section and edit the various boxes to match the following table.

Actions	Action	Menu
Action.Properties	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=Properties")	"Properties"
Actions.RemoveFromDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=RemoveFromDiagram")	"Remove from Diagram"
Actions.GotoTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM / product=TCUA /action=GoToTeamcenter")	"Go to Teamcenter"

Actions	Action	Menu
Actions.DeleteInTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=DeleteInTeamcenter")	"Delete in Teamcenter"
Actions.AssociateSignal	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=AssociateSignal")	"Associate Signal"



- Close the shape sheet and **Edit Master** window to save and update the master shape.
- In the related mapping file (.xml), enter a **ConnectionShape** tag for each connection master shape.

The **ConnectionShapes** section identifies which shapes are defined as connections. Each **ConnectionShape** tag relates a specific connection to a specific shape.

```
<ConnectionShapes>
  <ConnectionShape Name="Connection" Default="true"/>
</ConnectionShapes>
```

Define a relation master shape

Note:

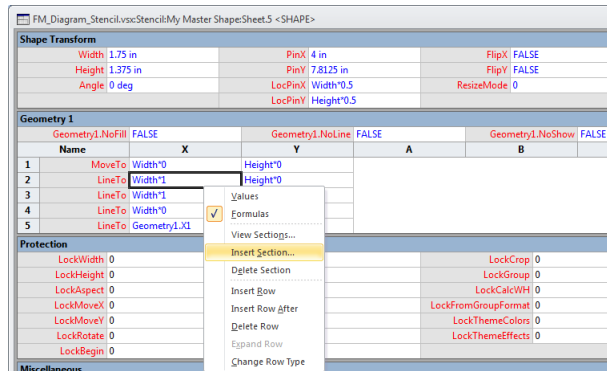
- You can define relation master shapes that show the names of the defining and complying objects for trace links. Also, such a relation shape can show an arrow pointing to the defining object.
- Relation master shapes and connection master shapes are mapped to connection objects in Teamcenter. Relation shapes differ in that they use relation rules to specify Teamcenter GRM relations.

- In the Teamcenter stencil file (.vsx or .vssx), open the stencil that contains the master shape.
- Right-click the master shape and choose **Edit Master**→**Edit Master Shape**.

If the menu command is disabled, the stencil may be in read-only mode.

- Right-click the master shape in the **Edit Master** window and choose **Show ShapeSheet**.

4. Right-click anywhere in the shape sheet and choose **Insert Section**.



5. In the **Insert Section** dialog box, select the **ShapeData** and **Actions** check boxes and click **OK**.
6. In the **Shape Data** section of the shape sheet, type **Object Name** in the first box under the **Label** box.

Do not change any of the rest of the boxes.

User-defined Cells	Value	Prompt	Type	Format
Shape Data	Prop.Row.1	"Object Name"	0	0

7. Modify the **Actions** section of the shape sheet to create the shortcut menu commands.

Right-click and insert rows in this section and edit the various boxes to match the following table.

Actions	Action	Menu	Tag name	Button face
Action.Properties	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=Properties")	"Properties"	""	""
Actions. RemoveFromDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=RemoveFromDiagram")	"Remove from Diagram"	""	""
Actions. GotoTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=GoToTeamcenter")	"Go to Teamcenter"	""	""
Actions. DeleteInTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=DeleteInTeamcenter")	"Delete in Teamcenter"	""	""

The section should look like this.

Shape Data	Label	Prompt	Type	Format
Prop.Row_1	"Object name"		0	
Actions	Action	Menu	TagName	
Actions.Properties	RUNADDONWARGS("QueueMarkerEvent","/comp=SiemensPLM /product=TCUA /action=Properties")	"Properties"		
Actions.RemoveFromDiagram	RUNADDONWARGS("QueueMarkerEvent","/comp=SiemensPLM /product=TCUA /action=RemoveFromDiagram")	"Remove From Diagram"		
Actions.GoToTeamcenter	RUNADDONWARGS("QueueMarkerEvent","/comp=SiemensPLM /product=TCUA /action=GoToTeamcenter")	"Go to Teamcenter"		
Actions.DeleteInTeamcenter	RUNADDONWARGS("QueueMarkerEvent","/comp=SiemensPLM /product=TCUA /action=DeleteInTeamcenter")	"Delete in Teamcenter"		

- Close the shape sheet and **Edit Master** window to save and update the master shape.
- In the related mapping file (.xml), enter a **RelationShape** tag for each relation master shape.

Within the **RelationShapes** section, the **RelationShape** tags identify the shapes defined as Teamcenter relations.

```
<RelationShapes>
  <RelationShape Name="TraceLink"/>
</RelationShapes>
```

Define a container master shape

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Teamcenter containers are available only in diagrams whose owners exist in the **Function** and **Logical Block** views.

- In Visio, open the Teamcenter stencil in which you created the container master shape.

You can browse to and select the stencil by clicking the **More Shapes** bar and choosing **Open Stencil**.

Note:

If the stencil's title bar does not display a red asterisk, right-click the title bar and choose **Edit Stencil**. The red asterisk indicates that you can edit the master shapes.

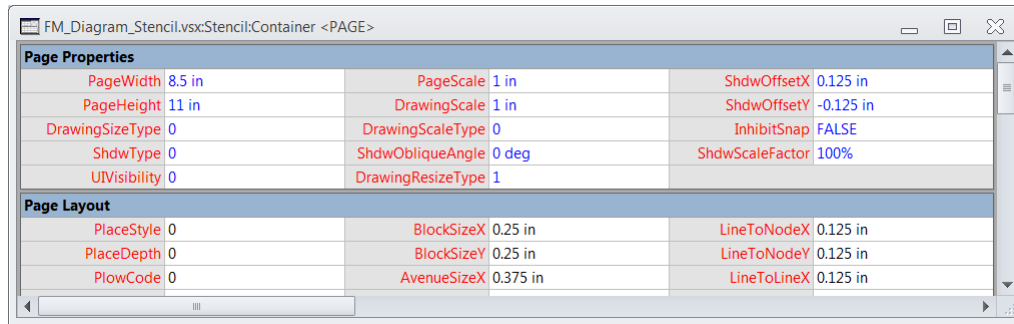
- Right-click the container master shape and choose **Edit Master**→**Edit Master Shape**.

A new pane opens to display the master shape for editing.

- In the editing pane:
 - Move the pointer on the master shape until it becomes a four-pointed arrow, and then click the shape.

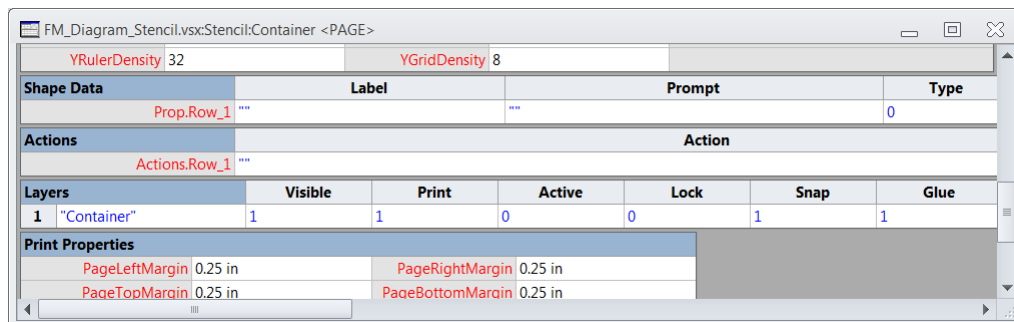
- b. Right-click the master shape and choose **Show ShapeSheet**.

The ShapeSheet opens in a new pane.



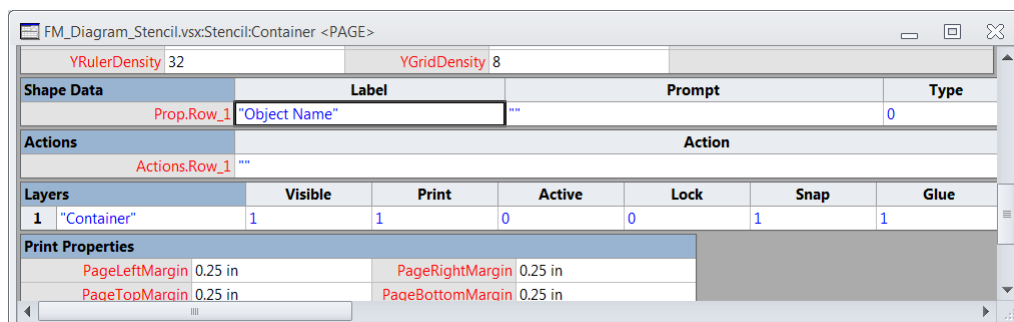
4. In the ShapeSheet, right-click anywhere and choose **Insert Section**.
5. In the **Insert Section** dialog box, select the **Shape Data**, **Actions**, and **Connection points** check boxes, and then click **OK**.

The sections are added to the ShapeSheet.



6. In the **Shape Data** section, type **Object Name** in the **Label** column.

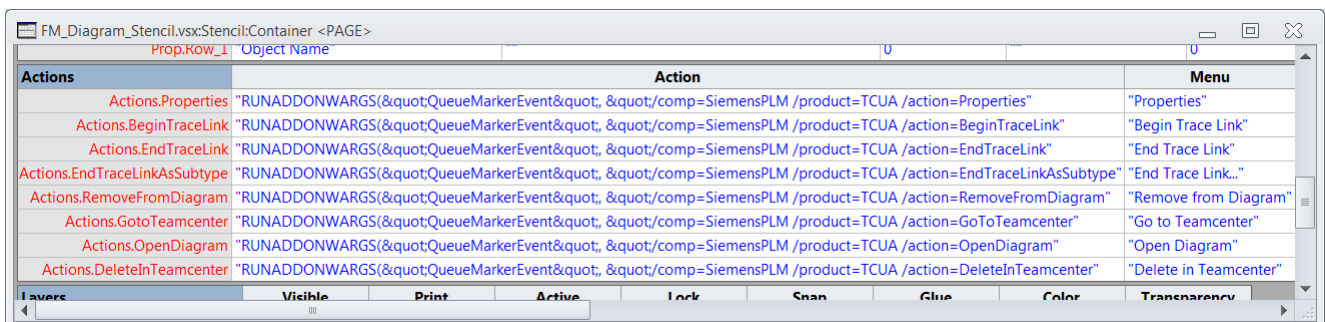
Do not make any other entries.



7. In the **Actions** section, enter the formulas for the shortcut menu options.

Insert additional rows as necessary.

Actions	Action	Menu
Actions.Properties	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=Properties")	"Properties"
Actions.BeginTraceLink	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=BeginTraceLink")	"Begin Trace Link"
Actions.EndTraceLink	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=EndTraceLink")	"End Trace Link"
Actions.EndTraceLinkAsSubtype	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=EndTraceLinkAsSubtype")	"End Trace Link..."
Actions.RemoveFromDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=RemoveFromDiagram")	"Remove from Diagram"
Actions.GotoTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=GoToTeamcenter")	"Go to Teamcenter"
Actions.OpenDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=OpenDiagram")	"Open Diagram"
Actions.DeleteInTeamcenter	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=DeleteInTeamcenter")	"Delete in Teamcenter"
Actions.CreateDiagram	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=CreateDiagram")	"Create Diagram"
Actions.CreateDiagramDiff	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=CreateDiagramDiff")	"Create Diagram..."
Actions.createAndPaste	RUNADDONWARGS("QueueMarkerEvent", "/comp=SiemensPLM /product=TCUA /action=createAndPasteNewObject")	"Create New Object and Paste"



- In the **Connection Points** section, enter formulas for connection points on the master shape.

You can specify as many connection points as you want. For example:

Connection points	X	Y	DirX / A	DirY / B	Type / C
1	Width*0.25	Height*1	0 in	-1 in	0
2	Width*0.5	Height*1	0 in	-1 in	0
3	Width*0.75	Height*1	0 in	-1 in	0
4	Width*1	Height*0.75	-1 in	0 in	0
5	Width*1	Height*0.5	-1 in	0 in	0
6	Width*1	Height*0.25	-1 in	0 in	0
7	Width*0.75	Height*0	0 in	1 in	0
8	Width*0.5	Height*0	0 in	1 in	0
9	Width*0.25	Height*0	0 in	1 in	0
10	Width*0	Height*0.25	1 in	0 in	0
11	Width*0	Height*0.5	1 in	0 in	0
12	Width*0	Height*0.75	1 in	0 in	0

9. Close the shape sheet and the editing pane, and then click **Yes** to save your edits.
10. In the related mapping file (.xml), enter a **ContainerShape** tag for the master shape.

The **ContainerShapes** section identifies which shapes are defined as containers. Each **ContainerShape** tag relates a specific object to a specific shape.

Container shapes are mapped to functions and logical blocks. You can specify multiple container shapes.

```
<ContainerShapes>
  <ContainerShape Name="Container"/>
</ContainerShapes>
```

Defining compound master shapes for showing multiple properties

Note:

Siemens Digital Industries Software assumes that you are familiar with Microsoft Visio and with XML.

Using the Visio grouping features, you can create master shapes that consist of two or more simple shapes. Because you can map each subshape's text block to a different Teamcenter property, a shape based on a compound master can show multiple properties of the object that it represents.

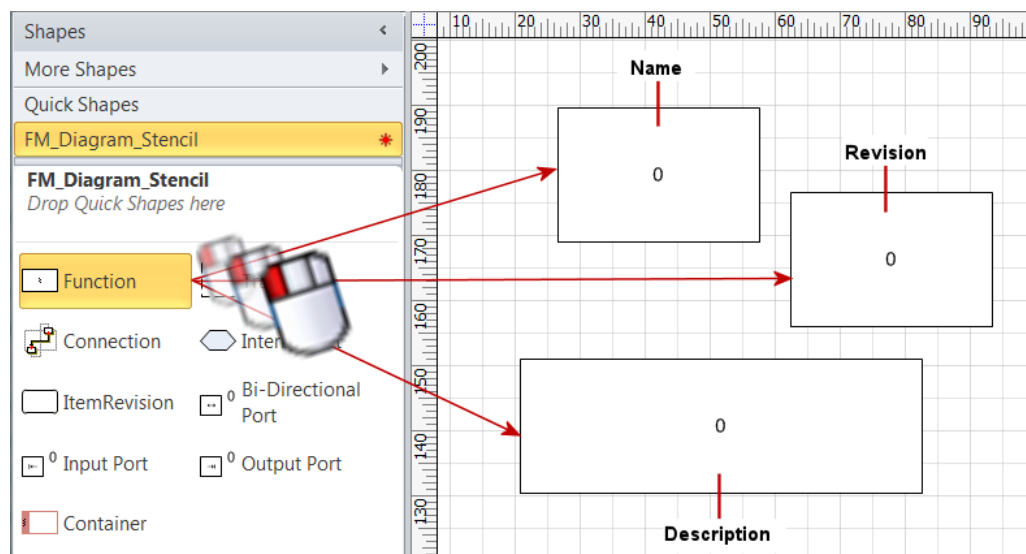
Example:

Assume that you want a function shape to show the object name, revision identifier, and description. To create a compound master shape for this purpose, your steps can be similar to those outlined below.

- Add three **Function** shapes from the Teamcenter stencil, and then name each shape.

Note:

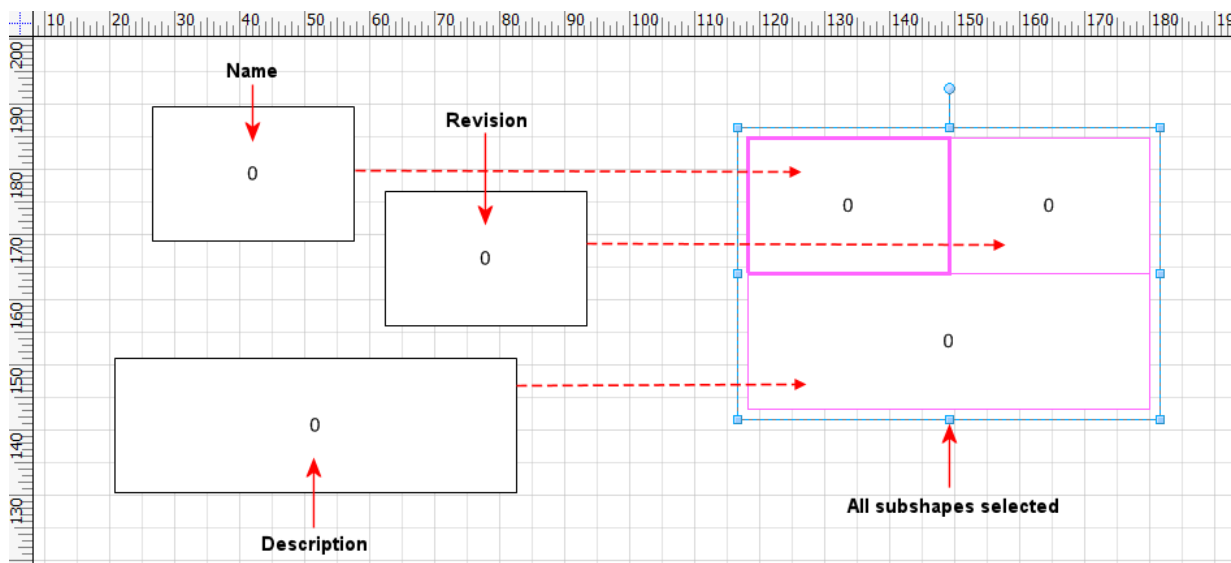
For this example, the subshapes are **Name**, **Revision**, and **Description**, each name relating to the property intended for its text block.



Tip:

You can name a subshape by right-clicking it and selecting **Data → Define Shape Data** to open the Visio **Define Shape Data** dialog box.

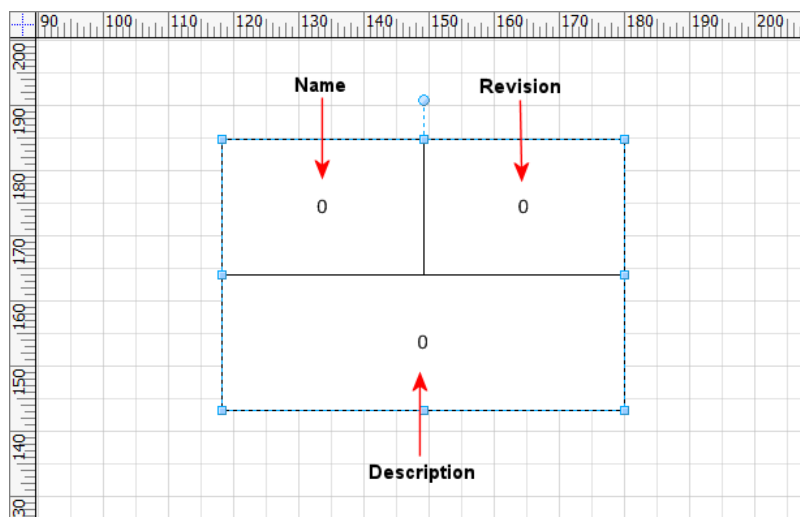
- Arrange the shapes contiguously, and then select all three simultaneously.



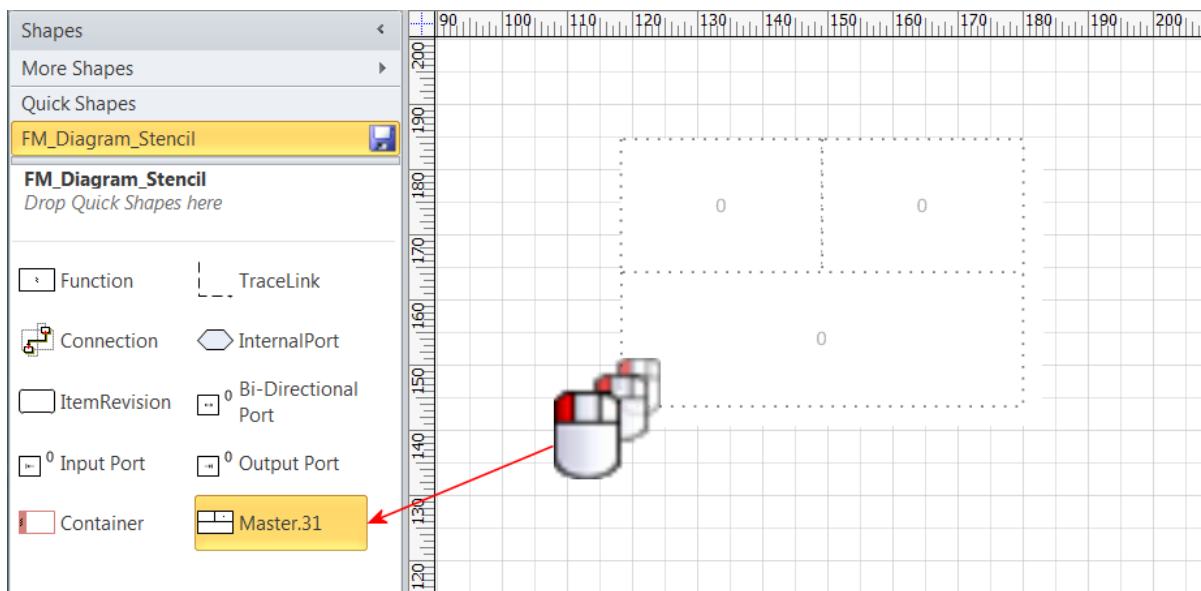
- On the Visio **Home** tab, click the **Group** button to join the individual blocks as one shape.

Caution:

Before grouping the subshapes, ensure that they all represent the same object type and that each subshape has its own name.



- Add the master shape by dragging the compound shape from the drawing pane to the stencil.



Tip:

- If necessary, open the stencil for editing by right-clicking the title bar and choosing **Edit Stencil**.
- Click the **Save Stencil** button on the title bar after you add and rename the master.

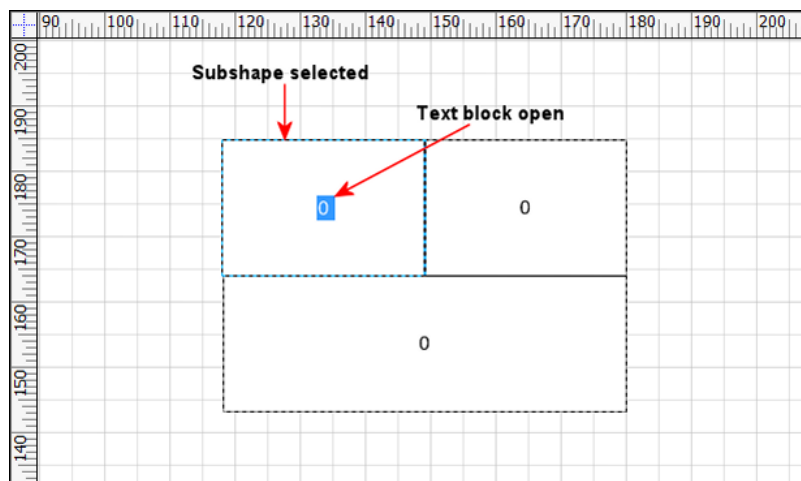
- In the master's ShapeSheet, navigate to the **Group Properties** section and set the **IsTextEditTarget** value to **FALSE**.

FM_Diagram_Stencil.vsx:Stencil:Compound Function:Sheet.5 <SHAPE>			
LangID	1033	Calendar	0
		LocalizeMerge	FALSE
Group Properties			
SelectMode	1	IsTextEditTarget	FALSE
DisplayMode	2	IsSnapTarget	TRUE
		IsDropTarget	FALSE
		DontMoveChildren	FALSE
Line Format			

Tip:

You can open the ShapeSheet by right-clicking the master and choosing **Edit Master**→**Edit Master Shape**.

- To test the compound master, drag it to the drawing pane to add a shape, and then verify that you can select each subshape individually and can open its text block for editing by pressing F2.



Note:

- New shapes based on compound masters are not resized automatically to fit their text-block contents. You can adjust them with the Visio resizing features.
- The component subshapes cannot display multiple properties.

After testing and verifying a compound master shape, you must map it to Teamcenter with these XML entries in the mapping file:

- A **DiagramShapeMap** entry, to declare the master shape name and flag it as a compound shape.
- A **typeMaps** entry, to relate the master shape to a Teamcenter object type.
- A **textPropertyMaps** entry, to map the subshape text blocks to object properties.

Note:

Special reserved tags are required in the mapping for a compound master shape.

Note:

To make Teamcenter properties available in Microsoft Visio, in addition to the above-mentioned steps, you must follow the steps mentioned in making Teamcenter properties available in Microsoft Visio.

Mapping master shapes to Teamcenter

Teamcenter mapping file contents

In a Teamcenter mapping file (.xml), the **DiagramShapeMaps** section contains the entries for all master shapes in the stencil. A **DiagramShapeMap** entry defines the mapping for one master shape. It declares the name of the master shape and contains other entries with their tags.

The following code maps the **Function** master shape in **FM_Diagram_prop_map.xml**, the default mapping file for the **FunctionalModeling** application domain:

```

1 <DiagramShapeMap DiagramShapeName="Function" isGroup="false">
  <typeMaps>
2    <typeMap BaseFoundationClass="Item" SubClassName=""/>
    <typeMap BaseFoundationClass="Item" SubClassName="Functionality"
      default="true"/>
3    <DiagramShapeResolverPropertyMaps>
    <DiagramShapeResolverProperty tcPropName="" tcPropvalue=""/>
    </DiagramShapeResolverPropertyMaps>
4    <createDescriptorPropertyMaps>
    <createDescriptorProperty tcPropName="object_name"
      tcPropvalue="Function block"/>
    </createDescriptorPropertyMaps>
5    <textPropertyMaps>
    <textPropertyMap tcPropName="bl_rev_object_name"
      tcTypeName="Fnd0FunctionalBOMLine" DiagramShapePropName="Text"/>
    <textPropertyMap tcPropName="object_name" tcTypeName="Functionality"
      DiagramShapePropName="Text"/>
    </textPropertyMaps>
  </typeMaps>
</DiagramShapeMap>

```

1	DiagramShapeMap	Declares the master shape with the DiagramShapeName parameter.

Caution:

If the master is a compound shape, you must set the **IsGroup** parameter to **true**.

2	typeMaps	Relates the master shape to a Teamcenter object type.
3	DiagramShapeResolver PropertyMaps	Makes a master shape unique among others that are mapped to the same object type.

4	createDescriptorProperty Maps	Specifies a required property and default value for diagram shapes that are based on the master shape.
5	textPropertyMaps	Specifies a property whose value is displayed in the text boxes of diagram shapes that are based on the master shape.

Declaring a master shape

You declare a master shape with the **DiagramShapeName** parameter of the **DiagramShapeMap** entry.

Caution:

If you are declaring a compound master shape, you must set the **isGroup** parameter to **true**.

```
<DiagramShapeMap DiagramShapeName="mstr_name" isGroup="t_f">
  ...
</DiagramShapeMap>
```

DiagramShapeName Declares the name of the master shape for this entry. Replace *mstr_name* with the exact name of the master shape in the stencil.

isGroup For a compound master shape, replace *t_f* with **true**.

Caution:

The **isGroup** value must be **true** if the master shape is a compound shape. A compound shape consists of two or more shapes that are grouped together.

For a master shape that is a single ungrouped shape, replace *t_f* with **false** or omit the parameter entirely.

Note:

For the **TraceLink** master shape, the **Link** parameter replaces **isGroup**. Enter **true** if the shape represents the GRM¹ trace link relation. Otherwise, enter **false**.

GRM relations appear in diagrams as connector shapes. They are governed by the relation rule for the diagram template that is associated with the mapping file.

A **DiagramShapeMap** entry contains a **typeMaps** entry, which itself contains other entries.

Relating a master shape to a Teamcenter object type

In a **typeMaps** entry, a **typeMap** tag relates the master shape to a Teamcenter object type.

```
<typeMaps>
  <typeMap BaseFoundationClass="par_cl" SubClassName="mstr_sc"
    default="t_f"/>
  ...
</typeMaps>
```

Note:

You can enter any number of **typeMap** tags, for example, to map two subclasses to the same master shape.

BaseFoundationClass Replace *par_cl* with the parent class of the object type represented by the master shape.

Master shape	BaseFoundationClass
Function ²	Item
LogicalBlock ³	Item
TraceLink	FND_TraceLink
Connection	PSConnection
InternalPort	Interfaces
Item	Item
ItemRevision	ItemRevision
Bi-Directional Port	Interfaces
Input Port	Interfaces
Output Port	Interfaces
Container	Item

SubClassName Replace *mstr_sc* with the subclass of the object type represented by the master shape.

² In FM_Diagram_Stencil.vsx
³ In LM_Diagram_Stencil.vsx

Master shape	SubClassName (FunctionalModeling)	SubClassName (LogicalModeling)
Function	Functionality	N/A
LogicalBlock	N/A	Fnd0LogicalBlock
TraceLink	Not required	Not required
Connection	Network	Fnd0LogicConn
InternalPort	Network_Port Connection_Terminal	Fnd0LogicIntrfce
Item	Requirement Paragraph	Requirement Paragraph
ItemRevision	FunctionalityRevision	Fnd0LogicalBlockRevision
Bi-Directional Port	Network_Port	Fnd0LogicIntrfce
Input Port	Network_Port	Fnd0LogicIntrfce
Output Port	Network_Port	Fnd0LogicIntrfce
Container	Functionality	Fnd0LogicalBlock

default

(Optional) Replace *t_f* with **true** if the subclass is the default object type for the master shape. Otherwise, enter **false** for an existing **default** parameter, or delete the entire parameter.

Caution:

Only one **typeMap** tag per subclass can contain **default="true"**.

If no **typeMap** tag for a subclass contains **default="true"**, objects are not created in the structure when you drop master shapes into the drawing pane.

A **typeMaps** entry also contains the following entries:

- **DiagramShapeResolverPropertyMaps**
- **createDescriptorPropertyMaps**
- **textPropertyMaps**

Making a master shape unique

In a **DiagramShapeResolverPropertyMaps** entry, the **DiagramShapeResolverProperty** tag distinguishes a master shape from others that are mapped to the same object type.

```
<DiagramShapeResolverPropertyMaps>
  <DiagramShapeResolverProperty tcPropName="rslv_prop_name"
    tcPropvalue="rslv_prop_val" tcPriority="rslv_priority"/>
</DiagramShapeResolverPropertyMaps>
```

Caution:

A master shape can have only one **DiagramShapeResolverProperty** tag.

tcPropName	Replace <i>rslv_prop_name</i> with the name of the property whose value distinguishes the master shape from others that are mapped to the same object type.
tcPropvalue	Replace <i>rslv_prop_val</i> with the value that distinguishes the master shape from others that are mapped to the same object type.
tcPriority	Replace <i>rslv_priority</i> with the number that represents the property's order of precedence over properties named in other DiagramShapeResolverProperty tags. The highest priority is zero (0).

Note:

A **DiagramShapeResolverProperty** tag without a **tcPriority** value is assigned the lowest priority (9999).

Example:

Assume that a Teamcenter stencil contains two different master shapes, and each **DiagramShapeMap** entry contains a **typeMap** tag with the **BaseFoundationClass=Interfaces** parameter value.

- One shape represents *internal* interfaces within single functions or logical blocks.
- One shape represents *external* interfaces between parent objects and their direct children.

The **DiagramShapeResolverProperty** tag maps the internal interface shape:

```
<DiagramShapeResolverPropertyMaps>
  <DiagramShapeResolverProperty tcPropName="bl_level_starting_0"
```

```
tcPropvalue="1 tcPriority="0"/>
</DiagramShapeResolverPropertyMaps>
```

Specifying required properties on a master shape

In a **createDescriptorPropertyMaps** entry, the **createDescriptorProperty** tag specifies a required property and its default value. These parameters apply to each Teamcenter object that is created when you drop the master shape in the Visio drawing pane.

```
<createDescriptorPropertyMaps>
  <createDescriptorProperty tcPropName="dscr_prop_name"
    tcPropvalue="dscr_prop_val"/>
</createDescriptorPropertyMaps>
```

Caution:
A master shape can have only one **createDescriptorProperty** tag.

- tcPropName**
- Replace *dscr_prop_name* with the name of the required property.
- tcPropvalue**
- Replace *dscr_prop_val* with the default value for the required property.
- For example, you can designate the new object as a function by replacing *dscr_prop_name* with **object_type** and replacing *dscr_prop_val* with **Function Block**.
- You can also assign default values for items, item revisions, and master forms. For item revisions and master forms, the **tcPropName** parameter value must be preceded by the appropriate prefix.

Object type	Prefix	Example
Item	None	tcPropName="object_name"
Item Revision	revision:	tcPropName="revision: item_revision_id"
Item Master Form	IMAN_master_form:	tcPropName="IMAN_master_form:serial_number"
	revision:IMAN_master_form_rev:	tcPropName="revision: IMAN_master_form_rev: project_id"

Specifying the property for the master shape text box

In a **textPropertyMaps** entry, a **textPropertyMap** tag specifies a property whose value is displayed in the text block of each shape based on the master.

```
<textPropertyMaps>
  <textPropertyMap tcPropName="prop_name" tcTypeName="obj_type"
    DiagramShapePropName="Text"/>
  ...
</textPropertyMaps>
```

Caution:

Teamcenter properties can be mapped only to the Visio **Text** property of the master shape. For all **textPropertyMap** tags, the **DiagramShapePropName** parameter value must be **Text**.

Note:

- Teamcenter properties of the **Boolean**, **Date**, **Double**, **Integer**, and **Text** types can be mapped to the master shape.
- A master shape can have any number of **textPropertyMap** tags.

tcPropName Replace *prop_name* with the name of the property to display in the diagram shape's text block.

tcTypeName Replace *obj_type* with the object type to which the property applies.

Master shape	tcTypeName (FunctionalModeling)	tcTypeName (LogicalModeling)
Function	Fnd0FunctionalBOMLine Functionality	N/A
LogicalBlock	N/A	Fnd0LogicalBlock Fnd0LogicalBOMLine
TraceLink	FND_TraceLink	FND_TraceLink
Connection	BOMLine	BOMLine
InternalPort	GDELine	GDELine
Item	BOMLine Item	BOMLine Item
ItemRevision	BOMLine FunctionalityRevision ItemRevision	BOMLine Fnd0LogicalBlockRevisi

Master shape	tcTypeName (FunctionalModeling)	tcTypeName (LogicalModeling)
		on ItemRevision
Bi-Directional Port	GDELine	GDELine
Input Port	GDELine	GDELine
Output Port	GDELine	GDELine
Container	Fnd0FunctionalBOMLine Functionality	Fnd0LogicalBlock Fnd0LogicalBOMLine

Additional mapping for creating occurrences and copying workspace objects

When you drag a block master shape from a stencil, the diagram shape represents a new occurrence (a structure element below the diagram owner). Also, you can copy a function revision or a logical block revision in My Teamcenter (a workspace object), and insert a shape for that object without adding a corresponding occurrence.

To cover each case, you map both a runtime property and a persistent property to the block master shape.

```
<textPropertyMaps>
  <textPropertyMap tcPropName="occ_prop" tcTypeName="occ_tp"
    DiagramShapePropName="Text"/>
  <textPropertyMap tcPropName="wks_prop" tcTypeName="wks_tp"
    DiagramShapePropName="Text"/>
</textPropertyMaps>
```

tcPropName	Replace <i>occ_prop</i> with the name of the occurrence property to display in the diagram shape's text block. Replace <i>wks_prop</i> with the name of the workspace object property to display in the diagram shape's text block.
tcTypeName	Replace <i>occ_tp</i> with the name of the occurrence type to which the property applies. Replace <i>wks_tp</i> with the name of the workspace object type to which the property applies.

For a compound master shape, you use reserved tags that map properties to the text blocks of the component subshapes.

- A **runtimePropMap** tag maps a property to shapes that represent occurrences.
- A **persistentPropMap** tag maps a property to shapes that represent workspace objects.

Enter each reserved tag below a **textPropertyMap** tag.

```
<textPropertyMap tcPropName="" tcTypeName="occ_tp"
  DiagramShapePropName="Text"/>
  <runtimePropMap shapeName="subshape" tcRuntimePropName="occ_prop"/>...
<textPropertyMap tcPropName="" tcTypeName="wks_tp"
  DiagramShapePropName="Text"/>
  <persistentPropMap shapeName="subshape"
tcPersistentPropName="wks_prop"/>...
```

tcPropName	Leave this value blank.
tcTypeName	Replace <i>occ_tp</i> with the name of the occurrence type to which the property applies. Replace <i>wks_tp</i> with the name of the workspace object type to which the property applies.
shapeName	Replace <i>subshape</i> with the name of the subshape to display the property in its text block. Caution: Use the name assigned in Visio to the shape as a separate component of the compound master.
tcRuntimePropName	Replace <i>occ_prop</i> with the name of the occurrence property to display in the subshape's text block.
tcPersistentPropName	Replace <i>wks_prop</i> with the name of the workspace object property to display in the subshape's text block.

Ensure that the property named in each reserved tag is of the type named in the preceding **textPropertyMap** tag.

You can enter both reserved tag types for each component subshape.

Example:

```
<textPropertyMap tcPropName="" tcTypeName="Fnd0FunctionalBOMLine"
  DiagramShapePropName="Text" >
  <runtimePropMap shapeName="Name" tcRuntimePropName="bl_rev_object_name"/>
  <runtimePropMap shapeName="Revision" tcRuntimePropName="bl_rev_item_revision_id"/>
  <runtimePropMap shapeName="Description" tcRuntimePropName="bl_rev_object_desc"/>
</textPropertyMap>
```

```
<textPropertyMap tcPropName = "" tcTypeName="Functionality"
    DiagramShapePropName="Text"/>
    <persistentPropMap shapeName="Name" tcPersistentPropName="object_name"/>
    <persistentPropMap shapeName="Revision" tcPersistentPropName="revision_id"/>
    <persistentPropMap shapeName="Description" tcPersistentPropName="object_desc"/>
</textPropertyMap>
```

- For the **Name** subshape, **bl_rev_object_name** maps the occurrence name and **object_name** maps the workspace object name.
- For the **Revision** subshape, **bl_rev_item_revision_id** maps the occurrence revision identifier, and **revision_id** maps the workspace object identifier.
- For the **Description** subshape, **bl_rev_object_desc** maps the occurrence description, and **object_desc** maps the workspace object description.

Making Teamcenter properties available in Microsoft Visio

To make additional Teamcenter properties available in Visio you must provide the following *extension* as the extension point: `com.teamcenter.rac.diagramming.tcPropertyListToPopulate`.

Example:

```
<!--
    Start tcPropertyListToPopulate extension point definition -->
<extension
    point="com.teamcenter.rac.diagramming.tcPropertyListToPopulate">
    <tcPropertyListToPopulate
        originatingClass="TCComponentBOMLine">
        <tcProperty>
            bl_rev_object_name
        </tcProperty>
    </tcPropertyListToPopulate>
    <tcPropertyListToPopulate
        originatingClass="TCComponentGDELine">
        <tcProperty>
            fnd0bl_gde_name
        </tcProperty>
        <tcProperty>
            bl_level_starting_0
        </tcProperty>
    </tcPropertyListToPopulate>
    <tcPropertyListToPopulate
        originatingClass="TCComponentTcRelation">
        <tcProperty>
            primary_object
        </tcProperty>
        <tcProperty>
            secondary_object
        </tcProperty>
    </tcPropertyListToPopulate>
    <tcPropertyListToPopulate
        originatingClass="TCComponentItem">
```

```

    <tcProperty>
      object_name
    </tcProperty>
  </tcPropertyListToPopulate>
  <tcPropertyListToPopulate
    originatingClass="TCComponentItemRevision">
    <tcProperty>
      object_name
    </tcProperty>
  </tcPropertyListToPopulate>
</extension>
<!-- End tcPropertyListToPopulate extension point definition -->

```

Assigning interface names automatically

When interface shapes are created on blocks in diagrams, the interface names can be assigned automatically. Interface names consist of a prefix and a number.

For each new interface, the number can be unique for individual blocks or for all blocks in the entire diagram. In either context, a counter sets the amount by which new numbers are incremented.

You enter the automatic-naming parameters in the **AutoNamePorts** tag, in the **ApplicationInfo** section of the mapping file.

```
<AutoNamePorts enabled="state" prefix="character" seed="start" counter="n" context="range"/>
```

Parameter	Value				
enabled	Replace <i>state</i> with one of the following values: <table> <tr> <td>true</td><td>Automatic naming is enabled.</td></tr> <tr> <td>false</td><td>Automatic naming is disabled.</td></tr> </table> The default value is false.	true	Automatic naming is enabled.	false	Automatic naming is disabled.
true	Automatic naming is enabled.				
false	Automatic naming is disabled.				
prefix	Replace <i>character</i> with an alphabetical character, for example, P. You can include a separator such as a hyphen (-), for example, P-.				
seed	Replace <i>start</i> with the first number of the automatic sequence, for example, 001 .				
counter	Replace <i>n</i> with the increment for each successive number of the automatic sequence, for example, 1 .				
context	Replace <i>range</i> with one of the following values: <table> <tr> <td>Block</td><td>Sequence numbers are unique for new interfaces that are created on the same block shape. New interfaces on other blocks use the same sequence numbers.</td></tr> </table>	Block	Sequence numbers are unique for new interfaces that are created on the same block shape. New interfaces on other blocks use the same sequence numbers.		
Block	Sequence numbers are unique for new interfaces that are created on the same block shape. New interfaces on other blocks use the same sequence numbers.				

Parameter	Value
	The default value is Block .
Diagram	Sequence numbers are unique for all new interfaces in the diagram.

For example, assume that the diagram has two blocks, and that two new interfaces are created on each block. Assume also that the **prefix** value is **P-**, the **seed** value is **001**, and the **counter** value is **1**.

- With the **Block** value, the interface names are **P-001** and **P-002** for both Block 1 and Block 2.
- With the **Diagram** value, the interface names are:

P-001 and **P-002** for Block 1

P-003 and **P-004** for Block 2

Setting interface direction for new connections

In the **ApplicationInfo** section, the **PortDirections** element defines the interface direction at a new connection's begin point and end point. To set the direction, you enter a value in the **StartPortDirection** tag and the **EndPortDirection** tag.

```
<PortDirections>
  <StartPortDirection>begin</StartPortDirection>
  <EndPortDirection>end</EndPortDirection>
</PortDirections>
```

Parameter	Value				
StartPortDirection	Replace <i>begin</i> with one of the following values: <table> <tr> <td>Input</td><td>The interface at the connection begin point can only consume a payload.</td></tr> <tr> <td>Output</td><td>The interface at the connection begin point can only generate a payload.</td></tr> </table> Output is the default value for this tag.	Input	The interface at the connection begin point can only consume a payload.	Output	The interface at the connection begin point can only generate a payload.
Input	The interface at the connection begin point can only consume a payload.				
Output	The interface at the connection begin point can only generate a payload.				
EndPortDirection	Replace <i>end</i> with one of the following values: <table> <tr> <td>Input</td><td>The interface at the connection end point can only consume a payload.</td></tr> </table>	Input	The interface at the connection end point can only consume a payload.		
Input	The interface at the connection end point can only consume a payload.				

Parameter	Value
	Input is the default value for this tag.
Output	The interface at the connection end point can only generate a payload.

Localizing Teamcenter stencils and mapping files

You can create stencils in locales other than English by using the same customizing steps. However, the boxes in the **Action** column of the **Actions** section should be entered as shown in the customizing topics for the various master shapes. Menu commands in the shortcut menus can be localized by adding the locale-specific menu names in the **Menu** column. The shape data **Object Name** box can also be localized.

Mapping files should be created in only the English locale. However, you can declare a master shape name in any locale as long as the stencil contains a master shape with the same name.

```
<DiagramShapeMaps>
  <DiagramShapeMap DiagramShapeName="locale_name">
    .....
  </DiagramShapeMaps>
```

Parameter	Value
DiagramShapeName	Replace <i>locale_name</i> with the exact name of the master shape in the stencil.

Note:

If a master shape consists of compound shapes, include the **isGroup** parameter with the value **true**. For example:

```
<DiagramShapeMap DiagramShapeName="locale_name" isGroup="true">
```

Extending Microsoft Visio events

Note:

- Microsoft Visio Professional must be installed on your computer. Visio Standard is not supported.
- Siemens Digital Industries Software assumes that you are familiar with Visio.

To customize Microsoft Visio events you must create a custom plug-in.

To create a custom plug-in that can have post-processing/pre-processing code to the corresponding Teamcenter event handler, do the following:

1. Create a new project (class library).
2. Add references to the following .dll files:
 - DiagrammingCommunication.dll
 - DiagrammingFramework.dll
 - DiagrammingFrameworkVisio.dll
 - FunctionalModelingVisio.dll
 - Thrift.dll
3. Create a class which is inherited from the appropriate handler that can be customized.

Following is the list of handlers that you can customize:

- FMVisioConnectionAddedEventHandler
 - FMVisioConnectionDeletedEventHandler
 - FMVisioCopyEventHandler
 - FMVisioDiagramSaveEventHandler
 - FMVisioShapeAddedEventHandler
 - FMVisioShapeDeleteEventHandler
4. If you want to customize the post processing for the event after the Teamcenter Visio processing is done, you must override the implementation of the PostProcessResponse method in the custom handler.
 5. If you want to customize the preprocessing for the event before the Teamcenter Visio processing is done, you must override the implementation of the PreProcessResponse method in the custom handler.
 6. Save the .dll file of the custom plug-in project in the *CustomPlugins* folder located in the installation folder of the add-in.

Sample code for customizing ShapeDeleted event

```
namespace CustomPlugin
{
    [Export(typeof(IShapeDeletedEventHandler))]
    public class CustomShapeDeleteEventHandler : FMVisioShapeDeleteEventHandler
    {
        protected override void PostProcessResponse(EventResponse response,
            DiagramModel model, DiagramView view, PropertyMapReader mapReader)
        {
            base.PostProcessResponse(response, model, view, mapReader);
            //Add your custom code here.
        }

        protected override void PreProcessResponse(EventResponse response,
            DiagramModel model, DiagramView view, PropertyMapReader mapReader)
        {
            base.PreProcessResponse(response, model, view, mapReader);
            //Add your custom code here.
        }
    }
}
```