Active Workspace Deployment
# Contents

## Part I: Planning and installation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1-1</td>
</tr>
<tr>
<td>What is Active Workspace?</td>
<td>1-1</td>
</tr>
<tr>
<td>Active Workspace components</td>
<td>1-2</td>
</tr>
<tr>
<td>Server Extensions</td>
<td>1-3</td>
</tr>
<tr>
<td>Client</td>
<td>1-3</td>
</tr>
<tr>
<td>Indexing</td>
<td>1-3</td>
</tr>
<tr>
<td>Visualization Server</td>
<td>1-4</td>
</tr>
<tr>
<td>System requirements</td>
<td>1-5</td>
</tr>
<tr>
<td>Deployment paths</td>
<td>1-7</td>
</tr>
<tr>
<td><strong>Installation</strong></td>
<td><strong>2-1</strong></td>
</tr>
<tr>
<td>Installation strategies</td>
<td>2-1</td>
</tr>
<tr>
<td>Parallel versus series installation of Active Workspace components</td>
<td>2-1</td>
</tr>
<tr>
<td>Teamcenter values needed to install Active Workspace components</td>
<td>2-2</td>
</tr>
<tr>
<td>Active Workspace values needed for parallel installation of components</td>
<td>2-3</td>
</tr>
<tr>
<td>Order of installation tasks</td>
<td>2-4</td>
</tr>
<tr>
<td>Installation overview</td>
<td>2-5</td>
</tr>
<tr>
<td>Server Extensions</td>
<td>2-6</td>
</tr>
<tr>
<td>Server Extensions overview</td>
<td>2-6</td>
</tr>
<tr>
<td>Server Extensions prerequisites</td>
<td>2-7</td>
</tr>
<tr>
<td>Install the Active Workspace patch on the Teamcenter servers</td>
<td>2-8</td>
</tr>
<tr>
<td>Install Server Extensions features</td>
<td>2-11</td>
</tr>
<tr>
<td>Install database triggers</td>
<td>2-18</td>
</tr>
<tr>
<td>Install Geolus (optional)</td>
<td>2-18</td>
</tr>
<tr>
<td>Set preference for Geolus server URL (optional)</td>
<td>2-19</td>
</tr>
<tr>
<td>Client</td>
<td>2-21</td>
</tr>
<tr>
<td>Client overview</td>
<td>2-21</td>
</tr>
<tr>
<td>Prerequisites for the Java EE Client</td>
<td>2-22</td>
</tr>
<tr>
<td>Install the Active Workspace patch on the Java EE Client build host</td>
<td>2-22</td>
</tr>
<tr>
<td>Build the Java EE Client web application</td>
<td>2-23</td>
</tr>
<tr>
<td>Deploy the Java EE Client web application</td>
<td>2-37</td>
</tr>
<tr>
<td>Log on using the Active Workspace interface</td>
<td>2-39</td>
</tr>
<tr>
<td>Indexing</td>
<td>2-40</td>
</tr>
<tr>
<td>Indexing installation overview</td>
<td>2-40</td>
</tr>
<tr>
<td>Dispatcher (optional)</td>
<td>2-42</td>
</tr>
<tr>
<td>Indexing Engine</td>
<td>2-64</td>
</tr>
<tr>
<td>Indexer</td>
<td>2-71</td>
</tr>
<tr>
<td>Visualization Server</td>
<td>2-92</td>
</tr>
<tr>
<td>Visualization Server overview</td>
<td>2-92</td>
</tr>
<tr>
<td>Visualization Data Server (optional)</td>
<td>2-93</td>
</tr>
</tbody>
</table>
Patching and upgrading Active Workspace ........................................... 3-1
Patch and upgrade deployment paths .................................................. 3-1
Patching to Active Workspace 2.4 ..................................................... 3-2
  Remove Active Workspace 2.1 features ........................................... 3-2
  Remove Active Workspace 2.1 Dispatcher translators .......................... 3-3
Patch Teamcenter .............................................................................. 3-5
Patch to Active Workspace 2.4 .......................................................... 3-6
Reinstall Active Workspace components ............................................ 3-9
Regenerate and redeploy the Active Workspace WAR file ...................... 3-10
Update Indexer settings ...................................................................... 3-12
Initiate Solr credentials ...................................................................... 3-13
Remerge Solr and Teamcenter schemas and reindex ............................. 3-19
Upgrading to Active Workspace 2.4 .................................................... 3-19
  Remove Active Workspace 2.1 features ........................................... 3-20
  Remove Active Workspace 2.1 Dispatcher translators .......................... 3-21
Upgrade to Active Workspace 2.4 and Teamcenter 10.1.4 ...................... 3-23
Reinstall Active Workspace components ............................................ 3-34
Remerge Solr and Teamcenter schemas and reindex ............................. 3-34
Regenerate and redeploy the Active Workspace WAR file ...................... 3-35
Upgrade the ACCT_TABLE database table .......................................... 3-37

Part II: Configuration and administration

Configuration ..................................................................................... 4-1
Modifying Active Workspace settings .................................................. 4-1
Configuring the interface .................................................................... 4-2
  Configure the User Assistance tile ................................................... 4-2
Enable a custom business object in Active Workspace ........................ 4-3
Configuring notification attributes ..................................................... 4-3
Configure table columns .................................................................... 4-4
  Configure the .war file generation for other locales .......................... 4-5
Configuring search ............................................................................ 4-7
  Configure the search user interface .................................................. 4-7
Configure object data indexing .......................................................... 4-10
TCFTSIndexer extensibility ............................................................... 4-24
Configuring structure indexing .......................................................... 4-44
Configuring structured content .......................................................... 4-61
  Setting security on structured content .............................................. 4-61
Configuring the properties of structured content ................................. 4-62
Cleaning up background working contexts ......................................... 4-62
Add custom objects to the Content tab and search .............................. 4-62
Configuring the visualization components ......................................... 4-63
  Specify the address for the Teamcenter SOA service ........................ 4-63
Configure viewer for Security Services-enabled environment ........................................ 4-64
Locales supported for Visualization Servers ............................................................. 4-65
Optimizing Visualization Server system performance .............................................. 4-66
Enable Visualization Server Manager logging ......................................................... 4-68
Configure automatic logon for Windows ...................................................................... 4-68
Configure the Visualization Pool Assigner for JMX metrics ...................................... 4-70
Specify viewer defaults using Teamcenter preferences .............................................. 4-71
Adjusting the display resolution for 3D models .......................................................... 4-71
Configuring Active Workspace with Teamcenter Security Services ......................... 4-72
  Configure logoff for Active Workspace ...................................................................... 4-72
  Configuring for a two-way SSL proxy server ............................................................ 4-73
  Configuring a logout landing page ............................................................................. 4-73
  Configure multiple application IDs .......................................................................... 4-74
Configuring relations .................................................................................................... 4-74
  Localize names that appear in the custom Relations Legend views ......................... 4-74
  Configure the style of nodes and edges ..................................................................... 4-75
Configuring requirements ............................................................................................ 4-76
  Prerequisites for requirements .................................................................................. 4-76
  Set preferences for requirements .............................................................................. 4-77
  Enable trace links for requirements ......................................................................... 4-78
Configuring product line engineering ......................................................................... 4-79
  Product line engineering general configuration ...................................................... 4-79
  Configuring analysis definition templates ................................................................ 4-79
Configure property formatters ..................................................................................... 4-80
  About property formatters ....................................................................................... 4-80
  Create property formatters ..................................................................................... 4-81
  Property formatter definition examples ..................................................................... 4-81
Enabling browser caching ............................................................................................. 4-85
Configuring workflows ................................................................................................. 4-86
  Workflow constants and conditions ........................................................................... 4-86
  Assigning participants using workflow handlers ...................................................... 4-87
  Automatically reassigning dynamic participants ..................................................... 4-88
  Configuring resource pool assignments .................................................................... 4-88
  Allowing access to the Inbox of other users .............................................................. 4-89
  Setting who can reassign tasks ................................................................................ 4-89
  Configuring the Inbox ............................................................................................... 4-89
Configuring change management .................................................................................. 4-92
  Automating the submission of changes to workflow ................................................ 4-92
  Setting the type of changes to be created .................................................................. 4-92
  Automating the assignment of participants ............................................................. 4-93
  Configuring how changes are derived ...................................................................... 4-93
  Defining deep copy rules for creating changes from another .................................... 4-93
  Configuring My Changes ......................................................................................... 4-94
Configuring digital signature ....................................................................................... 4-98
Configuring template files ............................................................................................ 4-100
Configuring custom content objects ........................................................................... 4-100
  When is configuration needed? ................................................................................ 4-100
  Active content technical overview .......................................................................... 4-100
  Understanding active content .................................................................................... 4-102
Using the Awb0BOMArchetypeToOccurrence type constant .......................... 4-103
Mapping type to model element .......................................................... 4-104
Marking archetypes to support structure .............................................. 4-104
Mapping properties to occurrence properties ....................................... 4-104
Implementing full text search of structures .......................................... 4-105
Configuring Document Management .................................................... 4-105
Configuring print reports .................................................................... 4-106
Defining reports with a watermark ....................................................... 4-106
Modify report definitions ..................................................................... 4-107
Customize the watermark by editing the style sheet associated with the report ........................................ 4-109
Add the watermark to an existing report by editing the style sheet associated with the report ........................................ 4-112
Create a report definition with a watermark ......................................... 4-114
Configuring batch printing .................................................................. 4-114

Administration ..................................................................................... 5-1
Monitor visualization components and processes .................................. 5-1
  Monitor visualization components and processes in Active Workspace 5-1
  Monitoring Visualization Server components using JMX ................... 5-3
Monitor browser activity with the F12 key .......................................... 5-9
Protecting against cross site request forgery ....................................... 5-10
Repin a tile to the home page ............................................................... 5-11
Utilities ............................................................................................... 5-13
  Introduction to utilities .................................................................. 5-13

Preferences .......................................................................................... 6-1
Introduction to preferences .................................................................... 6-1
Hosting preferences ............................................................................ 6-1

Business Modeler IDE constants ......................................................... 7-1
Global constants .................................................................................. 7-1
Business object constants ................................................................... 7-2
Property constants ............................................................................... 7-3

Workflow handlers ............................................................................... 8-1

Active Workspace in other Teamcenter clients .................................... 9-1
Host Active Workspace in stand-alone visualization ............................. 9-1
Host Active Workspace in Adobe applications .................................... 9-2
  Install the Active Workspace extension in Adobe Extension Manager CC 9-2
  Modify the configuration file ......................................................... 9-3

Troubleshooting .................................................................................. 10-1
General troubleshooting ...................................................................... 10-1
TcFTSIndexer troubleshooting ............................................................. 10-2
  Obtain TcFTSIndexer troubleshooting logs ...................................... 10-2
  Resolving TcFTSIndexer issues .................................................... 10-3
  Troubleshooting object data (objdata) ........................................... 10-6
Troubleshooting structure .................................................. 10-8
Visualization troubleshooting ............................................ 10-9

Figures

Environment Settings for Client Communication System panel ................. 2-81
Updating the Active Workspace client settings ..................................... 4-6
Active Workspace Client Settings panel ......................................... 4-6
Additional Client Settings panel .................................................. 4-7
Search prefilters ....................................................................... 4-8
Properties displayed in search results filtering ..................................... 4-12
Stand-alone mode ..................................................................... 4-16
Maximum connections setting ....................................................... 4-19
TcFTSIndexer synchronization ...................................................... 4-22
Active Workspace search deployment ............................................. 4-24
TcFTSIndexer components (stand-alone indexing mode) ....................... 4-25
TcFTSIndexer components (dispatcher indexing mode) ....................... 4-26
TcFTSIndexer extensibility ........................................................... 4-27
TcFTSIndexer (ObjData indexing) stand-alone mode ............................ 4-34
TcFTSIndexer ObjData flow ....................................................... 4-35
An example of CurrentMaxTotal values ........................................ 5-3
Example of CurrentMaxMin values ............................................... 5-3
Internet Explorer developer tools accessed with the F12 key .................. 5-10
Searching for a tile collection .................................................... 5-12
Re-pin a tile ........................................................................... 5-13
Environment Settings for Client Communication System panel ................ 10-5
Part I: Planning and installation
Chapter 1: Introduction

What is Active Workspace?

Active Workspace allows you to view and create data in a wide range of browsers on traditional desktop computers, laptops, tablets, and mobile devices. Your selected browser must support HTML5 and CSS3. The interface is compatible with traditional mouse-driven computers or the touch displays found on Windows 8.1 devices, iPads, and Android OS tablets. On end-user devices, Active Workspace requires minimal installation of software and requires no plug-ins, such as ActiveX or Java.

Administrators may configure the interface for a specific industry, group, role, or individual user. This ensures users see only the tasks and data most relevant to their needs and interactions. Active Workspace includes a powerful search that allows you to locate the data you need quickly and conveniently. Workflows allow you to carry out work assignments and manage changes to your product. The complexities of how the data is actually managed on the Teamcenter server remains behind the scene.
Active Workspace components

You deploy Active Workspace on top of the Teamcenter four-tier architecture. The components added provide client functionality, indexing, and visualization. You can deploy the components on a single machine or on multiple machines depending on business needs.
Server Extensions

Installing Active Workspace Server Extensions features adds binaries to TC_ROOT and adds data model changes to the database and TC_DATA. These additions enable Active Workspace functionality. Server Extensions features must be installed on each Teamcenter server that has the Teamcenter Foundation feature installed. This includes corporate servers, volume servers, and server manager hosts. The Server Extensions are installed using Teamcenter Environment Manager (TEM).

Installing Active Workspace Server Extensions is covered in Server Extensions.

Client

Client functionality is provided by a user-facing Java EE web application (awc.war) that is built using TEM and deployed on a Java EE web application server such as JBoss or WebSphere. Users connect to the client web application using a web browser.

Building, deploying, and testing the Java EE web application (awc.war) is covered in Client.

Indexing

The Indexing Server provides the basis for full-text search from the Active Workspace client.

Installing the Indexing components is covered in Indexing.

The components involved in indexing are:

- **Dispatcher** (optional)
  
  An asynchronous executor and load balancer of scheduled jobs that is used if Dispatcher-based Indexing Mode is deployed.

  Installing Dispatcher for use with Active Workspace indexing is covered in the Dispatcher (optional) topic.

- **Indexing Engine**
  
  This feature installs the Solr database, which is a full-text search engine that stores indexed Teamcenter data. It is installed by selecting the Indexing Engine feature in TEM.

  Installing and configuring the Solr database is covered in the Indexing Engine topic.

- **Indexer**
  
  This feature installs the TcFTSIndexer, which is a four-tier SOA client used to export Teamcenter data for importing into the Solr database. This service manages overall indexing processes. It is installed by selecting the Indexer feature in TEM. The TcFTSIndexer is run manually for the initial indexing and can be scheduled to run periodically for subsequent updates.

  Indexing can be deployed in two different modes:

  - **Standalone Indexing Mode**
    
    Indexing operations are run on one machine using a single Java process which connects to the Teamcenter server manager and the Solr search engine.
This mode supports indexing both object data and structure data. It is the only mode that supports object data. For custom types, this mode can be used if the type associated with TcFtsIndexer supports Standalone mode.

- **Dispatcher-Based Indexing Mode**

  Indexing operations are initiated using the Dispatcher and run across multiple machines. This provides load balancing for better performance if indexing steps are CPU or memory intensive.

  The Dispatcher-based mode supports structure data indexing. If you run the TcTFSIndexer in Dispatcher mode against object data, the indexer will revert to using Standalone mode. For custom types, this mode can be used if the type associated with TcFtsIndexer supports Dispatcher mode.

**Visualization Server**

The Visualization Server provides dynamic 3D visualization functionality to the Active Workspace client. The Visualization Server is optional. If you do not plan on using Visualization with Active Workspace, do not install its components.

The Visualization Server is comprised of:

- **Visualization Data Server** (optional)

  The Visualization Data Server improves Visualization performance by caching visualization data close to the Visualization Server Manager. It caches product structure, computes Massive Model Visualization (MMV) spatial hierarchies, and prepopulates JT files in the FCC for faster visualization streaming to the Active Workspace client. The Visualization Data Server is part of the MMV solution for Active Workspace.

  The Visualization Data Server is optional. You can use the Visualization Server Manager without this component. However, if MMV is used, the Visualization Data Server is required. Additionally, you need to index structure data for the product configurations that you want to view with the MMV flag. Refer to .

  A single Visualization Data Server can support one or more Visualization Server Managers.

- **Visualization Server Manager**

  The Visualization Server Manager streams visualization data to the Active Workspace client. It manages a pool of rendering processes, starting and stopping processes as needed.

Installing and running the Visualization Server is covered in *Visualization Server*. 
System requirements

For hardware and software support details, refer to the GTAC site (http://support.industrysoftware.automation.siemens.com/gtac.shtml). You can locate the requirements information in Hardware and Software Certifications → Operating System and 3rd party Certifications.

Active Workspace software components have the following system requirements:

- **Server Extensions**
  - The corporate server must have a 64-bit operating system and 64-bit JDK.
  - The Teamcenter four-tier web application (tc.ear) must be deployed on a Teamcenter supported J2EE web application server.

- **Client**
  The machine on which the WAR file is built must have:
  - A minimum of 16 GB of free physical RAM (unused by other applications) to avoid paging. Siemens PLM Software recommends that the machine used have a minimum of 24 GB.
  - Windows server operating system.
    - If you are running Teamcenter in a Linux environment, you can deploy the WAR file on a Linux server. But, the WAR file must be built on a Windows server machine.
  - 64-bit JDK 1.7 installed and the JAVA_HOME system environment variable must be set.

- **Indexing**
  - Solr must be installed on a machine with a 64-bit operating system.
  - Solr must be installed on a machine with 64-bit JDK or JRE.

- **Visualization Server**
  - **Visualization Data Server**
    - The machine hosting the Visualization Data Server should have a minimum of 16 GB of RAM, but may require more. The amount of RAM needed depends on the number of structures to be indexed and their size.
      A rough rule of thumb is to count the number of lines in the unconfigured structure to be indexed and allow at least 2000 bytes per line. For example, if there are 1 million lines in the unconfigured product index, then 1 million * 2000 = 2 GB of RAM.
      If you are not sure of the size of the structures, Siemens PLM Software recommends that you allow approximately 4 GB of RAM for each structure you are planning to cache in the Visualization Data Server. For example, if 4 structures are to be indexed, 16 GB of RAM is required.
    - Siemens PLM Software recommends that you install the Visualization Data Server on a machine with multiple processors.
The Visualization Data Server is a multi-threaded server program and is thus resource intensive; multiple processor will be utilized if they are available. Standard server class machine hardware is sufficient.

- Virtual machine deployment is supported.
- There are no graphics card requirements.
- You must deploy the Visualization Data Server on a high speed LAN near the Visualization Server Manager.
- Siemens PLM Software recommends that you deploy the Visualization Data Server near or on a machine hosting an FSC cache or FSC volume. If you deploy the Visualization Data Server remote (on a WAN) from the FSC volume, you should deploy an FSC cache on a LAN near or on the Visualization Data Server host machine.
- You must deploy an FMS Client Cache (FCC) component on the machine hosting the Visualization Data server.
- Structure indexing must be set up. The Visualization Data Server uses the structure indexing infrastructure of Active Workspace to keep cached product structure up to date.

**Visualization Server Manager**

- The Visualization Server Manager’s supported hardware and software configurations are:
  - Server class hardware using NVIDIA GRID K1 or K2 graphics cards as certified by NVIDIA.
  - Windows Server 2008 R2.

  **Note**

  Siemens PLM Software considers supported hardware as suitable for a production environment. Other hardware may work with Active Workspace, but Siemens PLM Software accepts problem reports only for issues that are reproducible on a supported configuration.

  For information about other NVIDIA server hardware compatible with the GRID K1 or K2 graphics cards, see [www.nvidia.com/buygrid](http://www.nvidia.com/buygrid).

- You can run the Visualization Server Manager on a virtual machine for demonstration or evaluation purposes. However, the Visualization Server Manager requires hardware graphics support. For full GPU performance and functionality on a virtual machine, assign an NVIDIA GRID K1 or K2 graphics card to the VM through a hypervisor such as Citrix XenServer.
# Deployment paths

To deploy Active Workspace, you follow either an installation process, a patching process, or an upgrade process. Use the following table to determine which process to use. The **Process** column provides links to the corresponding process.

<table>
<thead>
<tr>
<th>Initial release level</th>
<th>Target release level</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Workspace not yet installed</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4, or Active Workspace 2.4 with Teamcenter 10.1.4 Active Workspace 2.4 with Teamcenter Rapid Start 10.1.4</td>
<td>Install</td>
</tr>
<tr>
<td>Active Workspace 2.1 with Teamcenter 9.1.2.7</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.2 with Teamcenter 9.1.2.10</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.3 with Teamcenter 9.1.3.1</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.1 with Teamcenter 10.1.1.1</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.2 with Teamcenter 10.1.2.1</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.3 with Teamcenter 10.1.2.3</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.1 with Teamcenter 10.1.2.7</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Upgrade</td>
</tr>
<tr>
<td>Active Workspace 2.2 with Teamcenter 9.1.2.10</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Upgrade</td>
</tr>
<tr>
<td>Active Workspace 2.3 with Teamcenter 9.1.3.1</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Upgrade</td>
</tr>
</tbody>
</table>
Chapter 2: Installation

Installation strategies

Parallel versus series installation of Active Workspace components

There are two strategies for installing Active Workspace that you should consider. Which strategy you employ depends on the amount of planning you do.

- Install Active Workspace components in parallel.
  
  If you plan the deployment of your Active Workspace components before beginning installation, you can install each component concurrently. This is particularly useful when installing components on separate server machines.

  For example, you are installing the Server Extensions on host1, the Client on host2, the Indexing components on host3, and the Visualization components on host4. Installation on each of these hosts can begin at the same time.

  This requires that, prior to installation, you know certain values (such as hostnames, URLs) that will be used by the components. Planning is an essential first task. These values are listed in *Active Workspace values needed for parallel installation of components*.

  Additionally, there are values from your Teamcenter four-tier environment that you need to know before beginning any installation of the Active Workspace components. Those values are listed in *Teamcenter values needed to install Active Workspace components*.

- Install Active Workspace components in series.

  If you have not done sufficient planning of your Active Workspace deployment to determine the values prompted for in the table in *Active Workspace values needed for parallel installation of components*, you should install the components in series.

  You must still know the values from your Teamcenter four-tier environment listed in *Teamcenter values needed to install Active Workspace components* before installing the components in series.
Teamcenter values needed to install Active Workspace components

To install Active Workspace components, you must identify a few values from your Teamcenter four-tier environment. You cannot begin your Active Workspace installation until you know these values.

This table is reproduced in the Teamcenter 4-tier tab of the Active_Workspace_version_Install_Spreadsheet.xlsx spreadsheet. You can use this spreadsheet, which is distributed in the Tcversion_ActiveWorkspaceversion_doc.zip file on GTAC, to record your environment's installation parameters.

<table>
<thead>
<tr>
<th>Values from the Teamcenter four-tier environment</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teamcenter four-tier URL (<a href="http://host:port/tc">http://host:port/tc</a>)</td>
<td></td>
</tr>
<tr>
<td>FSC bootstrap URLs (<a href="http://host:4544">http://host:4544</a>)</td>
<td></td>
</tr>
<tr>
<td>Database system user login/password</td>
<td></td>
</tr>
<tr>
<td>TC_DATA</td>
<td></td>
</tr>
<tr>
<td>TC_ROOT (for Dispatcher-based indexing)</td>
<td></td>
</tr>
<tr>
<td>SPLM license server host and port (28000@host)</td>
<td>(for Dispatcher-based indexing)</td>
</tr>
</tbody>
</table>
Active Workspace values needed for parallel installation of components

The installation of some Active Workspace components require values that are set when installing other components. If you plan your environment before beginning its installation, you can install the separate Active Workspace components concurrently.

This table is reproduced in the Values for parallel install tab of the Active_Workspace_version_Install_Spreadsheet.xlsx spreadsheet. You can use this spreadsheet, which is distributed in the Tcversion_ActiveWorkspaceversion_doc.zip file on GTAC, to record your environment's installation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visualization Pool Assigner host and port (<a href="http://host:55566">http://host:55566</a>)</td>
<td>Defined when building the awc.war file.</td>
<td>Build the Java EE Client web application</td>
</tr>
<tr>
<td>Dispatcher Server URL (mri://host:2001)</td>
<td>Defined when installing the Dispatcher Server. (for Dispatcher-based indexing only)</td>
<td>Install Dispatcher Server</td>
</tr>
<tr>
<td>Dispatcher staging directory</td>
<td>Defined when installing the Dispatcher Server. (for Dispatcher-based indexing only)</td>
<td>Install Dispatcher Server</td>
</tr>
<tr>
<td>Indexing Engine URL (<a href="http://host:8983/solr">http://host:8983/solr</a>)</td>
<td>Install Indexing Engine (Solr)</td>
<td></td>
</tr>
<tr>
<td>Indexing Engine user name and password</td>
<td>Install Indexing Engine (Solr)</td>
<td></td>
</tr>
<tr>
<td>Visualization Data Server host and port (only needed if the Visualization Data Server is used)</td>
<td>Install the Visualization Data Server</td>
<td></td>
</tr>
</tbody>
</table>
Order of installation tasks

The *Installation overview* diagram describes the task order for both parallel and series installation. Each horizontal grouping represents a taskflow for deploying a specific component. A rectangle in a grouping represents a task to perform and is associated with a topic in this guide. An octagon represents a point in the taskflow at which you must wait for the indicated taskflows to be completed before proceeding.

- For parallel installation of components, you can begin each taskflow concurrently.
- For series installation of components, start with the taskflow at the top of the diagram and work your way down, one taskflow at a time.
## Installation overview

<table>
<thead>
<tr>
<th>Server Extensions</th>
<th>Prerequisites</th>
<th>Install Active Workspace patch on Teamcenter servers</th>
<th>Install Server Extensions features [1], [2]</th>
<th>Install database triggers</th>
<th>Install Geous (optional)</th>
<th>Set preference for Geous server URL (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Prerequisites</td>
<td>Install Active Workspace patch on Java EE Client build host</td>
<td>Build the Java EE Client web application</td>
<td>Deploy the Java EE Client web application</td>
<td>Verification: Log on using the Active Workspace interface</td>
<td></td>
</tr>
<tr>
<td>Dispatcher (optional)</td>
<td>Prerequisites</td>
<td>Install Dispatcher Server</td>
<td>Install Dispatcher Client (optional)</td>
<td>Install Active Workspace patch on Dispatcher hosts</td>
<td>Install Dispatcher Translators [2]</td>
<td>Install Axis for ReqMgmtWordToHtmlTrans Translator (optional)</td>
</tr>
<tr>
<td>Indexing Engine</td>
<td>Prerequisites</td>
<td>Install Active Workspace patch on Indexing Engine host</td>
<td>Install Indexing Engine (Solr)</td>
<td>Start SolR</td>
<td>Server Extensions tasks must be complete</td>
<td>Merge the Teamcenter and Solr schemas</td>
</tr>
<tr>
<td>Indexer</td>
<td>Prerequisites</td>
<td>Install Active Workspace patch (TcFTSIndexer) [2], [3]</td>
<td>Dispatcher (if used) and Indexing Engine tasks must be complete</td>
<td>Verification: TcFTSIndexer connectivity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Run initial index of object data</td>
<td>TcFTSIndexer scheduling</td>
<td>Configure Classification Search (optional)</td>
<td>Client tasks must be complete</td>
<td>Verification: Run search from the Active Workspace interface</td>
</tr>
<tr>
<td>Visualization Server (optional)</td>
<td>Prerequisites</td>
<td>Install Active Workspace Patch on Visualization Data Server host</td>
<td>Install the Visualization Data Server</td>
<td>Start the Visualization Data Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visualization Server Manager</td>
<td>Prerequisites</td>
<td>Install Active Workspace patch on Visualization Server Manager host</td>
<td>Install the Visualization Server Manager [4]</td>
<td>Start the Visualization Server Manager</td>
<td>Visualization Data Server (if used) and Client tasks must be complete</td>
<td>Verification: Test Visualization from the Active Workspace interface</td>
</tr>
</tbody>
</table>

[1] Requires the Search Engine URL (Solr DB URL) defined during when installing the Indexing Engine.

[2] Requires the Indexing Engine user name and password defined during when installing the Indexing Engine.


[4] Requires Assigner Host and Port defined when building the Java EE Client file.
Server Extensions

Server Extensions overview

[1] Requires the search engine URL (Solr DB URL) defined during when installing the Indexing Engine.

[2] Requires the Indexing Engine user name and password defined during when installing the Indexing Engine.

Installing Active Workspace Server Extensions features adds binaries to TC_ROOT and adds data model changes to the database and TC_DATA. These additions enable Active Workspace functionality. Server Extensions features must be installed on each Teamcenter server that has the Teamcenter Foundation feature installed. This includes corporate servers, volume servers, and server manager hosts. Server Extensions are installed using Teamcenter Environment Manager (TEM).

These tables are reproduced in the Server Extensions tab of the Active_Workspace_version_Install_Spreadsheet.xlsx spreadsheet. You can use this spreadsheet, which is distributed in the Tcversion_ActiveWorkspaceversion_doc.zip file on GTAC, to record your environment's installation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexing Engine user name and password</td>
<td>Install Indexing Engine (Solr)</td>
<td></td>
</tr>
<tr>
<td>Search engine URL (<a href="http://host:8983/solr)">http://host:8983/solr)</a></td>
<td>Install Indexing Engine (Solr)</td>
<td></td>
</tr>
<tr>
<td>Database system user name and password</td>
<td>Teamcenter database creation</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geolus server URL (only if Shape Search is used)</td>
<td>Install Geolus (optional)</td>
<td></td>
</tr>
</tbody>
</table>
Server Extensions prerequisites

Prerequisites for installing the Server Extensions:

- 64-bit operating system
- 64-bit JDK
- Supported version of Teamcenter (Teamcenter 10.1.4, Teamcenter 9.1.3.4, or Teamcenter Rapid Start 10.1.4) installed. The Teamcenter environment must include corporate server, a Java EE server manager, and Teamcenter Java EE web tier.

For information about installing and patching Teamcenter, refer to the Teamcenter documentation:

- Windows Server Installation → Additional configuration and maintenance → Installing Teamcenter patches
- UNIX and Linux Server Installation → Additional configuration and maintenance → Installing Teamcenter patches

Record the following values from your Teamcenter four-tier environment. You need these values when you install Active Workspace components.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teamcenter four-tier URL (<a href="http://host:port/tc">http://host:port/tc</a>)</td>
<td></td>
</tr>
<tr>
<td>FSC bootstrap URLs (<a href="http://host:4544">http://host:4544</a>)</td>
<td></td>
</tr>
<tr>
<td>Database system user login/password</td>
<td></td>
</tr>
<tr>
<td>TC_DATA</td>
<td></td>
</tr>
<tr>
<td>TC_ROOT (for Dispatcher-based indexing)</td>
<td></td>
</tr>
<tr>
<td>SPLM license server host and port (28000@host) (for Dispatcher-based indexing)</td>
<td></td>
</tr>
</tbody>
</table>
Install the Active Workspace patch on the Teamcenter servers

Before you install any Active Workspace features, you must add the Active Workspace patch to all Teamcenter servers that have the Teamcenter Foundation feature installed.

1. From GTAC, download the Active Workspace patch ZIP files:
   - If you are installing Active Workspace in a Teamcenter 9.1.3.4 environment, you download two files: `Tcrelease_ActiveWorkspace_rel-num_install.zip` and `Tcrelease_ActiveWorkspace_rel-num_platform.zip`.
   - If you are installing Active Workspace in a Teamcenter 10.1.4 or a Teamcenter Rapid Start 10.1.4 environment, you download one file: `Tcrelease_ActiveWorkspace_rel-num_platform.zip`.

2. If you are installing Active Workspace in a Teamcenter 10.1.4 or a Teamcenter Rapid Start 10.1.4 environment, skip this step. If you are installing Active Workspace in a Teamcenter 9.1.3.4 environment, unzip `Tcrelease_ActiveWorkspace_rel-num_install.zip` into the corporate server `TC_ROOT\install` directory.

3. For Teamcenter 9.1.3.4, Teamcenter 10.1.4, and Teamcenter Rapid Start 10.1.4 environments, unzip `Tcrelease_ActiveWorkspace_rel-num_platform.zip` to a location on the corporate server.

4. If any server managers are running on the corporate server to be patched, shut them down before proceeding with the patch.

5. Launch TEM for the corporate server.
6. In the **Maintenance** panel, select **Updates Manager** and click **Next**.

![Maintenance Panel]

Select the type of maintenance to be performed. Individual configurations can be modified by adding/removing features or by uninstalling the configuration all together.

- Configuration Manager
- Updates Manager
- Migrate Teamcenter to another JRE
7. Complete the **Apply Updates** panel.

**Apply Updates**

Select an update to apply. A backup of the files being updated will be created prior to applying the update. If there are services currently running, they must be stopped prior to starting the update process.

- **Update kit location**
  
  C:\kits\Tc1014_avv24_wntx64\wntx64

- **Backup directory**

  C:\appstc_root\install\installpatches\backup

- **Original Media Location**

  C:\kits\Tc10.1.0_wntx64

- In the **Update kit location** box, type or browse to the location to which you unzipped Tcrelease_AWCrel-num_platform.zip.

- In the **Backup directory** box, type or browse to the location in which backup files are to be created.

- In the **Original Media Location** box, type or browse to the location of the full Teamcenter release installation media, for example, Teamcenter 10.1 or Teamcenter Rapid Start 10.1. TEM does not display this box if you are installing Active Workspace in a Teamcenter 9.1.3.4 environment.

- Click **Next**.

8. The **Status Message** window informs you that TEM stops running Teamcenter services to perform the update. Once the update is completed, these services are restarted. Any users currently on the system are dropped. Click **Close**.

9. In the **Operating System User** panel, type the password of the operating system account used to update Teamcenter and click **Next**.

10. In the **Teamcenter Administrative User** panel, type the user’s password and click **Next**.

11. In the **Confirmation** panel, click **Start**.

12. During the update, a **Status Message** window opens. It informs you that you must manually patch TC_DATA after TEM completes its updates. Click **Close**.

13. When the update is complete, click **Close** to close TEM.
14. Browse to the location to which you unzipped `Tcrelease_AWCrel-num_platform.zip`. Under this location, browse to `tc`.

15. Unzip the `data.zip` file to the corporate server’s `TC_DATA` folder.

**Install Server Extensions features**

<table>
<thead>
<tr>
<th>Server Extensions</th>
<th>Prerequisites</th>
<th>Install Active Workspace patch on Teamcenter servers</th>
<th>Install Server Extensions features [1], [2]</th>
<th>Install database triggers</th>
<th>Install Geolus (optional)</th>
<th>Set preference for Geolus server URL (optional)</th>
</tr>
</thead>
</table>

[1] Requires the search engine URL (Solr DB URL) defined during when installing the Indexing Engine.

[2] Requires the Indexing Engine user name and password defined during when installing the Indexing Engine.

Server Extensions must be installed on every host that has the **Teamcenter Foundation** feature installed.

1. Launch Teamcenter Environment Manager (TEM) for the corporate server to which Server Extensions features are to be added.

2. In the **Maintenance** panel, select **Configuration Manager** and click **Next**.

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

4. In the **Old Configuration** panel, select the corporate server configuration and click **Next**.

5. In the **Feature Maintenance** panel, select **Add/Remove Features** and click **Next**.

```
Feature Maintenance

Please select the feature to configure below and click the Next button.

**Teamcenter**
- Add/Remove Features

**FMS Server Cache**
- Modify FMS Master Imports
- Update FMS Master

**Teamcenter Foundation**
- Update Security
- Update Database (Full Model - System downtime required)
- Update Database (Live Update - System downtime not required)
```
6. In the **Features** panel, expand **Base Install → Active Workspace → Server Extensions**. Select the Active Workspace server extension features to include.

The **Active Workspace** feature is required. The others are optional and are selectable only if you select **Active Workspace**.

- **Active Workspace** *(required)*
  Adds binaries to `TC_ROOT` and data model changes to the database and `TC_DATA`. It adds the **Active Workspace** (`aw2_template.xml`) and **Relationship Viewer** (`relationshipviewer_template.xml`) templates to the database.

- **Active Collaboration** *(optional)*
  Adds the **Active Collaboration** (`s2clsocial_template.xml`) template to the database.
If you are installing this feature, you should also install the **Active Collaboration Client** feature when building the Client web application.

- **Active Content Structure** (optional)

  If you plan to index structure data, you must install this feature. It provides functionality and data model extensions necessary for indexing structure data. It adds the **Active Content Structure** (*activeworkspacebom_template.xml*) template to the database.

  If you are installing this feature, you should also install the **Active Content Client** feature when building the Client web application.

- **Active Workspace Reporting** (optional)

  Provides the ability to view report templates, generate reports based on selected criteria, style sheets, or both, and view them in HTML, Excel, or raw XML formats. It adds the **Active Workspace Reporting** (*rb0reportingaw_template.xml*) template to the database.

  If you are installing this feature, you should also install the **Active Reporting Client** feature when building the Java EE Client web application.

  Additionally, to allow for asynchronous report generation, install the **AsyncService** translator in the Dispatcher Server.

- **Active Workspace Subscription** (optional)

  Allows users to manage subscriptions and notifications. It adds the **Active Workspace Subscription** (*sub0subscription_template.xml*) template to the database.

  If you are installing this feature, you should also install the **Subscription Client** feature when building the Client web application.

- **Digital Signatures Server** (optional)

  Adds digital signatures functionality to the server. It does not add a database template.

  Before selecting this feature, see the additional setup requirements for digital signatures in *Configuring digital signature*.

  This feature is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4.

  If you are installing this feature, you should also install the **Digital Signatures Client** feature when building the Java EE Client web application.

- **Shape Search** (optional)

  Provides the functionality data model extensions for the shape search functionality. It adds the **Shape Search** (*shapesearch_template.xml*) template to the database and requires Geolus to be installed and configured.

  This feature is not available with Teamcenter Rapid Start 10.1.4.

- **Active Workspace Requirements Management** (optional)

  Allows users to author a requirement structure in the **Content** tab of the Active Workspace client. Adds the **Active Workspace Requirements Management** (*arm0activeworkspacereqmgmt_template.xml*) template to the database.
This feature is not available with Teamcenter Rapid Start 10.1.4.

If you are installing this feature, you should also install:

- **Extensions→Systems Engineering and Requirements Management→Requirements Management** feature on the corporate server.

- **Requirements Management ACE Client** feature when building the Client web application.

**Active Workspace Systems Engineering** (optional)

Allows users to manage Systems Engineering diagrams from the Active Workspace client. It adds the **Active Workspace Systems Engineering** (ase0activeworkspacesyseng_template.xml) template to the database.

This feature is not available with Teamcenter Rapid Start 10.1.4.

If you are installing this feature, you should also install:

- **Extensions→Systems Engineering and Requirements Management→Systems Engineering** feature on the corporate server.

- **Requirements Management ACE Client** feature when building the Client web application.

**Active Workspace Visualization** (optional)

Enables visualization features related to Active Workspace, such as saving or viewing a bookmark. It adds the **Active Workspace Visualization** (awv0activeworkspacevis_template.xml) template to the database.

The **Active Workspace Visualization** feature is not available for installation unless the **Active Content Structure** feature is also installed.

If you plan to use the Visualization Data Server with Massive Model Visualization (MMV), you must install the **Active Workspace Visualization** feature.

**MRO→As-Built** (optional)

Provides searching and BOM extensions necessary to support MRO As-Built capabilities. This feature is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4.

In addition to the **Active Workspace** feature, this feature requires:

- **Extensions→Maintenance Repair and Overhaul→As-Built Management**

Add these templates to the database:

- **MRO Core, Active Workspace BOM Interface** (smr1mrocoreaw_template.xml)

- **As-Built** (sab1asbuiltaw_template.xml)

**MRO→As-Maintained** (optional)

Provides searching and BOM extensions necessary to support MRO As-Maintained capabilities.
This feature is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4. In addition to the Active Workspace feature, this feature requires:

- Extensions → Maintenance Repair and Overhaul → As-Maintained Management

This feature requires the following extensions:

- MRO Core, Active Workspace BOM Interface (smr1mrocoreaw_template.xml)
- As-Maintained (sam1asmaintainedaw_template.xml)

**MRO → Service Event** (optional)

Provides searching and BOM extensions necessary to support MRO Service Event Management capabilities.

This feature is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4. In addition to the Active Workspace feature, this feature requires:

- Base Install → Active Workspace → Server Extensions → MRO → As-Maintained
- Extensions → Maintenance Repair and Overhaul → As-Maintained Management
- Extensions → Maintenance Repair and Overhaul → Service Event Management

This feature adds the following templates to the database:

- Transaction Processing, Active Workspace BOM Interface (stp1transactionprocessingaw_template.xml)
- Service Processing, BOM Interface (spr1servicenprocessingaw_template.xml)
- Service Event (sem1serviceeventmgmtaw_template.xml)

**Product Line Engineering → Measurable Attributes and Targets for Active Workspace** (optional)

Configures Active Workspace to interact with Measurable Attributes and Targets, providing functionality such as retrieving the associated measurable attributes from a parent object or parent line in ACE.

This feature is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4. In addition to the Active Workspace feature, this feature requires:

- Extensions → Product Line Engineering → Measurable Attributes and Targets

This feature adds the following template to the database:

- Measurable Attributes and Targets for Active Workspace (att0attrtargetmgmtaw_template.xml)

If you are installing this feature, you should also install the Measurable Attributes and Targets Client feature in the Client web application.

**Product Line Engineering → Product Line Engineering Request** (optional)
Provides the ability to validate key performance measurements of objects by systems engineers and validation engineers.

This feature is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4.

In addition to the **Active Workspace** feature, this feature requires:

- **Extensions** → **Product Line Engineering** → **Product Line Engineering Request**

  Adds this template to the database

- **Product Line Engineering Request** (crt1validationcontractaw_template.xml)

  If you are installing this feature, you should also install the [Product line Engineering Request Client](#) feature in the Client web application.

7. The **Features** window informs you that all Teamcenter services and processes (with the exception of any FSCs) must be shut down before continuing. Shut down these services and processes and then click **OK**.

8. In the **Teamcenter Administrative User** panel, type the user’s password and click **Next**.

9. In the **Indexing Engine User** panel, type the Solr administrator’s user name and password, and click **Next**. The Solr administrator’s user name and password are defined when installing the **Indexing Engine (Solr)**.

   These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used).

   **Indexing Engine User**

   Enter the user and password for the Indexing Engine user in the fields provided below.
   The credentials entered must match for every feature that will use the Indexing Engine.

<table>
<thead>
<tr>
<th>User</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>solr_admin</td>
<td></td>
</tr>
</tbody>
</table>
10. In the **Active Workspace Server Extensions Settings** panel, type the Solr search engine URL. The format is:

   http://host:port/solr

   *host* is the machine designated to run Solr. This is the machine on which the Indexing Engine is installed.

   *port* is the port value used by Solr. The default is **8983**.

11. The **Database Template Summary** panel lists the templates to be installed. The templates listed vary depending on the features selected.

<table>
<thead>
<tr>
<th>Template Name</th>
<th>Template File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Workspace/Data Model</td>
<td>aws2_template.xml</td>
</tr>
<tr>
<td>Relationship Viewer/Data Model</td>
<td>relationshipviewer_template.xml</td>
</tr>
<tr>
<td>Active Collaboration/Data Model</td>
<td>s2c:social_template.xml</td>
</tr>
<tr>
<td>Active Content Structure/Data Model</td>
<td>activeworkspacebom_template.xml</td>
</tr>
<tr>
<td>Active Workspace Reporting/Data Model</td>
<td>rb0reportingaw_template.xml</td>
</tr>
<tr>
<td>Active Workspace Subscription/Data Model</td>
<td>sub0subscription_template.xml</td>
</tr>
<tr>
<td>Shape Search/Data Model</td>
<td>shapesearch_template.xml</td>
</tr>
<tr>
<td>Measurable Attributes and Targets/Data Model</td>
<td>att0attrtargetmgmt_template.xml</td>
</tr>
<tr>
<td>Active Workspace Requirements Management</td>
<td>arm0activeworkspaceregmgmt_template.xml</td>
</tr>
<tr>
<td>Active Workspace Systems Engineering/Data Model</td>
<td>ase0activeworkspacesyseng_template.xml</td>
</tr>
<tr>
<td>Active Workspace Visualization/Data Model</td>
<td>aw0activeworkspacevis_template.xml</td>
</tr>
<tr>
<td>Measurable Attributes and Targets for Active</td>
<td>att1attrtargetmgmtaw_template.xml</td>
</tr>
<tr>
<td>MRO Core, Active Workspace BOM Interface/...</td>
<td>smr1mrccoreaww_template.xml</td>
</tr>
<tr>
<td>As-Built/Data Model</td>
<td>sab1asbuiltaw_template.xml</td>
</tr>
<tr>
<td>As-Maintained/Data Model</td>
<td>sam1asmaintainedaw_template.xml</td>
</tr>
<tr>
<td>Transaction Processing, Active Workspace</td>
<td>stp1transactionprocessingaw_template.xml</td>
</tr>
<tr>
<td>Service Processing, Active Workspace BOM</td>
<td>spr1serviceprocessingaw_template.xml</td>
</tr>
<tr>
<td>Service Event/Data Model</td>
<td>sem1servicetgntmgmtaw_template.xml</td>
</tr>
</tbody>
</table>

12. In the **Confirmation** panel, click **Start**.

13. When the installation completes, click **Close**.
Install database triggers

The TcFTSIndexer requires that you install database triggers to detect additions, modifications, and deletions to the database when performing run-time (synchronous) indexing. The TcFTSIndexer uses the same triggers as the Site Consolidation feature. The SQL scripts that must be run are named sitecons_install_* and are located in the Teamcenter_installation_kit\tc\install\sitecons\ directory.

Note

If the Site Consolidation feature is already installed, you do not need to perform this step.

1. Log on to the Teamcenter database using the appropriate SQL tools.

   On Oracle databases, log on as the schema user. On Microsoft SQL or IBM DB2 databases, log on as the DBA of the database.

   You must have the create trigger privilege to run this script. If this privilege does not exist, log on to the database as System or sysdba and grant the privilege to the appropriate user.

   For example, to grant this privilege to the infodba user on Oracle, type:

   ```
   grant Create trigger to infodba identified by infodba;
   ```

2. Run the script that is appropriate for your database platform.

   For example, on Oracle, type:

   ```
   @E:\Tc10.1.2\wntx64\tc\install\sitecons\sitecons_install_tables_and_triggers.sql
   ```

Install Geolus (optional)

The Shape Search feature uses the Geolus shape search engine from Siemens PLM Software. If you plan on using Shape Search, you must install and configure Geolus. For details, refer to the Geolus documentation.
Set preference for Geolus server URL (optional)

The Shape Search feature uses the Geolus shape search engine from Siemens PLM Software. You must define the URL used for communication between Shape Search and Geolus. This is done by creating the GeolusServer preference.

Note

This step is required only if you installed the Shape Search feature in Install Server Extensions features.

1. Log on to the rich client and choose Edit→Options.
2. In the bottom-left of the Options dialog box, click Search.
3. In the upper-left of the Preferences By Search panel, click Create a preference definition.
4. In the right-hand panel, define the preference:

![Preference definition interface]

- **Name**: Type **GeolusServer**.
- **Category**: Set to **DecisionApps.ShapeSearch.Preferences**.
- Type a useful description in the **Description** box.
- **Value**: Type the Geolus server URL. It has the format:

  \[
  \text{protocol://gServer:}gPort/\text{gContext}
  \]

  *protocol* can be **http** or **https**.

  *gServer* is the machine name or IP address of the machine running the Geolus server. It must be accessible by all Teamcenter clients that need to connect to it.

  *gPort* is the port number that the server uses to handle HTTP or HTTPS requests.

  *gContext* is the context root of the Geolus server.

5. Click **Save** at the bottom of the right-hand panel.

6. Close the **Options** dialog box.

7. Restart the Teamcenter server.
Client

Client overview

Client functionality is provided by a user-facing Java EE web application (awc.war) that is built using TEM and deployed on a Java EE web application server such as JBoss or WebSphere. Users connect to the client web application using a web browser.

These tables are reproduced in the Client tab of the Active_Workspace_version_Install_Spreadsheet.xlsx spreadsheet. You can use this spreadsheet, which is distributed in the Tcversion_ActiveWorkspaceversion_doc.zip file on GTAC, to record your environment's installation parameters.

<table>
<thead>
<tr>
<th>Values required before beginning the Client taskflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Teamcenter four-tier URL</td>
</tr>
<tr>
<td>JDK Home</td>
</tr>
<tr>
<td>FSC bootstrap URLs (<a href="http://host:4544">http://host:4544</a>)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Values to record during the Client taskflow execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Visualization Pool Assigner host and port</td>
</tr>
<tr>
<td>Active Workspace Client URL</td>
</tr>
</tbody>
</table>
Prerequisites for the Java EE Client

The machine on which you build the Active Workspace client web application must meet the following requirements. These requirements apply to the machine on which it is built, not to the machine on which it is deployed.

- A minimum of 16 GB of free memory to successfully generate the awc.war file. Siemens PLM Software recommends that the machine used have a minimum of 24 GB.
- Windows Server operating system.
  If you are running Teamcenter in a Linux environment, you can deploy the Java EE Active Workspace client web application on a Linux server, however, the awc.war file must be built on a Windows Server machine.
- JDK 64-bit is installed and the JAVA_HOME system environment variable must be set.

Install the Active Workspace patch on the Java EE Client build host

Because of the special requirements for the machine used to build the Active Workspace client web application, this machine may not have a Teamcenter corporate server installed. In this case, you need to do the following:

1. Install the base Teamcenter release with no features (Teamcenter 10.1, Teamcenter 9.1, or Teamcenter Rapid Start 10.1):
   a. Launch TEM from the installation media for the base Teamcenter version.
   b. In the Installer Language dialog box, select a language and click Next.
   c. In the Welcome to Teamcenter panel, click Install.
   d. In the Configuration panel, type an ID and Description, and then click Next.
   e. In the Solutions panel, make no selections and click Next.
f. In the Features panel, enter the installation root directory for the Active Workspace client features in the Installation Directory box. Do not select features. Click Next.

g. In the Confirmation panel, click Start.

h. When the installation is complete, click Close.

2. Patch your environment to the supported Teamcenter patch level (Teamcenter 10.1.4, Teamcenter 9.1.3.4, or or Teamcenter Rapid Start 10.1.4). Refer to the general patch instructions in the Teamcenter documentation as well as the readme file for the patch.

In the Teamcenter documentation:

• Windows Server Installation→Additional configuration and maintenance→Installing Teamcenter patches

• UNIX and Linux Server Installation→Additional configuration and maintenance→Installing Teamcenter patches

3. Add the Active Workspace patch to the installation; follow the steps in Install the Active Workspace patch on the Teamcenter servers.

Build the Java EE Client web application

1. Launch TEM for the installation to which the Active Workspace Client features are to be installed.

2. In the Maintenance panel, select Configuration Manager and click Next.

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

4. In the Old configuration panel, select the configuration to which the Active Workspace Client features are to be installed and then click Next.

5. In the Feature Maintenance panel, select Add/Remove Features and then click Next.
6. In the **Features** panel, expand **Base Install → Active Workspace → Client** and select the features to include in the Active Workspace client web application.

The **Active Workspace Client (Java EE)** feature is required. The others are optional and are selectable only if you select **Active Workspace Client (Java EE)**.

- **Active Workspace Client (Java EE)** (required)
  
  Selecting this feature builds the **awc.war** file.

- **Visualization Server Pool Assigner** (optional)
  
  Select this feature only if you are going to use Visualization in the Active Workspace Client.

- **Active Collaboration Client** (optional)
  
  Adds workflow functionality to the client interface.
If you are installing this feature in the Client, you should install the **Active Collaboration** Server Extensions feature in the corporate server.

- **Active Content Client** (optional)
  
  Adds structure search functionality to the client interface.

  If you are installing this feature in the Client, you should install the **Active Content Structure** Server Extensions feature in the corporate server.

- **Active Reporting Client** (optional)
  
  Adds the ability to view report templates, generate reports based on selected criteria, style sheets, or both, and view them in HTML, Excel, or raw XML formats in the client.

  If you are installing this feature in the Client, you should install the **Active Workspace Reporting** Server Extensions feature in the corporate server.

  Additionally, to allow for asynchronous report generation, install the **AsyncService** translator in the Dispatcher Server.

- **Change Management Client** (optional)
  
  Adds ability to work with Change Management objects in the client.

  This is not available with Teamcenter Rapid Start 10.1.4.

  If you are installing this feature in the Client, you should install the **Extensions→Enterprise Knowledge Foundation→Change Management** feature in the corporate server.

- **Digital Signatures Client** (optional)
  
  Adds digital signatures functionality to the client interface.

  Before selecting this feature, refer to the additional setup requirements for Digital Signatures in *Configuring digital signature*.

  This is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4.

  If you are installing this feature in the Client, you should install the **Digital Signatures Server** Server Extensions feature in the corporate server.

- **Service Manager Client** (optional)
  
  Allows users to view disposition and utilization history of physical parts in the Active Workspace client.

  This is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4.

- **Solid Edge Integration** (optional)
  
  Adds the Solid Edge integration to the client.

- **Subscription Client** (optional)
  
  Allows users to manage subscriptions and notifications in the client.

  If you are installing this feature in the Client, you should install the **Active Workspace Subscription** Server Extensions feature in the corporate server.
• **Architecture Modeler Client** (optional)

  In addition to the **Active Workspace Client (Java EE)** feature, requires the **Active Content Client** feature.

  This is not available with Teamcenter Rapid Start 10.1.4.

• **Requirements Management ACE Client** (optional)

  In addition to the **Active Workspace Client (Java EE)** feature, requires the **Active Content Client** feature.

  This is not available with Teamcenter Rapid Start 10.1.4.

  If you are installing this feature in the Client, you should install the **Active Workspace Requirements Management** and **Active Workspace Systems Engineering** Server Extensions features in the corporate server.

• **Product Line Engineering—Measurable Attributes and Targets for Active Workspace** (optional)

  Provides functionality such as retrieving the associated measurable attributes from a parent object or parent line in ACE.

  This is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4.

  If you are installing this feature in the Client, you should install the **Measurable Attributes and Targets for Active Workspace** Server Extensions feature in the corporate server.

• **Product Line Engineering—Product Line Engineering Request** (optional)

  Provides the ability to validate key performance measurements of objects by systems engineers and validation engineers.

  This is not available with Teamcenter 9 or Teamcenter Rapid Start 10.1.4.

  If you are installing this feature in the Client, you should install the **Product Line Engineering Request** Server Extensions feature in the corporate server.
7. In the **Prerequisite Checks** panel, you run checks to ensure that the prerequisites needed to build the Active Workspace client web application are met. Enter a location in which to create the log files for the prerequisite checks and click **Run**.

![Prerequisite Checks panel](image)

A number of tests will now be performed on the current environment to check for prerequisites.

Enter a working directory where logs can be created

<table>
<thead>
<tr>
<th>Test</th>
<th>Results</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free RAM Availability C...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JDK Version Compatibility</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8. Inspect the test results. If any checks result in a status of **Fail**, you must correct the issues before proceeding. The **Next** button is unavailable until all checks have a status of **Pass** or **Warning**. Click **View** for an analysis of a failure or warning. Click **Review** for a summary of all checks.

![Prerequisite Checks](image)

The following prerequisites are checked:

- **Free RAM Availability** checks that the machine has the recommended minimum of 16 GBs of free physical RAM. If this recommendation is not met, the result is a warning and you are not prevented from proceeding.

- **JDK Version** checks that the `JAVA_HOME` system environment variable is set and that a 64-bit JDK 1.7 or later is installed. If both of these conditions are not met, the result is a failure. You cannot proceed until this is corrected.
9. Once all checks result in either **Pass** or **Warning**, the **Next** button is available. Click **Next**.
10. If the prerequisite checks resulted in any warnings, the **Confirm Warnings** window displays the warnings. If you wish to continue, click the check box next to the warning and then click **OK**.
11. Complete the **Active Workspace Client Settings (J2EE)** panel. In the **Teamcenter 4-tier URL** box, type the URL for the deployed four-tier Teamcenter web application. The format of this URL is:

```
protocol://host:port/tc-web-app-name
```

*protocol* can be **http** or **https**.

*host* is the machine running the web application server on which the Teamcenter web application is deployed.

*port* is the port used by the web application server.

*tc-web-app-name* is the name of the Teamcenter web application. The default is **tc**.
12. The **JDK Home** box is the installation location of a supported version of 64-bit JDK 1.7. This box is prepopulated with the value of the **JAVA_HOME** system environment variable. You can change this to a different JDK as needed.

For supported versions of third-party software, refer to the certification matrix on GTAC:

http://support.industrysoftware.automation.siemens.com/gtac.shtml
13. The Active Workspace client uses FMS to download and upload files. You define the FSC servers that used by selecting either **Use as Bootstrap URLs** or **Use Assigned FSC URLs**.

**Active Workspace Client Settings (J2EE)**
Specify Active Workspace client installation settings below.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teamcenter 4-tier URL</td>
<td>https://&lt;HOSTNAME&gt;:7001/tc</td>
</tr>
<tr>
<td>JDK Home</td>
<td>C:\Program Files\Java\jdk1.7.0_17</td>
</tr>
<tr>
<td><strong>Use as Bootstrap URLs</strong></td>
<td></td>
</tr>
<tr>
<td>Bootstrap Client IP</td>
<td></td>
</tr>
<tr>
<td><strong>Use Assigned FSC URLs</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Use as Bootstrap URLs**
FSC servers are automatically assigned by the FMS system. This is the most common scenario.

In the **Bootstrap URLs** box, enter a comma-separated list of one or more FMS bootstrap URLs. Optionally, provide the **Bootstrap Client IP** value to be used for the assignment.

The list of bootstrap URLs should contain well-known FSCs that can be consulted for the actual FSC assignment. The assignment is delegated to the FMS network, which uses mapping logic and configuration data to determine the FSCs that should be used by the requestor (refer to the clientmap configuration in the FMSMaster configuration file). The client mapping logic uses an IP address to determine the assignment. By default, the IP address from the HTTP connection of the requestor is used unless a Bootstrap Client IP value is provided. The **Bootstrap Client IP** value should only be used when there are proxies that may hide the actual requestor’s IP address from the FSC servers, or if you want the assignment to be performed based on a particular IP and not that of the requestor. (The client/requestor is the host on which the awc.war file is deployed.)

**Use Assigned FSC URLs**
FSC servers are explicitly declared.

In the **Assigned FSC URLs** box, enter a comma-separated list of one or more FSC URLs. The URL values entered are directly used for file operations. This allows you to explicitly declare the FSC servers that should be used. Select this only if you want explicit control of the FSCs used.
14. Select **Enable TcSS Support** to enable Security Services. Enter the Security Services application ID and logon URL.

For details about configuring Security Services, see *Security Services Installation/Customization* in the Teamcenter collection.

Note:

When configuring Active Workspace for Security Services, be sure to only install the language packs for the Security Services that Active Workspace supports.

Active Workspace always runs in applet free mode even when configured with TcSS Login Service that is deployed in applet mode. Active Workspace provides single sign on capability when used along with rich clients with an applet-based deployment.

If Active Workspace is deployed on a different URL, you must configure Security Services with multiple application IDs. For more information, see the readme file for your Security Services release.

15. At the bottom of the **Active Workspace Client Settings (J2EE)** panel, click **Advanced** to open the **Additional Client Settings** dialog box.
16. In the **Additional Client Settings** dialog box, you can select additional locales to include in the *awc.war* file configuration.

![Additional Client Settings dialog box]

17. Click **OK** to close the **Additional Client Settings** dialog box.
18. Click **Next** in the **Active Workspace Client Settings (J2EE)** panel.

19. If you selected to install the **Visualization Server Pool Assigner** feature, the **Visualization Pool Assigner Settings** panel appears. These settings are used only if you are going to use Visualization in the Active Workspace client.

![Visualization Server Pool Assigner Settings](image)

- **Local Assigner Settings** section defines the host and port used by the Visualization Server Pool Assigner.
  - **Host** is the machine on which this Visualization Server Pool Assigner runs. This is the machine on which the Active Workspace client web application containing this Visualization Server Pool Assigner is deployed. The machine building the Active Workspace client web application may or may not be the machine on which it is deployed. This can be the machine name or IP address.
  - **Port** is the port value used by this Visualization Server Pool Assigner.

- **Add pool assigner** should be selected only if more than one Visualization Server Pool Assigner is to be used. If just one Visualization Server Pool Assigner is used, do not select this option.

- **Peer Assigners** table is a list of other Visualization Server Pool Assigners that this Visualization Server Pool Assigner knows about.
  - **Assigner Host** is the host on which a peer Visualization Server Pool Assigners is deployed. This can be the machine name or IP address.
  - **Assigner Port** is the port value used by the peer Visualization Server Pool Assigner.

20. In the **Confirmation** panel, click **Start**.
21. When the installation is complete, click **Close**.

The **awc.war** file is created in the `TC_ROOT\aws2\stage\out\` directory on the machine used to build the **awc.war** file.

**Deploy the Java EE Client web application**

In this step, you deploy the **awc.war** file created in *Build the Java EE Client web application*. The **awc.war** file hosts static client resources and provides web proxies to connect to Teamcenter SOA and file services.

You deploy the **awc.war** file on a supported Java EE web application server such as JBoss or WebSphere. Deployment on Microsoft Internet Information Services (IIS) is not supported for the **awc.war**. For supported versions of third-party software, refer to the certification matrix on GTAC:

http://support.industrysoftware.automation.siemens.com/gtac.shtml

The process of deploying the **awc.war** file varies by web application server vendor. Generally, the **awc.war** file is copied directly to an auto deploy directory or is copied and deployed manually into a web application server using either a web-based console or command line utilities.

The **awc.war** file is created in the `TC_ROOT\aws2\stage\out\` directory on the machine used to build the **awc.war** file.

**JBoss configuration**

If you are using a JBoss server, you must edit its configuration file to allow Active Workspace to be referenced outside of local host environments.

1. Open **standalone.xml** (the name of your deployment’s configuration file) in an editor.

2. Search for the following line:

   `<inet-address value="$\{jboss.bind.address:127.0.0.1\}"/>

   This line is located in the `<interface name="public">` section of the file.

3. Replace the existing line with the following:

   `<any-address/>

4. Save and close the configuration file.

**WebSphere configuration**

- If you are using a WebSphere server, you must modify the web container to work around a known issue with WebSphere’s handling of internal HTTP routing (resulting in an **SRVE0190E** error). Refer to the following WebSphere documentation for details on this modification:

- Following is an example of the additional custom properties for WebSphere 8.5:

  - com.ibm.ws.webcontainer.mapFiltersToAsterisk=true
  - com.ibm.ws.webcontainer.removetrailingservletpathslash=true
  - com.ibm.ws.webcontainer.invokeFiltersCompatibility=true
  - com.ibm.ws.webcontainer.invokefilterscompatibility=true

- If during deployment the administration console either freezes or returns an error that says it is unable to deploy the WAR file, try changing the Java heap size settings. For example, set the initial heap size to 4096 MB and the maximum heap size to 6144 MB. Refer to the following WebSphere documentation for instructions on setting the Java heap size:

Log on using the Active Workspace interface

From a client machine in your environment, log on using the Active Workspace client. This tests the environment and the Active Workspace client web application.

1. Log on to a client machine in your environment.

2. Open a supported web browser.
   
   For supported versions of third-party software, refer to the certification matrix on GTAC:
   

3. Go to the URL:
   

   - *host* is the machine running the web application server on which the Active Workspace client web application is deployed.
   - *port* is the port used by the web application server.
   - *web-app-name* is the WAR file name, awc.

4. Type a user name and password and click **Sign in**. Verify that you can log on without errors.
Indexing

Indexing installation overview

[2] Requires the Indexing Engine user name and password defined when installing the Indexing Engine.

[3] Requires Dispatcher Server URL and staging directory if Dispatcher-based indexing is used.

Active Workspace indexing provides the basis for full-text search from the Active Workspace client. The components involved in indexing are:

- **Dispatcher** (optional)
  
  An asynchronous executor and load balancer of scheduled jobs. It is used if Dispatcher-based Indexing Mode is deployed.

- **Indexing Engine**
  
  This feature installs the Solr database, which is a full-text search engine that stores indexed Teamcenter data. It is installed by selecting the **Indexing Engine** feature in TEM.

- **Indexer**
  
  This feature installs the TcFTSIndexer which is a four-tier SOA client used to export Teamcenter data for importing into the Solr database. This service manages overall indexing processes. It is installed by selecting the **Indexer** feature in TEM. The TcFTSIndexer is run manually for the initial indexing and can be scheduled to run periodically for subsequent updates.

  Indexing can be deployed in two different modes:

  - **Standalone Indexing Mode**
Indexing operations are run on one machine using a single Java process which connects to the Teamcenter server manager and the Solr search engine.

This mode supports indexing both object data and structure data. It is the only mode that supports object data. For custom types, this mode can be used if the type associated with TcFtsIndexer supports Standalone mode.

- **Dispatcher-Based Indexing Mode**

  Indexing operations are initiated using the Dispatcher and run across multiple machines. This provides load balancing for better performance if indexing steps are CPU or memory intensive.

  The Dispatcher-based mode supports structure data indexing. If you run the TcTFSIndexer in Dispatcher mode against object data, the indexer will revert to using Standalone mode. For custom types, this mode can be used if the type associated with TcFtsIndexer supports Dispatcher mode.
Dispatcher (optional)

Dispatcher overview

![Diagram showing the Dispatcher (optional) process flow]

[2] Requires the Indexing Engine user name and password defined when installing the Indexing Engine.

Dispatcher is an asynchronous executor and load balancer of scheduled jobs. It is used only if Dispatcher-based Indexing Mode is deployed.

These tables are reproduced in the Dispatcher (optional) tab of the Active_Workspace_version_Install_Spreadsheet.xlsx spreadsheet. You can use this spreadsheet, which is distributed in the Tcversion_ActiveWorkspaceversion_doc.zip file on GTAC, to record your environment's installation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexing Engine user name and password</td>
<td>Install Indexing Engine (Solr)</td>
<td></td>
</tr>
<tr>
<td>TC_ROOT</td>
<td>Teamcenter installation</td>
<td></td>
</tr>
<tr>
<td>TC_DATA</td>
<td>Teamcenter installation</td>
<td></td>
</tr>
<tr>
<td>SPLM license server port and host</td>
<td>SPLM license server installation</td>
<td></td>
</tr>
<tr>
<td>Teamcenter four-tier URL</td>
<td>Teamcenter web tier app deployment</td>
<td></td>
</tr>
<tr>
<td>FSC bootstrap URLs (<a href="http://host:4544">http://host:4544</a>)</td>
<td>Teamcenter installation</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispatcher root directory</td>
<td>Install Dispatcher Server</td>
</tr>
<tr>
<td>Dispatcher staging directory</td>
<td>Install Dispatcher Server</td>
</tr>
<tr>
<td>Dispatcher URL</td>
<td>Install Dispatcher Server</td>
</tr>
</tbody>
</table>
Prerequisites for installing the Dispatcher:

- Windows Server operating system (if the Requirements Management PDF translator is used)
- .NET 4.5 (if the Requirements Management PDF translator is used)

Structure indexing can be run in either Dispatcher-Based Mode or Standalone Mode. Refer to the indexing mode options in **Indexing** to choose an indexing mode. If you are going to use the Dispatcher Mode to run the TcFTSIndexer, you must install the **Dispatcher Server** feature.

For more detailed information on the Dispatcher see **Dispatcher Server Installation** in the Teamcenter documentation.

1. Launch Teamcenter Environment Manager (TEM).
2. In the **Installer Language** window, select a language and click **OK**.
3. In the **Welcome to Teamcenter** panel, click **Install**.
4. In the **Configuration** panel, type an ID and a description and click **Next**.
5. In the **Solution** panel, select **Dispatcher (Dispatcher Server)** and click **Next**.

![Dispatcher (Dispatcher Server)](image)

**Dispatcher Solution (Dispatcher Server)** functions as an independent compute server that translates files from one format to other formats. It consists of Scheduler, Modules, and an optional Admin client. For better load balancing, each Module resides on a separate machine and is connected to Scheduler. Scheduler sends translation tasks to Modules and they invoke one or more translators to perform translations. Dispatcher Solution allows you to perform translations either using a Web server (Web mode) or through remote method invocation (RMI) mode. Translations using RMI mode are generally faster. Standard installation supports RMI mode.
6. In the **Features** panel, expand **Extensions→Enterprise Knowledge Foundation** and note that **Dispatcher Server** is selected. Enter an installation root in the **Installation Directory** box and click **Next**.

![Features panel](image)

7. In the **File Client Cache (FCC)** panel, define the **FMS_HOME** environment variable, and click **Next**.

![File Client Cache (FCC) panel](image)
8. In the **FCC Parents** panel, enter the URLs for the FSC servers that the Dispatcher uses as FCC parents and click **Next**.

![FCC Parents Panel](image)  
FCC Parents  
Please enter this FCC’s parent FSC’s information. The FSC’s will be used based on the defined priorities.  
FSC assignment mode: **clientmap**  
Protocol | Host | Port | Path | Priority | Transport  
--- | --- | --- | --- | --- | ---  
http | 4544 |  |  | | \lan

9. In the **Operating System User** panel, type the password for the operating system account used to install the Dispatcher and click **Next**.

![Operating System User Panel](image)  
Operating System User  
The following user will be granted ‘Log on as a service’ right and all services will run as this user  
**User**: SRVRHOST\infodba  
**Password**:  
**Confirm**:  

10. In the **Teamcenter Installation Location** panel, provide the **TC_ROOT** and **TC_DATA** directories of the corporate server and click **Next**.

![Teamcenter Installation Location Panel](image)  
Teamcenter Installation Location  
Enter the location of the Teamcenter TC_ROOT and TC_DATA directories.  
**TC_ROOT Directory**:  
**TC_DATA Directory**:  

---

Chapter 2: Installation

---

2-46  
Active Workspace Deployment  
AW001 2.4
11. In the **Flex License Client** panel, provide the port and host of the Teamcenter license server if it is not already populated and click **Next**.
12. Complete the **Dispatcher Components** panel.

- In the **Dispatcher Root Directory** box, enter the location in which to install the selected Dispatcher components.

- Select **Install Scheduler** (default) and define the **Scheduler Port** value. Record the scheduler port value, it is needed later in Active Workspace installation steps.

- Select **Install Module**, provide the **Staging Directory** location, and the **Module Port** value. Record the staging directory location, it is needed later in Active Workspace installation steps.

- **Maximum Tasks** specifies the overall maximum number of tasks to run in a given Dispatcher module. The default value is **3**.

- **Admin Client** is not required.
13. In the **Dispatcher Settings** panel, define settings for logging, installation of Translation Service documentation, and the method used to start the scheduler and module (Windows service or command line).

The defaults work for Active Workspace, though you may find it more convenient to start the scheduler and module from the command line.

![Dispatcher Settings panel](image)

14. In the **Select Translators** panel, do not select translators at this time. You install the translators needed later. Click **Next**.

15. In the **Confirmation** panel, click **Start**.

16. When the installation is complete, click **Close**.

17. Patch to the supported Teamcenter version, Teamcenter 10.1.4, Teamcenter 9.1.3.4, or Teamcenter Rapid Start 10.1.4.

For information about installing and patching Teamcenter, refer to the Teamcenter documentation:

- *Windows Server Installation*→*Additional configuration and maintenance*→*Installing Teamcenter patches*

- *UNIX and Linux Server Installation*→*Additional configuration and maintenance*→*Installing Teamcenter patches*
**Install Dispatcher Client (optional)**

If you are going to use the `ReqMgmtWordToHtmlTrans` translator, you must install the **Dispatcher Client** feature and create a Dispatcher request rule. (This feature is not required for Dispatcher-based indexing.)

This topic provides steps to meet this requirement. For more detailed information about the Dispatcher, see *Dispatcher Server Installation→Installing and configuring components→Installing and configuring the dispatcher client* in the Teamcenter documentation.

The **Dispatcher Client** feature requires the **Teamcenter Foundation** feature. Therefore, the following steps assume that the **Dispatcher Client** feature is being added to an existing corporate server.

1. Launch the TEM from the corporate server to which the **Dispatcher Client** feature is to be added.

2. In the **Maintenance** panel, select **Configuration Manager** and click **Next**.

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

4. In the **Old Configuration** panel, select the configuration to which the Dispatcher Client will be added and click **Next**.

5. In the **Feature Maintenance** panel, select **Add/Remove Features** and click **Next**.
6. In the **Features** panel, expand **Extensions→Enterprise Knowledge Foundation**, select **Dispatcher Client**, and then click **Next**.

![Features Panel]

- Enterprise Knowledge Foundation
  - Remote Workflow
  - Teamcenter Client for Microsoft Office
  - Dispatcher Client for Rich Client
  - Render Document for Rich Client
  - Change Management
  - Contract Data Management
  - Dispatcher Server
  - Finish Management
  - Material Management
  - Stock Material
  - Work Package Management
  - Change and Schedule Management Interface
  - **Dispatcher Client** (Selected)

Installs an integration of the Dispatcher Server to enable rich client users to translate Teamcenter data files to desired visualization formats.

7. The **Features** window informs you that all Teamcenter service and processes, with the exception of any FSC services, must be shut down before continuing. Once these are stopped, click **OK** in the **Features** window.

8. In the **Teamcenter Administrative User** panel, type the user's password and click **Next**.

![Teamcenter Administrative User Panel]

**Teamcenter Administrative User**
Enter the administrative password for the Teamcenter application.

User: infodba
Password: [Enter]
9. A **Status Message** window informs you that any Dispatcher Windows service currently installed on this machine will be removed. Click **Close**.

10. In the **Select Translators** panel, do not select translators at this time. You install the translators needed later. Click **Next**.

11. In the **Teamcenter Administrative User** panel, type the password for the user and click **Next**.

12. Complete the **Dispatcher** panel.

   ![Dispatcher Panel](image)

   - Select **RMI** (default).
   - **Staging Directory** is the location you defined earlier in the **Dispatcher Components** panel.
   - Type and confirm a password for the **dcproxy** user. This is a Teamcenter user created to run the Dispatcher services.
13. In the next **Dispatcher** panel, specify logging and advanced settings. The defaults are sufficient for Active Workspace.

![Dispatcher Panel]

14. In the **Database Template Summary** panel, the **Translation Service Database Module/Data Model** template is listed. This template is added to the database as part of the **Dispatcher Client** feature. Click **Next**.

15. In the **Confirmation** panel, click **Start**.

16. When the installation is complete, click **Close**.

17. Log on to the rich client and open the **Access Manager Application**.
18. Add a rule under Has Class( POM_application_object ) —> Working with the following settings:
   - Set the Condition box to Has Class.
   - Set the Value box to DispatcherRequest.
   - In the ACL Name box, type DispatcherRequest.
   - Type = User
   - Accessor = dcproxy
   - Write = enabled (blue check mark)
   - Delete = enabled (blue check mark)

19. Patch to the supported Teamcenter version, Teamcenter 10.1.4, Teamcenter 9.1.3.4, or Teamcenter Rapid Start 10.1.4.

   For information about installing and patching Teamcenter, refer to the Teamcenter documentation:
   - Windows Server Installation — Additional configuration and maintenance — Installing Teamcenter patches
   - UNIX and Linux Server Installation — Additional configuration and maintenance — Installing Teamcenter patches

**Install the Active Workspace patch on the Dispatcher hosts**

You must install the Active Workspace patch on all Dispatcher hosts before you can install any Active Workspace translators.

1. Patch your environment to the supported Teamcenter patch level (Teamcenter 10.1.4, Teamcenter 9.1.3.4, or Teamcenter Rapid Start 10.1.4). Refer to the general patch instructions in the Teamcenter documentation as well as the readme file for the patch.

   In the Teamcenter documentation:
   - Windows Server Installation — Additional configuration and maintenance — Installing Teamcenter patches
   - UNIX and Linux Server Installation — Additional configuration and maintenance — Installing Teamcenter patches

2. Add the Active Workspace patch to the installation. To do this, follow the steps in *Install the Active Workspace patch on the Teamcenter servers*. 

   ![Diagram of Active Workspace Deployment]
Install Dispatcher translators

This topic provides steps for installing the following translators:

- **TcFtsIndexerTrans**
  
The installation of this translator is optional and is only required if you want to use the Dispatcher to run the indexing functionality. This provides the capability to use multiple machines as resources for indexing data.

- **ReqMgmtWordToHtmlTrans**
  
The installation of this translator is optional. It converts requirements content that has been edited and saved in Microsoft Word from Teamcenter (stored as a full-text dataset), so that it can be viewed in the rich text editor in Active Workspace.

  After installation, there are additional configuration steps.

- **AsyncService**
  
The installation of this translator is optional. This translator provides asynchronous reporting and printing.

  Additional configuration for asynchronous services is provided in the Teamcenter help collection in *Dispatcher Server Translator's Reference*→*Implementing Teamcenter-specific translators*→*Configuring asynchronous services*.

  For email notifications, you must set preferences listed at the end of this procedure.

The Teamcenter configuration must be patched to the supported Teamcenter version and Active Workspace before these translators can be added.

1. Launch TEM for the Teamcenter configuration that contains the Dispatcher Server.

2. In the Maintenance panel, select Configuration Manager and click Next.

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

4. In the Old configuration panel, select the configuration containing the Dispatcher Server and then click Next.
5. In the **Feature Maintenance** panel, select **Modify Dispatcher Settings** under **Dispatcher Server** and then click **Next**.

6. In the **Teamcenter Installation Location** panel, the TC_ROOT and TC_DATA directories are populated. Make no changes and click **Next**.

7. In the **Flex License Client** panel, the license server information is populated. Make no changes and click **Next**.

8. In the **Teamcenter Administrative User** panel, type the user’s password and click **Next**.

9. In the **Dispatcher Components** panel, the values will be populated with those defined at installation. Make no changes and click **Next**.

10. In the **Dispatcher Settings** panel, the values are populated with those defined at installation. Make no changes and click **Next**.

11. In the **Select Translators** panel, select translators:
   
   • To use Dispatcher-based indexing, select **TcFtsIndexerTrans** under **Active Workspace Indexer Translator**.
   
   • Optionally, if you are going to use Requirements Management, select the **ReqMgmtWordToHtmlTrans** translators. It converts requirements content that has been edited and saved in Microsoft Word from Teamcenter (stored as a full-text dataset), so that it can be viewed in the rich text editor in Active Workspace.

   After installation, there are additional configuration steps.

   • Optionally, to provide asynchronous reporting and printing, select the **AsyncService** translator.
Additional configuration for asynchronous services is provided in the Teamcenter help collection in *Dispatcher Server Translator’s reference—Implementing Teamcenter—Configuring asynchronous services.*

For email notifications, you must set preferences listed at the end of this procedure.
12. In the **Translator Settings** panel, the **User** and **Password** boxes appear only if you selected the **ReqMgmtWordToHtmlTrans** translator. This user must be in the **dba** group.

**Translator Settings**

Configure the selected Translators to be enabled in Module

- **Document Management (DocMgt) Translators**
  - Enter the user and password for ReqMgmtWordToHtmlTrans to login to Teamcenter in the
  - **User**
  - (NOTE: This user must have DBA privilege)
  - **Password**

- **TC FTS Indexer Translators**
  - **Teamcenter 4-tier URL**: `http://<HOSTNAME>:7001/tc`
  - **Teamcenter Retry Count**: 5
13. In the **Translator Settings** panel, the **Teamcenter 4-tier URL** and **Teamcenter Retry Count** boxes appear only if you selected the **TcFtsIndexerTrans** translator.

![Translator Settings Panel]

- **Teamcenter 4-tier URL**
  
  The URL for the Teamcenter web tier application. The format is:

  \[
  \text{http://host:port/web-app-name}
  \]

  *host* is the machine running the web application server on which the Teamcenter web application is deployed.

  *port* is the port used by the web application server.

  *web-app-name* is the name of the Teamcenter web application. The default is **tc**.

- **Teamcenter Retry Count**
  
  Specifies the number of tries to connect to the Teamcenter server.
14. In the **Active Workspace Translator Selection** panel, select the type of translator to use for indexing.

Select the same indexer types here and for the **Indexer** feature.

- **Active Workspace Object Data Translator**
  Translator for indexing object data. This is the standard search indexer.

- **Active Content Structure Translator**
  Translator for indexing structure data.

15. In the **Active Workspace Object Data Translator Settings** panel, set the maximum number of object data translators. **Maximum Limit** controls the maximum number of individual operations configured in a module. The default is 3, the minimum value is 1. This panel appears only if you selected the **Active Workspace Object Data Translator**.

16. In the **Active Content Structure Translator Settings** panel, set the maximum number of structure translators. The default is 3, the minimum value is 1. This panel appears only if you selected the **Active Workspace Content Structure Translator**.
17. In the **Indexing Engine User** panel, type the Solr administrator’s user name and password. This is the user and password defined when installing the Indexing Engine (Solr). These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used). After typing the user name and password, click **Next**.

![Indexing Engine User panel](image)

18. In the **Confirmation** panel, click **Start**.

19. When the installation is complete, click **Close**.

20. If you installed the AsyncService translator and are setting up email notifications, you must set these preferences:

- **MAIL_OSMAIL_ACTIVATED** = true
- **MAIL_INTERNAL_MAIL_ACTIVATED** = true
- **MAIL_SERVER_CHARSET** = ISO-8859-1
- **MAIL_SERVER_NAME** = mail-server-name
- **MAIL_SERVER_PORT** = 25
- **MAIL_SUBSCRIPTION_NOTIFY_SUB_GROUP_TOO** = FALSE
- **WEB_DEFAULT_SITE_SERVER** = *host:*port
  
  *host* is the machine on which the web application server is running.

  *port* is the port value used by the web application server.

- **WEB_DEFAULT_SITE_DEPLOYED_APP_NAME** = tc-web-tier-app
  
  *tc-web-tier-app* is the name of the Teamcenter web tier application, for example, *tc*. 
Install Axis for ReqMgmtWordToHtmlTrans translator

This step is required only if you are using the ReqMgmtWordtoHtmlTrans translator with Active Workspace. Before installing Axis, the Dispatcher Server and Dispatcher Client must already be installed.

1. From the installation media for the base Teamcenter release, navigate to \additional_applications\dispatcher\axis and locate the axis.zip file.

2. Extract axis.zip to the dispatcher_root\DispatcherClient\lib directory.

   The dispatcher_root is the Dispatcher root installation directory, not the TC_ROOT installation directory.

3. In a plain text editor, open the dispatcher_root\DispatcherClient\conf\DispatcherClient.config file.

4. Uncomment the following line by removing the # character:

   #Axis.Port=8080

5. Ensure that the Axis port is not used by any other application.

6. Save and close the file.

7. Launch the rich client and choose Edit→Options.

8. Search for the SyncDispatcherClientUrl site-level preference.

9. Set the value of the preference to the Axis URL. The format of this URL is:

   http://host:port/axis2/services/AxisDispatcherClient.AxisDispatcherClientHttpSoap12Endpoint

   host is the machine on which Axis is installed.

   port is the port value used by Axis. You defined this by setting the Axis.Port property in the DispatcherClient.config file.

10. Start the Dispatcher Scheduler, Module, and Client:

    a. Run dispatcher_root\Scheduler\bin\runscheduler.bat to start the Dispatcher Scheduler.

    b. Run dispatcher_root\Module\bin\runmodule.bat to start the Dispatcher Module.

    c. Run dispatcher_root\DispatcherClient\bin\runDispatchClient.bat to start the Dispatcher Client.
11. Open a web browser and go to the Axis URL:

   \texttt{http://host:port/axis2/services/AxisDispatcherClient.AxisDispatcherClientHttpSoap12Endpoint}

   \textit{host} is the machine on which Axis is installed.

   \textit{port} is the port value used by Axis. You defined this by setting the \texttt{Axis.Port} property in the \texttt{DispatcherClient.config} file.

12. Verify that \textbf{Submit} is one of the available options.
Indexing Engine

Indexing Engine overview

Installing the Indexing Engine installs the Solr database. This is a full-text search engine that stores indexed Teamcenter data. It is installed by selecting the Indexing Engine feature in TEM.

These tables are reproduced in the Indexing Engine tab of the Active_Workspace_version_Install_Spreadsheet.xlsx spreadsheet. You can use this spreadsheet, which is distributed in the Tcversion_ActiveWorkspaceversion_doc.zip file on GTAC, to record your environment’s installation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC_DATA</td>
<td>Teamcenter installation</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solr home directory</td>
<td>Install Indexing Engine (Solr)</td>
<td></td>
</tr>
<tr>
<td>Search Engine URL (<a href="http://host:8983/solr">http://host:8983/solr</a>)</td>
<td>Install Indexing Engine (Solr)</td>
<td></td>
</tr>
<tr>
<td>Indexing Engine user name and password</td>
<td>Install Indexing Engine (Solr)</td>
<td></td>
</tr>
</tbody>
</table>
Indexing Engine prerequisites

Prerequisites for installing the Indexing Engine:

- 64-bit operating system
- JDK 64-bit or JRE 64-bit

Install the Active Workspace patch on the Indexing Engine host

It is highly likely that the indexing features will be installed on a machine that does not have a Teamcenter corporate server. In this case, you need to do the following:

1. Install the base Teamcenter release with no features (Teamcenter 10.1, Teamcenter 9.1, or Teamcenter Rapid Start 10.1):
   a. Launch TEM from the installation media for the base Teamcenter release.
   b. In the Installer Language dialog box, select a language and click Next.
   c. In the Welcome to Teamcenter panel, click Install.
   d. In the Configuration panel, type an ID and Description, and then click Next.
   e. In the Solutions panel, make no selections and click Next.
   f. In the Features panel, enter the installation root directory for the Indexing Engine feature in the Installation Directory box. Do not select features. Click Next.
   g. In the Confirmation panel, click Start.
   h. When the installation is complete, click Close.

2. Patch your environment to the supported Teamcenter patch level (Teamcenter 10.1.4, Teamcenter 9.1.3.4, or Teamcenter Rapid Start 10.1.4). Refer to the general patch instructions in the Teamcenter documentation as well as the readme file for the patch.

In the Teamcenter documentation:
• Windows Server Installation→Additional configuration and maintenance→Installing Teamcenter patches

• UNIX and Linux Server Installation→Additional configuration and maintenance→Installing Teamcenter patches

3. Add the Active Workspace patch to the installation. To do this, follow the steps in Install the Active Workspace patch on the Teamcenter servers.

Install Indexing Engine (Solr)

1. Launch the TEM for the server to which the Indexing Engine feature is to be added. This server must be patched to the appropriate Teamcenter and Active Workspace release levels.

2. In the Maintenance panel, select Configuration Manager and click Next.

3. In the Configuration Maintenance panel, select Perform maintenance on an existing configuration and click Next.

4. In the Old Configuration panel, select the configuration to which the indexing features are to be added and click Next.

5. In the Feature Maintenance panel, select Add/Remove Features and click Next.
6. In the **Features** panel, expand **Base Install→Active Workspace→Indexing Server**, select the **Active Workspace Indexing Engine** feature, and then click **Next**.

![Features panel](image)

This feature installs the indexing engine that is used by the Active Workspace indexer feature to provide indexing capability.

7. In the **Indexing Engine User** panel, type the Solr administrator’s user name and password. These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used). After typing the user name and password, click **Next**.

![Indexing Engine User panel](image)
8. Complete the **Indexing Engine Configuration** panel.

**Indexing Engine Configuration**

Browse to the folder containing the SOLR schema files. The 'tsi/solr_schema_files' directory will be generated under TC_DATA as part of Active Workspace Server Extensions install and contains the SCLR schema files. If the latest 'tsi/solr_schema_files' directory is not already on this machine then please copy or map the directory.

**SOLR schema files location**

In the **SOLR schema files location** box, type or browse to the path that contains the Solr schema files. This allows the Solr and Teamcenter schemas to be automatically merged during installation of the Indexing Engine. If you leave this box blank, the schemas are not automatically merged; you must manually merge them after installing the Indexing Engine.

The Solr schema files are created on the corporate server when installing the Server Extensions. There are two schema files located in `TC_DATA/ftsisolr_schema_files`:

- **TC_ACE_SOLR_SCHEMA.xml**
- **TC_SOLR_SCHEMA.xml**

If you have not yet installed the Server Extensions on the corporate server, you can complete the installation of the Indexing Engine, but the schemas are not automatically merged. You must manually merge them after installing the Indexing Engine and the Server Extensions.

**Install indexing engine as a service**

Select this to install the Indexing Engine as a service. If this is not selected, you must start the Indexing Engine manually.

9. In the **Operating System User** panel, type the user’s password and click **Next**.

10. In the **Confirmation** panel, click **Start**.

11. When the installation is complete, click **Close**.
**Start Solr**

Solr is installed on the machine on which the **Indexing Engine** feature is installed. It is located in `INDEXING-ENGINE-ROOT\solr-4.4.0`.

If you selected to install the Indexing Engine as a service, the **Active Workspace Indexing Service** was created and started during installation. Its **Startup Type** is **Automatic**.

If you did not install the Indexing Engine as a service, you must start it manually by running:

```
INDEXING-ENGINE-ROOT\solr-4.4.0\runSolr.bat
```

To verify that Solr is running:

1. Open a web browser and check that you can access the Solr administration page:

   ```
   http://host:port/solr/admin
   ```

   *host* is the machine on which Solr is installed.

   *port* is the port value used by Solr. The default is **8983**.

2. Log on with the Solr administrator user name and password that you defined when installing the Indexing Engine.

**Server Extensions tasks must be complete**

Before you can proceed to the next task in the **Indexing Engine** taskflow, all tasks in the **Server Extensions** taskflow must be complete.
Merge the Teamcenter and Solr schemas

Before indexing data, you must merge the Solr schema with the Teamcenter schema. If you did not provide the location of the directory containing the schema files (TC_DATA\fts\solr_schema_files) during Indexing Engine installation, you must merge the schemas manually.

1. If Solr is running, stop it.

2. Locate the TC_SOLR_SCHEMA.xml and TC_ACE_SOLR_SCHEMA.xml files in the TC_DATA\fts directory on the corporate server. To manually merge the schemas, these files must be available on the machine on which Solr is installed.

   TC_ACE_SOLR_SCHEMA.xml is present only if the Active Content Structure feature is installed.

   Solr is installed on the same machine as the Indexing Engine feature. If you installed this feature on a machine other than the corporate server, you must copy these files to a temporary location on the Solr host.

3. Open a command prompt and change to the SOLR_HOME directory.

   If you installed the indexing features in the corporate server configuration, the SOLR_HOME directory is TC_ROOT\solr-4.4.0 directory.

   If you installed the indexing features on a separate machine, the SOLR_HOME directory is indexing-engine-install-dir\solr-4.4.0.

4. Run:

   TcSchemaToSolrSchemaTransform.bat LOCAL-DIR

   LOCAL-DIR is the local directory containing the TC_SOLR_SCHEMA.xml and TC_ACE_SOLR_SCHEMA.xml files.

   This updates the Solr schemas using the XML files.

5. Restart Solr either by restarting the Active Workspace Indexing Service or by running the following command:

   INDEXING-ENGINE-ROOT\solr-4.4.0\runSolr.bat
Indexer

Indexer overview

Installing the Indexer feature installs the TcFTSIndexer, which is a four-tier SOA client used to export Teamcenter data for importing into the Solr database. This service manages overall indexing processes. It is installed by selecting the Indexer feature in TEM. The TcFTSIndexer is run manually for the initial indexing and can be scheduled to run periodically for subsequent updates.

Indexing can be deployed in two different modes:

- **Standalone Indexing Mode**
  
  Indexing operations are run on one machine using a single Java process that connects to the Teamcenter server manager and the Solr search engine.

  This mode supports indexing both object data and structure data. It is the only mode that supports object data. For custom types, this mode can be used if the type associated with TcFtsIndexer supports Standalone mode.

- **Dispatcher-Based Indexing Mode**
  
  Indexing operations are initiated using the Dispatcher and run across multiple machines. This provides load balancing for better performance if indexing steps are CPU or memory intensive.

  The Dispatcher-based mode supports structure data indexing. If you run the TcFTSIndexer in Dispatcher mode against object data, the indexer will revert to using Standalone mode. For custom types, this mode can be used if the type associated with TcFtsIndexer supports Dispatcher mode.

[2] Requires the Indexing Engine user name and password defined when installing the Indexing Engine.

[3] Requires Dispatcher Server URL and staging directory if Dispatcher-based indexing is used.
These tables are reproduced in the **Indexer** tab of the **Active_Workspace_version_Install_Spreadsheet.xlsx** spreadsheet. You can use this spreadsheet, which is distributed in the **Tcversion_ActiveWorkspaceversion_doc.zip** file on GTAC, to record your environment's installation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexing Engine user name and password</td>
<td><strong>Install Indexing Engine (Solr)</strong></td>
<td></td>
</tr>
<tr>
<td>Teamcenter four-tier URL</td>
<td>Teamcenter web tier app deployment</td>
<td></td>
</tr>
<tr>
<td>Dispatcher Server URL (only if Dispatcher-based indexing is used)</td>
<td><strong>Install Dispatcher Server</strong></td>
<td></td>
</tr>
<tr>
<td>Dispatcher staging directory (only if Dispatcher-based indexing is used)</td>
<td><strong>Install Dispatcher Server</strong></td>
<td></td>
</tr>
<tr>
<td>Dispatcher root (only if Dispatcher-based indexing is used)</td>
<td><strong>Install Dispatcher Server</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>TcFTSIndexer directory</td>
<td><strong>Install Indexer (TcFTSIndexer)</strong></td>
</tr>
</tbody>
</table>
Install the Active Workspace patch on the Indexer host

It is highly likely that the indexing features will be installed on a machine that does not have a Teamcenter corporate server. In this case, do the following:

1. Install the base Teamcenter release (Teamcenter 10.1, Teamcenter 9.1, or Teamcenter Rapid Start 10.1):
   a. Launch TEM from the installation media for the base Teamcenter release.
   b. In the **Installer Language** dialog box, select a language and click **Next**.
   c. In the **Welcome to Teamcenter** panel, click **Install**.
   d. In the **Configuration** panel, type an **ID** and **Description**, and then click **Next**.
   e. In the **Solutions** panel, make no selections and click **Next**.
   f. In the **Features** panel, enter the installation root directory for the Visualization Server Manager in the **Installation Directory** box. Do not select features. Click **Next**.
   g. In the **Confirmation** panel, click **Start**.
   h. When the installation is complete, click **Close**.

2. Patch your environment to the supported Teamcenter patch level (Teamcenter 10.1.4, Teamcenter 9.1.3.4, or Teamcenter Rapid Start 10.1.4). Refer to the general patch instructions in the Teamcenter documentation as well as the readme file for the patch.

   In the Teamcenter documentation:
   - *Windows Server Installation*→*Additional configuration and maintenance*→*Installing Teamcenter patches*
   - *UNIX and Linux Server Installation*→*Additional configuration and maintenance*→*Installing Teamcenter patches*

3. Add the Active Workspace patch to the installation. To do this, follow the steps in **Install the Active Workspace patch on the Teamcenter servers**.
Install Indexer (TcFTSIndexer)

Install Active Workspace Patch on Indexer host → Install Indexer (TcFTSIndexer) [2], [3] → Dispatcher (if used) and Indexing Engine tasks must be complete → Verification: Test TcFTSIndexer connectivity → Test TcFTSIndexer connectivity → Run initial index of object data → Configure classification search (optional) → Client tasks must be complete → Verification: Run search from the Active Workspace interface

[2] Requires the Indexing Engine user name and password defined when installing the Indexing Engine.

[3] Requires Dispatcher Server URL and staging directory if Dispatcher-based indexing is used.

1. Launch the TEM for the server to which the Indexer feature is to be added. This server must be patched to the appropriate Teamcenter and Active Workspace release levels.

2. In the Maintenance panel, select Configuration Manager and click Next.

3. In the Configuration Maintenance panel, select Perform maintenance on an existing configuration and click Next.

4. In the Old Configuration panel, select the configuration to which the indexing features are to be added and click Next.

5. In the Feature Maintenance panel, select Add/Remove Features and click Next.
6. In the **Features** panel, expand **Base Install → Active Workspace → Indexing Server**, select the **Active Workspace Indexer** feature, and then click **Next**.

![Features panel](image)

**Features**
The features listed below are installed in the current configuration. Select a feature to add it to this configuration. Deselect a feature to remove it; those features which are grayed out are not removable.

- Active Workspace
- Client
- Indexing Server
  - Active Workspace Indexing Engine
  - Active Workspace Indexer
- Server Extensions
- Visualization Server
- Server Enhancements
- Extensions

This feature configures the Active Workspace indexer settings.

7. In the **Teamcenter Administrative User** panel, type the user’s password and click **Next**.

8. In the **Indexing Engine User** panel, type the Solr administrator’s user name and password. This is the user and password defined when installing the **Indexing Engine (Solr)**. These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used). After typing the user name and password, click **Next**.

![Indexing Engine User](image)

**Indexing Engine User**
Enter the user and password for the Indexing Engine user in the fields provided below. The credentials entered must match for every feature that will use the Indexing Engine.

<table>
<thead>
<tr>
<th>User</th>
<th>solr_admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td></td>
</tr>
</tbody>
</table>
9. In the **Active Workspace Indexer Settings** panel:

![Active Workspace Indexer Settings](image)

- Choose the indexing method to use:
  - **Standalone indexing environment**:
    - Indexing operations are run on one machine using a single Java process (TcFTSIndexer orchestrator), which connects to the Teamcenter server manager and the Solr search engine. This Java process is an SOA client that tracks the number of **TcServer** connections and executes flows based on the configuration.
    - This mode supports indexing both object data and structure data. It is the only mode that supports object data. For custom types, this mode can be used if the type associated with TcFtsIndexer supports Standalone mode.
    - This indexing method does not use the Dispatcher.
  - **Dispatcher-based indexing environment**:
    - Indexing operations are initiated using the Dispatcher and run across multiple machines. This provides load balancing for better performance if indexing steps are CPU or memory intensive.
    - The Dispatcher-based mode supports structure data indexing. If you run the TcFTSIndexer in Dispatcher mode against object data, the indexer will revert to using Standalone mode. For custom types, this mode can be used if the type associated with TcFtsIndexer supports Dispatcher mode.

- **Teamcenter 4-tier URL**
  - The URL for the deployed J2EE Teamcenter web application. The format of this URL is:
    ```
    http://host:port.awc
    ```
    - `host` is the machine running the web application server on which the Teamcenter web application is deployed.
    - `port` is the port used by the web application server.
• **Dispatcher Server URL**
  
  (Dispatcher-based indexing only) This is the URL used by the Dispatcher Schedule. It was defined in the Dispatcher Components TEM panel when installing the Dispatcher Server. The format of this URL is:

  \texttt{rmi://host:port}

  \textit{host} is the machine on which the Dispatcher Scheduler is running.

  \textit{port} is the port value used by the Dispatcher Scheduler. The default is 2001.

• **Staging Directory**
  
  (Dispatcher-based indexing only) Type or browse to the dispatcher staging directory. It was defined in the Dispatcher Components TEM panel when installing the Dispatcher Server.

10. Click the Advanced at the bottom of the Active Workspace Indexer Settings panel to open the Active Workspace Indexer Advanced Setting dialog box.

11. In the Active Workspace Indexer Advanced Setting dialog box, you can set:

   ![Specify settings for Teamcenter maximum allowable connections and connection retry count below.](image)

   - **Maximum Teamcenter Connections**
     
     Specifies the maximum number of connection between the Teamcenter server and the indexer. Usually, this number should be less than the number of warm TcServers available in Teamcenter server manager. The minimum value allowed for Maximum Teamcenter Connections is 2.

   - **Teamcenter Retry Count**
     
     Specifies the number of tries to connect to the Teamcenter server. Minimum value is 1.

12. Click OK to close the Active Workspace Indexer Advanced Setting dialog box.

13. Click Next in the Active Workspace Indexer Settings panel.
14. In the **Active Workspace Indexer Type Selection** dialog box, select the indexer types to install.

If you are using Dispatcher-based indexing, select the same indexer types for this and for the **TcFtsIndexerTrans** translator.

- **Active Workspace Object Data Indexer**
  Indexes Teamcenter data. This is the standard search indexer.

- **Active Content Structure Indexer**
  Indexes structure data.
15. In the **Active Workspace Object Data Indexer Settings** panel, set the extraction start and end times.

![Active Workspace Object Data Indexer Settings](image)

**Active Workspace Object Data Indexer Settings**

Configure the settings for extraction start and end times in the fields provided below.

<table>
<thead>
<tr>
<th>Extraction Time</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>dd-MMM-yyyy</td>
<td>HH:mm</td>
</tr>
<tr>
<td>Example: 01-Jan-2014</td>
<td>Example: 13:15</td>
<td></td>
</tr>
<tr>
<td><strong>No End Time</strong></td>
<td>Date</td>
<td>Time</td>
</tr>
<tr>
<td>Example: 01-Jan-2014</td>
<td>Example: 13:15</td>
<td></td>
</tr>
</tbody>
</table>

If ‘No End Time’ is specified then indexing will be done for all items that have been created from the specified start time until now.

- **Start Time**

  All data modified after this date and time will be extracted for indexing; data older than this will not be extracted. This value is only used during first-time indexing or re-indexing.

  **Date**

  Format is *dd-MMM-yyyy*. Values for month are based on locale, for example, 01-Jan-2014.

  **Time**

  Format is *HH:mm* in 24-hour format.

- **No End Time**

  If selected, all data modified from the start time to the present is indexed.

- **End Time**

  If selected, specifies the end date for extracting data. All data modified after this date will not be extracted for indexing. This value is only used during first-time indexing or re-indexing.

  **Date**

  Format is *dd-MMM-yyyy*. Values for month are based on locale, for example, 01-Jan-2014.

  **Time**

  Format is *HH:mm* in 24-hour format.
16. Click the **Advanced** button at the bottom of the *Active Workspace Object Data Indexer Settings* panel to open the **Advanced Settings** dialog box.

17. In the **Advanced Settings** dialog box, you can set:

- **Maximum Query Timespan**
  
  Specifies the maximum span of a Teamcenter query in minutes. Maximum value is 50000; minimum value is 5000; default value is 20000.

- **Export Batch Size**
  
  Specifies the maximum number of Teamcenter objects handled in one thread. Maximum value is 20000; minimum value is 1; default value is 1000.

18. Click **OK** to close the **Advanced Settings** dialog box.

19. Click **Next** in the *Active Workspace Object Data Indexer Settings* panel.

20. In the **Confirmation** panel, click **Start**.

21. When the installation is complete, click **Close**.

**Dispatcher (if used) and Indexing Engine tasks must be complete**

Before you can proceed to the next task in the *Indexer* taskflow, all tasks in the **Dispatcher (optional)** (if used) and **Indexing Engine** taskflows must be complete.
Test TcFTSIndexer connectivity

1. Ensure that the Teamcenter user running the runFTSIndexer can log on to the database. The default user that runs the utility is infodba, as defined in the Tc.user setting in the TC_ROOT/TcFTSIndexer/conf/TcFtsIndexer.properties file.

2. When running in a Security Services protected environment, this user must also be a valid user in the LDAP server against which Security Services authenticates.
   a. If you are using multiple TCCS Security Services application IDs, make sure they are configured correctly.

   You can configure multiple application IDs using the Environment Settings for Client Communication System panel in Teamcenter Environment Manager (TEM).

   Environment Settings for Client Communication System panel

   b. Ensure that the user defined by the Tc.user setting in the TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer.properties file is a valid user in the LDAP server and the Teamcenter database. Create a user in both if needed or select an existing valid active user to run the runTcFTSIndexer utility.
Note
If you create a new user, create an encrypted password file by setting an environment variable, within the console, to the password value, for example:

```bash
set mytcenv=mytcenv
```

Then run the `encryptPass.bat/sh` utility with the `-tc` argument specifying the environment variable name created, for example:

```bash
encryptPass -tc password
```

Then run the utility for the LDAP password used by the TcFTSIndexer user. The `encryptPass.bat/sh` utility is located in the `TC_ROOT\TcFTSIndexer\bin` directory.

After creating the encrypted password file insure that the password variable is removed.

3. Ensure that the following are running:
   - Teamcenter database
   - Solr
   - Web application server hosting the Teamcenter web tier application
   - Web application server hosting the Active Workspace web application
   - Server manager
   - Dispatcher Scheduler (only if Dispatcher-based indexing is used)
     Start the Dispatcher Scheduler by running:
     ```bash
     DispatcherRoot\Scheduler\bin\runscheduler.bat
     ```
   - Dispatcher Module (only if Dispatcher-based indexing is used)
     Start the Dispatcher module by running:
     ```bash
     DispatcherRoot\Module\bin\runmodule.bat
     ```

4. On the machine on which the **Indexer** feature is installed, open a command prompt.

5. Navigate to the bin directory of the TcFTSIndexer, for example, `TC_ROOT\TcFTSIndexer\bin`. 
6. To test the TcFTSIndexer’s connectivity, run the command:

`runTcFTSIndexer -task=TYPE:test`

Add the `dispatcher` argument only for Dispatcher-based indexing.

For `TYPE`, substitute:

- **objdata**
  
  Object data does not support Dispatcher-based indexing. Object data requires that **Active Workspace Object Indexer** is selected for the Indexer. This is done in the **Active Workspace Indexer Type Selection** TEM panel when installing the **Indexer feature**.

- **structure**

  Structure data requires that **Active Content Structure Indexer** is selected for the Indexer and, if Dispatcher-based indexing is used, that **Active Content Structure Translator** is selected for the **TcFtsIndexerTrans** translator. This is done in the **Active Workspace Indexer Type Selection** TEM panel when installing the **Indexer feature** and in the **Active Workspace Translator Selection** TEM panel when installing the **TcFtsIndexerTrans** translator.

**Example: Test Standalone Indexer for object data**

1. For stand-alone indexing, to test object data connectivity, type the following command:

`runTcFTSIndexer -task=objdata:test`

Following is an example of the output:

```
2015-03-30 12:27:37,837 INFO - Downloading 'FTSIndexer Files' Dataset files for 'objdata' indexer...
OS name is Windows Server 2008 R2
JVM vendor is Oracle Corporation
Data model is 64
2015-03-30 12:27:41,896 INFO - Solr server status: OK
2015-03-30 12:27:46,963 INFO - Transform configuration: OK
2015-03-30 12:27:46,968 INFO - Total time for all Steps 7 sec
2015-03-30 12:27:46,969 INFO - Done processing Type: objdata FlowAction: test
```

2. Verify that there were no errors. If you encounter errors, refer to **Resolving TcFTSIndexer issues**.

**Example: Test Standalone Indexer for structure data**

1. For stand-alone indexing, to test structure data connectivity, type the following command:

`runTcFTSIndexer -task=structure:test`

Following is an example of the successful output:

```
2015-03-30 12:32:19,668 INFO - Running TcFtsIndexer Type: structure FlowAction: test
2015-03-30 12:32:117 INFO - FMS status : OK.
2015-03-30 12:32:41,896 INFO - Solr server status: OK
2015-03-30 12:32:34,927 INFO - Downloading 'FTSIndexer Files' Dataset files for 'structure' indexer...
2015-03-30 12:32:36,455 INFO - Downloading 'FTSIndexer ACE Files' Dataset files for 'structure' indexer...
2015-03-30 12:32:36,576 INFO - Download transform files: OK
2015-03-30 12:32:36,576 INFO - All tests successful.
2015-03-30 12:32:37,222 INFO - Total time for all Steps 4 sec
2015-03-30 12:32:37,223 INFO - Overall Time 5.181 sec
```
Chapter 2: Installation

2015-03-30 12:32:37,224 INFO - Done processing Type: structure FlowAction: test

Following is an example of the output with errors:

OS name is Windows Server 2008 R2
JVM vendor is Oracle Corporation
Data model is 64
2014-07-27 18:32:08,601 INFO - Solr server status: OK
2014-07-27 18:32:17,137 INFO - Downloading 'FTSIndexer Files' Dataset files for 'structure'
indexer...
2014-07-27 18:32:31,265 INFO - Downloading 'FTSIndexer ACE Files' Dataset files for 'structure'
indexer...
2014-07-27 18:32:39,172 ERROR - U147714b22d47c0a8c2800003 has errors in Step teststructurestep Flow
testflow
2014-07-27 18:32:39,176 INFO - Total time for all Steps 48 sec

2. Verify that there were no errors. If you encounter errors, refer to Resolving TcFTSIndexer issues.

Example: Test Dispatcher-based Indexer for structure data

1. For Dispatcher-based indexing, to test structure data connectivity, type the following command:
   
   runTcFtsIndexer -dispatcher -task=structure:test

   Following is an example of the successful output:

   2015-03-30 12:41:56,158 INFO - Running TcFtsIndexer Type: structure FlowAction: test
   2015-03-30 12:41:29,957 INFO - Teamcenter 4 tier server status: OK
OS name is Windows Server 2008 R2
JVM vendor is Oracle Corporation
Data model is 64
2015-03-30 12:41:29,930 INFO - FMS status: OK
2015-03-30 12:41:30,465 INFO - Solr server status: OK
2015-03-30 12:41:33,245 INFO - Downloading 'FTSIndexer Files' Dataset files for 'structure'
indexer...
2015-03-30 12:41:34,309 INFO - Downloading 'FTSIndexer ACE Files' Dataset files for 'structure'
indexer...
2015-03-30 12:41:34,348 INFO - Download transform files: OK
2015-03-30 12:41:34,450 INFO -
2015-03-30 12:41:35,500 INFO - Dispatcher configuration: OK
2015-03-30 12:41:35,575 INFO - Total time for all Steps 5 sec
2015-03-30 12:41:35,576 INFO - Overall Time 5.877 sec
2015-03-30 12:41:35,576 INFO - Done processing Type: structure FlowAction: test

2. Verify that there were no errors. If you encounter errors, refer to Resolving TcFTSIndexer issues.
Run initial index of object data

1. If there were no errors resulting from the TcFTSIndexer connectivity test, you are ready to run the initial index. Ensure that the following are running:
   - Teamcenter database
   - Solr
   - Web application server hosting the Teamcenter web tier application
   - Web application server hosting the Active Workspace
   - Server manager

2. On the machine on which the **Indexer** feature is installed, open a command prompt.

3. Navigate to the bin directory of the TcFTSIndexer, for example, `TC_ROOT\TcFTSIndexer\bin`.

4. For object data indexing, type the following command:
   ```
   runTcFTSIndexer -task=objdata:reindex
   ```

5. The initial index may take some time to run if there is existing data in the database.

   Following is an example of the output:

   ```
   2015-03-30 13:20:18,166 INFO - Downloading 'FTSIndexer Files' Dataset files for 'objdata' indexer...
   OS name is Windows Server 2008 R2
   JVM vendor is Oracle Corporation
   Data model is 64
   2015-03-30 13:20:20,353 INFO - FMS status : OK.
   2015-03-30 13:20:25,366 INFO - Solr server status: OK
   2015-03-30 13:20:27,011 INFO - [01-Jan-2015 12:00 TO 15-Jan-2015 09:20] Query complete. 0 objects found in 1.093 seconds
   2015-03-30 13:20:27,419 INFO - [26-Feb-2015 01:20 TO 11-Mar-2015 23:40] Query complete. 0 objects found in 0.233 seconds
   2015-03-30 13:20:27,795 INFO - [12-Feb-2015 04:00 TO 26-Feb-2015 01:20] Query complete. 178 objects found in 0.31 seconds
   2015-03-30 13:20:28,063 INFO - [29-Jan-2015 06:40 TO 12-Feb-2015 04:00] Query complete. 0 objects found in 0.194 seconds
   2015-03-30 13:20:28,578 INFO - [15-Jan-2015 09:20 TO 29-Jan-2015 06:40] Query complete. 0 objects found in 0.39 seconds
   2015-03-30 13:20:48,669 INFO - Export of 178 objects in (secs) 20.007
   ```
2. Verify that there were no errors.
TcFTSIndexer scheduling

1. Navigate to the TC_ROOT\TcFTSIndexer\bin directory.
   This is on the machine on which the indexing features (Active Workspace Indexer and Active Workspace Indexing Engine) are installed.

2. If you are using multiple types of indexing (for example, object indexing and structure indexing), the TcFTSIndexer process must be started in service mode as shown in this step. If you are using object indexing only, skip to the next step.
   To start the TcFTSIndexer in service mode, use the -service option. Using the -service option keeps the TcFTSIndexer process running so that multiple types of indexing operations can run concurrently. For example:

   runTcFTSIndexer -service

   If you are using the Dispatcher for indexing, use the -dispatcher option along with the -service option. For example:

   runTcFTSIndexer -dispatcher -service

3. Type runTcFTSIndexer -task=objdata:sync -interval=x, where x is the number of seconds to wait before rerunning the sync operation.
   The interval value, x, must be greater than 0.
   For example, to wait 25 seconds, type:

   runTcFTSIndexer -task=objdata:sync -interval=25

   If the -service option was previously used, the command is sent to the service and executed. Otherwise, the action is executed in the current process.

4. For structure indexing synchronization, the TcFTSIndexer process must be started in service mode, for example:

   runTcFTSIndexer -task=structure:sync -interval=300

5. To stop the TcFTSIndexer process:
   a. Type runTcFTSIndexer -stop.
b. Type `runTcFTSIndexer -status`.

   When this command returns that no flows are running, the process is stopped. You may need to wait a period and repeat this command in order to give the process time to stop.

6. You can use the `start` command to run the TcFTSIndexer service in a new window. In this example, the `-service` argument is not optional, but the `-dispatcher` argument is.

   a. Run TcFTSIndexer in a new window.

      `start runTcFTSIndexer [-dispatcher] -service`

   b. Start object data synchronization using the desired interval. In the following example the interval is 25 seconds.

      `runTcFTSIndexer -task=objdata:sync -interval=25`

   c. Start the structure data synchronization using the desired interval. In the following example the interval is 5 minutes.

      `runTcFTSIndexer -task=structure:sync -interval=300`

**Configure Classification Search (optional)**

This step is required only if you want to search for classification data in Active Workspace.

1. Activate the **Search Index** view in the Teamcenter Classification application by setting the `ICS_searchindex_view_visible` preference to `true`.

   For details about the preferences specific to Active Workspace, see *Preferences*.

   For information about working with preferences, see the *Preferences and Environment Variables Reference* in the Teamcenter collection.

2. Define the classification properties you want to index by doing one of the following:

   - To select all classes, attributes, and filters, run the `smlutility` utility.
   - To define specific classification properties:
a. In the Teamcenter Classification application, select the class you want to index and add the **Search Index** view to the class.

b. With the class selected, click **View Attribute** and specify the attributes you want indexed.

c. To specify that an attribute be used as a search filter, select **Filter**.

3. Turn on the Classification Search feature by setting the `Awp0SearchClassifySearchEnabled` business object constant (at the workspace object level) to `true`.

   For details about the constants specific to Active Workspace, see *Business Modeler IDE constants*.

   For information about working with constants, see *Business Modeler IDE* in the Teamcenter collection.

4. Specify that Classification data be indexed for searching by setting the `Awp0SearchIsClassifyDataIndexed` business object constant (at the workspace object level or below) to `true`.

   The object type on which it is set and the one below it are indexed for searching.

5. Update the custom template as described in *Run the search indexer after updating the custom template*.

   Each time you modify existing classification classes or views, or create new ones on top of them, you must repeat step 5.

**Client tasks must be complete**

Before you can proceed to the next task in the **Indexer** taskflow, all tasks in the **Client** taskflow must be complete.
Run search from the Active Workspace interface

You can test the initial indexing by logging on with the Active Workspace interface and searching for data.

1. Ensure that the following are running:
   - Web application server hosting the Active Workspace WAR file
   - Web application server hosting the Teamcenter web tier application
   - Server manager
   - Teamcenter database

2. Open a supported web browser.
   For supported versions of third-party software, refer to the certification matrix on GTAC:
   
   http://support.industrysoftware.automation.siemens.com/gtac.shtml

3. Go to the URL:
   
   
   *host* is the machine running the web application server on which the *awc.war* file is deployed.

   *port* is the port used by the web application server.

4. Log on as *infodba*.

5. To test search, type *infodba* in the search box and click the Search button.
   
   If you see results related to the *infodba* user and you do not receive any errors, the Active Workspace deployment is operating properly.

   Even if your Teamcenter environment is a new installation, there is data owned by *infodba*.
Visualization Server

Visualization Server overview

[4] Requires Assigner Host and Port defined when building the awc.war file.

The Visualization Server provides dynamic 3D visualization functionality to the Active Workspace client. The Visualization Server is optional. If you do not plan on using Visualization with Active Workspace, do not install its components.

The Visualization Server is comprised of:

- **Visualization Data Server** (optional)
  
  The Visualization Data Server improves Visualization performance by caching visualization data close to the Visualization Server Manager. It caches product structure, computes Massive Model Visualization (MMV) spatial hierarchies, and prepopulates JT files in the FCC for faster visualization streaming to the Active Workspace client. The Visualization Data Server is part of the MMV solution for Active Workspace.

  The Visualization Data Server is optional. You can use the Visualization Server Manager without this component. However, if MMV is used, the Visualization Data Server is required. In order to use the Visualization Data Server with MMV, the **Active Workspace Visualization Server** Extension feature needs to be installed. This features adds the awv0activeworkspacevis_template.xml template to the database. Additionally, you need to index structure data for the product configurations that you want to view with the MMV flag. Refer to Configuring structure indexing.

  A single Visualization Data Server can support one or more Visualization Server Managers.

- **Visualization Server Manager**

  The Visualization Server Manager streams visualization data to the Active Workspace client. It manages a pool of rendering processes, starting and stopping processes as needed.
**Visualization Data Server (optional)**

**Visualization Data Server overview**

The Visualization Data Server improves Visualization performance by caching visualization data close to the Visualization Server Manager. It caches product structure, computes Massive Model Visualization (MMV) spatial hierarchies, and prepopulates JT files in the FCC for faster visualization streaming to the Active Workspace client. The Visualization Data Server is part of the MMV solution for Active Workspace.

The Visualization Data Server is optional. You can use the Visualization Server Manager without this component. However, if MMV is used, the Visualization Data Server is required. In order to use the Visualization Data Server with MMV, the **Active Workspace Visualization** Server Extension feature needs to be installed. This features adds the awv0activeworkspacevis_template.xml template to the database. Additionally, you need to index structure data for the product configurations that you want to view with the MMV flag. Refer to [Configuring structure indexing](#).

A single Visualization Data Server can support one or more Visualization Server Managers.

These tables are reproduced in the **Vis Data Server (optional)** tab of the **Active_Workspace_version_Install_Spreadsheet.xlsx** spreadsheet. You can use this spreadsheet, which is distributed in the **Tcversion_ActiveWorkspaceversion_doc.zip** file on GTAC, to record your environment's installation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCC parents</td>
<td>Teamcenter installation</td>
<td></td>
</tr>
<tr>
<td>Teamcenter four-tier URL</td>
<td>Teamcenter installation</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visualization Data Server host and port</td>
<td>Install the Visualization Data Server</td>
<td></td>
</tr>
</tbody>
</table>
Prerequisites for the Visualization Data Server

Visualization Data Server prerequisites:

- The machine hosting the Visualization Data Server should have a minimum of 16 GB of RAM, but may require more. The amount of RAM needed depends on the number of structures to be indexed and their size.

  A rough rule of thumb is to count the number of lines in the unconfigured structure to be indexed and allow at least 2000 bytes per line. For example, if there are 1 million lines in the unconfigured product index, then 1 million * 2000 = 2 GB of RAM.

  If you are not sure of the size of the structures, Siemens PLM Software recommends that you allow approximately 4 GB of RAM for each structure you are planning to cache in the Visualization Data Server. For example, if 4 structures are to be indexed, 16 GB of RAM is required.

- Siemens PLM Software recommends that you install the Visualization Data Server on a machine with multiple processors.

  The Visualization Data Server is a multi-threaded server program and is thus resource intensive; multiple processor will be utilized if they are available. Standard server class machine hardware is sufficient.

- Virtual machine deployment is supported.

- There are no graphics card requirements.

- You must deploy the Visualization Data Server on a high speed LAN near the Visualization Server Manager.

- Siemens PLM Software recommends that you deploy the Visualization Data Server near or on a machine hosting an FSC cache or FSC volume. If you deploy the Visualization Data Server remote (on a WAN) from the FSC volume, you should deploy an FSC cache on a LAN near or on the Visualization Data Server host machine.

- You must deploy an FMS Client Cache (FCC) component on the machine hosting the Visualization Data server.

- Structure indexing must be set up. The Visualization Data Server uses the structure indexing infrastructure of Active Workspace to keep cached product structure up to date.
Install the Active Workspace patch on the Visualization Data Server host

Siemens PLM Software recommends that the Visualization Server Manager and the Visualization Data Server are installed on machines that do not have a Teamcenter corporate server. In this case, do the following:

1. Install the base Teamcenter version (Teamcenter 10.1, Teamcenter 9.1, or Teamcenter Rapid Start 10.1):
   a. Launch TEM from the installation media for the base Teamcenter release with no features.
   b. In the Installer Language dialog box, select a language and click Next.
   c. In the Welcome to Teamcenter panel, click Install.
   d. In the Configuration panel, type an ID and description, and then click Next.
   e. In the Solutions panel, make no selections and click Next.
   f. In the Features panel, enter the installation root directory for the Visualization Server Manager in the Installation Directory box. Do not select features. Click Next.
   g. In the Confirmation panel, click Start.
   h. When the installation is complete, click Close.

2. Patch your environment to the supported Teamcenter patch level (Teamcenter 10.1.4, Teamcenter 9.1.3.4, or Teamcenter Rapid Start 10.1.4). Refer to the general patch instructions in the Teamcenter documentation as well as the readme file for the patch.

   In the Teamcenter documentation:
   - Windows Server Installation→Additional configuration and maintenance→Installing Teamcenter patches
   - UNIX and Linux Server Installation→Additional configuration and maintenance→Installing Teamcenter patches

3. Add the Active Workspace patch to the installation. To do this, follow the steps in Install the Active Workspace patch on the Teamcenter servers.
Install the Visualization Data Server

To install the Visualization Data Server, do the following:

1. Launch TEM for the Visualization Data Server installation.
2. In the Maintenance panel, select Configuration Manager and click Next.
3. In the Configuration Maintenance panel, select Perform maintenance on an existing configuration and then click Next.
4. In the Old configuration panel, select the Visualization Data Server configuration and then click Next.
5. In the Feature Maintenance panel, select Add/Remove Features, and then click Next.
6. In the Features panel, expand Base Install → Active Workspace → Visualization Server, select the Visualization Data Server feature and click Next.
7. In the **Prerequisite Checks** panel, you run checks to ensure that the required RAM needed for the Visualization Data Server is available. Enter a location in which to create the log files for the prerequisite check and click **Run**.
8. Inspect the test result. If the check result in a status of **Fail**, you must correct the issues before proceeding. The **Next** button is unavailable until the check has a status of **Pass** or **Warning**. Click **View** for an analysis of a failure or warning. Click **Review** for a summary of all checks.

**Free RAM Availability** checks that the machine has the recommended minimum of 16 GBs of free physical RAM. If this recommendation is not met, the result is a warning and you are not prevented from proceeding.
9. Once the check result is either **Pass** or **Warning**, the **Next** button is available. Click **Next**.
10. If the prerequisite check resulted in a warning, the Confirm Warnings window displays the warnings. If you wish to continue, click the check box next to the warning and then click OK.
11. In the **File Client Cache (FCC)** panel, set the **FMS_HOME** environment variable. The FCC must be installed on the same machine as the Validation Data Server.

![File Client Cache (FCC)](image)

12. In the **FCC Parent** panel, define the list of FSC parents that the Visualization Data Server will connect to. You must provide protocol, host, and port of the FSC parent. To add rows to the table, click **Add**.

The Visualization Data Server and the Visualization Server Manager should use the same FSC.

![FCC Parents](image)

13. In the **Teamcenter Administrative User** panel, enter the user’s password and then click **Next**.
14. Complete the **Visualization Data Server Configuration** panel.

**Visualization Data Server Configuration**

The Visualization Data Server is used to cache product structure and prepopulate JT files for faster visualization. To configure the data server, provide input to the required fields below.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Port</td>
<td>9990</td>
</tr>
<tr>
<td>Teamcenter 4-tier URL</td>
<td>http://&lt;HOSTNAME&gt;:7001/tc</td>
</tr>
</tbody>
</table>

- **Server Port**
  This is the port number on which the Visualization Data Server listens.

- **Teamcenter 4-tier URL**
  This is the URL of the Teamcenter web tier application. The format is:

  \[http://host:port/tc-web-app\]

  *host* is the machine running the web application server on which the Teamcenter web application is deployed.

  *port* is the port value used by the web application server.

  *tc-web-app* is the name of the Teamcenter web application. The default is **tc**.

15. In the **Confirmation** panel, click **Start**.

16. When the installation is complete, click **Close**.
Start the Visualization Data Server

To start the Visualization Data Server Manager, run:

\texttt{TC\_ROOT\VisDataServer\Program\VisDataServer.exe}

After the Visualization Data Server is started, it automatically detects and caches product configurations that have been indexed with the MMV flag. These cached product configurations are now ready for fast visualization with the MMV technology.

A product configuration is ready for MMV visualization after it has been indexed, the Visualization Data Server has detected, downloaded and cached the structure, and prepopulated the FMS system. If you attempt to visualize a product configuration that is not yet completely indexed and cached in the Visualization Data Server, the viewer will default to the regular non-MMV mode. Changes in the product configuration need to be re-indexed and re-read by the Visualization Data Server before they can be displayed by the viewer.

Additional configuration for the Visualization Data Server is available in the \texttt{etc/VisDataServer.properties} file. This includes detailed logging and fine tuning of other settings. If you make changes to the properties file, you need to restart the Visualization Data Server.
Visualization Server Manager

Visualization Server Manager overview

![Visualization Server Manager Workflow Diagram]

[4] Requires Assigner Host and Port defined when building the awc.war file.

The Visualization Server Manager streams visualization data to the Active Workspace client. It manages a pool of rendering processes, starting and stopping processes as needed.

These tables are reproduced in the Vis Server Manager (optional) tab of the Active_Workspace_version_Install_Spreadsheet.xlsx spreadsheet. You can use this spreadsheet, which is distributed in the Tcversion_ActiveWorkspaceversion_doc.zip file on GTAC, to record your environment's installation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCC parents</td>
<td>Teamcenter installation</td>
<td></td>
</tr>
<tr>
<td>Visualization Pool Assigner host and port</td>
<td>Build the Java EE Client web application</td>
<td></td>
</tr>
<tr>
<td>Visualization Data Server host and port</td>
<td>Install the Visualization Data Server</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Where value is defined</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Prerequisites for the Visualization Server Manager

Visualization Server Manager prerequisites:

- The Visualization Server Manager’s supported hardware and software configurations are:
  - Server class hardware using NVIDIA GRID K1 or K2 graphics cards as certified by NVIDIA.
  - Windows Server 2008 R2.

  **Note**
  Siemens PLM Software considers supported hardware as suitable for a production environment. Other hardware may work with Active Workspace, but Siemens PLM Software accepts problem reports only for issues that are reproducible on a supported configuration.

  For information on other NVIDIA server hardware compatible with the GRID K1 or K2 graphics cards, see www.nvidia.com/buygrid.

- You can run the Visualization Server Manager on a virtual machine for demonstration or evaluation purposes for models where MMV is not being used. However, the Visualization Server Manager requires hardware graphics support for production deployments and to support MMV. For full GPU performance and functionality on a virtual machine, assign an NVIDIA GRID K1 or K2 graphics card to the VM through a hypervisor such as Citrix XenServer.

Install the Active Workspace patch on the Visualization Server Manager host

Siemens PLM Software recommends that the Visualization Server Manager and the Visualization Data Server are installed on machines that do not have a Teamcenter corporate server. In this case, do the following:

1. Install the base Teamcenter version (Teamcenter 10.1, Teamcenter 9.1, or Teamcenter Rapid Start 10.1):
   a. Launch TEM from the installation media for the base Teamcenter release with no features.
b. In the **Installer Language** dialog box, select a language and click **Next**.

c. In the **Welcome to Teamcenter** panel, click **Install**.

d. In the **Configuration** panel, type an ID and description, and then click **Next**.

e. In the **Solutions** panel, make no selections and click **Next**.

f. In the **Features** panel, enter the installation root directory for the Visualization Server Manager in the **Installation Directory** box. Do not select features. Click **Next**.

g. In the **Confirmation** panel, click **Start**.

h. When the installation is complete, click **Close**.

2. Patch your environment to the supported Teamcenter patch level (Teamcenter 10.1.4, Teamcenter 9.1.3.4, or Teamcenter Rapid Start 10.1.4). Refer to the general patch instructions in the Teamcenter documentation as well as the readme file for the patch.

   In the Teamcenter documentation:

   • *Windows Server Installation*→*Additional configuration and maintenance*→*Installing Teamcenter patches*

   • *UNIX and Linux Server Installation*→*Additional configuration and maintenance*→*Installing Teamcenter patches*

3. Add the Active Workspace patch to the installation. To do this, follow the steps in *Install the Active Workspace patch on the Teamcenter servers*.

### Install the Visualization Server Manager

| Visualization Server Manager | Prerequisites | Install Active Workspace patch on Visualization Server Manager host | Install the Visualization Server Manager [4] | Start the Visualization Server Manager | Visualization Data Server (if used) and Client tasks must be complete | Verification: Test Visualization from the Active Workspace interface |

[4] Requires Assigner Host and Port defined when building the **awc.war** file.

1. Launch TEM for the Visualization Server Manager installation.

2. In the **Maintenance** panel, select **Configuration Manager** and click **Next**.

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

4. In the **Old configuration** panel, select the Visualization Server Manager configuration and then click **Next**.
5. In the **Feature Maintenance** panel, select **Add/Remove Features** and then click **Next**.

6. In the **Features** panel, expand **Base Install→Active Workspace→Visualization Server**, select the **Visualization Server Manager** feature and click **Next**.

![Feature Maintenance Panel]

7. In the **File Client Cache (FCC)** panel, set the **FMS_HOME** environment variable.

![File Client Cache (FCC) Panel]
8. In the **FCC Parent** panel, define the list of FSC parents that the Visualization Server Manager will connect to. You must provide protocol, host, and port of the FSC parent. To add rows to the table, click **Add**.

If you are using a Visualization Data Server, the Visualization Data Server and the Visualization Server Manager should use the same FSC.
9. In the top section of the **Visualization Server Manager** panel, provide values for the Visualization Server Manager.

### Visualization Server Manager

The Visualization Server Manager process controls the Teamcenter Visualization servers by starting and timing out servers. It informs a Server Assigner of its actions so that the Assigner can assign available servers to user sessions.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Host Alias</td>
<td>VisServerManager362</td>
</tr>
<tr>
<td>Server Host</td>
<td>lm6s003</td>
</tr>
<tr>
<td>Server Port</td>
<td>8090</td>
</tr>
<tr>
<td>Max Servers in Sub-Pool</td>
<td>30</td>
</tr>
<tr>
<td>Min Warm Servers</td>
<td>4</td>
</tr>
</tbody>
</table>

### Visualization Data Server Configuration

- **Add Visualization Data Server**

<table>
<thead>
<tr>
<th>Host</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9990</td>
</tr>
</tbody>
</table>

- **Local Host Alias**

Specifies the alias for the local Visualization Server Manager.

- **Server Host**

Specifies the host where the Visualization Server Manager is running. This should be the local machine name and must be resolvable by the Visualization Pool Assigner machine (the machine running the awc.war file). Do not use `localhost` or `127.0.0.1`.

- **Server Port**

Specifies the port on which the Visualization Server is listening.

- **Max Servers in Sub-Pool**

Specifies the maximum number of Visualization server processes allowed to run in this pool (for a single-host configuration) or in this subpool (for a multihost configuration).

- **Min Warm Servers**

Specifies the minimum number of Visualization server processes in this pool that are started but not assigned.

**Note**

If necessary to maintain the minimum number of warm servers, while not exceeding the maximum number of server processes, the server manager times out servers in use.
10. If you are using a Visualization Data Server, provide the values for this server in the bottom section of the **Visualization Server Manager** panel.

You use a Visualization Data Server to improve performance by caching product structure and JT parts files.

![Visualization Server Manager]

**Visualization Server Manager**

The Visualization Server Manager process controls the Teamcenter Visualization servers by starting and timing out servers. It informs a Server Assigner of its actions so that the Assigner can assign available servers to user sessions.

<table>
<thead>
<tr>
<th><strong>Configuration</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Host Alias</td>
<td>VisServerManager362</td>
</tr>
<tr>
<td>Server Host</td>
<td>lm6s003</td>
</tr>
<tr>
<td>Server Port</td>
<td>8090</td>
</tr>
<tr>
<td>Max Servers in Sub-Pool</td>
<td>30</td>
</tr>
<tr>
<td>Min Warm Servers</td>
<td>4</td>
</tr>
</tbody>
</table>

- **Add Visualization Data Server**
  Select this if you are using a Visualization Data Server.

- **Host**
  Type the name of the host on which the Visualization Data Server is installed.

- **Port**
  Type the port value used by the Visualization Data Server.
11. In the **Visualization Server Manager Settings** panel, define the settings for how the Visualization Server Manager communicates with the pool assigners.

![Visualization Server Manager Settings](image)

**Local Server Manager Settings**
- **Override local node settings**
  - **Host**: Config115644VM0
  - **Port**: 55577

**Visualization Server Pool Assigners**

<table>
<thead>
<tr>
<th>Assigner Host</th>
<th>Assigner Port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>55566</td>
</tr>
</tbody>
</table>

- **Override local node settings**
  Select to override the host name and port value. Enter the **Host** and **Port** values of the local machine.

- **Visualization Server Pool Assigners**
  Lists the pool assigners that this Visualization Server Manager will use.

  - **Assigner Host**
    Host name of the machine where the pool assigner is running. This is the machine on which the AWC WAR file is deployed.

  - **Assigner Port**
    Port value of the pool assigner.

12. In the **Operating System User** panel, type the user’s password and click **Next**.

13. In the **Confirmation** panel, click **Start**.

14. When the installation is complete, click **Close**.
**Start the Visualization Server Manager**

To start the Visualization Server Manager, run:

\[TC_ROOT\vispoolmanager\run_visservermgr.cmd\]

When running `run_visservermgr.cmd`, you can use Windows Remote Desktop Connection to log on to the machine on which the Visualization Server Manager is installed if you have an NVIDIA card with a driver version of 340.66 or later. Other remote access products, such as VNC, also can be used.

After running `run_visservermgr.cmd`, you can lock the machine, but you must remain logged on. If you log out, the Visualization Server Manager is shut down.

For information about configuring Windows for automatic log on, see *Configure automatic logon for Windows*.

**Visualization Data Server (if used) and Client tasks must be complete**

Before you can proceed to the next task in the *Visualization Server* taskflow, all tasks in the *Client* taskflow must be complete.

**Test Visualization from the Active Workspace client interface**

You can test the Visualization Server by logging on with the Active Workspace interface and viewing Visualization data, for example, a JT file.
1. Ensure that the following are running:
   • Visualization Server Manager
   • Web application server hosting the Active Workspace web application
   • Web application server hosting the Teamcenter web tier application
   • Server manager
   • Teamcenter database

2. Open a supported web browser.
   For supported versions of third-party software, refer to the certification matrix on GTAC:
   
   http://support.industrysoftware.automation.siemens.com/gtac.shtml

3. Go to the URL:

   **http://host:port/awc**

   *host* is the machine running the web application server on which the **awc.war** file is deployed.

   *port* is the port used by the web application server.

4. Type a user name and password and click **Sign in**.

5. Search for and open an object that has an attached JT file.

6. Click the **Viewer** tab to display the JT file.
Active Workspace Launcher

Install Client for Office and the Active Workspace Launcher

To use Teamcenter Client for Microsoft Office with Active Workspace, you must install both Teamcenter Client for Microsoft Office and the Teamcenter Active Workspace Launcher. Follow this installation process for the client side, especially if you use Microsoft Office on the Windows platform.

Note

- Microsoft Office must be installed prior to installing the Teamcenter Client for Microsoft Office.
- Requirements editing, which differs from opening a Word attachment to a Teamcenter object, has other specific installation requirements, including .NET 4.5.

1. Install Microsoft Office.

2. Install Teamcenter Client for Microsoft Office by following the steps in the Teamcenter help collection in Windows client installation→Installing Microsoft Office interfaces→ Installing Teamcenter Client for Microsoft Office.

3. Locate the installer for Teamcenter Active Workspace Launcher (setup.exe) in the Teamcenter service pack installation media in the Tcrelease\wntx64\wntx64\additional_applications\TcClientAppLauncher directory. This installer is an MSI-based InstallShield installer that can run in GUI mode or silent mode. Administrator privilege is required to install Teamcenter Active Workspace launcher.

   This launches the Teamcenter Active Workspace Launcher installation wizard.

5. In the Welcome dialog box, click Next.

6. In the License Agreement dialog box, accept the license terms, then click Next.

7. In the Ready to Install the Program dialog box, click Install.

8. In the InstallShield Wizard Complete dialog box, click Finish.

9. To confirm the installation, locate Teamcenter Active Workspace Launcher listed in the Programs and Features dialog box.

Troubleshoot the Teamcenter Active Workspace Launcher installation

If the appropriate Microsoft Office application fails to launch when opening an attachment, the .awoai file may be associated to be opened in Microsoft Word directly. To resolve this, try these two tasks:

- Uninstall and reinstall the Teamcenter Active Workspace Launcher and try again.

- Look at the HKEY_CLASSES_ROOT\awoai\shell\Open\command listing in the registry editor to verify the .awoai file is correctly associated to the TcClientApplauncher.exe command.
Uninstall the Teamcenter Active Workspace Launcher

1. Double-click **Uninstall** in the **Programs and Features** dialog box.
   This launches the **Teamcenter Active Workspace Launcher** installation wizard.

2. Click **Yes** in the installation wizard to remove the Teamcenter Active Workspace Launcher.

3. Click **Finish** when the installation wizard successfully uninstalls the Teamcenter Active Workspace Launcher.
Chapter 3: Patching and upgrading Active Workspace

Patch and upgrade deployment paths

If you already have Active Workspace 2.1, 2.2, or 2.3 installed and are moving to Active Workspace 2.4, you follow either a patching process or an upgrade process:

- **Patch**
  
  If your initial and target environments are both on the same Teamcenter major release, you follow a patching process. For example, if your initial environment is Teamcenter 9.1.2.10 and your target environment is Teamcenter 9.1.3.4, you follow a patching process.

- **Upgrade**
  
  If the target Teamcenter version to which you are moving is a later major release than your initial environment, you follow an upgrade process. For example, if your initial environment is Teamcenter 9.1.x and your target environment is Teamcenter 10.1.x, you follow an upgrade process.

Use the following table to determine which process to use. The **Process** column provides links to the corresponding process.

<table>
<thead>
<tr>
<th>Initial version level</th>
<th>Target version level</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Workspace 2.1 with Teamcenter 9.1.2.7</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.2 with Teamcenter 9.1.2.10</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.3 with Teamcenter 9.1.3.1</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.1 with Teamcenter 10.1.1.1</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.2 with Teamcenter 10.1.2.1</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.3 with Teamcenter 10.1.2.3</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Patch</td>
</tr>
<tr>
<td>Active Workspace 2.1 with Teamcenter 9.1.2.7</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Upgrade</td>
</tr>
<tr>
<td>Active Workspace 2.2 with Teamcenter 9.1.2.10</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Upgrade</td>
</tr>
<tr>
<td>Active Workspace 2.3 with Teamcenter 9.1.3.1</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
<td>Upgrade</td>
</tr>
</tbody>
</table>
Patching to Active Workspace 2.4

If you are moving from Active Workspace 2.1, 2.2, or 2.3 to Active Workspace 2.4 and are not upgrading to the next major Teamcenter version, you follow a patching process. For example, if your initial environment is Teamcenter 9.1.i and your target environment is Teamcenter 9.1.j, you follow a patching process.

The following table provides examples of initial and target Active Workspace and Teamcenter versions for which the patching process is used.

<table>
<thead>
<tr>
<th>Initial version level</th>
<th>Target version level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Workspace 2.1 with Teamcenter 9.1.2.7</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4</td>
</tr>
<tr>
<td>Active Workspace 2.2 with Teamcenter 9.1.2.10</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4</td>
</tr>
<tr>
<td>Active Workspace 2.3 with Teamcenter 9.1.3.1</td>
<td>Active Workspace 2.4 with Teamcenter 9.1.3.4</td>
</tr>
<tr>
<td>Active Workspace 2.1 with Teamcenter 10.1.1.1</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
</tr>
<tr>
<td>Active Workspace 2.2 with Teamcenter 10.1.2.1</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
</tr>
<tr>
<td>Active Workspace 2.3 with Teamcenter 10.1.2.3</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
</tr>
</tbody>
</table>

Remove Active Workspace 2.1 features

Note: Complete the steps in this topic only if your initial Active Workspace version is 2.1. If your initial Active Workspace version is 2.2 or later, skip this topic.

In Active Workspace 2.2, new indexing and visualization components were introduced. You must remove the Active Workspace 2.1 Visualization Server Manager, Visualization Server Pool Assigner, and Active Workspace Indexer features before moving to Active Workspace 2.4. These features may be on separate servers or in combinations on servers. After moving to Active Workspace 2.4, you must reinstall these features.

1. Start Teamcenter Environment Manager (TEM) for the Active Workspace 2.1 server containing the features to be removed.
2. In the Maintenance panel, select Configuration Manager and click Next.
3. In the Configuration Maintenance panel, select Perform maintenance on an existing configuration and click Next.
4. In the Old Configuration panel, select configuration and click Next.
5. In the Feature Maintenance panel, select Add/Remove Features and click Next.
6. In the Features panel, clear the check box for the features to remove: Visualization Server Manager, Visualization Server Pool Assigner, and Active Workspace Indexer. Then click Next.
7. In the **Confirmation** panel, click **Start**.

8. When the process is complete, click **Close**.

**Remove Active Workspace 2.1 Dispatcher translators**

**Note**

Complete the steps in this topic only if your initial Active Workspace version is 2.1. If your initial Active Workspace version is 2.2 or later, skip this topic.

In Active Workspace 2.2, a new indexing translator was introduced. This requires you to remove the Active Workspace 2.1 translators (tcftsindexer_queryuids, tcftsindexer_tieexport, tcftsindexer_transform, and tcftsindexer_load) before moving to Active Workspace 2.4. After moving to Active Workspace 2.3, if you plan to use Dispatcher-based indexing, you must install the TcFtsIndexerTrans translator.

1. Start TEM for the Dispatcher Server on which the translators are installed.

2. In the **Maintenance** panel, select **Configuration Manager** and click **Next**.

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

4. In the **Old Configuration** panel, select the configuration and click **Next**.

5. In the **Feature Maintenance** panel, under **Dispatcher Server**, select **Modify Dispatcher Settings** and click **Next**.
6. In the **Teamcenter Installation Location** panel, the **TC_ROOT** and **TC_DATA** directories are populated. Make no changes and click **Next**.

7. In the **Flex License Client** panel, the license server information is populated. Make no changes and click **Next**.

8. In the **Teamcenter Administrative User** panel, type the user’s password and click **Next**.

9. In the **Dispatcher Components** panel, the values are populated with those defined at installation. Make no changes and click **Next**.

10. In the **Dispatcher Settings** panel, the values are populated with those defined at installation. Make no changes and click **Next**.

11. In the **Select Translators** panel, clear the check boxes for **TCFTS Indexer Translators**. Then click **Next**.
12. In the **Confirmation** panel, click **Start**.

13. When the process is complete, click **Close**.

**Patch Teamcenter**

Patch your environment to the supported Teamcenter patch level (either Teamcenter 10.1.4 or Teamcenter 9.1.3.4). Refer to all of the following *before* adding the Teamcenter patch:

- Readme file for the Teamcenter patch
• Readme file for the Active Workspace patch that you plan on adding

Because Active Workspace is installed in your environment, there may be steps that you must perform before you can add a new Teamcenter patch. If such steps are required, the readme file for the Active Workspace patch that you are moving to provides them.

• General patch instructions in the Teamcenter documentation
  
  o Windows Server Installation→Additional configuration and maintenance→Installing Teamcenter patches
  
  o UNIX and Linux Server Installation→Additional configuration and maintenance→Installing Teamcenter patches

Patch to Active Workspace 2.4

1. From GTAC, download the Active Workspace patch zip files:

   • If you are patching Active Workspace in a Teamcenter 9.1.3.4 environment, you download two files: Tcrelease_ActiveWorkspacerel-num_install.zip and Tcrelease_ActiveWorkspacerel-num_platform.zip.

   • If you are patching Active Workspace in a Teamcenter 10.1.4 environment, you download one file: Tcrelease_ActiveWorkspacerel-num_platform.zip.

2. If you are patching Active Workspace in a Teamcenter 10.1.4 environment, skip this step. If you are patching Active Workspace in a Teamcenter 9.1.3.4 environment, unzip Tcrelease_ActiveWorkspacerel-num_install.zip into the corporate server TC_ROOT\install directory.

3. For both Teamcenter 9.1.3.4 and Teamcenter 10.1.4 environments, unzip Tcrelease_ActiveWorkspacerel-num_platform.zip to a location on the corporate server.

4. If any server managers are running on the corporate server to be patched, shut them down before proceeding with the patch.

5. Start TEM for the corporate server.

6. In the Maintenance panel, select Updates Manager and click Next.
7. Complete the **Apply Updates** panel.

**Apply Updates**

Select an update to apply. A backup of the files being updated will be created prior to applying the update. If there are services currently running, they must be stopped prior to starting the update process.

- In the **Update kit location** box, type or browse to the location to which you unzipped `Tcrelease_ActiveWorkspaceRel-num_platform.zip`.
- In the **Backup directory** box, type or browse to the location in which backup files are to be created.
- In the **Original Media Location** box, type or browse to the location of the full Teamcenter release installation media, for example, Teamcenter 10.1.

TEM does not display this box if you are installing Active Workspace in a Teamcenter 9.1.3.4 environment.
- Click **Next**.
8. The **Status Message** popup window informs you that TEM will stop any running Teamcenter services in order to perform the update. Once the update is complete, these services are restarted. Any users currently on the system are dropped. Click **Close**.

9. In the **Operating System User** panel, type the password of the operating system account used to update Teamcenter and click **Next**.

10. In the **Optional Configuration Enhancements** panel, select **Yes** to install the enhancements and click **Next**.

![Optional Configuration Enhancements](image)

The installer has detected that there are additional enhancements that can be optionally installed as part of your 'tc' configuration. Click on the 'View Enhancement Info' button to read more about these enhancements and the benefits of each. Note that these enhancements also contain template updates. If you choose to install these enhancements, the installer will automatically handle installing the template updates as well.

**Optional Enhancements**

- New type constant AwpIDatasetTypeToBeIndexedInline
- Active Workspace 2.2 release
- Relationship viewer for Teamcenter business objects and their relations

If you would like to install these enhancements, select 'Yes' and click 'Next'. By selecting 'Yes', understand that all of the enhancements listed above will be installed.

- **Yes**
- **No**

11. In the **Teamcenter Administrative User** panel, type the user’s password and click **Next**.

12. In the **Indexing Engine User** panel, type the Solr administrator’s user name and password, and click **Next**. These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used).
13. In the **Confirmation** panel, click **Start**.

14. During the update, a **Status Message** popup window opens. It informs you that you must manually patch the **TC_DATA** directory and regenerate the **awc.war** file after TEM completes its updates. Click **Close**.

15. When the update is complete, click **Close** to close TEM.

16. Browse to the location to which you unzipped **TcRelease_AWCrel-num_platform.zip**. Under this location, browse to **tc**.

17. Unzip the **data.zip** file to the corporate server’s **TC_DATA** directory.

### Reinstall Active Workspace components

**Note**

Complete the steps in this topic only if your initial Active Workspace version is 2.1. If your initial Active Workspace version is 2.2 or later, skip this topic.

In Active Workspace 2.2, new indexing and visualization components were introduced. You must remove the Active Workspace 2.1 **Visualization Server Manager**, **Visualization Server Pool Assigner**, and **Active Workspace Indexer** features and the Active Workspace 2.1 translators (**tcftsindexer_queryuids**, **tcftsindexer_tieexport**, **tcftsindexer_transform**, and **tcftsindexer_load**) before moving to Active Workspace 2.4. These features may be on separate servers or in combinations on servers. After moving to Active Workspace 2.4, you must reinstall these features and, if Dispatcher-based indexing is used, install the **TcFtsIndexerTrans** translator.

1. On the **awc.war** file build machine, install the **Visualization Server Pool Assigner** feature. Perform this task only if Visualization is used with Active Workspace.

   See [Build the Java EE Client web application](#) for installing the new feature.

   Installing this feature rebuilds the **awc.war** file. Therefore, after installing this feature, you must redeploy the file.

   See [Deploy the Java EE Client web application](#) for deploying the **awc.war** file.

2. On the Dispatcher Server machine, install the **TcFtsIndexerTrans** translator. Perform this task only if Dispatcher-based indexing will be used.

   The four translators used in Active Workspace 2.1 are replaced by this single translator.

   See [Install Dispatcher translators](#).
3. On the indexer machine, install the **Indexer** feature on the indexer machine.  
   See *Install Indexer (TcFTSIndexer)* for installing the new feature.

4. On the Visualization server machine, install the **Visualization Server Manager** feature on the Visualization server machine. Perform this task only if Visualization is used with Active Workspace.  
   See *Install the Visualization Server Manager* for installing the new feature.

**Regenerate and redeploy the Active Workspace WAR file**

After patching, upgrading, or adding features to the Active Workspace WAR file, you must regenerate and redeploy it.

1. To regenerate the Active Workspace WAR file:
   a. Start TEM on the WAR file build machine.
   b. In the Maintenance panel, select **Configuration Manager** and click **Next**.
   c. In the Configuration Maintenance panel, select **Perform maintenance on an existing configuration** and click **Next**.
   d. In the Old Configuration panel, select the configuration and click **Next**.
e. In the **Feature Maintenance** panel, select **Update Active Workspace client settings** and click **Next**.
f. In the **Active Workspace Client Setting (J2EE)** panel, you can modify parameters as needed, but no changes are necessary if you are only regenerating the WAR file. Click **Next**.

![](image)

**Active Workspace Client Settings (J2EE)**

Specify Active Workspace client installation settings below.

- **Teamcenter 4-tier URL**: http://Config115644VM0:8030/tc
- **JDK Home**: C:\Program Files\Java\jdk1.7.0_17
- **Use as Bootstrap URLs**
  - **Bootstrap URLs**: http://Config115644VM0:4544
  - **Bootstrap Client IP**: 
- **Use Assigned FSC URLs**
  - **Assigned FSC URLs**: http://Config115644VM0:4544
- **Enable TcSS Support**
  - **Enable TcSS Support**
  - **TcSS Application ID**: Teamcenter
  - **TcSS Login URL**: 

  ![Advanced...]


g. In the **Confirmation** panel, click **Start**.

h. When the installation is complete, click **Close**.

2. Note that, for Active Workspace 2.4, the Java EE Client WAR file is created in `TC_ROOT\aws2\stage\out\awc.war`. In older versions of Active Workspace, the Java EE Client WAR file is created in `TC_ROOT\aws2\stage\Teamcenter\target\awc.war`.

Siemens PLM Software recommends that you delete the directory, `TC_ROOT\aws2\stage\Teamcenter\target`, that contains the old WAR file in order to avoid confusion.

3. To redeploy the Active Workspace WAR file, see *Deploy the Java EE Client web application*.

When deploying the new WAR file, keep in mind that it is created in `TC_ROOT\aws2\stage\out\awc.war`.

**Update Indexer settings**

After patching Teamcenter and Active Workspace, you must update the Indexer settings.
1. Start TEM for the Indexer.

2. In the **Maintenance** panel, select **Configuration Manager**, and then click **Next**.

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

4. In the **Old Configuration** panel, select the configuration in which the Indexer is installed, and then click **Next**.

5. In the **Feature Maintenance** panel, under **Active Workspace Indexer**, select **Update Active Workspace Indexer settings**, and then click **Next**.

6. In the **Teamcenter Administrative User** panel, type the user's password in the **Password** box, and then click **Next**.

7. In the **Active Workspace Indexer Settings** panel, make no changes. Click **Next**.

8. In the **Active Workspace Indexer Type Selections** panel, make no changes. Click **Next**.

9. In the **Confirmation** panel, click **Start**.

10. When the installation is complete, click **Close**.

**Initiate Solr credentials**

If your initial Active Workspace environment was 2.1 or 2.2, you need to initiate the credentials for the Solr administrative user after patching to the target Teamcenter and Active Workspace versions. This is not necessary if your initial environment was Active Workspace 2.3 or later.

There are four places this needs to be done. The credentials entered in each of these locations must be identical.

- Server Extensions
- Dispatcher Server translators (if Dispatcher-based indexing is used)
- Indexing Engine
- Indexer

**Enter Solr credentials for the Server Extensions**

1. Start TEM for the corporate server on which the Server Extensions features are installed.
2. In the **Maintenance** panel, select **Configuration Manager**, and then click **Next**.
3. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**, and then click **Next**.
4. In the **Old Configuration** panel, select the configuration in which the Server Extension features are installed, and then click **Next**.
5. In the **Feature Maintenance** panel, under **Data Model**, select **Update Indexing Engine user credentials**, and then click **Next**.

6. In the **Indexing Engine User** panel, type the Solr administrative user's name and password. These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used). After typing the user name and password, click **Next**.
7. In the **Confirmation** panel, click **Start**.

8. When the installation is complete, click **Close**.

**Enter Solr credentials for the Dispatcher Server**

1. Start TEM for the Dispatcher Server.

2. In the **Maintenance** panel, select **Configuration Manager**, and then click **Next**.

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

4. In the **Old Configuration** panel, select the configuration in which the Dispatcher Server is installed, and then click **Next**.

5. In the **Feature Maintenance** panel, under **Dispatcher Server**, select **Modify Dispatcher Settings**, and then click **Next**.
Feature Maintenance

Please select the feature to configure below and click the Next button.

Teamcenter
- Add/Remove Features
Client Communication System
- Use Configurations and Environments
- Modify Configurations
- Modify 4-tier server settings
- Modify FCC Parent settings
- Remove Configurations
Dispatcher Server
- Modify Dispatcher Settings
Active Workspace Object Data Translator
- Update Active Workspace Indexer settings
Active Content Structure Translator
- Modify Active Content Structure translator settings

6. In the Operating System User panel, type the user's password in the Password and Confirm boxes, and then click Next.

7. In the Teamcenter Installation Location panel, make no changes. Click Next.

8. In the Flex license Client panel, make no changes. Click Next.

9. In the Teamcenter Administrative User panel, type the user's password in the Password box, and then click Next.

10. In the Dispatcher Components panel, make no changes. Click Next.

11. In the Dispatcher Settings panel, make no changes. Click Next.

12. In the Select Translators panel, make no changes. Click Next.

13. In the Translator Settings panel, if the ReqMgmtWordToHtmlTrans translator is installed, you must enter the user's password. Make no other changes. Click Next.
14. In the **Active Workspace Translator Selection** panel, make no changes. Click **Next**.

15. In the **Indexing Engine User** panel, type the Solr administrative user’s name and password. These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used). After typing the user name and password, click **Next**.

16. In the **Confirmation** panel, click **Start**.

17. When the installation is complete, click **Close**.

**Enter Solr credentials for the Indexing Engine**

1. Start TEM for the Indexing Engine.

2. In the **Maintenance** panel, select **Configuration Manager**, and then click **Next**.

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

4. In the **Old Configuration** panel, select the configuration in which the Indexing Engine is installed, and then click **Next**.
5. In the Feature Maintenance panel, under Active Workspace Indexing Engine, select Update Indexing Engine user credentials, and then click Next.

6. In the Indexing Engine User panel, type the Solr administrative user's name and password. These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used). After typing the user name and password, click Next.

7. In the Confirmation panel, click Start.

8. When the installation is complete, click Close.

Enter Solr credentials for the Indexer

1. Start TEM for the Indexer.

2. In the Maintenance panel, select Configuration Manager, and then click Next.

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

4. In the Old Configuration panel, select the configuration in which the Indexer is installed, and then click Next.

5. In the Feature Maintenance panel, under Active Workspace Indexer, select Update search engine user credentials, and then click Next.
6. In the **Indexing Engine User** panel, type the Solr administrative user's name and password. These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used). After typing the user name and password, click **Next**.

7. In the **Confirmation** panel, click **Start**.

8. When the installation is complete, click **Close**.

### Remerge Solr and Teamcenter schemas and reindex

After patching to the target Active Workspace and Teamcenter releases, the Teamcenter and Solr schemas will be out of sync; it is necessary that you remerge the schemas.

1. Remerge the Solr and Teamcenter schemas. See *Merge the Teamcenter and Solr schemas*.

2. Test indexer connectivity. See *Test TcFTSIndexer connectivity*.

3. Reindex data. See *Run initial index of object data*.

### Upgrading to Active Workspace 2.4

If you are moving from Active Workspace 2.1, 2.2, or 2.3 with Teamcenter 9.1.x to Active Workspace 2.4 with Teamcenter 10.1.4, you follow an upgrade process. The following table provides examples of initial and target Active Workspace and Teamcenter versions for which the patching process is used.
Remove Active Workspace 2.1 features

Note

Complete the steps in this topic only if your initial Active Workspace version is 2.1. If your initial Active Workspace version is 2.2 or later, skip this topic.

In Active Workspace 2.2, new indexing and visualization components were introduced. You must remove the Active Workspace 2.1 Visualization Server Manager, Visualization Server Pool Assigner, and Active Workspace Indexer features before moving to Active Workspace 2.4. These features may be on separate servers or in combinations on servers. After moving to Active Workspace 2.4, you must reinstall these features.

1. Start Teamcenter Environment Manager (TEM) for the Active Workspace 2.1 server containing the features to be removed.

2. In the Maintenance panel, select Configuration Manager and click Next.

3. In the Configuration Maintenance panel, select Perform maintenance on an existing configuration and click Next.

4. In the Old Configuration panel, select configuration and click Next.

5. In the Feature Maintenance panel, select Add/Remove Features and click Next.

6. In the Features panel, clear the check box for the features to remove: Visualization Server Manager, Visualization Server Pool Assigner, and Active Workspace Indexer. Then click Next.

<table>
<thead>
<tr>
<th>Initial version level</th>
<th>Target version level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Workspace 2.1 with Teamcenter 9.1.2.7</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
</tr>
<tr>
<td>Active Workspace 2.2 with Teamcenter 9.1.2.10</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
</tr>
<tr>
<td>Active Workspace 2.3 with Teamcenter 10.1.2.3</td>
<td>Active Workspace 2.4 with Teamcenter 10.1.4</td>
</tr>
</tbody>
</table>
7. In the **Confirmation** panel, click **Start**.

8. When the process is complete, click **Close**.

### Remove Active Workspace 2.1 Dispatcher translators

**Note**

Complete the steps in this topic only if your initial Active Workspace version is 2.1. If your initial Active Workspace version is 2.2 or later, skip this topic.

In Active Workspace 2.2, a new indexing translator was introduced. This requires you to remove the Active Workspace 2.1 translators (tcftsindexer_queryuids, tcftsindexer_tieexport, tcftsindexer_transform, and tcftsindexer_load) before moving to Active Workspace 2.4. After moving to Active Workspace 2.3, if you plan to use Dispatcher-based indexing, you must install the TcFtsIndexerTrans translator.

1. Start TEM for the Dispatcher Server on which the translators are installed.

2. In the **Maintenance** panel, select **Configuration Manager** and click **Next**.

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

4. In the **Old Configuration** panel, select the configuration and click **Next**.

5. In the **Feature Maintenance** panel, under **Dispatcher Server**, select **Modify Dispatcher Settings** and click **Next**.
6. In the **Teamcenter Installation Location** panel, the **TC_ROOT** and **TC_DATA** directories are populated. Make no changes and click **Next**.

7. In the **Flex License Client** panel, the license server information is populated. Make no changes and click **Next**.

8. In the **Teamcenter Administrative User** panel, type the user’s password and click **Next**.

9. In the **Dispatcher Components** panel, the values are populated with those defined at installation. Make no changes and click **Next**.

10. In the **Dispatcher Settings** panel, the values are populated with those defined at installation. Make no changes and click **Next**.

11. In the **Select Translators** panel, clear the check boxes for **TCFTS Indexer Translators**. Then click **Next**.
12. In the Confirmation panel, click Start.

13. When the process is complete, click Close.

Upgrade to Active Workspace 2.4 and Teamcenter 10.1.4

If the target Teamcenter version to which you are moving is a later major release than your initial environment, you follow an upgrade process. For example, if your initial environment is Teamcenter 9.1.x and your target environment is Teamcenter 10.1.x, you follow an upgrade process.

If your initial version level is Teamcenter 9.1.x with Active Workspace 2.1, 2.2, or 2.3 and your target version level is Teamcenter 10.1.4 with Active Workspace 2.4, you must upgrade to get to the target version. This must be done in a single upgrade process. You cannot upgrade to Teamcenter 10.1.4
in one process and then patch to Active Workspace 2.4 in a separate process; this causes the upgrade to fail.

Because Active Workspace is installed in your environment, there may be steps that you must perform before you can upgrade. If such steps are required, the readme file for the Active Workspace patch that you are moving to provides them.

The following steps illustrate an upgrade of a simple environment. Siemens PLM Software recommends that you refer to the Teamcenter Upgrade Guide before attempting an upgrade.

The panels you see in TEM are dependent on the features that you have installed. You may see additional panels in your environment.

1. From GTAC (http://support.industrysoftware.automation.siemens.com/gtac.shtml), download the following installation files:
   - Teamcenter 10.1 base version
     There are two ZIP files, Tc10.1.0_platform_1_of_2.zip and Tc10.1.0_platform_2_of_2.zip. Unzip these two files into a single location.
   - Teamcenter service pack or patch (Tc-relnum_platform.zip)
     For example, Tc10.1.4_wntx64.zip
     Unzip this file into its own location.
   - Active Workspace patch (Tc-relnum_ActiveWorkspace-relnum_platform.zip)
     For example, Tc10.1.4_ActiveWorkspace2.4_wntx64.zip
     Unzip this file into its own location.

2. Start TEM from the unzipped Active Workspace installer.

3. In the Installer Language window, select a language and click OK.

4. In the Welcome to Teamcenter panel, click Upgrade.
5. In the **Media Locations** panel, provide the locations of the installation media and ensure that the paths in the **Update Location** table are correct.

   ![Media Locations Panel]

   **Media Locations**
   Enter the location of the installation media for Teamcenter 10.1.0 in the 'Original Media Location' field provided below. In the 'Update Location' table, click the 'Add' button to enter the path to additional patch locations that you would like to include for this installation.

   **Original Media Location**
   C:\install_images\Tc10.1.0_win64

   **Update Location**
   C:\install_images\Tc10.1.2_patch_3_win64\win64
   C:\install_images\Tc10.1.2.3_ActiveWorkspace2.3_win64\win64

   ![Update Location Table]

   a. In the **Original Media Location** box, provide the location of the unzipped Teamcenter 10.1 installation media, Tc10.1.0\platform.

   b. In the **Update Location** table, note that the Active Workspace patch, Tc\rel-num\ActiveWorkspace\rel-num\platform\wntx64, is already listed. This is the location from which you started TEM.

   c. Click **Browse** and select the unzipped Teamcenter patch, Tc\rel-num\platform\wntx64.

   d. The locations in the **Update Location** table must be in the correct order. You can reorder the table by selecting a location and clicking **Shift Up** or **Shift Down**. The correct order is:

      - Teamcenter patch, Tc\rel-num\platform\wntx64
      - Active Workspace patch, Tc\rel-num\ActiveWorkspace\rel-num\platform\wntx64

   e. Once the installation media locations are in the correct order in the **Update Location** panel, click **Next**.

6. In the **Old Application Root** panel, provide the location of TC_ROOT for the server to be upgraded and then click **Next**.

7. In the **Old Configuration** panel, select the configuration to be upgraded and click **Next**.

8. In the **Configuration** panel, provide an **ID** and a **Description** and then click **Next**.

9. In the **New Application Root** panel:
Check that no features have a status of **WARNING**. If any do, click **Browse** and navigate to the appropriate feature file.

- In the **New Application Root Directory** box, provide the location in which to install the target **TC_ROOT**.

10. In the **Operating System User** panel, provide the password of the operating system user that is running the upgrade and then click **Next**.

11. In the **Teamcenter Administrative User** panel, provide the password and then click **Next**.

12. In the **Upgrade Information** panel:
• Verify that the **Old TC_DATA Location** box has the correct value.

• Provide a location for the new **TC_DATA** in the **New TC_DATA Location**.

• Enter the database user name and password.

13. In the **Confirm** window, click **Confirm**.

14. In the **Upgrade Database Features** panel, verify that all features have a status of **Upgrade** and the **Path** value is correct. To correct any issues, click **Browse** and locate the appropriate feature file.
15. In the **Upgrade Options** panel, select either:

   - **Stop database upgrade at first error**
     
     This option stops Teamcenter upgrade when the first error occurs. Siemens PLM Software strongly recommends this option when upgrading production systems.

   - **Stop database upgrade after accumulating all errors**
     
     This option allows the database upgrade to continue when an error is encountered, and report all errors at the end of the upgrade process. Siemens PLM Software strongly recommends using this option only when upgrading a test environment. Errors during database upgrade could render the database unusable.

16. In the **Flex License Client** panel, enter the license server and license file information

17. In the **Password Security** panel, click **Next**.

18. In the **TcServer Character Encoding Settings** panel, click **Next**.

19. In the **Active Workspace Client Settings (J2EE)** panel, type the **Teamcenter 4-tier URL**. This is the URL for the deployed J2EE Teamcenter web application. The format of this URL is:

   ```
   protocol://host:port/ear-file-name
   ```

   `protocol` can be **http** or **https**.

   `host` is the machine running the web application server on which the Teamcenter web application is deployed.

   `port` is the port used by the web application server.
ear-file-name is the name of the Teamcenter web application ear file. The default is tc.

20. In the Active Workspace Client Settings (J2EE) panel, the JDK Home box is the installation location of a supported version of 64-bit JDK 1.7. This box will be prepopulated with the value of the JAVE_HOME system environment variable. You can change this to a different JDK as needed.

For supported versions of third-party software, refer to the certification matrix on GTAC:
http://support.industrysoftware.automation.siemens.com/gtac.shtml
21. The Active Workspace client uses FMS to download and upload files. You define the FSC servers that will be used by choosing one of the following:

- **Use as Bootstrap URLs**: FSC servers are automatically assigned by the FMS system. This is the most common scenario.

  In the **Bootstrap URLs** box, enter a comma-separated list of one or more FMS bootstrap URLs. Optionally, provide the **Bootstrap Client IP** value to be used for the assignment.

  The **Bootstrap Client IP** value should only be used when there are proxies that might hide the actual requestor’s IP address from the FSC servers, or if you want the assignment to be performed based on a particular IP and not that of the requestor. (The client/requestor is the host on which the awc.war file is deployed.) The list of Bootstrap URLs should contain well-known FSCs that can be consulted for the actual FSC assignment. The assignment is delegated to the FMS network, which uses mapping logic and configuration data to determine the FSCs that should be used by the requestor (refer to the **clientmap** configuration in the FMS Master configuration file). The **client mapping** logic uses an IP address to determine the assignment. By default, the IP address from the HTTP connection of the requestor will be used unless a **Bootstrap Client IP** value is provided.

- **Use Assigned FSC URLs**: FSC servers are explicitly declared.
This should only be selected if you want explicit control of the FSCs used. In the **Assigned FSC URLs** box, enter a comma-separated list of one or more FSC URLs. The URL values entered will be directly used for file operations.

22. Select **Enable TcSS Support** to enable Teamcenter Security Services. Enter the Teamcenter Security Services application ID and logon URL.

For details about configuring Teamcenter Security Services, see *Security Services Installation/Customization* in the Teamcenter collection.

When configuring Active Workspace for Teamcenter Security Services, be sure to only install the language packs for the Security Services that Active Workspace supports.

If Active Workspace is deployed on a different URL, you must configure Teamcenter Security Services with multiple application IDs. For more information, see the **readme** file for your Teamcenter Security Services release.
23. In the **Indexing Engine User** panel, type the Solr administrator's user name and password. This is the user and password defined when installing the **Indexing Engine (Solr)**. These credentials must match for the Indexing Engine, the Indexer, Server Extensions, and the TcFTS Indexer Translator (if used). After typing the user name and password, click **Next**.

24. In the **Indexing Engine Configuration** panel, provide the path to the directory that contains the Solr schema files. This allows the Solr and Teamcenter schemas to be automatically merged during installation of the Indexing Engine. If you leave this box blank, the schemas are not automatically merged; you must manually merge them after installing the Indexing Engine.
The Solr schema files are created on the corporate server when installing the Server Extensions. There are two schema files located in `TC_DATA\fts\solr_schema_files`:

- `TC_ACE_SOLR_SCHEMA.xml`
- `TC_SOLR_SCHEMA.xml`

If you have not yet installed the Server Extensions on the corporate server, you can complete the installation of the Indexing Engine, but the schemas are not automatically merged.

25. In the **Pre-Upgrade Diagnostics** panel, provide a logging directory, click **Run**, and verify that all diagnostic tests have a status of **Passed**.

26. In the **Confirmation** panel, click **Start**.

27. When the upgrade is complete, click **Close**.
Reinstall Active Workspace components

**Note**

Complete the steps in this topic only if your initial Active Workspace version is 2.1. If your initial Active Workspace version is 2.2 or later, skip this topic.

In Active Workspace 2.2, new indexing and visualization components were introduced. You must remove the Active Workspace 2.1 **Visualization Server Manager**, **Visualization Server Pool Assigner**, and **Active Workspace Indexer** features and the Active Workspace 2.1 translators (`tcftsindexer_queryuids`, `tcftsindexer_tieexport`, `tcftsindexer_transform`, and `tcftsindexer_load`) before moving to Active Workspace 2.4. These features may be on separate servers or in combinations on servers. After moving to Active Workspace 2.4, you must reinstall these features and, if Dispatcher-based indexing is used, install the **TcFtsIndexerTrans** translator.

1. On the awc.war file build machine, install the **Visualization Server Pool Assigner** feature. Perform this task only if Visualization is used with Active Workspace.

   See [Build the Java EE Client web application](#) for installing the new feature.

   Installing this feature rebuilds the awc.war file. Therefore, after installing this feature, you must redeploy the file.

   See [Deploy the Java EE Client web application](#) for deploying the awc.war file.

2. On the Dispatcher Server machine, install the **TcFtsIndexerTrans** translator. Perform this task only if Dispatcher-based indexing will be used.

   The four translators used in Active Workspace 2.1 are replaced by this single translator.

   See [Install Dispatcher translators](#).

3. On the indexer machine, install the **Indexer** feature on the indexer machine.

   See [Install Indexer (TcFTSIndexer)](#) for installing the new feature.

4. On the Visualization server machine, install the **Visualization Server Manager** feature on the Visualization server machine. Perform this task only if Visualization is used with Active Workspace.

   See [Install the Visualization Server Manager](#) for installing the new feature.

Remerge Solr and Teamcenter schemas and reindex

After patching to the target Active Workspace and Teamcenter releases, the Teamcenter and Solr schemas will be out of sync; it is necessary that you remerge the schemas.

1. Remerge the Solr and Teamcenter schemas. See [Merge the Teamcenter and Solr schemas](#).

2. Test indexer connectivity. See [Test TcFTSIndexer connectivity](#).

3. Reindex data. See [Run initial index of object data](#).
Regenerate and redeploy the Active Workspace WAR file

After patching, upgrading, or adding features to the Active Workspace WAR file, you must regenerate and redeploy it.

1. To regenerate the Active Workspace WAR file:
   a. Start TEM on the WAR file build machine.
   b. In the Maintenance panel, select Configuration Manager and click Next.
   c. In the Configuration Maintenance panel, select Perform maintenance on an existing configuration and click Next.
   d. In the Old Configuration panel, select the configuration and click Next.
   e. In the Feature Maintenance panel, select Update Active Workspace client settings and click Next.

Feature Maintenance

Please select the feature to configure below and click the Next button.

- Modify 4-tier server settings
- Modify FCC Parent settings
- Remove Configurations
- 2-tier server configurations
- Modify 2-tier server settings
- Teamcenter Rich Client 2-tier
  - Modify settings
- Dispatcher Server
  - Modify Dispatcher Settings
- Dispatcher Client
  - Modify Dispatcher Client Settings
- **Active Workspace Client (Java EE)**
  - Update Active Workspace client settings
- Doc Model
  - Update Indexing Engine user credentials
- Active Workspace Indexing Engine
  - Update Indexing Engine user credentials
f. In the **Active Workspace Client Setting (J2EE)** panel, you can modify parameters as needed, but no changes are necessary if you are only regenerating the WAR file. Click **Next**.

![Active Workspace Client Settings (J2EE)](image)

**Active Workspace Client Settings (J2EE)**
Specify Active Workspace client installation settings below.

- **Teamcenter 4-tier URL**: http://Config115644VMO:8080/tc
- **JDK Home**: C:\Program Files\Java\jdk1.7.0_17 ...
- **Use as Bootstrap URLs**
  - **Bootstrap URLs**: http://Config115644VMO:4544
  - **Bootstrap Client IP**
- **Use Assigned FSC URLs**
  - **Assigned FSC URLs**: http://Config115644VMO:4544
- **Enable TcSS Support**
  - **Enable TcSS Support**
  - **TcSS Application ID**: Teamcenter
  - **TcSS Login URL**

![Advanced...]

---

g. In the **Confirmation** panel, click **Start**.

h. When the installation is complete, click **Close**.

2. Note that, for Active Workspace 2.4, the Java EE Client WAR file is created in `TC_ROOT\aws2\stage\out\awc.war`. In older versions of Active Workspace, the Java EE Client WAR file is created in `TC_ROOT\aws2\stage\Teamcenter\target\awc.war`.

Siemens PLM Software recommends that you delete the directory, `TC_ROOT\aws2\stage\Teamcenter\target`, that contains the old WAR file in order to avoid confusion.

3. To redeploy the Active Workspace WAR file, see *Deploy the Java EE Client web application*.

When deploying the new WAR file, keep in mind that it is created in `TC_ROOT\aws2\stage\out\awc.war`.
Upgrade the ACCT_TABLE database table

If you are performing site consolidation in an environment that uses Active Workspace and are upgrading to Teamcenter 9.1.3.4 or 10.1.4, you must manually upgrade the ACCT_TABLE database table after upgrading the corporate server. Both site consolidation unconfigured synchronization and object data indexing use this table. Starting with Teamcenter 9.1.3.4 and 10.1.4, the table has a new column, app_id. During the corporate server upgrade, data collisions exist and the new app_id column is not added. After the upgrade, you must run the txml_acct_table_upgrade utility to add the new column to the table.

More information is provided in the Teamcenter help collection in Site Consolidation→Preparing for site consolidation→Deploying Teamcenter components and Site Consolidation→Preparing for site consolidation→Upgrade the ACCT_TABLE database table.
Part II: Configuration and administration
Chapter 4: Configuration

Modifying Active Workspace settings

You can use TEM in maintenance mode to change many Active Workspace settings.

1. Launch TEM from your installed environment.

2. On the Maintenance panel, select Configuration Manager and then click Next.

3. On the Configuration Maintenance panel, select Perform maintenance on an existing configuration and then click Next.

4. On the Old Configuration panel, select the configuration where the Active Workspace components are installed and then click Next.

   The Feature Maintenance panel appears. The options in the panel vary based on the components installed in the selected configuration.

5. Select one of the following options:
   - **Update Active Workspace Indexer settings** (under the Active Workspace Indexer group)
     Updates the Active Workspace search indexer settings.
   - **Update Active Workspace client settings** (under the Active Workspace Client group)
     Updates the Active Workspace client settings and regenerates the awc.war file based on the new settings.
   - **Update Visualization Pool Assigner settings** (under the Visualization Pool Assigner group)
     Updates the Visualization Pool Assigner settings and regenerates the awc.war based on the new settings.
   - **Update Visualization Server Manager settings** (under the Visualization Server Manager group)
     Updates the Visualization Server Manager settings and regenerates the awc.war file based on the new settings.
   - **Modify Dispatcher settings** (under the Dispatcher Server group)
     Updates the Dispatcher Server translator settings.

   **Note**

   To update the Solr URL, modify the **AWS_FullTextSearch_Solr_URL** preference.
To update the Solr username and password, you must update the **Indexing Engine User** panel in several locations.

These credentials must match for the Solr indexing engine, the TcFTS Indexer, Server Extensions, and the TcFTS Indexer Translator if Dispatcher is used.

- Under **Data Model→Update Indexing Engine** user credentials.
  
  Use this to update the Solr credentials for each Teamcenter server with Server Extensions features installed.

- Under **Active Workspace Indexer→feature.awsftsclient.seusermaint**.
  
  This is for the TcFTS indexer.

- Under **Active Workspace Indexing Engine→Update Active Workspace Indexing Engine user credentials**.
  
  This is for the Solr indexing engine. This is the option changes the credentials in Solr.

- If you use Dispatcher services, the panel is accessed from **Dispatcher Server→Modify Dispatcher Settings**.

6. Click **Next** and navigate through the remaining panels to perform the update.

   The panels that display are the same as those presented during the initial installation of Active Workspace.

   For details on these panels, refer to the **Installation** topics.

### Configuring the interface

#### Configure the User Assistance tile

You must configure the **User Assistance** tile to enable the display of the **Active Workspace User Assistance** documentation.

You can choose to display any one of the available localized versions of the **Active Workspace User Assistance** documentation.

This configuration must be performed even to display the English version of the **Active Workspace User Assistance** documentation.

1. Contact GTAC for the location of the **UserAssistance.pdf** file for the desired locale. Save a copy of the file to a local machine.

2. In Teamcenter, locate the **awc.war** file generated by Teamcenter Environment Manager (TEM).

3. Open the **awc.war** file with a ZIP application (such as 7-Zip or WinZip).
4. Within the `awc.war` file, locate the `UserAssistance.pdf` file in the following location, where `locale-directory` is the directory for your desired language.

    `awc.war\thinclient\docs\locale-directory`

5. Replace the existing PDF file with the `UserAssistance.pdf` file for the desired locale that you received from GTAC.

6. Click OK and exit the archive.

7. Redeploy the `awc.war` file.

   **Note**

   The User Assistance tile must be configured in this way any time the `awc.war` file is rebuilt and deployed.

### Enable a custom business object in Active Workspace

1. Perform the following steps to create the business object in the Business Modeler IDE:
   a. Import the Active Workspace (`aws2`) template into your Business Modeler IDE project.
   b. Create the new business object and create custom properties on the business object.
   c. Use the Operation Descriptor tab to ensure the custom properties appear on the creation page in Active Workspace.
   d. Ensure that the custom business object and its properties are included in searches by applying the `Awp0SearchIsIndexed` business object constant and property constant.

      If you have installed Active Content Structure and you want to make the custom business object available on the Content tab, add the custom business object to the `Awb0SupportsStructure` global constant.

   e. Install the custom template to the server.

      For details about using the Business Modeler IDE, see Business Modeler IDE in the Teamcenter help collection.

2. Update the search indexer by merging the Teamcenter schema with the Solr schema and reindexing.

3. Add the custom business object type to the `AWC_DefaultCreateTypes` preference.

4. If you want unique page layout for the business object, set up style sheets for the business object’s create page, summary page, and information panel.

   For details about style sheets, see Active Workspace Customization.

### Configuring notification attributes

Active Workspace provides clear and consistent notifications of alerts, errors, and warnings. You can, however, edit the `noty.css` file to modify the notification attributes.
The `noty.css` file is located in `awc.war\thinclient\lib\noty\css`.

## Configure table columns

Active Workspace provides two utilities, `export_uiconfig` and `import_uiconfig`, that let administrators specify table column configurations.

### Note

Column configuration using the `export_uiconfig` and `import_uiconfig` utilities is currently limited to My Changes and Inbox tables.

1. Use the `export_uiconfig` utility to generate an XML file containing the column configuration used by Active Workspace.

   The XML attribute URI in the exported file refers to the table corresponding to the column configuration. A list of URIs and supported locations follows:

<table>
<thead>
<tr>
<th>URI</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>fnd0alltasks</td>
<td>All in Inbox</td>
</tr>
<tr>
<td>fnd0mytasks</td>
<td>My Tasks</td>
</tr>
<tr>
<td>fnd0surrogatescope</td>
<td>Surrogate Tasks</td>
</tr>
<tr>
<td>fnd0Inbox</td>
<td>Team, Tracking, and any custom defined tabs added under Inbox</td>
</tr>
<tr>
<td>fnd0Change</td>
<td>All tabs under Changes</td>
</tr>
</tbody>
</table>

2. Edit the XML file to set the column configurations for Active Workspace inbox tables and change tables, by group, role, and sublocation as indicated by `ClientScope`.

3. Use the `import_uiconfig` to import the edited XML file.

   You can use the `export_uiconfig` utility again to generate an XML file containing the imported column configuration for site, group, and role.

### Example 1, configure the My Changes table display:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Import>
  <Client abbreviation="AWClient" name="AWClient">
    <ClientScope hostingClientName="" name="Change" url="fnd0Change">
      <ColumnConfig columnConfigId="changeColConfig" sortBy="1" sortDirection="Ascending">
        <ColumnDef objectType="ChangeItemRevision" propertyName="awp0Item_item_id" width="100"/>
        <ColumnDef objectType="ChangeItemRevision" propertyName="item_revision_id" width="100"/>
        <ColumnDef objectType="ChangeItemRevision" propertyName="object_name" width="100"/>
        <ColumnDef objectType="ChangeItemRevision" propertyName="CMMaturity" width="100"/>
        <ColumnDef objectType="ChangeItemRevision" propertyName="creation_date" width="100"/>
        <ColumnDef objectType="ChangeItemRevision" propertyName="last_mod_date" width="100"/>
      </ColumnConfig>
    </ClientScope>
  </Client>
</Import>
```

The following image shows the specifications in the preceding example.
Example 2, configure the Inbox table display:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Import>
  <Client abbreviation="AWClient" name="AWClient">
    <ClientScope hostingClientName="" name="Inbox" uri="fnd0Inbox">
      <ColumnConfig columnConfigId="inboxColConfig" sortBy="1" sortDirection="Ascending">
        <ColumnDef objectType="WorkspaceObject" propertyName="object_name" width="100"/>
        <ColumnDef objectType="EPMTask" propertyName="job_name" width="100"/>
        <ColumnDef objectType="EPMTask" propertyName="fnd0Assignee" width="100"/>
        <ColumnDef objectType="EPMTask" propertyName="fnd0WorkflowInitiator" width="100"/>
        <ColumnDef objectType="EPMTask" propertyName="due_date" width="100"/>
        <ColumnDef objectType="WorkspaceObject" propertyName="last_mod_date" width="100"/>
      </ColumnConfig>
    </ClientScope>
  </Client>
</Import>
```

For information about these utilities, see `export_uiconfig` and `import_uiconfig` in the Utilities section.

For information about end-user column configuration, see Active Workspace User Assistance.

**Configure the .war file generation for other locales**

You can generate the `awc.war` file to support the following language locales: de, en, es, fr, it, ja_JP, ko_KR, pl_PL, pt_BR, ru_RU, zh_CN, and zh_TW.

1. Run Teamcenter Environment Manager (TEM).
2. Select Perform maintenance on an existing configuration and click Next.
3. In the Feature Maintenance panel, choose Update Active Workspace client settings.
Updating the Active Workspace client settings

4. Click Next.

The Active Workspace Client Settings panel is displayed.

5. In the Active Workspace Client Settings panel, click Advanced.

The Additional Client Settings panel is displayed.
6. Select the languages you want to include in the generated WAR file and click **OK**.

7. Click **Next** in the **Active Workspace Client Settings** panel to generate the **awc.war** file.

8. Deploy the generated **awc.war** file into your web application server (for example, in WebLogic, JBoss, or WebSphere).

   **Note**

   This default WAR file requires no compile step and should be leveraged by customers who do not require the customization of Active Workspace and use the provided languages. If you require additional or other combination of languages, general Active Workspace customization, or a smaller WAR file, you can generate a new WAR file.

---

### Configuring search

Configure the search user interface

**Configure search suggestions**

When performing a search in Active Workspace, the system displays suggested search values as you type your text in the search box. By default, keywords that make up at least .5 percent of the indexed data are included as search suggestions. For example, if there are 1000 indexed items, a keyword must exist at least five times before it is used as a suggestion. You can change the default value as follows.

1. Navigate to the **solrconfig.xml** file in your Solr installation directory, for example, `TC_ROOT\solr-version\awc\solr\collection1\conf\solrconfig.xml`

2. Scroll to the bottom of the **solrconfig.xml** file and look for the following line:

   ```xml
   <float name=threshold" tag="float>0.005</float>
   ```
Chapter 4: Configuration

3. Modify the **0.005** value as desired.

   To lessen the restriction (that is, create more search suggestions), decrease the value (for example, change it to **0.003**). To tighten the restriction (create less search suggestions), increase the value (for example, change it to **0.008**).

4. Save the file.

5. Reboot Solr.

To turn off the search suggestion feature, you must edit the `solrconfig.xml` file in the Solr installation directory.

1. Open the `solrconfig.xml` file in a text editor.

2. Near the end of the file, locate the `tcfssuggest` configuration section.

   ```xml
   <requestHandler class="org.apache.solr.handler.component.SearchHandler" name="/tcfssuggest">
     <!-- Request handler for Teamcenter search suggestions -->
     <lst name="defaults">
       <str name="spellcheck">true</str>
       <str name="spellcheck.dictionary">tcfssuggest</str>
       <str name="spellcheck.onlyMorePopular">true</str>
       <str name="spellcheck.count">5</str>
       <str name="spellcheck.collate">true</str>
     </lst>
     <arr name="components">
       <str>tcfssuggest</str>
     </arr>
   </requestHandler>
   ```

3. Comment out or remove the following line:

   ```xml
   <str>tcfssuggest</str>
   ```

4. Save and close the file.

5. Restart Solr.

**Define search prefilters**

Search prefilters appear above the search box and are properties used to filter data at run-time when you perform a search. Click a prefilter to see a context menu of search terms to filter. In the following example, **Select Owner** and **Select Type** are search prefilters.

![Search prefilters](image)

**Search prefilters**

You can define different prefilters based on user, role, or group using the `AWS_SearchPreFilter_Property1` and `AWS_SearchPreFilter_Property2` preferences.

In the example, the **Select Owner** search filter results from setting the `AWS_SearchPreFilter_Property1` preference with the following value:
The **Select Type** search filter results from the following value in the `AWS_SearchPreFilter_Property2` preference:

```
WorkspaceObject.object_type
ItemRevision
DocumentRevision
```

When a user uses a prefilter on a search, the selections are saved to the `AWS_SearchPreFilter_Property1_SelectedValue` and `AWS_SearchPreFilter_Property2_SelectedValue` preferences. When the user logs off and then back on, the prefilters are retrieved from these preferences and displayed in the client.

**Search for common words**

You may notice as you search that common words such as *the*, *and*, and *a* are ignored in searches. These common words are not indexed for searching in order to reduce the overall size of the search index. The list of words that are ignored by default can be found in the Solr documentation:

https://wiki.apache.org/solr/AnalyzersTokenizersTokenFilters#solr.StopFilterFactory

You may want to allow users to search for these words. You can do this by disabling the Solr *stop words* function.

**Warning**

Disabling the stop words function in this way makes the search index much larger and can degrade search performance.

1. Navigate to the Solr `schema.xml` file, for example, `TC_ROOT\solr-4.4.0\awc\solr\collection1\conf\schema.xml`.

2. Search for the lines identifying `stopwords_language.txt` files and comment out those lines, for example:

   ```xml
   <!--filter class="solr.StopFilterFactory" ignoreCase="true" words="lang/stopwords_en.txt"/ -->
   ```

   This allows all words to be included in searches, including the ones in the stop words list.

3. Reboot Solr.

4. Perform a reindex using the `runTcFTSIndexer` utility:

   ```
   TC_ROOT\TcFTSIndexer\bin\runTcFTSIndexer -task=objdata:reindex
   ```

**Search for synonyms**

Solr allows you to search for synonyms. For example, if you search for *MB*, search results also are returned for *megabyte*. Information about Solr use of synonyms can be found in the Solr documentation:

https://wiki.apache.org/solr/AnalyzersTokenizersTokenFilters#solr.StopFilterFactory

You can add words to the synonyms file to enable additional synonym searching.
1. Navigate to the Solr **synonyms.txt** file, for example, `TC_ROOT\solr\version\awc\solr\collection1\conf\synonyms.txt`.

2. Add synonyms on a single line separated by commas, for example:
   
   GB, gib, gigabyte, gigabytes

3. Reboot Solr.

**Configure object data indexing**

**Re-index search data**

The initial search index is performed during the installation of the client.

You should re-index your existing data using the `runTcFTSIndexer` utility after any of the following occurs:

- The business objects or properties to be indexed are modified in the Business Modeler IDE and the template is redeployed.
- A new template is added.
- An existing template that affects indexing is updated.
- Data is imported using TC XML utilities.
- A change is made to the Start Of Week preference. You can change this preference in the rich client by choosing Edit → Options → Calendar → Start Of Week.

  **Note**

  For information about object data indexing flow actions, see the `runTcFTSIndexer` utility.

To re-index your existing data:

1. If the `runTcFTSIndexer -task=objdata:sync -interval=seconds` command was run, you should temporarily shut it down by using the following command:
   
   `runTcFTSIndexer -task=objdata:sync -stop`

2. Perform one of the following:

   - To clear the existing full-text search data and perform a new, full index, open a Teamcenter command prompt, navigate to `TC_ROOT\TcFTSIndexer\bin`, and type the following command:
     
     `runTcFTSIndexer -task=objdata:reindex`

   - To perform a new, full index without deleting the existing indexed data, type the following command:
     
     `runTcFTSIndexer -task=objdata:index`

   By default, the indexer uses the current date and time as an end point. To specify a different date and time, use the TcFTSIndexer settings in TEM.
Note
To index data from the date defined in the
TC_FTS_INDEXER_HOME\conf\TcFtsIndexer_objdata.properties file, you must delete the TcFtsIndexerSessionObjData.cache file from TC_FTS_INDEXER_HOME\cache.

3. If you shut down the command in step 1 of this procedure, you can now start it back up.

Define index data and filters
By default, children of the ItemRevision business object are marked for indexing as well as specific properties on those business objects. Using the Business Modeler IDE, you can create a custom template to define other business objects and properties to index. You can also define search filters using global, business object, and property constants. For information about working with constants, see Business Modeler IDE in the Teamcenter collection.

You can define only certain property types for search and filtering:

<table>
<thead>
<tr>
<th>Properties supported for search and filtering</th>
<th>Properties not supported for search and filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Date</td>
<td>• String properties with long string as storage</td>
</tr>
<tr>
<td>• String</td>
<td>• Numeric properties</td>
</tr>
<tr>
<td>• References</td>
<td>• Array properties</td>
</tr>
<tr>
<td>• Logical</td>
<td></td>
</tr>
</tbody>
</table>

Note
My Changes and Inbox have separate filtering that you can set up.

1. Import the Active Workspace (aws2) template into your Business Modeler IDE project.
   For details about importing templates into Business Modeler IDE, see Business Modeler IDE in the core Teamcenter help collection.

2. In the Business Modeler IDE, open the business object you want to index.

3. To define a business object for indexing, use the Awp0SearchIsIndexed business object constant.
   The indexing values are inherited in the hierarchy. For example, if you mark ItemRevision business objects for indexing, all item revisions under it (part revision, document revision, and so on) are automatically marked for indexing. You can, however, choose to not index a lower level object by setting the property to false to override its inherited value.

4. To define a property for indexing, use the Awp0SearchIsIndexed property constant.
   The indexing values are inherited in the hierarchy. For example, if you mark object_type on ItemRevision business objects for indexing, object types for all item revisions under it (part revision, document revision, and so on) are automatically marked for indexing. You can, however, choose to not index a property on a lower level object by deselecting it manually.
5. To define the file types whose contents will be indexed, use the `Awp0IndexableFileTypes` global constant.

For a list of the file types that can be indexed, refer to the description of the `Awp0IndexableFileTypes` constant in the *Global constants* topic.

6. Define the search filters using the appropriate global constant, business object constant, or property constants.

For example, to specify a property to display on the list of search results filters, use the `Awp0SearchCanFilter` property constant. To set the order that the property appears in the list of filters, use the `Awp0SearchFilterPriority` property constant. The indexing values are inherited in the hierarchy.

![Image of search results with filters]

**Properties displayed in search results filtering**

**Note**

When you set the `Awp0SearchCanFilter` property constant to true on a property, that property only displays in the search results filtering list if all the objects returned also have that property. If all properties of the returned objects were displayed, the filter list would be far too long to be useful. Because owning user, last modification date, release status, and similar properties are common to all objects, they always appear in the search results filter list.
Tip

By default, Solr displays a maximum of 100 objects under each property in the filters list. To remove the limitation and allow all objects to be listed, open the \awc\solr\collection1\conf\solrconfig.xml file, search for `<requestHandler name="/tcfts" class="solr.SearchHandler">`, and add the following line to the `defaults` section:

```
<str name="facet.limit">-1</str>
```

The setting appears as follows:

```
<requestHandler name="/tcfts" class="solr.SearchHandler">
    <!-- Request handler for Teamcenter search -->
    <lst name="defaults">
        <str name="echoParams">explicit</str>
        <int name="rows">10</int>
        <str name="df">TC_0Y0_FTS</str>
        <str name="q.op">OR</str>
        <str name="facet.limit">-1</str>
    </lst>
</requestHandler>
```

7. Update your custom template into the database.

Change the user running the TcFTSIndexer

By default, `infodba` is the user that runs the search indexer (TcFTSIndexer). If you do not want `infodba` persistently logged on as the authorized indexing user, you can change to a different user. The indexing user must have read access to datasets and their associated files to index their text content.

1. Navigate to the \TC_ROOT\TcFTSIndexer\conf\TcFTsIndexer.properties file.

2. Change the user in the `Tc.user` setting to an authorized Teamcenter user with database privileges.

3. In the console, set an environment variable to the password value.

   ```
   set mytcenv=password
   ```

4. Create an encrypted password file for this user by running the `encryptPass.bat/sh` utility with the `-tc` argument and specifying the environment variable created in the previous step, for example:

   ```
   encryptPass -tc mytcenv
   ```

   The `encryptPass.bat/sh` utility is located in the \TC_ROOT\TcFTSIndexer\bin directory.

5. After you create the encrypted password file, remove the environment variable value.

   ```
   set mytcenv=
   ```

Run the search indexer after updating the custom template

Any time you make a change to your custom template, you must redeploy it to the database and run the search indexer.

1. Use TEM to install your custom template into the database.

   For details, see Business Modeler IDE.
Note

If changes are done using Hot Deploy or Live Update, you must manually run the following utilities after the Hot Deploy or Live Update has been run:

```
bmide_modedtool.bat -u=username -p=password -g=dba -tool=all -mode=upgrade
"-target_dir="TC_DATA"
```

These commands must be run in a Teamcenter command window that has TC_ROOT and TC_DATA defined.

2. Merge the Teamcenter and Solr schemas.

3. Run the search indexer.

Note

Users can search only in the master language set on the server, which is defined by the SiteMasterLanguage global constant. If you change the SiteMasterLanguage global constant value, you must run the search indexer with the -reindex option because the data needs to be re-indexed with the new set of analyzers for the language.

Configuring search index properties

The TcFTSIndexer search properties are stored in the TcFtsIndexer_objdata.properties file located at TC_ROOT\TcFTSIndexer\conf.

Most of the search properties are defined during the installation of the TcFTSIndexer client (see step 1 in Install Server Extensions features for details about the properties). However, you can define several additional properties in the TcFtsIndexer_objdata.properties file, such as objdatatransformstep.indexDatasetFileContent. This property controls the indexing of dataset file contents. Set the property to true (index dataset file contents) or false (do not index dataset file contents).

Search deployment best practices

Prerequisites for search deployment

Following are prerequisites for search deployment:

• An understanding TcFTSIndexer architecture.

• A general understanding of the TcFTSIndexer and its available configurations. Look at configurations in the TC_ROOT\TcFTSIndexer\conf directory.

• An understanding of all options provided by the runTcFTSIndexer -? command.

• Knowledge of Solr shards (https://cwiki.apache.org/confluence/display/solr/SolrCloud).

• Ensure the Teamcenter database is properly tuned and does not have slow SQL call messages in the syslog files. See the Teamcenter Deployment Guide on GTAC for information on database tuning. The Teamcenter Deployment Guide is located at GTAC→Documentation→Teamcenter.

• Ensure the Teamcenter database is on a dedicated machine.
• Ensure Solr is installed on a secure machine and is only accessible through the Teamcenter server.

Selecting hardware for TcFTSIndexer components

When selecting hardware for TcFTSIndexer components, following are general hardware considerations:

• See the Teamcenter Deployment Guide on GTAC for hardware considerations for the database (including tuning) and for the pool manager.

• Solr hardware requirements:
  


Some of the hardware considerations for TcFTSIndexer are based on memory and CPU usage of the TcFTSIndexer components:

• TcFTSIndexer orchestrator
  
  Runs a Java process that runs the query, TIE export, transform, and load operations in stand-alone mode. In this case, the orchestrator is CPU and memory intensive for large amounts of data. It should be on a dedicated machine.

• Pool manager
  
  For object data indexing, the TcFTSIndexer orchestrator connects to the pool manager to run query, TIE export, and load operations. These operations are executed in multiple tcserver instances and connect to the database for SQL calls. For parallel processing, the pool manager should be on a high-end dedicated machine.

• Solr
  
  Solr is CPU-intensive during query calls and can be memory intensive during large dataset file indexing. For this reason it should be on a separate high-end machine.

Search deployment strategy

Stand-alone indexing mode is the only supported mode for Object data indexing.

In this configuration TcFTSIndexer executes indexing operations in a single Java process (TcFTSIndexer orchestrator) that connects to the pool manager and the Solr search engine. This Java process is a Java SOA client that tracks the number of tcserver connections and runs flows based on the configuration.

The Solr search engine and the pool manager are not required to be on the same machine as the orchestrator and are recommended to be on separate machines.

TcFTSIndexer components can be set up independently on different machines, or in a simple case on a single machine. It has many configuration strategies that can be defined to meet specific customer needs. General considerations are:
• Number of machines available
• Machine specifications
• Available time for initial indexing
• Whether the component being installed is CPU/memory intensive
• Long-term maintenance

Stand-alone mode strategy:

For stand-alone mode, all components can be on one machine or separated as shown in the following diagram.

**Solr shards**

For large index scalability and redundancy, SolrCloud can be configured to support multiple Solr cores. One core equates to one piece of the total collection of indexed objects, and a number of cores can be utilized. Each core can have a group of shards assigned to it, where one shard is the leader and the rest are replicas of each other. If the leader fails, the replicas promote a
new leader with almost no downtime to the user. For best performance, each core should be set up on a separate machine. Each shard can be set up on a separate machine depending on the level of physical redundancy. This performance gain is only noticeable on indexes that contain 10 million or more objects. Siemens PLM Software recommends for highest performance output to use a single core configuration when indexes are less than 10 million. For more information, see https://cwiki.apache.org/confluence/display/solr/SolrCloud.

**Enabling SolrCloud authentication**

When authentication is enabled on the Solr jetty server in a SolrCloud setup, individual Solr nodes may not be able to communicate with each other.

To use SolrCloud with authentication on a proxy server:

   For example, the remaining steps assume you install the server in D:\Apache24.

2. Open the D:\Apache24\conf\httpd.conf file.
   Enable `mod_proxy` and `mod_proxy_http` by uncommenting these lines:
   ```
   LoadModule proxy_module modules/mod_proxy.so
   LoadModule proxy_http_module modules/mod_proxy_http.so
   ```

3. Add the proxy configuration information in the `httpd.conf` file.
   ```
   # Proxy Setup
   ProxyPass /siemens http://localhost:8983/solr
   ProxyPassReverse /siemens http://localhost:8983/solr
   ProxyPass /solr http://localhost:8983/solr
   
   <Location /siemens>
   AuthType Basic
   AuthUserFile "D:\Apache24\solr\password"
   AuthName ActiveWorkspaceRealm
   require valid-user
   </Location>
   ```
   **Note**
   The `http://localhost:8983/solr` URL specified must be the **Leader1** location.


5. In a command shell, navigate to D:\Apache24\bin and execute this command to create the password.
   ```
   htpasswd -c D:\Apache24\solr\password solr_admin
   ```
   You are prompted to enter and confirm the password.

6. Comment out all the authentication related settings from all `webdefault.xml` files of the leader and replica cores in SOLR_HOME/awc/etc/webdefault.xml.
   ```
   <!--
   <security-constraint>
   <web-resource-collection>
   <web-resource-name>Solr authenticated application</web-resource-name>
   <url-pattern>/*</url-pattern>
   ```
7. Log on to the Teamcenter rich client and change the AWS_FullTextSearch_Solr_URL
preference value to the new proxy server.
In this case the value is http://localserver/siemens, Where localserver is the IP/Host name
where Apache httpd server is running
8. Start the Apache httpd server.
D:\Apache24\bin>httpd
9. Test authentication:
Type the http://localserver/siemens/ URL in the browser to be prompted for authentication.
10. Test the indexer using the command:
    runTcFtsIndexer.bat -task=objdata:test

Configure Solr for HTTPS communication
By default, Solr uses HTTP, but you can configure Solr to use HTTPS. TcServer and TcFtsIndexer
are programmed to accept self-signed certificates as well as Certificate Authority (CA) signed
certificates.
1. Choose a certificate signed by a trusted CA for Solr, or create an self-signed certificate by using a
Java keytool or openssl.
2. Store the generated certificate in a keystore.
3. Copy the keystore file, for example, IdentityKeystore.jks, to the SOLR_HOME/etc/ directory
and make the following modifications to the jetty.xml file:

    <Call name="addConnector">
      <Arg>
        <New class="org.eclipse.jetty.server.ssl.SslSelectChannelConnector">
          <Arg>
            <New class="org.eclipse.jetty.http.ssl.SslContextFactory">
              <Set name="keyStore"><SystemProperty name="jetty.home" default="."/>/etc/IdentityKeystore.jks</Set>
              <Set name="keyStorePassword">secret</Set>
              <Set name="needClientAuth"><SystemProperty name="jetty.ssl.clientAuth" default="false"></Set>
            </New>
          </Arg>
        </New>
      </Arg>
      <Set name="port">8984</Set>
    </Call>

4-18 Active Workspace Deployment
4. Update the Solr URL to use the HTTPS by modifying the **AWS_FullTextSearch_Solr_URL** preference.

   For example, the preference will be **https://localhost:8984/solr** for the configuration above.

### Optimize usage of TcFTSIndexer machines

#### Introduction to optimizing TcFTSIndexer machines

First, index a small set of data to understand how long it takes to index the whole set of data. You can then optimize the machine usage. This helps in reconfiguring the TcFTSIndexer settings and when adding new machines based on need.

The TcFTSIndexer orchestrator supports query, TIE export, transform, and load as stand-alone processes that run in a sequential pattern. These processes have a single level of configuration in stand-alone mode, which is the maximum connections setting (**tc.maxConnections** property). This setting specifies the maximum number of Teamcenter connections to be open at a given time.

Usually, this number should be less than the number of warmed-up **tcserv** instances available in the pool manager. The minimum value is **1** and the default value is **3**.

To change the value using Teamcenter Environment Manager (TEM), in the Feature Maintenance panel select **Update Active Workspace Indexer settings** and click **Next** until you reach the Active Workspace Indexer Settings panel. Click **Advanced** to set the maximum connections.

![Maximum connections setting](image)

#### Optimize instances of TcFTSIndexer

Once the number of machines and the components on each machine are defined, the next step is to optimize the number of instances of each TcFTSIndexer operation to run on each machine.

1. Identify a small list of objects to index. Because indexing is based on time slices, identify a start and end time that contain around 5% of total objects. This can be achieved by running queries for indexable objects in the Teamcenter rich client.

2. After setting the start and end time in the
   
   `TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer_objdata.properties` file, run the following **runTcFTSIndexer** command to make sure all the components are running and the TcFTSIndexer orchestrator can connect to them:

   ```bash
   runTcFTSIndexer -task=objdata:test
   ```
3. Start the Windows Task Manager on each machine that has any indexing component installed, such as Solr, pool manager, TcFTSIndexer orchestrator, and database.

4. Run the following command to index the 5% set of objects:

   ```
   runTcFTSIndexer -task=objdata:reindex
   ```

   While executing the initial index of 5% of total objects, memory consumption and CPU processes must be monitored on machines that have any indexing component installed. This lets you evaluate whether there is room for more tcserver connections. Monitoring of the database and the machine it is running on is also crucial as to not overload either one.

   Record the overall time taken for reindexing using the report generated at the end of the TcFTSIndexer orchestrator execution. This report is generated on the console and stored in the `TC_ROOT\TcFTSIndexer\logs\TcFTSIndexer.log` file.

5. After the initial run, if all machines used less than 90% of their resources, increase the `tc.maxConnections` value.

   **Note**

   To dynamically modify the `tc.maxConnections` setting while an indexing run is active:

   a. Open a new Teamcenter command window and navigate to the `TC_ROOT\TcFTSIndexer\bin` directory.

   b. Run the following `runTcFTSIndexer` utility command:

      ```
      runTcFTSIndexer -maxConnections=connections
      ```

      Where `connections` is the number of connections you want, but no more than the number of `TcServers` available.

6. Iterate through the previous two steps until the `tc.maxConnections setting` is optimized by reaching more than 90% resource consumption on any machine.

7. After you optimize the `tc.maxConnections` setting, you can then optimize the `exportbatchsizestep.baseBatchSize` property.

   **Note**

   This step is optional for small environments with fewer than one million objects to index, but is recommended for medium to large environments with more than 1 million objects to index.

   - The default `exportbatchsizestep.baseBatchSize` value is 1000.
   - Increasing the value for `exportbatchsizestep.baseBatchSize` produces performance gains by increasing the number of Teamcenter objects per thread.
   - Internal testing produced optimized results with the value set between 2000 and 5000 depending on the size of the index.
   - Increasing the value to a value that is too large can adversely impact performance.

8. Once the TcFTSIndexer is optimized and sized correctly, you can start a reindex of the full data.
**TcFTSIndexer synchronization process**

After re-indexing, the synchronization operation can be run on the TcFTSIndexer orchestrator at regular intervals to keep indexed data up-to-date, for example:

```
runTcFTSIndexer -task=objdata:sync -interval=seconds
```

Synchronization is run as a single Java SOA client that connects to the pool manager and Solr to achieve the delta indexing. Synchronization takes an additional argument for the interval. This interval specifies the waiting period before the synchronization operation can be run again.

Synchronization performs the following steps:

1. Query for objects that failed to be indexed during the previous synchronization.

   If there are any failed objects, a warning message is written to the logs.

   **There are NNNN failed objects. Some of these errors may get fixed in subsequent sync calls. If the errors persist, these errors have to be fixed manually and "indexsyncfailures flow" has to be rerun to index these failures.**

   In this case, run TcFsIndexer with the **-task=objdata:indexsyncfailures** option.

   The next sync picks the failed objects and tries to recover them.

2. Query for all revision rule changed objects, run the list of objects through TIE export and transform, and load to Solr.

3. Query for deleted objects, connect to Solr, and delete the objects.

4. Query for new objects, run the list of objects through TIE export and transform, and load to Solr.

5. Query for objects affected by an Access Manager rule change, run the list of objects through TIE export and transform, and load to Solr.

6. Query for modified objects, run the list of objects through transform, and load to Solr.

7. Refresh objects.

   During this step the system queries for indexed objects that have exported dates older than the last processed date of the **:FTS_REFRESH** application ID. These objects must be indexed again.

   The **refreshBatchSize** property value configured in the TcFtsIndexer_objdata.properties file specifies the maximum number of objects that are queried for and indexed. By default this value is set to 10000. If the objects to be refreshed exceeds 10,000, the additional objects are handled in the next sync call.

8. Query for replication-pending objects.

   These are the objects that would have failed during TIE export, transform, or load during the previous steps.

   Indexing these objects is retried three times. If they continue to fail they are marked as import failed and are addressed in next sync call.

9. Clean up the object data that has been consumed by all the applications that have subscribed.
The number of objects being synchronized are output on the console and in the `TC_ROOT\TcFTSIndexer\logs\TcFTSIndexer.log` file. The interval of running synchronization must be defined based on the typical number of objects being handled by the synchronization case. If the number of objects is high, reduce the synchronization interval. Also consider how fast users need to get access to these changed objects.

If a synchronization operation is in progress and it is time for the next scheduled synchronization based on the interval, the system skips the next scheduled synchronization and starts when the current synchronization operation completes.

### TcFTSIndexer synchronization

#### Sample indexing and query performance data
- Metadata and file content in Standalone mode

<table>
<thead>
<tr>
<th>Machine</th>
<th>CPU cores&lt;sup&gt;1&lt;/sup&gt;</th>
<th>Core clock speed (GHz)</th>
<th>RAM</th>
<th>HDD (GB)</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business logic server</td>
<td>12</td>
<td>2.0</td>
<td>64</td>
<td>3x600</td>
<td>TcFTSIndexer&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>Database</td>
<td>8</td>
<td>2.9</td>
<td>81.9</td>
<td>5x600</td>
<td>Oracle</td>
</tr>
</tbody>
</table>

- Indexing objects in Solr:
  - 165,232 objects: 17.83 minutes / 556,048 objects per hour.
  - Index of item revisions, datasets, and dataset file content.

---

1. Cores are hyper-threaded
2. TcFTSIndexer: Query, Export, Transform, Load, pool manager, Solr, Weblogic
70,500 files indexed.

Average file size: 396 KB.

30 properties indexed per object.

- **Tc.max connections** = 28.
- **baseBatchSize** = 1000.
- The pool manager is set to 30 warmed-up TcServer instances.

**Query performance:**

<table>
<thead>
<tr>
<th>165,232 objects in Solr</th>
<th>Search value</th>
<th>Objects found</th>
<th>Initial search (seconds)</th>
<th>Second search (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wildcard search</td>
<td>*</td>
<td>All</td>
<td>8.8</td>
<td>3.0</td>
</tr>
<tr>
<td>String searches</td>
<td>auto*</td>
<td>4786</td>
<td>5.4</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>engine*</td>
<td>16472</td>
<td>6.7</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>item*</td>
<td>38246</td>
<td>6.1</td>
<td>2.1</td>
</tr>
</tbody>
</table>

- Metadata only in Standalone mode

**Note**

This example uses the same machine parameters as the previous example.

**Indexing objects in Solr:**

- 12,244,466 objects: 2103.864 minutes / 349,203 objects per hour.
- Index of item revisions and datasets; no documents are indexed during this run.
  
  - 28 properties indexed per object.

- **Tc.max connections** = 28.
- **baseBatchSize** = 5000.
- The pool manager is set to 30 warmed-up TcServer instances.

**Query performance:**

<table>
<thead>
<tr>
<th>12,244,466 objects in Solr</th>
<th>Search value</th>
<th>Objects found</th>
<th>Initial search (seconds)</th>
<th>Second search (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wildcard search</td>
<td>*</td>
<td>5.5 Million+</td>
<td>27.9</td>
<td>7.7</td>
</tr>
<tr>
<td>String searches</td>
<td>motor</td>
<td>12734</td>
<td>21.8</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td>engine*</td>
<td>38670</td>
<td>22.3</td>
<td>3.8</td>
</tr>
<tr>
<td></td>
<td>car*</td>
<td>119516</td>
<td>22.5</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td>vehicle*</td>
<td>584179</td>
<td>24.3</td>
<td>5.0</td>
</tr>
</tbody>
</table>
TcFTSIndexer extensibility

TcFTSIndexer architecture

TcFTSIndexer search deployment diagram

Active Workspace search deployment

Indexing work occurs in the enterprise tier:

- Selected product data is indexed in Solr, an open source search platform from Apache.
- The master product data is not stored in Solr. It is always loaded from Teamcenter.
- The TcFTSIndexer orchestrator indexes external and Teamcenter objects into Solr. It connects to the pool manager to query and extract Teamcenter data to be indexed into Solr.
- In dispatcher indexing mode, Dispatcher scheduler and Dispatcher module are used to distribute indexing tasks across multiple machines to load-balance indexing into Solr.

TcFTSIndexer deployment options

TcFTSIndexer supports two options to index data:
• **Stand-alone indexing mode**

In this configuration, TcFTSIndexer executes indexing operations in a single Java process (TcFTSIndexer orchestrator) that connects to the pool manager and the Solr search engine. This Java process is a Java SOA client that tracks the number of tcserver connections and executes flows based on the configuration. This is the only mode supported for object data indexing.

**TcFTSIndexer components (stand-alone indexing mode)**

- **TcFTSIndexer orchestrator**
  - Manages the overall indexing process and Teamcenter connections.
  - In dispatcher mode, integrates with Dispatcher to submit requests and handle requests processed by Dispatcher.
  - Handles the synchronization of changes to Teamcenter data (additions, deletions, and modifications) to the index.
  - Loads data into Solr.
  - Provides a framework to customize new indexer types.

- **Teamcenter pool manager**
  - TcFTSIndexer orchestrator connects to the pool manager for communication with Teamcenter for its operations.

- **Solr**
The load operation adds, deletes, and modifies indexed data to Solr, and the TcFTSIndexer orchestrator commits data to Solr at regular intervals.

- Dispatcher indexing mode

Steps within a flow can be executed in Dispatcher if they are CPU or memory intensive. This helps to offload and load balance these kinds of steps to other machines to get better performance. Steps can be configured to work in dispatcher mode and executed in Dispatcher when the `-dispatcher` argument is specified. This mode can be used if the type associated with the TcFtsIndexer supports Dispatcher mode.

### TcFTSIndexer components (dispatcher indexing mode)

- **Dispatcher scheduler**
  - Load balances the submitted requests.
  - Submits requests to individual module machines.
  - Returns status to the TcFTSIndexer orchestrator.

- **Dispatcher module**
  - Executes and tracks Dispatcher requests.
  - Manages the number of instances of each type of request running at any given time.
  - Sends status back to Dispatcher scheduler about the executed request.
TcFTSIndexer extensibility framework

TcFTSIndexer is a Java application that executes types, flows, and steps.

- **Types**
  
  Represent the different integrations or customizations into TcFTSIndexer, for example, object data and structure. Types contain flows.

- **Flows**
  
  Represent the supported operations for a given type. For example, some flows for object data (ObjData) are reindex, index, clear, recover, and synchronization. Flows contain steps that are chained together.

- **Steps**
  
  Contain methods that define a certain behavior. Each step should have the input and outputs defined. These steps can be reused across different flows and types based on the input and output requirements. Steps are executed in sequence as defined in a flow. Output of one step becomes the input of the next step.

  For example, the object data (ObjData) reindex flow has query, TIE export, transform, and load steps.

  There are three types of steps:

  - Simple step
    
    Executes a step based on the input and returns the output data for the next step to process.

  - Split step
Splits a list of input data into multiple simple steps based on the size of the list. In the case of object data (ObjData), the query is split into time slices using this step. A flow can have multiple split steps.

- Aggregate step
  
  Waits on all the split steps to finish and combines all the output data from the processing of split steps. In a flow, every split step has to have an associated aggregate step.

**TcFTSIndexer installation layout**

TcFTSIndexer installation has the following directory structure (located at $TC\_ROOT\TcFTSIndexer$):

- **bin**
  
  Contains scripts to start up TcFTSIndexer.

- **cache**
  
  Stores all relevant cache files.

- **conf**
  
  Contains all the configuration files.

- **docs**
  
  Contains Javadocs for published APIs.

- **lib**
  
  Contains all the JAR files needed to run TcFTSIndexer.

- **logs**
  
  Contains the indexer log files.

- **sample**
  
  Contains sample files to help with customizations.

**TcFTSIndexer customization**

**Overview of indexer customization**

TcFTSIndexer is a Java application that can execute types, flows, and steps. TcFTSIndexer:

- Is an SOA client that connects to Teamcenter to extract data and index the data into Solr.
- Allows modification of any existing steps and flows to meet customer requirements.
- Can be customized to extract external system data and index into Solr.
- Provides utilities that can be used in step customization.
Details of these utilities are in Javadocs available in the TC_ROOT\TcFTSIndexer\docs\javadocs directory.

Indexer customization prerequisites

- A working TcFTSIndexer Installation in stand-alone mode.
- A high-level understanding of TcFTSIndexer architecture.
- An understanding of input and output objects associated with each step in a flow that is being customized.
- An understanding of properties associated with the flow.
- Review sample code for steps in the TC_ROOT\TcFTSIndexer\sample directory.
- Refer to Javadocs for published methods and classes discussed.
- Refer to the TC_ROOT\TcFTSIndexer\sample\TcFtsIndexer_sample1.properties files and the TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer_objdata.properties files for example configurations.
- For the new requirements, create a high-level design of the functionality and:
  - Create a list of steps with their input and output objects defined. Check if there are existing steps that can be reused.
  - Chain these steps to create a new flow or modify an existing flow.
  - Determine if the flow is part of an existing type or a new type.

Adding new types

A type is defined by creating a property file that includes all flows and step configurations. Create this property file in the TC_ROOT\TcFTSIndexer\conf directory. The file name must have the following syntax: TcFtsIndexer_TypeName.properties. TcFTSIndexer reads the property file name and determines the type name associated with these properties.

The type uses the default behavior unless specific behavior is required. In this case a type class can be defined as follows in the TcFtsIndexer_TypeName.properties file:

```
type=full-class-name
```

For example:

```
com.siemens.teamcenter.ftsi.objdata.TcFtsIndexerObjDataType
```

This custom implementation class must extend the TcFtsIndexerType class and typically requires the override of the following method:

```
public void initialize( Object zData )
```

This method is called during the execution of the type and helps in any initialization of objects specific to this type or in the validation of inputs. For example, the object data type implementation class overrides the initialize method to make sure only one object data type is executed at any given time.
**Adding flows**

A flow has an internal name and an associated flow action that are run by end users as commands. The mapping between the flow action and the internal flow name is defined as:

\[ \text{internal-flow-name}.action=\text{flow-action} \]

For example:

```
reindexflow.action=reindex
```

Flow uses the default behavior unless there is a need to support flow-specific behavior. A custom flow class can be defined by creating a property as:

\[ \text{internal-flow-name}=\text{full-class-name} \]

For example:

```
reindexflow=com.siemens.teamcenter.ftsi.objdata.TcFtsIndexeObjDataFlow
```

This custom implementation class must extend the `TcFtsIndexerFlow` class and typically requires an override of the following method:

\[ \text{public void initialize( Object zData )} \]

This method is called during the execution of the flow and helps in any initialization of objects specific to this flow or in the validation of inputs. For example, the object data flow implementation class overrides the initialize method to cache Teamcenter preferences in a cache file for performance reasons.

Users can type `runTcFTSIndexer.bat/sh -?` command to get information of all supported flows. The flow description can be provided as follows:

\[ \text{internal-flow-name}.description=description-text \]

For example:

```
reindexflow.description=Clears the existing indexed data and performs index of data.
```

**Adding steps**

- **Implementation class**

  Each step must be associated with an implementation class. This class must have clearly defined input and output parameters so that it can be reused in other flows if needed. Steps can be defined in the property file as:

  \[ \text{step-name}=\text{full-class-name} \]

  For example:

  ```
  querytimesslicestep=com.siemens.teamcenter.ftsi.objdata.steps.query.QueryTimeSliceStep
  ```

  There are three choices based on types of steps:

  - **Simple step**

    For a simple step that converts an input object to an output object, implement a class that extends the `TcFtsIndexerAbstractStep` class. (Refer to the Javadocs for details.) This requires implementation of the `process` method:

    ```java
    public ITcFtsIndexerStep.Status process( IStepInfo zStepInfo, IMessage zMessage ) throws Exception
    ```
The **process** method for each step is called in sequence for all steps defined in a flow. Output of the process is sent as input to the next step. The process takes an **IStepInfo** class as an argument. This class tracks all the statuses associated with the step being processed. Steps can set tracking information and also use helper methods to access the staging directory using the **IStepInfo** class. All **StepInfo** objects are printed as a status at the end of indexing or during `-status` command execution.

The **IMessage** object is a message object that holds on to input/output objects and is accessible across steps in a flow. The message object contains a data object that can be any Java object. In the case of simple steps, the same message object is passed around.

TcFTSIndexer parses all the arguments passed in through the command line and creates a list of arguments that are not framework related. These arguments are considered type-specific arguments. This list is sent to the first step in the flow for processing as message data.

Steps can get this message data object and update it. They can also set new message data objects. Steps can be stopped from further processing by throwing an exception or returning the **NoData** status.

See `TC_ROOT\TcFTSIndexer\sample` directory for sample steps.

- **Split step**
  This step splits a list of objects as the message data input to the individual elements and creates a new message object for each element in the list. TcFTSIndexer creates parallel threads equal to the size of the input message data list and executes the next step with these new message objects. The split step can be defined as:

  ```
  step-name=com.siemens.teamcenter.ftsi.core.TcFtsIndexerSplitStep
  ```

  For example, object data indexing splits a list of lists (`List<List<String>>`). This list contains a list of UIDs of specific batch size. This step breaks the list of lists into multiple threads processing each batch size list of UIDs.

- **Aggregate step**
  This step waits on all split threads to complete and combines the successful split step messages data into a new message object and passes it to the next step. The number of aggregate steps must match the split steps. The aggregate step can be defined as:

  ```
  step-name=com.siemens.teamcenter.ftsi.core.TcFtsIndexerAggregateStep
  ```

  For example, the object data aggregate step waits on all the threads created by the split step to process a batch size of UIDs. After all the split steps are aggregated, the data is committed to Solr in the next step.

- **Steps in a flow**
  
  TcFTSIndexer executes the steps in the order specified. The sequence of steps defined in a flow must be defined in the property file as:

  ```
  internal-flow-name.steps=step-name-1,step-name-2,step-name-n
  ```

  For example:

  ```
  reindexflow.steps=tcftsindexervalidatestep,querytimeslicestep,
  querysplitstep,objcdataqueriesstep,exportbatchsizesstep,exportsplitstep,
  objdataexportstep,objcdatatransformsstep,objcdataloadstep,batchexportaggregatestep,
  postexportbatchaggregatestep,querytimeaggregatestep,postquerytimeaggregatestep
  ```
• Status of steps

By default, the `-status` argument outputs the status of all steps. To control the output to only certain important steps, the following property format can be used:

```
internal-flow-name.status=step-name-1,step-name-4,step-name-7
```

For example:

```
reindexflow.status=objdataquerystep, objdataexportstep, objdatatransformstep, objdataloadstep
```

• Create dispatcher step

A step can be used in stand-alone or dispatcher mode. By default, a step only works in stand-alone mode. Users have to decide if a step should be run in dispatcher mode based on the CPU and memory usage. If the step must be configured for dispatcher mode, the following property must be added:

```
step-name.usedispatcher=true
```

For example:

```
objdatatransformstep.usedispatcher=true
```

Deploy a new type

1. Ensure that the TcFTSIndexer is installed and working in stand-alone mode.

2. Create a new `TcFtsIndexer_type-name.properties` file and add the associated properties to the `TC_ROOT\TcFTSIndexer\conf` directory.

3. Implement the classes for types, flows, and steps as defined in the property file.

4. Compile the code by adding JAR files in the `TC_ROOT\TcFTSIndexer\lib` directory to the classpath.

5. Create a custom JAR file with all the custom implementation classes.

6. Copy the new custom JAR file into the `TC_ROOT\TcFTSIndexer\lib` directory.

7. Run the `runTcFTSIndexer -?` command to check if the new custom type and flows appear in the console.

8. Fix any configuration and implementation related issues based on the output console messages.

9. Run the flow using the following command:

```
runTcFTSIndexer -task=type:flow-action
```

Integrate a new type with dispatcher

If there are steps that need to be offloaded to a different machine for CPU or memory optimizations, mark them as `step-name.usedispatcher=true` in the property file. Each step must implement the dispatcher-specific interfaces. These methods are called by the TcFTSIndexer framework from the TcFTSIndexer orchestrator or the Dispatcher module depending on the context.

1. Make sure that the TcFTSIndexer is installed and working in dispatcher mode.
2. Create a dispatcher flow that calls the step to be run in the Dispatcher module in the *TcFtsIndexer_type-name.properties* file.

3. Copy the custom *TcFtsIndexer_type-name.properties* file to the *dispatcher-location\Module\Translators\TcFTSIndexer\conf* directory.

4. Copy the custom JAR file to the *dispatcher-location\Module\Translators\TcFTSIndexer\lib* directory.

5. Create a new translator service in the *dispatcher-location\Module\conf\translator.xml* file that executes the new dispatcher flow. See the existing example *objdatatransformstep* service for details.


7. Run the main flow from TcFTSIndexer orchestrator using the `-dispatcher` argument.

**Modify an existing type**

Back up the existing type before any modifications; work with a copy of the existing type.

1. Make sure TcFTSIndexer is installed and working in stand-alone mode.

2. Copy *TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer_type-name1.properties* to *TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer_type-name2.properties*.

3. Change any type-specific definitions to the new name in the file.

4. Delete unnecessary flows and steps for the new type.

5. Add, modify, or delete any steps associated with a flow.

6. Implement the new steps based on the property file definition.

7. Create a custom JAR file with all the implementation classes.

8. Copy the new custom JAR file into the *TC_ROOT\TcFTSIndexer\lib* directory.

9. Run the *runTcFTSIndexer -?* command to check if the new custom type and flows appear in the console.

10. Fix any configuration and implementation related issues based on the output console messages.

11. Run the flow using the following command:

    `runTcFTSIndexer -task=type:flow-action`
**ObjData indexing**

**Indexing in ObjData stand-alone mode**

1. The TcFTSIndexer orchestrator connects to Teamcenter and does the initial validation. It reads the configuration for start and end time, and then breaks down the time into configurable time slices for query.

2. The TcFTSIndexer orchestrator creates a new thread that connects to Teamcenter and gets the list of UIDs.

3. The TcFTSIndexer orchestrator reads the output UID file and chunks the data into a manageable size, and then calls TIE export by connecting to the pool manager. This step creates the Teamcenter XML file.

4. The TcFTSIndexer orchestrator converts the Teamcenter XML file to a Solr input XML file with the security read expressions.

5. The TcFTSIndexer orchestrator loads the Solr input XML file into Solr.

6. The TcFTSIndexer orchestrator connects to the pool manager and calls to confirm export.
Object data reindexing steps

Following are steps in the ObjData reindex flow. These steps are reused in the ObjData recover and synchronization flows. Input and output objects related to each step are available in Javadocs.

- **tcftsindexervaldatatestep**
  
  Validates the configuration, connects to pool manager, Solr, and Dispatcher in dispatcher mode.

- **querytimeslicestep**
Breaks down the start and end time into smaller configurable time slices and creates a list of time slices.

- **querysplitstep**
  Is a split step that uses the list from the **querytimeslicestep** step and creates multiple **objdataquerystep** steps. The number of **objdataquerystep** steps matches the list size.

- **objdataquerystep**
  Queries for all indexable objects in the time slice and returns a list of UIDs to index.

- **exportbatchsizestep**
  Breaks down the list of UIDs to a configurable batch size and creates a list of configured batch size UIDs.

- **exportsplitstep**
  Splits a list of configured batch size UIDs into multiple **objdataexportstep** steps.

- **objdataexportstep**
  Calls TIE export using the UIDs from the **exportsplitstep** step and creates the Teamcenter XML file.

- **objdatasaxtransformstep**
  Converts the Teamcenter XML file to a Solr XML file.

- **objdataloadstep**
  Loads the Solr XML into Solr and calls to confirm export.

- **batchexportaggregatestep**
  Waits on all steps to be processed that are created by the **exportsplitstep** split step. Essentially it waits on all UIDs for a given time slice to be indexed.

- **postexportbatchaggregatestep**
  Updates the time slice as success in the cache file.

- **querytimeaggregatestep**
  Waits on all time slices to be finished. It waits on all splits created by the **querysplitstep** step to be processed.

- **postquerytimeaggregatestep**
  Creates the Solr suggestions after all the objects are indexed.

**ObjData indexing support for saved queries**

A sample **savedquery** flow action is added to the **ObjData** type to support queries for UIDs through saved queries. Sample code for the **savedquerystep** step is available in the `TC_ROOT\TcFTSIndexers\sample` directory. This sample step is
integrated into the saved query flow. Flow and step configurations are available in the

`TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer_objdata.properties` file. Properties related to sample saved queries are named `savedquerystep.*`. See the properties description for usage and details.

This saved query can be executed by running:

```
runTcFTSIndexer -task=objdata:savedquery
```

Additional saved query requirements can be handled by making changes to sample code and creating a new step. This new step can replace the existing `savedquerystep` step.

**Run sample flow**

TcFTSIndexer comes with a sample flow. This sample flow can be run using the following steps:

1. Copy the `TC_ROOT\TcFTSIndexer\sample\TcFtsIndexer_sample1.properties` file to the `TC_ROOT\TcFTSIndexer\conf` directory.

2. Run the command:

```
runTcFTSIndexer -task=sample1:sampleflow1
```

This sample flow contains:

- Four simple step implementation classes (`SampleStep1.java`, `SampleStep2.java`, `SampleStep3.java`, and `SampleStep4.java`). Code is available in the `TC_ROOT\TcFTSIndexer\sample` directory.

- One split step.

- One aggregate step.

- The `SampleStep4.java` file that is reused to create a new `samplestep5` step.

This flow is chained as follows:

1. **samplestep1**

   This simple step creates a list of lists (`List<List<String>>`) and adds it to the message data. This message data list has four lists that contain sample strings of size 3. This message data list is set as tracking information in the `StepInfo` message to show status. Following is the sample step:

   ```
   [Batch 1 MD 1, Batch 1 MD 2, Batch 1 MD 3,
    Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3,
    Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3,
    Batch 4 MD 1, Batch 4 MD 2, Batch 4 MD 3]
   ```

2. **samplesplitstep**

   This is a split step that creates four threads and these threads call the `samplestep2` step with message data input as the inner list of strings of size 3.

3. **samplestep2**

   This step is called four times using separate threads with inner list of size 3:

   ```
   Thread1 -> [Batch 1 MD 1, Batch 1 MD 2, Batch 1 MD 3]
   Thread2 -> [Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3]
   ```
4. samplestep3

This step is called by each of the four threads and the same message objects are sent from the previous step. An additional string is added to it.

Thread1 -> [Batch 1 MD 1, Batch 1 MD 2, Batch 1 MD 3, Message Data for Step3]
Thread2 -> [Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Message Data for Step3]
Thread3 -> [Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3, Message Data for Step3]
Thread4 -> [Batch 4 MD 1, Batch 4 MD 2, Batch 4 MD 3, Message Data for Step3]

One thread execution of this step is returned as NoData status and the other throws an exception. This stops further processing of the steps on these threads and only the remaining two threads move on to the next step.

5. samplestep4

This step is called by only two threads and the message associated with these threads is passed from the previous step.

Thread2 -> [Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Message Data for Step3]
Thread3 -> [Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3, Message Data for Step3]

6. sampleaggregatestep

This step waits on all four threads to finish and aggregates the message data associated with successful threads. Because one thread failed and another had no data, the combined message data created is a list of two entries, and these two entries are the lists from previous successful messages.

[[[Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Message Data for Step3]
[Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3, Message Data for Step3]].

This step retrieves the message object before the samplestep1 split step message object and passes it to the next step.

7. samplestep5

This step gets the message object from the previous aggregate step with message data.

Thread 5 -> [[Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Message Data for Step3]
[Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3, Message Data for Step3]].

The tracking information set on the message data is the same as the one set in the samplestep1 step.

This sample flow outputs the message data at every step and provides a consolidate status at the end:
Message Data for Step3], Count=3]  
U1470d4f6e9e3297a28dd0012  0.14 Done [MessageList=[Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3, Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Batch 1 MD 1, Batch 1 MD 2, Batch 1 MD 3], Count=4]  
U146ee0c1848927a28dd0015  0.14 Done [MessageList=[Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Batch 1 MD 1, Batch 1 MD 2, Batch 1 MD 3], Count=4]  
U146f3ec056927a28dd0014  0.15 Error [MessageList=[Batch 1 MD 1, Batch 1 MD 2, Batch 1 MD 3], Count=0]  
---------------------------------------------samplestep3--------------------------------------------
Status: Created: 0 Started: 0 Done: 3 Error: 1  
Total Time 0.57 Total Count 8
---------------------------------------------samplestep4--------------------------------------------
Status: Created: 0 Started: 0 Done: 2 Error: 0  
Total Time 0.03 Total Count 8
---------------------------------------------samplestep5--------------------------------------------
Status: Created: 0 Started: 0 Done: 1 Error: 0  
Total Time 0.00 Total Count 12

Step Summary samplestep1
Status: Created: 0 Started: 0 Done: 1 Error: 0
samplestep2
Status: Created: 0 Started: 0 Done: 4 Error: 0
samplestep3
Status: Created: 0 Started: 0 Done: 3 Error: 1
samplestep4
Status: Created: 0 Started: 0 Done: 2 Error: 0
samplestep5
Status: Created: 0 Started: 0 Done: 1 Error: 0

Total time for all Steps 0 sec
Overall Time 0.307 sec
Done processing Type: samplel FlowAction: sampleflow

Indexing non-Teamcenter data

Create external system objects

Before performing this step, be sure you know the structure of the objects you are creating (that is, property name and type).

1. In the Business Modeler IDE, create a new Business Modeler IDE template project. Make sure you select foundation and aws2 as the Dependent Templates.

2. In the newly created project, create a new business object using the Awp0AWCExternalSystemObject business object as the parent object.

3. Add the required properties as Runtime properties.

The following attribute types are supported: Boolean, Date, Double, Integer, and String.
4. Set the **Awp0SearchIsStored** property constant to **true** for the object property content to display in Active Workspace.

5. Set the **Awp0SearchIsIndexed** property constant to **true** to indicate that the field is searchable.

6. Open the **Awp0AWCEnternalSystemObject** business object and set the **Awp0SearchIsIndexedExt** business object constant to **true** to indicate that external business objects are indexed for searching.

7. Deploy your template to the Teamcenter server.

   For instructions, see *Business Modeler IDE*.

8. Create the XRT files used to format and display the non-Teamcenter data in Active Workspace.

   For each object type added, you must create two datasets with XRT files for the proper display of objects in Active Workspace: one for the **Summary** panel and one for the **Info** panel (you can review **Awp0ItemRevSummary** and **Awp0ItemRevInfoSummary** for an example of the syntax).

9. Add the following Teamcenter preferences to link these XRT files to the external system objects added:

   - **ExternalSystemObjectName.CellProperties**
   - **ExternalSystemObjectName.INFORENDERING**
   - **ExternalSystemObjectName.SUMMARYRENDERING**
   - **InfoXRTDatasetName.INFOREGISTEREDTO**
   - **SummaryXRTDatasetName.SUMMARY_REGISTEREDTO**

   **Note**

   If you add an **AWC** prefix to the new preferences, they only apply to Active Workspace; other Teamcenter clients are unaffected.

**Add indexing for external objects**

**Add data indexing to TcFTSIndexer**

Before starting on the external objects indexing, understand how **ObjData** indexing works. Also ensure you have a working TcFTSIndexer installation for **ObjData** indexing. After determining the details of how to get the data from your external system, create a **type**, **flow**, and **steps**. This flow reuses some of the steps defined in the **objdata:savedqueryFlow** flow.

1. Create a new type by adding a new property file to the **TC_ROOT\TcFTSIndexer\conf\** directory. For example, create a **TcFtsIndexer_externaldata.properties** file. (This creates an **externaldata type**.)

2. Create a flow action, for example, **extdataflow.action=extdata**

3. Copy the steps to reuse from the **objdata:savedquery** flow found in the **TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer_objdata.properties** file. Steps that can be reused are defined as follows. Details for these steps are in Javadoc associated with each class.
The exact steps and details of the flow to create are data and system dependant, for example:

- In a system where there are large numbers of objects to index, return data as IDs in query calls. This allows for greater control over how the objects are gathered from the system and how large are the chunks of data to process.
- In an environment where there is already an existing XML export, return data as XML in export calls.
- In other cases, a direct export of data into the Solr input XML format may be preferable.

**External data is returned as IDs in query calls to the external system**

1. Define a flow with the steps as follows.
   
   ```
   extdataflow.steps=tccfsexistvalidatstep,extquerystep,exportbatchsizestep,exportsplitstep,extexportstep,exttransformstep,objcdataoladstep,batchexportaggregatstep,postexportbatchaggregatstep,
   postquerytimeaggregatstep=com.siemens.teamcenter.fsi.objdata.steps.postprocess.PostQueryTimeAggregateStep
   ```

   The three steps in bold (extquerystep, extexportstep, and exttransformstep) must be created in the property file and implemented.

2. Implement a step class that connects to external systems and returns the IDs of objects to index as a list in message data. Look at the `TC_ROOT\TcFtsIndexerTcFtsIndexerSavedQueryStep.java` file for sample code that executes a saved query to return a list of UIDs in message data.

   The output is a list of IDs, for example:
   
   ```
   extquerystep=com.siemens.teamcenter.fsi.externaldatalsteps.query.QueryStep
   ```

3. Implement an export step that connects to an external system and exports an XML file for the input IDs. This step must return the full path to the XML file as a string in the message data.

   The input is a list of IDs and the output is a string that is the full path to the external system XML file, for example,
   
   ```
   extexportstep=com.siemens.teamcenter.fsi.externaldatalsteps.export.ExportStep
   ```

4. Implement a step that takes the XML file from the previous step and converts it to a Solr input XML file. The full path to the XML file that was output from the export step is sent as input to this step as message data. This step must return the full path to the Solr XML file as a string in the message data object. The input is a string that is the full path to the external system XML file.

   The output is a string that is the full path to the Solr XML file, for example:
   
   ```
   exttransformstep=com.siemens.teamcenter.fsi.externaldatalsteps.transform.TransformStep
   ```

**External data is returned as XML in export calls to the external system (no query)**

1. Define a flow with the steps as follows:
   
   ```
   extdataflow.steps=tccfsexistvalidatstep,extexportstep,exttransformstep,objcdataoladstep,postexportbatchaggregatstep,postquerytimeaggregatstep
   ```

   The two steps in bold (extexportstep and exttransformstep) must be created in the property file and implemented.
2. Implement an export step that connects to an external system and exports the XML file for all objects to be indexed. This step must return the full path to the XML file as a string in the message data object.

   The output is a string that is full path to the external system XML file, for example:

   ```
   extexportstep=com.siemens.teamcenter.ftsi.externaldata.steps.export.ExportStep
   ```

3. Implement a step that takes the XML file from the previous step and converts it to a Solr input XML file. The full path to the XML file that was the output of the export step is sent as input to this step as message data. This step must return the full path to the Solr XML file as a string in the message data object.

   The input is a string that is the full path to the external system XML file, and output is a string that is the full path to the Solr XML file, for example:

   ```
   exttransformstep=com.siemens.teamcenter.ftsi.externaldata.steps.transform.TransformStep
   ```

**External data is returned as Solr input XML from external system (no query and export)**

1. Define a flow with the steps as follows:

   ```
   extdataflow.steps=tcftsindexervalidatestep,extrtransformstep,
   objdataloadstep,postexportbatchaggregatetep,postquerytimeaggregatetep
   ```

   The step in bold (extrtransformstep) needs to be created in the property file and implemented.

2. Implement a step that connects to the external system and creates a Solr input XML file for all objects to index. This step must return the full path to the Solr XML file as a string in the message data object.

   The output is a string that is the full path to the Solr XML file. The basic format of the Solr XML to generate is as follows:

   ```
   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
   <add>
     <doc>
       <field name="property name">Value of Prop</field>
       ...
       <field name="property name">Value of Prop</field>
       </doc>
     ...
     <doc>
       ...
     </doc>
   </add>
   ```

   Each object is represented by a `<doc>` element. The properties of the object are represented by the `<field>` element. The name attribute represents the property name defined in the Business Modeler IDE and is in the following format: `TC_0Y0_name-of-the-object_0Y0_propertyname`.

   The following elements are required:

   - **id**
     Specifies the unique ID of the object. To prevent object ID collisions, prefix the ID with a unique identifier.

   - **TC_GROUP_FIELD_ID**
     Represents the ID. This is an internally used value; it should be the same as id.
- **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSysObjClassName**
  Identifies the name of the object defined in the Business Modeler IDE used to properly generate the object type in Active Workspace.

- **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSystemName**
  Identifies the name of the external system from which the object came.

- **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSystemURI**
  Defines the URI in which to display the object.

- **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0PropertyNameList**
  Specifies the element for each Solr name of the object properties.
  The value format is: **TC_0Y0_object-name_0Y0_property-name**

- **TC_PRIV_am_rule_str**
  Defines the permission read expression.
  Permission information is indexed with each object. The base security read expression string to allow an object to be searchable by all users is **EAKT(EACT+)EAYT+**. Detailed permission information may require assistance from Siemens PLM Software.

- **TC_0Y0_WorkspaceObject_0Y0_object_type**
  Allows objects of this class name to be filtered using the Type category in the Active Workspace client. The value must be the same as the **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSysObjClassName** field.

An example of the general XML follows:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><add>
  <doc>
    <field name="TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSysObjClassName">The object class name defined in the BMIDE</field>
    <field name="TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSystemName">The name of the external system</field>
    <field name="TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0PropertyNameList">TC_0Y0_A4_address_0Y0_a4configuration</field>
    ...
    <field name="TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0PropertyNameList">TC_0Y0_A4_address_0Y0_a4type_displayName</field>
    <field name="id">A4_F0D44</field>
    <field name="TC_Group_Field_Id">A4_F0D44</field>
    <field name="TC_0Y0_A4_address_0Y0_a4configuration">Feature Dictionary Revision</field>
    ...
    <field name="TC_0Y0_A4_address_0Y0_a4type_displayName">Address</field>
    <field name="TC_PRIV_am_rule_str">EAKT(EACT+)EAYT+</field>
  </doc>
</add>
```

SOA method calls to Teamcenter are not allowed in this method.
Configure TcFTSIndexer

1. Add JAR files in the `TC_ROOT\TcFTSIndexer\lib` directory into the classpath and compile the implemented code and package it into a .jar file.

2. Copy the .jar file created to the `TC_ROOT\TcFTSIndexer\lib` directory.

3. Copy the `TcFtsIndexer_type.properties` file created when you added data indexing to `TcFTSIndexer` the `TC_ROOT\TcFTSIndexer\conf` directory.

4. Run the `runTcFTSIndexer -task=type:flow-action` command to execute the flow, for example:
   
   ```
   runTcFTSIndexer -task= externaldata:extdata
   ```

Configuring structure indexing

Introduction to structure indexing

The Structure feature is made up of several components:

- **Template**
  Contains the model and schema, the `tcserver` libraries, and so on.

- **Active Workspace client contributions**
  Contains the client exposure of the server features.

- **Indexer and translator additions**
  Contains indexing framework support and additions.

Indexing is part of the Active Content Structure feature that provides support for fast structure (BOM) navigation and in-context search capability leveraged by the Active Workspace client. For any given product configuration, there are two indexes maintained:

- An index within the Teamcenter database that is used for structure modeling and navigation.

- An index within Solr used for in-context searches. (Structure data is stored within the same Solr instance as object data but within a separate collection.)

The `runTcFTSIndexer` utility maintains both Teamcenter and Solr indexes. Structure indexing support is added into the TcFTSIndexer installation during the Teamcenter Environment Manager (TEM) installation process. It is expected that Active Workspace has already been installed and that you have verified it is working.

All Active Workspace indexing is expected to be orchestrated by a single TcFTSIndexer instance. This is the index orchestrator that performs orchestration functions for all types of indexing (object data, structure, and so on). It is expected that the orchestrator (TcFTSIndexer) is run with the `-service` argument, and separate commands are issued to control the indexing operations for the different indexing types. A given indexing action can be executed once or executed on a given interval.

If you plan to use the Dispatcher to distribute transformations, it is optional to dispatch the structure indexing transformations. Structure indexing transforms are quite efficient and can be run within the index orchestrator.
Tip

If Dispatcher is configured and you do not want to dispatch structure index transformation, you must change the value of the `structuretransformstep.usedispatcher` property to `false` in the `TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer_structure.properties` properties file.

Indexing design and structured content

Teamcenter maintains a separate index for structured content to ensure search results are always up-to-date. The administrator defines the structures and configurations to index, minimizing response times whenever a user searches for data.

The following diagram shows the indexing framework mechanism.

![Indexing Framework Diagram](image)

The following diagram shows the process that occurs when you create an index for structured content.

![Indexing Process Diagram](image)

To define structured content search indexes, you define property constants on structure elements in the Business Modeler IDE to specify the properties to index. The Teamcenter indexer then indexes those properties when it creates or updates the structured content index. You use the same property constants as you use with the basic Active Workspace search but attaches them to structure elements. Changing the structure filters follows the same process as other search filters.
You must reindex the structure are regular intervals to ensure that the search results reflect changes to the structure. Manual edits and system operations (for example, release and revise) on structure are reflected only after the indexer processes the changes.

Each top level of a structure is indexed for all the specified revision rules and effectivities. For example:

- **Latest Working, Released** creates one set of occurrences.
- **Released** creates one set of occurrences.
- **Released, Effectivity Milestone 1(1/1/2013)** creates one set of occurrences.

For best performance, Siemens PLM Software recommends you index only those structures and configurations that are frequently accessed.

**Note**

Whenever there is a change to a revision rule or a saved variant rule, you *must* reindex the structure.

For each searchable revision rule, the system creates a unique set of occurrences that includes all possible variations in the structure (sometimes referred to as a 150% BOM). Each indexed occurrence is stored with a bit mask that identifies the valid variant rule for the occurrence. The variant mask is exported in TC XML format and indexed in the Teamcenter indexer. Teamcenter can then process the bit mask to build a 100% BOM for any given variant rule.

The following diagram shows the process that occurs when a user changes index content and Teamcenter updates the index.

Teamcenter maintains the indexed status of each structure in the `Awb0BOMIndexAdminData` business object on the `awb0IndexState` property.
Index structure content data

Index the structure content by running the **bomindex_admin** utility, for example:

```
bomindex_admin -u=username -p=password -g=dba -logfile=C:\Scratch\log\log1.txt -function=create -inputfile=C:\Scratch\log\bomindex_admin_input.txt
```

Maintain the structure indexes by running the **runTcFTSIndexer** utility using the `structure` type.

Structure index life cycle

Over the course of an index life cycle there are 16 possible states. These states represent the current processing or condition the index is in.

Following is the life cycle of an index:

1. Created
   - Product configuration parameters are defined but the index data is not yet generated.

2. Active
   - The index data is generated and synchronized periodically.

3. Delete pending
   - The product configuration is marked for deletion, or the delete processing is occurring.

4. Deleted
   - There are no longer any artifacts relating to the index configuration.

The current index state and configuration details for product configurations are maintained within the Teamcenter database as unique **BOMIndexAdminData** table entries. As indexes for product configurations are created, maintained, and then eventually deleted, it is the **BOMIndexAdminData** table entry that tracks the configuration and state information. The data contained in the **BOMIndexAdminData** table is often referred to as product configuration, BIAD, or BIADInfo.

The **BOMIndexAdminData** entry for a given product configuration is created (and deleted) using the **bomindex_admin** command line utility.

The **runTcFTSIndexer** is responsible for orchestrating the index generation and synchronization of the indexes defined by the **BOMIndexAdminData** entries.

Structure indexing utilities

Overview of the bomindex_admin utility

The **bomindex_admin** utility is used to create the **BOMIndexAdminData** data to define a specific product configuration to be indexed. This utility is also used to mark a given product configuration ([**BOMIndexAdminData/BIAD**]) for deletion.

The **bomindex_admin** utility supports the following options:

```
bomindex_admin
   -u=user-id -p=password -g=group -function=action-to-perform -inputfile=input file -logfile=log-file
```

The actions supported by the function parameter are:
• **create**
  Creates the BIAD for a given product configuration.

• **delete**
  Deletes the BIAD for a given product configuration.

• **list**
  Lists the BIADs defined in the system.

A product configuration has many parameters, and they are passed to the utility using a text file specified using the `-inputfile` parameter.

The input file is required to have lines created in the following form:

```
item-query-string | item-revision-ID | base-revision-rule | effectivity-unit | effectivity-end-item-query-string |
effectivity-date (dd-mmm-yyyy hh:mm:ss) | variant-rules | MMV
```

The variant rules are comma delimited, and follow this format:

```
SVR-name:owning-item-query-string:owning-itemrevision-ID
```

The topline item revision is the default owner.

The MMV flag is optional and is used to designate that this product configuration will be indexed for Massive Model Visualization (MMV). MMV is a visualization technology that uses Visibility Guided Rendering (VGR) to increase performance and scalability for the viewing of extremely large 3D models, such as cars, airplanes, and ships. Use of MMV requires deployment of the Visualization Data Server.

When performing the same function (action) on multiple product configurations, each product configuration can be specified on separate lines of the input file. Create them one at a time to help manage any errors that may occur during the creation process.

**Structure indexing using TcFTSIndexer**

The TcFTSIndexer is used to orchestrate the required processing to create, synchronize, and eventually delete the indexes defined by the `BOMIndexAdminData` tables (BIADs). TcFTSIndexer can be run in a number of ways and is also used to perform object data indexing.

The TcFTSIndexer has an extensibility model that allows different indexing types to define the processing required using a sequence of steps organized in flows. Steps can be reused by multiple flows and different indexing types. This is different than the traditional QETL model where every action is plugged in as a part of query, extract, transform, or load.

Key concepts of the extensibility model are:

• **Type**
  Specifies the name of the indexing type (for example, `objdata`, `structure`).

• **Action**
  Specifies a name that ties a command line option to start a flow (for example, `sync` starts the synchronization flow).
• **Flow**
  Specifies the flow to execute (for example, `reindexflow`, `syncflow`).

• **Step**
  Specifies a single step that is executed as part of a flow.

With these concepts in mind, structure indexing is accomplished with:

• The **structure** indexing type.

• The **test**, **show**, **sync**, and **recoverfailures** actions.

• The **testflow**, **showflow**, **syncflow**, and **recoverfailuresflow** flows.

• A number of steps that are the building blocks of the various flows

The general syntax for starting any action is using the `runTcFTSIndexer` utility is:

```
runTcFTSIndexer -task=type:action [additional-arguments]
```

The `-help` argument lists all available actions. Do not use any actions described as Internal use only.

```
runTcFTSIndexer -help
```

The main actions provided by structure indexing are:

• **-task=structure:test**
  Performs basic tests, such as for Teamcenter logon, FMS connectivity, verifying or downloading of transform files, Solr schema, and so on. This command must be run exclusively; it cannot be run concurrently with other exclusive actions.

• **-task=structure:show**
  Shows a summary of all configured product configurations.

• **-task=structure:sync**
  Performs normal synchronization and delete actions for all product configurations. This command must be run exclusively. This command queues up all the synchronization actions for the product configurations. The queued synchronization actions are processed as resource permits.

• **-task=structure:syncone product-config-UID**
  Performs normal synchronization and delete actions for a single product configuration UID. This command must be run exclusively. This command queues up all the synchronization actions for the product configurations. The queued synchronization actions are processed as resource permits.

• **-task=structure:recoverfailures**
  Changes all product configurations with failed states to the **ReadyToIndex** state or the **MarkedForDeletion** state.

• **-task=structure:resetall**
Chapter 4: Configuration

Downloads the latest transform and schema files, resets all active product configurations to the ReadyToIndex state, and resets all deleted product configurations to the MarkedForDeletion state. This command must be run exclusively.

- **-task=structure:reset product-config-UID**

Resets the given PRODUCT_CONFIG_UID setting to the ReadyToIndex or MarkedForDeletion state. This command must be run exclusively.

Managing structure indexes

Overview of managing structure indexes

To manage a given structure index, it first must be created using the **bomindex_admin** utility with the **-function=create** option.

Once the product configuration is created, the TcFTSIndexer service automatically maintains the indexes keeping them synchronized with the Teamcenter data that defines the structure. The index is synchronized periodically based on the interval the synchronization operation runs using the following **runTcFTSIndexer** utility command:

```
runTcFTSIndexer -task=structure:sync -interval=seconds
```

Once the product configuration is no longer needed, the **bomindex_admin** utility is used to mark the product configuration for deletion using the **-function=delete** option. On the next synchronization interval, the TcFTSIndexer process deletes the indexes and then finally deletes the associated BOMIndexAdminData object that defined the product configuration.

Create a structure index

To create an index for a specific product configuration, use the **bomindex_admin** utility with the **-function=create** and **-inputfile** options. Determine the product configuration that defines the index. Create an input file with the format specified, configuring the index.

The input file line format is as follows:

```
item-query-string | item-revision-ID | base-revision-rule | effectivity-unit | effectivity-end-item-query-string | effectivity-date (dd-mmm-yyyy hh:mm:ss) | variant-rules
```

The variant rules are comma delimited, and follow this format:

```
SVR-name:owning-item-query-string:owning-item-revision-ID
```

The topline item revision is the default owner.

Following is an example of an input file (**bomindex_admin_input.txt**):

```
item_id=HDD-0527 | B | Any Status; Working | 5 | item_id=HDD-0527 | 31-May-2013 00:00:00 | vrule1: item_id=OwnItem1:B, vrule2:, vrule3: item_id=OwnItem3:A
```

Following is an example of running the **bomindex_admin** utility:

```
bomindex_admin -u=username -p=password -g=dba -function=create -inputfile=bomindex_admin_input.txt -logfile=bomindex_admin.log
```

Assuming there are no errors, this creates the required BOMIndexAdminData entries for the specified product configuration. At this point, the configuration required to generate and maintain the indexes exists, but there is no actual index data.
Save the input file for later use when the product configuration must be deleted.

Delete a structure index

When a given product configuration is no longer needed, mark the index for deletion so that the index artifacts can be cleaned up. Use the bomindex_admin utility to mark the product configuration for deletion using the same input file contents when the index was created.

Following is an example input file named bomindex_admin_input.txt:

```
item_id=HDD-0527 | B | Any Status; Working | 5 |item_id=HDD-0527 | 31-May-2013 00:00:00 | vrule1:item_id=OwnItem1:B,vrule2:,vrule3:item_id=OwnItem3:A

Following is the example bomindex_admin utility execution:

bomindex_admin -u:username -p:password -g:dba -function=delete
-logfile=bomindex_admin_input.txt

Assuming there are no errors, this marks the BOMIndexAdminData entries for deletion. On the next sync action, the index data and BOMIndexAdminData table entry are deleted. At the end of the sync action (delete), the show output is printed to the console:

```
AC gcNR4APF0cIC $JCig8 $M1Cg9C
AC gcUNanThWcIeC $JClg8 $M1Cg9C
```

Maintain structure indexes

View the current states of the structure indexes

Little should be required to maintain the structure indexes. The TcFTSIndexer service handles all processing as part of the -task=structure:sync -interval=seconds processing.

It is good practice to verify index states occasionally by using the -task=structure:show action. The show action prints a summary of information to the service console of all product configurations configured for indexing. Details of the configured indexers is shown in output as well as counts and status. When synchronization is not actively processing an index, all states should be 8. (See the following example.) If synchronization is actively processing a given index, the index can also be in various intermediate states. Within the show output, verify the state, the last update date, and that the TC counts and Solr counts match.

Run the following runTcFTSIndexer utility command:

```
runTcFTSIndexer -task=structure:show
```

The following example output in the service console shows all indexes are in state 8. The counts match and the last update dates are current.

```
2014-08-07 13:17:45,442 INFO - Running TcFtsIndexer Type: structure FlowAction: show
--- BOM Index Summary ---
Status Product Config UID Window UID State Name TC Count Solr Count
AC wkUN927DoR4_1D SkimFizjM1SdMD 8 HDD-0527/A;1-Hard Drive Assemb 364 364
AC wkUN927DoR4_1D bCCyVKhM1S0nA 8 HDD-0527/A;1-Hard Drive Assemb 364 364
AC wkUN927DoR4_1D oMQnKj3M1y1EC 8 HDD-0527/B;1-Hard Drive Assemb 195 195
AC wkUN927DoR4_1D 1wEFyXgN010TB 8 HDD-0527/B;1-Hard Drive Assemb 195 195
AC wseN927DoR4_1D SDftNHSN1y6wc 8 JCB-Fastrac/B;1-Tractor 9,968 9,968
AC wwRN927DoR4_1D 3GovLJUYN1yOyB 8 JCB-Fastrac/B;1-Tractor 9,968 9,968
```
The show action does require available Teamcenter connections. If the show command takes a long time to run it usually is the result of other actions within the indexer using the connections. Once a connection becomes available, the show action runs. The show output currently is only displayed in the TcFTSIndexer service console. The show output is also printed at the end of many of the structure indexer actions.

View the status of synchronization in progress

Another useful TcFTSIndexer option is -status. This prints status information about any currently running flows. The output in the syncdispatchstep sections provides details about current structure:sync processing.

Run the following command:

```
runTcFTSIndexer -status
```

The following output was gathered while the structure:sync command was in process. For structure:sync actions, the most useful information is contained in the syncdispatchstep output sections. Run the runTcFTSIndexer.bat -status command to see output similar to the following example:

```
------------------------------------------ syncdispatchstep ------------------------------------------
TaskId       Time     Status       StepInfo
U14977b3eddc0a802641379  0.00  Started  {ProductConfigUID=goZrkWxogSdyB, WindowUID=mSCFWDOMISBXC, Process=sync (update), Status=SOA call in progress, CurrentState=8 (SolrIndexGenSuccess), StateHistory=[8], LastStatusUpdate=2014-10-27 17:02:42 -0500}
                          Status: Created: 0 Started: 1 Done: 0 Error: 0 Total Time 0.00 Total Count 0
Step Summary            syncdispatchstep
                          Status: Created: 0 Started: 1 Done: 0 Error: 0 Total time for all Steps 0 sec Overall Time 5.314 sec
------------------------------------------ syncdispatchstep ------------------------------------------
TaskId       Time     Status       StepInfo
U1493c842c6c70a802641381  0.00  Started  {ProductConfigUID=goCrkWxogSdyB, WindowUID=ybqRdpVm1Sn1A, Process=sync (update), Status=Waiting for 1 permits, CurrentState=8 (SolrIndexGenSuccess), StateHistory=[8], LastStatusUpdate=2014-10-27 17:02:39 -0500}
                          Status: Created: 0 Started: 1 Done: 0 Error: 0 Total Time 0.00 Total Count 0
Step Summary            syncdispatchstep
                          Status: Created: 0 Started: 1 Done: 0 Error: 0 Total time for all Steps 0 sec Overall Time 5.469 sec
...                      
```

Because the sections and status output is scattered you may want to filter the output to show only the status lines. Run the runTcFTSIndexer.bat -status | find "ProductConf" command to see output similar to the following example:

```
U14977b3eddc0a802641379  0.00  Started  {ProductConfigUID=goZrkWxogSdyB, WindowUID=mSCFWDOMISBXC, Process=sync (update), Status=SOA call in progress, CurrentState=8 (SolrIndexGenSuccess), StateHistory=[8, 5, 8], LastStatusUpdate=2014-10-27 17:02:54 -0500}
U1493c842c6c70a802641381  0.00  Started  {ProductConfigUID=goCrkWxogSdyB, WindowUID=ybqRdpVm1Sn1A, Process=sync (update), Status=Waiting for 1 permits, CurrentState=8 (SolrIndexGenSuccess), StateHistory=[8], LastStatusUpdate=2014-10-27 17:02:39 -0500}
U14949b7e5e270a802641383  0.00  Started  {ProductConfigUID=goFrkWxogSdyB, WindowUID=OSe35nOMICsEB, Process=sync (update), Status=Waiting for 1 permits, CurrentState=8 (SolrIndexGenSuccess), StateHistory=[8], LastStatusUpdate=2014-10-27 17:02:54 -0500}
U1499a46af6c70a802641387  0.00  Started  {ProductConfigUID=gsSrkwXogSdyB, WindowUID=4WeUga39MIyOVC, Process=sync (update), Status=Waiting for 1 permits, CurrentState=8 (SolrIndexGenSuccess), StateHistory=[8],
```
This output shows the various product configurations, what initial state they were in, the states they were propagated through (for this sync action), and other status information about what types of operation are currently pending such as waiting on connection, SOA call in progress, or done.

### Managing failures

When a product configuration ends up in a failed state, it remains in that state until the administrator runs the `structure:recoverfailures` action. When that is run, any failed product configurations are returned to the initial state and the index is regenerated (re-indexed) on the next synchronization.

Run the following command:

```
runTcFTSIndexer -task=structure:recoverfailures
```

Following is a truncated example output. It shows a failed product is now in state 0.

<table>
<thead>
<tr>
<th>Status</th>
<th>Product Config UID</th>
<th>Window UID</th>
<th>State</th>
<th>Name</th>
<th>TC Count</th>
<th>Solr Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>gcRNanawXWcIEc</td>
<td>s3pvF9yM1ipFC</td>
<td>0</td>
<td>HDD-0527/B;1-Hard Drive Assemble</td>
<td>81</td>
<td>81</td>
</tr>
<tr>
<td>AC</td>
<td>gcRMr4APWcIEc</td>
<td>9Axq$7ymM1y6BD</td>
<td>8</td>
<td>SUB-479/A;1-holland-assy</td>
<td>2,254</td>
<td>2,254</td>
</tr>
<tr>
<td>AC</td>
<td>gcUNanawXWcIEc</td>
<td>$JCJg8_SMiCq9C</td>
<td>8</td>
<td>flipone_assembly/A;1</td>
<td>14</td>
<td>14</td>
</tr>
</tbody>
</table>

On the next synchronization interval, that product configuration's indexes regenerated, and in this case, it succeeds. Following is the truncated output:

<table>
<thead>
<tr>
<th>Status</th>
<th>Product Config UID</th>
<th>Window UID</th>
<th>State</th>
<th>Name</th>
<th>TC Count</th>
<th>Solr Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>gcRNanawXWcIEc</td>
<td>s3pvF9yM1ipFC</td>
<td>8</td>
<td>HDD-0527/B;1-Hard Drive Assemble</td>
<td>81</td>
<td>81</td>
</tr>
<tr>
<td>AC</td>
<td>gcRMr4APWcIEc</td>
<td>9Axq$7ymM1y6BD</td>
<td>8</td>
<td>SUB-479/A;1-holland-assy</td>
<td>2,254</td>
<td>2,254</td>
</tr>
<tr>
<td>AC</td>
<td>gcUNanawXWcIEc</td>
<td>$JCJg8_SMiCq9C</td>
<td>8</td>
<td>flipone_assembly/A;1</td>
<td>14</td>
<td>14</td>
</tr>
</tbody>
</table>

If a product configuration continues to fail, output generated during the synchronization processing, TcFTSIndexer logs, and tcserver syslog files should help diagnose the underlying issue.

A common source of errors is stopping the TcFTSIndexer while synchronization operations are in progress. If you wish to stop the TcFTSIndexer you should never kill the process while actions are being processed. Use the `-stop` option to stop the scheduling of any flows, then verify that all flows have stopped using the `-status` option, and then finally shut down the TcFTSIndexer process using Ctrl+C in the service console window.

### Restarting the indexer

Sometimes you may observe that the FTS indexer is stuck. The symptoms of this condition are one or more of the following:

- The indexer is not picking a product for indexing.

- When you run `TcFTSIndexer.bat` with the `-status` argument, it reports status as `Status=SOA call in progress`. This status does not change over a long period of time.

- One or more Teamcenter server process crashes is observed in the logs.

These symptoms indicate the Teamcenter server crashed while processing and the crash was not reported back to the FTS indexer by the pool manager or Web tier.
To resolve this condition, you should stop synchronization, then (after an interval) restart the TcFTSIndexer service and reset the stuck configuration.

Structure indexing details

Structure index states

There are various categories of index states:

- **Initial**
  
  Indicates either new index creation or that an existing index is marked for deletion. In the state output, the create state is **0 - ReadForIndexing** and the marked for deletion state is **10 - MarkedForDeletion**. These states are usually set by the `bomindex_admin` utility, but can also be set by certain TcFTSIndexer actions. When the next `sync` action occurs, any product configuration in these states is propagated to other states during processing.

- **Transitional**
  
  Tracks the intermediate progress while processing a given product configuration. Transitional states should naturally propagate to final or terminal (failed) states during `sync` processing. Transitional states are only expected to be encountered during `sync` processing. If a transitional state is encountered at the start of `sync` processing, that indicates the given product configuration was not able to complete index processing for an unknown reason and that product configuration is promoted to the closest failure state.

- **Final**
  
  Indicates index processing for the given product configuration completed normally (**8 - SolrIndexGenSuccess**). Although you might expect to have a final deleted success state as well, the final step in index delete processing is to delete the product configuration data for the index, along with the state information.

- **Terminal**
  
  Track failures at particular steps in index processing. Once these states are reached, no further processing occurs for the given product configuration. Failure states are set either:

    1. Directly as a result of a `processBOMIndex` SOA call when the `tcserror` process had an issue processing the index. Examine the `tcserror` syslogs for details about the failure. Examine the `TcFtsIndexer.log` file for the `ERROR` message and any other information that was sent back with the error.
    
    2. Or the failure was detected in the TcFTSIndexer process due to a failure or an unexpected event. Examine the `TcFtsIndexer.log` file for `ERROR` information.

  To recover from terminal states (failures) use the `structure:recoverfailures` action, which resets all failed product configurations to the appropriate initial state and the next `sync` action attempts to process them again.

You must be careful that TcFTSIndexer is never stopped while the `structure:sync` action is running. If it is stopped prematurely any indexes that were still being processed are in transitional states and are set to terminal states on the next `sync` action.
Complete structure index state list

Run the following `runTcFTSIndexer` utility command to see the list of structure index states:
```
runTcFTSIndexer -task=structure:show
```

Following is the complete list of index states:

0 – ReadyToIndex
Indicates an initial index state usually set by the `bomindex_admin -function=create` command. This state is also set by the `structure:reset`, `structure:resetall`, or `structure:recoverfailures` actions for active but failed indexes.

1 – IndexGenStarted
Indicates a transitional state set by the `processBOMIndex` SOA operation. It is set while the Teamcenter database index generation is in progress.

2 – IndexGenSuccess
Indicates a transitional state set by the `processBOMIndex` SOA operation. It is set when the Teamcenter database index generation is complete.

3 – IndexGenFailure
Indicates a terminal failure state set by the `processBOMIndex` SOA operation. It is set if the Teamcenter database index generation failed.

4 – IndexExportStarted
Indicates a transitional state set by the `processBOMIndex` SOA operation. It is set while the Teamcenter database index export is in progress.

5 – IndexExportSuccess
Indicates a transitional state set by the `processBOMIndex` SOA operation. It is set while the Teamcenter database index export is complete and the TC XML download is starting.

6 – IndexExportFailure
Indicates a terminal failure state set by the `processBOMIndex` SOA operation. It is set if the Teamcenter database index export or TC XML download fails.

7 – SolrIndexGenStarted
Indicates a transitional state set by structure indexer. It is set while the TcFTSIndexer is transforming and uploading the index data to Solr.

8 – SolrIndexGenSuccess
Indicates a final resting state set by structure indexer. It is set when the TcFTSIndexer has successfully updated Solr with the index data.

9 – SolrIndexGenFailure
Indicates a terminal failure state set by structure indexer. It is set if the TcFTSIndexer had a failure while transforming or uploading the index data.
10 – MarkedForDeletion
Indicates an initial delete state usually set by the bomindex_admin -function=delete command. It is also set by the structure:recoverfailures action for any deleted indexes that failed during delete processing.

11 – IndexDelStarted
Indicates a transitional state set by the processBOMIndex SOA operation. It is set while the Teamcenter database index data is being deleted.

12 – IndexDelSuccess
Indicates a transitional state set by the processBOMIndex SOA operation. It is set when the Teamcenter database index data delete is complete.

13 – IndexDelFailure
Indicates a terminal failure state set by the processBOMIndex SOA operation. It is set if the Teamcenter database index data delete failed.

14 – SolrIndexDelStarted
Indicates a transitional state set by the structure indexer. It is set while the TcFTSIndexer is deleting the Solr index data.

15 – SolrIndexDelSuccess
Indicates a transitional state set by the structure indexer prior to object deletion. After this state, the BOMIndexAdminData entry is deleted.

16 – SolrIndexDelFailure
Indicates a terminal failure state set by the structure indexer. It is set if the TcFTSIndexer had a failure while deleting the Solr index data.

17 – IndexGenSyncStarted
Indicates a transitional state set by the processBOMIndex SOA operation when an index was previously synchronized and is currently being synchronized again. (This is a synchronization update as opposed to an initial synchronization.)

Structure index state propagation
This is the actual state propagation that occurs when processing indexes. Not all of these states are necessarily seen in the indexer as many are expected to be transitioned during backend server processing. Following is a complete set of state propagations for various types of sync processing:

• sync (initial)
  0 - ReadyToIndex
  1 – IndexGenStarted (On failure, processing goes to 3 – IndexGenFailure and then stops.)
  2 – IndexGenSuccess
  4 – IndexExportStarted (On failure, processing goes to 6 – IndexExportFailure and then stops.)
  5 – IndexExportSuccess
7 – **SolrIndexGenStarted** (On failure, processing goes to 9 – **SolrIndexGenFailure** and then stops.)
8 – **SolrIndexGenSuccess** (Processing then stops.)

- **sync** (update)

8 – **SolrIndexGenSuccess**
17 - **IndexGenSyncStarted** (This is a transitional state set by the processBOMIndex SOA operation.)
4 – **IndexExportStarted** (On failure, processing goes to 6 – **IndexExportFailure** and then stops.)
5 – **IndexExportSuccess**
7 – **SolrIndexGenStarted** (On failure, processing goes to 9 – **SolrIndexGenFailure** and then stops.)
8 – **SolrIndexGenSuccess** (Processing then stops.)

- **sync** (delete)

10 - **MarkedForDeletion**
11 – **IndexDelStarted** (On failure, processing goes to 13 – **IndexDelFailure** and then stops.)
12 – **IndexDelSuccess**
14 – **SolrIndexDelStarted** (On failure, processing goes to 16 – **SolrIndexDelFailure** and then stops.)
15 – **SolrIndexDelSuccess** (The entry is deleted.)

**Show structure index output**

Run the following `runTcFTSIndexer` utility command to see the structure index status:

```
runTcFTSIndexer -task=structure:show
```

Following is sample output:

```
--- BOM Index Summary ---
Status   Product Config UID    Window UID    State Name                                TC Count  Solr Count
AC      wUN927DoR4_1D     StimEfrzjM1SdMD   8  HDD-0527/A;1-Hard Drive Assemb      364        364
AC      wUN927DoR4_1D     bCCyVUKdM1SoNA   8  HDD-0527/A;1-Hard Drive Assemb      364        364
AC      wkAN927DoR4_1D     oMOQnKJ5M1yEFc   8  HDD-0527/B;1-Hard Drive Assemb      195        195
AC      wkdN927DoR4_1D     lwfEyXsK5M1lOTB   8  HDD-0527/B;1-Hard Drive Assemb      195        195
AC      wseN927DoR4_1D     SDfkn5MNy6wC     8  JCB-Fastrac/B;1-Tractor              9,968      9,968
AC      wWn927DoR4_1D      35DVLJ3M1yCyB     8  JCB-Fastrac/B;1-Tractor              9,968      9,968
```

Following is an explanation of the columns:

- **Status**
  
  Indicates the general category of the index state. The output is sorted by the status category.

  - Active index status codes:
    
    - **AI**
      
      Flagged for initial index generation or re-index.
    
    - **AP**
      
      Indicates that `sync` process in progress.
    
    - **AC**

Indicates that the **sync** process is complete.

- **AF**
  Indicates that the **sync** process failed.

**Deleted index status codes:**

- **DI**
  Flagged for deletion.

- **DP**
  Indicates that deletion is in progress.

- **DF**
  Indicates that the deletion failed.

**Product Config UID**

Indicates the UID of the **BomIndexAdminData** object that identifies the product configuration details.

**Window UIDs**

Indicates the BOM window UID.

**State**

Indicates the state of the index. A state legend is included in the show output.

**Name**

Indicates the name of the top line of the product structure.

**TC Count**

Indicates the number of occurrences found in the database for this product configuration.

**Solr Count**

Indicates the number of occurrences found in Solr for this product configuration. (The counts should match.)

**Last update date**

Indicates the time when the index was last updated.

**Status syncdispatch output**

Run the following command:

```
runTcFTSIndexer -status | find "ProductConf"
```

Following is an example of the output in the **syncdispatchstep** section filtered to show only the structure status information:

```
U1477b73edd1c0a802641379  0.00  Started {ProductConfigUID=goZRkWxogg5DyB, WindowUID=nmSCEWD0M156X, Process=sync {update},
```
Status=SOA call in progress, CurrentState=8 (SolrIndexGenSuccess), StateHistory=[8, 5, 8], LastStatusUpdate=2014-10-27 17:02:54 -0500

Following is an explanation of the columns:

- **ProductConfigUID**
  Indicates the UID of the BomIndexAdminData object that identifies the product configuration details.

- **WindowUID**
  Indicates the BOM window UID.

- **Process**
  Indicates the type of processing that is occurring for the given product configuration. Following are available values:
  
  o **Dispatching**
    Indicates that dispatching is occurring. This is a transient message.

  o **sync (initial)**
    Indicates the first time the index is generated (for example, during re-index).

  o **sync (update)**
    Indicates an incremental sync update is occurring.

  o **sync (delete)**
    Indicates sync delete processing.

  o **promoting to failure (found in intermediate state)**
    Indicates that the given product configuration did not complete processing on the last sync action and is being promoted to a failure state.

  o **ignored (previously failed)**
    Indicates that the given product configuration was previously promoted to a failure state and is being skipped during this processing.

- **Status**
  Indicates fine-grained detail about the current processing. Possible values include:

  o **Waiting for # permits**
    Indicates that processing is waiting for the required number of permits to begin processing.
- **Waiting for connection**
  Indicates that processing is waiting for a Teamcenter connection.

- **SOA call in progress**
  Indicates that a Teamcenter SOA call is in process.

- **Download**
  Indicates that export files are being downloaded.

- **Transform**
  Indicates that export data is being transformed into Solr files.

- **Solr**
  Indicates that Solr is being updated.

- **Done**
  Indicates that process completed without error.

- **Failed**
  Indicates that an error occurred.

- **CurrentState**
  Indicates the current state of the given product configuration.

- **StateHistory**
  Indicates the history of the states recorded while processing this product configuration during this sync process. Some expected values are \[8, 5, 8\] (sync update) and \[0, 2, 5, 8\] (initial sync).

### Important structure content indexing files

The following files are used in structure content indexing:

- **Configuration file**

  `TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer_structure.properties` file

  This properties file configures the structure indexing including the type, flows, steps, and actions. Most of the content in this file should not be modified. There is one set of properties that configures control logic to limit the number of concurrent sync operations, weighing the different type of sync actions (initial versus update) differently. The reason for the different weighting is that initial sync processing is a much more resource intensive operation on the `tcserver` instance and database processes.

  Default values are shown in the file as follows:

  ```
  # control the number of concurrent structure syncs to limit server and oracle load
  permits.total=7
  permits.required.initialsync=3
  permits.required.updatesync=1
  ```
For one of the given types of sync actions to be allowed to run (initial or update), it must be able to obtain the required number of permits for that type. There is a limited (total) number of permits that can be used at any given time. Any sync operation that cannot obtain the required permits waits until other sync processing finishes making more permits available. This controls the load on the servers during the sync processing.

With the defaults provided, these are all the combinations of types of sync process that can concurrently run:

- 0 initials, sync actions and up to 7 updates sync actions
- 1 initials sync actions and up to 4 updates sync actions
- 2 initials sync actions and only 1 updates sync action

When making changes to these values also consider the maximum number of connections properties found in the TcFtsIndexer.properties file.

- Logging configuration file
  
  $TC_ROOT\TcFTSIndexer\conf\log4j.properties

  This file is used to change the logging level. (The default logging level for the TcFTSIndexer process is INFO.)

- Log file
  
  $TC_ROOT\TcFTSIndexer\logs\TcFtsIndexer.log

- Cache files
  
  $TC_ROOT\TcFTSIndexer\cache\*.cache

  These files may need to be deleted if Teamcenter preferences are changed that affect the Solr installation and FMS.

- Dispatcher files
  
  $dispatcher-root\Module\conf\translator.xml
  $dispatcher-root\Module\Translators\TcFTSIndexer\conf\TcFtsIndexer_structure.properties
  $dispatcher-root\Module\Translators\TcFTSIndexer\cache\*.cache
  $dispatcher-root\Module\Translators\TcFTSIndexer\conf\log4j.properties
  $dispatcher-root\Module\Translators\TcFTSIndexer\logs\TcFtsIndexer.log

  The translator.xml file configures the translator service (structuretransformstep).

### Configuring structured content

### Setting security on structured content

For nonindexed structures, standard Teamcenter is applied, for example, the Access Manager rule processing. When using indexing, Access Manager rules are included in the index as a read expression

For detailed information about setting ACLs, see Access Manager in the Teamcenter collection.
Configuring the properties of structured content

You can display the properties of the underlying object (called the archetype) in a structure element object tile. For example, you can show the release status of an item revision on the tile of an element in a structure using the following syntax: awb0Archetype.release_status_list.

For more information about defining object tiles, see Configuring the interface.

Cleaning up background working contexts

Teamcenter deletes background working contexts to free up disk space, according to their age and the maximum number of allowed contexts per user. To control this clean up, set the following preferences:

- **AWBBackgroundContextCleanupDays**
  
  Controls the maximum time working contexts are kept. For example, if this preference is set to 30 days, when a user opens a structure, all working contexts older than 30 days are deleted.

- **AWBBackgroundContextMaxCountPerUser**
  
  Controls the maximum number of working contexts kept for each user. For example, if this preference is set to 50, when a user opens a structure, the oldest working contexts from the set of that user’s contexts beyond the limit of 50 are deleted.

These settings applies to all users and it is not possible to set a different cleanup period for individual users.

Add custom objects to the Content tab and search

To display your custom business object in the Content tab of Active Workspace, you must add it to the Awb0SupportsStructure global constant in the Business Modeler IDE. This procedure also allows instances of the custom business object to be returned by in context searches if the object is indexed.

1. In the Business Modeler IDE, ensure the **Active Content Structure** template is loaded and then open the Global Constants Editor.
2. Select the **Awb0SupportsStructure** constant and then click **Edit**.
3. In the **Add** dialog box of the **Edit** window, enter the name of the class corresponding to the first custom object you want to display.
4. Repeat the previous step for each of the other custom objects you want to display.
5. Click **Finish** on both the **Modify Global Constant** and the **Add Value** dialog boxes to save the additions.
Note

On some systems, you may have to reopen each additional object in the Edit window and resave the entry for each added item. Failure to save your entries on the first attempt does not necessarily indicate they are incorrect.

6. To enable the custom objects to be returned by in context searches, set the Awp0SearchIsIndexed business object constant to true on each custom object.

Configuring the visualization components

Specify the address for the Teamcenter SOA service

You can specify the address for the Teamcenter SOA service which is the context root for the Teamcenter instance. In some reverse proxy deployments, the Visualization Pool Manager cannot access the outside reverse proxy address due to a firewall. This address is used to provide a direct path from the Visualization Pool Manager to the main Teamcenter SOA stack.

1. In a text editor, open the web.xml file that is contained within the awc.war file.

2. In the VisPoolProxy section, locate the soaPath parameter.

3. Update the value of the soaPath parameter with the URL of the Teamcenter server you want to use for loading 3D models.

Example

<init-param>
<param-name>soaPath</param-name>
<param-value>http:tc-serverURL:port#/tc/</param-value>
</init-param>
Configure viewer for Security Services-enabled environment

Before you can use the Active Workspace viewer when Security Services (SSO) is enabled, the following must be configured:

- You need just one login service deployed in Applet mode to be used by Visualization as well as the Active Workspace client.

- The Teamcenter thin client must be enabled for Security Services and configured with a Login Service that is deployed in applet mode. See Security Services Installation/Customization in the Teamcenter help collection.

  Note
  
  You must configure the Identity Service with two applID values: one for Active Workspace and another for Teamcenter thin client.

  The Teamcenter server must be configured for single sign-on with both applID values separated by a comma or space. For example:

<table>
<thead>
<tr>
<th>applID value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Workspace</td>
<td>applIDAW</td>
</tr>
<tr>
<td>Teamcenter thin client</td>
<td>applIDTC</td>
</tr>
<tr>
<td>Teamcenter server</td>
<td>applIDAW, applIDTC</td>
</tr>
<tr>
<td>Teamcenter rich client (not required for viewer)</td>
<td>applIDTC</td>
</tr>
</tbody>
</table>

- Once configured, verify you can log onto the Teamcenter thin client from the Visualization Server machine. You may need to enable Java applets for the browser.

- Specify the address for the Teamcenter SOA service.

  Note
  
  Ensure you add a trailing slash (/) to the end of the URL.
Note

• After working in one Active Workspace session for more than 10 hours, the viewer may fail to launch. Log off Active Workspace and log on again to open the viewer.

• If you remain logged on overnight, Security Services may time out. If it does, log on again.

• If your password has certain special characters, such as the percent sign (%), the viewer does not work. Change your password to one that omits special characters.

Locales supported for Visualization Servers

You can configure the Active Workspace client to display the user interface in any of the supported Teamcenter locales. However, some visualization data such as Product and Manufacturing Information (PMI) requires a Visualization Server configured for the same locale as the information. For visualization data to display correctly in the Active Workspace client, you must have at least one Visualization Server Manager configured to run in each locale for which you have data. With this system in place, visualization processes are then routed to the appropriate server based on locale.

Visualization Server Managers support the following languages:

• English
• French
• German
• Italian
• Spanish
• Chinese (Simplified)
• Chinese (Traditional)
• Japanese
• Korean
• Russian

You can configure a Visualization Server Manager with any one of these languages. If you want to configure a cluster of Visualization Servers to support more than one language, you need at least one Visualization Server Manager per language.

To change the language of a Visualization Server Manager, set Windows to the required language, location, and locale. You can adjust these settings using the Region and Language options found in the Windows Control Panel. You must adjust the Date and time formats, the Current location, and the Current language for non-Unicode programs values. After changing your Windows settings, reboot the system. When the Visualization Server Manager is started again, it inherits the new language configuration of the operating system.
If all Visualization Server Managers are configured to use the same language, all clients use the available language regardless of browser preferences.

If you have a Visualization Server system configured for two or more different languages, Siemens PLM Software highly recommends that at least one of the Visualization Server Managers be in English, even though this may require a minimum of three Visualization Server Managers. When the server system is configured with multiple languages, if you have a Visualization Server Manager in English, the English locale is the default. In this scenario, if a client requests use of the visualization system but that client is not in one of the pre-configured languages, the client is automatically routed to the default English server. If there is no English server in this case, there is no default server and the client is rejected.

**Note**

Visualization Servers do not currently support the Polish, Czech, and Brazilian Portuguese locales. If your organization works with visualization data from any of these locales, make sure to have at least one Visualization Server Manager set to English.

**Optimizing Visualization Server system performance**

The Visualization Pool Assigner and Visualization Server Manager have several threshold configuration parameters that you can adjust to optimize your server system. If a user attempts to load a 3D model while the values of these parameters are exceeded, Active Workspace displays the following message:

The visualization servers are busy. Please try again later to inform your system administrator that more capacity is needed.

- **maxBytesPerSec**

The Visualization Pool Assigner and Visualization Server Manager both possess a `maxBytesPerSec` configuration parameter. This parameter represents the maximum number of bytes per second that this node may transmit or receive before it rejects requests from users to load new models. The default value is 125000000 bytes per second. If your server system is consuming too much network bandwidth, you may want to consider adjusting this parameter.

To modify this configuration parameter for the Visualization Pool Assigner, do the following:

1. In a text editor, open the `web.xml` file that is contained within the `awc.war` file.

2. Add the `maxBytesPerSec` parameter to the `VisPoolProxy` servlet section of the file as shown:

```xml
<init-param>
  <param-name>launchMode</param-name>
  <param-value>Assigner</param-value>
</init-param>
<init-param>
  <param-name>cacheConfigFile</param-name>
  <param-value>/com/teamcenter/thinclient/resources/TreeCache.xml</param-value>
</init-param>
<init-param>
  <param-name>maxBytesPerSec</param-name>
  <param-value>125000000</param-value>
</init-param>
```
3. Adjust the value of the `maxBytesPerSec` parameter for your server environment.

To modify this configuration parameter for the Visualization Server Manager, do the following:

1. In a text editor, open the `jetty-service.properties` file from the in the Visualization Server Manager’s installation.

2. Adjust the value of the `VisPoolproxy.maxBytesPerSec` parameter for your server environment.

- **maxUsageThreshold**

The Visualization Pool Assigner and Server Manager possess a `maxUsageThreshold` configuration parameter. This parameter represents the maximum usage and load ratio that any Visualization Server Manager or VisView process may possess before the Visualization Pool Assigner refuses to allocate new open model requests to that Visualization Server Manager or VisView process. The Visualization Pool Assigner and Server Manager do not assign any more users to a particular VisServer if its system load (an amalgam of network usage, CPU usage, memory usage, and GPU memory usage) exceeds the given threshold specified with this parameter. The range of values accepted is 0.0 (no load) to 1.0 (full load).

To modify this configuration parameter for the Visualization Server Manager, do the following:

1. In a text editor, open the `web.xml` file that is contained within the `awc.war` file.
   
   a. Add the `maxUsageThreshold` parameter to the `VisPoolProxy` servlet section of the file as shown:

   ```xml
   <servlet>
   <servlet-name>VisPoolProxy</servlet-name>
   <servlet-class>com.teamcenter.vis.proxy.Proxy</servlet-class>
   <init-param>
     <param-name>launchMode</param-name>
     <param-value>Assigner</param-value>
   </init-param>
   <init-param>
     <param-name>cacheConfigFile</param-name>
     <param-value>/com/teamcenter/thinclient/resources/TreeCache.xml</param-value>
   </init-param>
   <init-param>
     <param-name>maxUsageThreshold</param-name>
     <param-value>.7</param-value>
   </init-param>
   <load-on-startup>1</load-on-startup>
   </servlet>
   
   b. Adjust the value of the `maxUsageThreshold` parameter for your server environment.

2. In a text editor, open the `jetty-service.properties` file in the Visualization Server Manager’s installation.

3. Adjust the value of the `VisPoolproxy.maxUsageThreshold` parameter for your server environment.
Enable Visualization Server Manager logging

The Visualization Server Manager logs capture system information that you can use to identify and resolve problems.

- Create the following environment variables:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCVIS_LOGGING_ENABLE</td>
<td>True</td>
</tr>
<tr>
<td>TCVIS_LOGGING_LEVEL</td>
<td>Any of the following:</td>
</tr>
<tr>
<td></td>
<td>ERROR</td>
</tr>
<tr>
<td></td>
<td>INFO</td>
</tr>
<tr>
<td></td>
<td>DEBUG</td>
</tr>
<tr>
<td>TCVIS_LOGGING_PATH</td>
<td>A valid path to the location where you want the system to write the log files. If this environment variable is not set, the generated VisView####.log files (one log file per VisView process) is placed into the jetty’s TEMP directory.</td>
</tr>
<tr>
<td></td>
<td>If TCVIS_LOGGING_ENABLE and TCVIS_LOGGING_LEVEL are set properly, this is an optional parameter.</td>
</tr>
</tbody>
</table>

**Example**

With TCVIS_LOGGING_LEVEL set to DEBUG, the log files include information about how the system is using OpenGL. When setting up the Visualization Server hardware, you can use this information to ensure the graphics hardware is operating as expected.

For example, the information in the sample output below indicates the system is not configured properly, because it is utilizing software rendering.

```
Running on GL_VERSION: 1.1.0 (1.1.0 supported by driver)
Running on GL_VENDOR: Microsoft Corporation
Running on GL_RENDERER: GDI Generic
```

**Note**

When a Visualization Server process crashes, a log file with crash information is always written. If Visualization Server logging is not enabled, the log file is written to the system TEMP directory.

Configure automatic logon for Windows

After starting the Visualization Server Manager, you must remain logged on to the system. The Visualization Server Manager requires access to the graphics card and consequently cannot run as a Windows service due to operating system limitations. In the event of a system reboot, you can use Windows automatic logon feature to automatically log on to the machine, whereupon the Visualization Server Manager is restarted with a script.
This procedure is applicable for Windows 7, 8, 8.1, and Server 2012 R2.

**Caution**

Enabling automatic logon is not a good practice because it bypasses security. When Windows is configured to automatically log on, anyone with physical access to the machine can restart it to gain entry to the system. Only use automatic logon when the system is in a secure environment.

1. Press the Windows key+R to display the Run dialog box.
2. In Open, type `netplwiz`.
3. Click OK.

   The User Accounts dialog box is displayed.
4. Select a user account from the list.
5. Clear the **Users must enter a user name and password to use this computer** check box.
6. Click **Apply**.
   The **Automatically sign in** dialog box is displayed.

7. In the **Password** and **Confirm Password** boxes, type the user’s password.

8. Click **OK**.
   Now the specified user is automatically logged on when Windows is started.

9. Create a script or batch file to launch the Visualization Server Manager. Include the following command to lock the workstation:

   ```
   rundll32.exe user32.dll LockWorkStation
   ```

10. Create a new task with Windows Task Scheduler to run the script or batch file at log on.

### Configure the Visualization Pool Assigner for JMX metrics

The Visualization Pool Assigner exposes a variety of useful information through JMX, but not all of this information is available automatically. Due to stability risks, the following JMX metrics are unavailable for the Visualization Pool Assigner without special configuration:

- `computerCpuUsageRatio`
- `computerMaxBandwidthBytesPerSec`
- `computerMemUsageRatio`
- `computerNetworkUsageRatio`
- `computerTotalMemMB`
- `loadRatioAbsolute`
- `loadRatioRelative`
- `visSysCpuUsageRatio`
- `visSysMemUsageRatio`
- `visSysNetworkUsageRatio`

Although these metrics appear in a JMX client, their values are not populated with useful information. To enable these JMX metrics for your Visualization Pool Assigner, you must copy a provided DLL into the path of the Java servlet container hosting your Visualization Pool Assigner.

1. Locate the `awc.war` file installed by TEM on your file system.

2. Open the `awc.war` file using an unzipping program.

3. Navigate into the `WEB-INF\lib\` area and locate the `vis-proxy` jar file.

4. Open `vis-proxy` using an unzipping program and extract `Metrix.dll`.

5. Add the location of `Metrix.dll` to the path of the server that hosts your Visualization Pool Assigner. The easiest way to do this is to find the script that starts the server, or to create a new script to start the server if one does not already exist. In the script, prepend the location of the `Metrix.dll` to your path.
If you place **Metrix.dll** in **C:\foo**, add the following command to the script:

```plaintext
set path=C:\foo;%path%
```

6. **Restart your Visualization Pool Assigner.**

   If the Visualization Pool Assigner fails to find **Metrix.dll**, the following warning is printed to the console on startup:

   **The Metrix library could not be loaded. Some system performance metrics are unavailable to JMX clients.**

   If the Visualization Pool Assigner succeeds in finding **Metrix.dll**, no warning is displayed and the JMX metrics are populated with meaningful data.

### Specify viewer defaults using Teamcenter preferences

You can change the default settings for many of the Active Workspace visualization features by changing their preference values in the Teamcenter rich client.

<table>
<thead>
<tr>
<th>Visualization feature</th>
<th>Default</th>
<th>Teamcenter preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D navigation mode</td>
<td>Rotate</td>
<td><strong>AWC_visNavigationMode</strong></td>
</tr>
<tr>
<td>Front view orientation</td>
<td>+Y</td>
<td><strong>AWC_visStdViewOrientationFront</strong></td>
</tr>
<tr>
<td>Left view orientation</td>
<td>+X</td>
<td><strong>AWC_visStdViewOrientationLeft</strong></td>
</tr>
<tr>
<td>Top view orientation</td>
<td>–Z</td>
<td><strong>AWC_visStdViewOrientationTop</strong></td>
</tr>
<tr>
<td>Display mode (shaded with edges on or off)</td>
<td>Shaded with edges off</td>
<td><strong>AWC_visShading</strong></td>
</tr>
<tr>
<td>Material</td>
<td>Flat</td>
<td><strong>AWC_visMaterial</strong></td>
</tr>
<tr>
<td>Floor visibility</td>
<td>Off</td>
<td><strong>AWC_visFloorOn</strong></td>
</tr>
<tr>
<td>Floor offset</td>
<td>0.0</td>
<td><strong>AWC_visFloorOffset</strong></td>
</tr>
<tr>
<td>Floor orientation</td>
<td>XZ</td>
<td><strong>AWC_visFloorPlaneOrientation</strong></td>
</tr>
<tr>
<td>Floor grid visibility</td>
<td>When the floor is on, the grid is displayed.</td>
<td><strong>AWC_visGridOn</strong></td>
</tr>
<tr>
<td>Reflection visibility</td>
<td>When the floor is on, the reflection is not displayed.</td>
<td><strong>AWC_visReflectionOn</strong></td>
</tr>
<tr>
<td>Shadow visibility</td>
<td>When the floor is on, the shadow is not displayed.</td>
<td><strong>AWC_visShadowOn</strong></td>
</tr>
<tr>
<td>Trihedron visibility</td>
<td>On</td>
<td><strong>AWC_visTrihedronOn</strong></td>
</tr>
</tbody>
</table>

### Adjusting the display resolution for 3D models

When you display and manipulate a 3D model in Active Workspace, the Visualization Server renders images of the model and sends them to the client viewer for display. The images are typically the same size as the client viewer. However, if the client viewer is larger than the desktop resolution of the Visualization Server, the images are the same size as the server resolution and scaled to fit the client viewer.
If you expand the client viewer to the point where it exceeds the server resolution, the scaled images may appear soft, especially the edges of 3D models. To compensate for this, increase the desktop resolution on the Visualization Server.

It is recommended that the system administrator consider the maximum client resolution that is needed to show crisp images, and set the server desktop resolution accordingly. Note that there is no fixed overhead to setting a high server desktop resolution, as clients typically use a lower resolution and the corresponding server rendering and network image transfer happen at that lower client resolution.

**Configuring Active Workspace with Teamcenter Security Services**

**Configure logoff for Active Workspace**

There are three possible scenarios for Active Workspace logoff. The logoff behavior changes based on whether Teamcenter Security Services is being configured or not, and if so, how it is configured for authentication.

If Active Workspace is being configured to use Teamcenter Security Services behind an authenticating gateway such as SiteMinder, WebSeal, IIS or Apache, this is a special case that requires additional configuration.

This is the default behavior to expect for each scenario.

- **Active Workspace is using Teamcenter authentication without Teamcenter Security Services:**
  1. User clicks the **logout** button.
  2. Teamcenter clears the **tcs** session.
  3. Active Workspace presents its logon page after successful logout.

- **Active Workspace is configured using Teamcenter Security Services, and Teamcenter Security Services is configured to authenticate using an LDAP server:**
  - **Active Workspace is launched in standalone:**
    1. User clicks the **logout** button.
    2. Teamcenter clears the **tcs** session and the Teamcenter Security Services session.
    3. Active Workspace presents the logon page after successful logoff.
  - **Active Workspace is participating in a Single Sign-On session with another Teamcenter client that has launched a session agent applet:**
    1. User clicks the **logout** button.
    2. Teamcenter clears the **tcs** session.
    3. User is presented a page stating **You are logged out of Teamcenter application however your TcSS session is still active.**
4. This page will also have a **Login Again** button. When the user clicks this button, they are directed to Active Workspace home page without challenge as long as Teamcenter Security Services session is still valid.

- **Active Workspace is configured with Teamcenter Security Services, and Teamcenter Security Services is behind an authenticating gateway:**
  1. User clicks the **logout** button.
  2. Teamcenter clears the **tcserver** session, but it does not close the authenticating gateway session by default.
  3. **Active Workspace does not present a page by default.**

  **Note**
  The third scenario requires additional configuration during the initial installation of Active Workspace. You must configure a logoff landing page.

**Configuring for a two-way SSL proxy server**

If Active Workspace is configured behind a gateway/proxy server, you must provide the URL for the gateway/proxy server by providing a value for the **localServerURL** parameter in the **web.xml** file that is contained within the **awc.war** file.

```xml
<init-param>
  <!-- Points to the host/port of the local AW server. If undefined, the value will be attempted to be generated based on incoming requests.

  ***In the case of reverse proxy, please set this value.***
  Example: http://localhost:7001

  Please only include the host and port, the application context will be determined automatically.
  -->

  <param-name>localServerURL</param-name>
  <param-value></param-value>
</init-param>
```

The value for this parameter is not defined by default—it must be added within the **AuthenticationFilter** filter definition.

**Configuring a logout landing page**

A logout landing page must be configured when Active Workspace is configured with TcSS which in turn is configured behind an authenticating gateway, such as SiteMinder, WebSeal, IIS or Apache. In this scenario the authentication is handled by the gateway and when a user clicks on the logout button, Teamcenter redirects the user to this landing page. This could be any page, such as an internal corporate site that provides access to various systems. To change the default logout landing page, you must provide the URL for the server by providing a value for the **tcSSOLogoutURL** parameter in the **web.xml** file that is contained within the **awc.war** file.

```xml
<init-param>
  <!-- This parameter is only used if TcSS is configured.

  This parameter defines the URL to redirect the user after logout is invoked. If undefined, the user will be returned to Active Workspace.
  -->

  <param-name>tcSSOLogoutURL</param-name>
  <param-value></param-value>
</init-param>
```
Configure multiple application IDs

In a single-signon Teamcenter deployment, you can configure a single instance of a Teamcenter server to support more than one web tier. For example, an Active Workspace client and a traditional thin client can be deployed for the same server instance. Each client needs a distinct application ID (AppID) configured in Security Services to associate with their return URL. To accommodate this situation, the Security Services identity server supports compound AppIDs.

For example, an Active Workspace client uses an AppID named TCAWC and a legacy thin client uses an AppID named TCthin. In the tc_profilevars file, configure the compound AppID as TCAWC,TCthin or TCAWC TCthin. The comma or space separates the individual AppIDs. The Teamcenter server sends that entire string as a token validation call parameter.

In addition to making the change in the tc_profilevars file, in the Security Services Identity Service you must configure an AppID for each of the clients (for example, TCAWC and TCThin) and include the appropriate return URLs.

Note

The Teamcenter thin client must be enabled for Security Services and configured with a Login Service that is deployed in applet mode, as described in Security Services Installation/Customization in the Teamcenter help collection.

Active Workspace needs to be configured with the same Login Service running in Applet mode. You must configure the Identity Service with two AppID values: one for Active Workspace and another for the Teamcenter thin client. The Teamcenter server must be configured for single sign-on with both AppID values separated by a comma or space. For example:

<table>
<thead>
<tr>
<th></th>
<th>AppID value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Workspace</td>
<td>appIDAW</td>
</tr>
<tr>
<td>Teamcenter thin client</td>
<td>appIDTC</td>
</tr>
<tr>
<td>Teamcenter server</td>
<td>appIDAW, appIDTC</td>
</tr>
<tr>
<td>Teamcenter rich client (not required for viewer)</td>
<td>appIDTC</td>
</tr>
</tbody>
</table>

Configuring relations

Localize names that appear in the custom Relations Legend views

To localize the custom view or the names that appear in the custom Relations Legend views that you created:

1. In Teamcenter rich client, search for the dataset RelationsB*.
2. In the Search pane, right-click `RelationBrowserConf` and select Named References.

3. In the Named References dialog box, select RB_UIConfigure.xml and click Export.

4. Open the RB_UIConfigure.xml file and update the value of the view name property to localize the name of the custom view and the filter name property to update the name of the relations and object names. For example, to localize the names of relations and objects that are in the Design view in Chinese, update the value of the filter name property with the localized name as follows:

   ```xml
   <view name="Design">
       <ruleName>GenericRule</ruleName>
       <defaultLayout>IncrementalHierarchic</defaultLayout>
       <group name="relations">
           <filter name="Attach" parameterSet="Attach" color="(64,100,142)"/>
           <filter name="Impact" parameterSet="Impact" color="(243,130,37)"/>
           <filter name="Master" parameterSet="Master" color="(167,153,80)"/>
           <filter name="Structure" parameterSet="BOMStructure" color="(138,66,8)"/>
           <filter name="Traceability" parameterSet="Traceability" color="(255,182,121)"/>
           <filter name="WhereUsed" parameterSet="WhereUsed" color="(149,179,215)"/>
       </group>
       <group name="objects">
           <filter name="Change" parameterSet="Change" color="(243,130,37)"/>
           <filter name="File" parameterSet="Dataset" color="(202,216,234)"/>
           <filter name="Physical" parameterSet="Physical" color="(138,66,8)"/>
           <filter name="Requirement" parameterSet="Requirement" color="(64,100,142)"/>
       </group>
   </view>
   ```

5. Check out the RelationBrowserConf dataset.

6. Delete the original RB_UIConfigure.xml file from the Named References dialog box and import the new RB_UIConfigure.xml file.

7. Click Close to close the Named References dialog box.

8. Check in the RelationBrowserConf dataset.

Configure the style of nodes and edges

You can change the style of nodes and edges such as change the line style as follows:

1. In Teamcenter rich client, search for the RelationsB* dataset.

2. In the Search pane, right-click RelationBrowserStyle and choose Named References.

3. In the Named References dialog box, select GraphStyle.xml and click Export.

4. Open the GraphStyle.xml file and make the necessary changes. For example, to change the line style of the traceability relation to dash, update the linestyle option under the Traceability Style section from SOLID to DASH.

   ```xml
   <EdgePresentation id="TraceabilityStyle">
       <parameter name="lineStyle">
           <value>DASH</value>
       </parameter>
   </EdgePresentation>
   ```
5. Checkout the **RelationBrowserStyle** dataset.

6. Delete the original **GraphStyle.xml** file from the **Named References** dialog box and import the new **GraphStyle.xml** file.

7. Click **Close** to close the **Named References** dialog box.

8. Check in the **RelationBrowserStyle** dataset.

### Configuring requirements

#### Prerequisites for requirements

To work with requirements, the **Active Workspace Requirements Management feature** must be installed.

The following are required to enable all the features of requirements, including editing with Microsoft Word.

**Note**

The functionality for editing requirements in Word differs from opening a Word **attachment** to a Teamcenter object. To enable opening attachments, you must also install Teamcenter **Active Workspace launcher** after the step for installing .NET 4.5.

- During installation, the following must be installed:
  - Microsoft Office
Microsoft Office must be installed prior to installing Teamcenter Client for Microsoft Office. The installation must include the .NET Programmability Support installation option.

- Teamcenter Client for Microsoft Office
  Teamcenter Client for Microsoft Office has additional installation requirements, including .NET 3.5 SP1. For more information, see Client for Microsoft Office in the Teamcenter online help.

- Teamcenter Extensions for Microsoft Office
  For more information, see Extensions for Microsoft Office in the Teamcenter online help.

- .NET 4.5
  This is installed after .NET 3.5 SP1 and Teamcenter Client for Microsoft Office. .NET 4.5 is also a prerequisite of Dispatcher.

- If Dispatcher is installed, these prerequisites pertain to requirements:
  - Windows Server operating system
  - .NET 4.5

  - Set the required preferences.
  - Enable trace links.
  - Convert requirements created prior to Active Workspace 2.3, using the req_bulk_upgrade_docm_to_html utility.

  **Note**
  To use this utility, the ReqMgmtWordToHtmlTrans translator must be installed. This translator also converts requirement content imported to Teamcenter or edited and saved in Microsoft Word from Teamcenter (stored as a full-text dataset), so that it can be viewed in the rich text editor in Active Workspace.

  The translator does not convert Word files (Microsoft Word datasets) that are stored in the system and may be related to requirement objects or other objects.

  Dispatcher must be running for the translator to perform conversions.

**Set preferences for requirements**

Set the following Teamcenter preferences for using requirements in Active Workspace.

- **AWC_req_do_html_conversion**
  Performs the HTML conversion of requirements content when requirements are imported and updated in Teamcenter Systems Engineering.

- **AWC_req_viewer_page_size**
  Defines the number of requirements that can be viewed at a time in the Active Workspace Content—Viewer tab.
• **AWC_ReqEditor**
  Defines the type of editor that opens when paragraphs, requirements, or requirements specifications are opened for editing.

• **Default_Transient_Server**
  Specifies the default transient file server location. Environment variable settings override this FCC configuration file setting.
  This preference is required to enable the editing of requirements in Microsoft Word.

• **Fms_BootStrap_Urls**
  Determines which FMS server cache manages file downloads. When searching for an assigned FMS server cache to manage file downloads, the thin client contacts the FSC servers defined in this preference in the order listed. The server responds with the FSC server assigned to the thin client, and all subsequent communication is with that assigned server cache. Environment variable settings override this FCC configuration file setting.
  This preference is required to enable the editing of requirements in Microsoft Word.

• **WEB_default_site_deployed_app_name**
  Appends the name of the application to be launched to the server URL defined in the **WEB_default_site_server** preference.
  This preference is required to enable the editing of requirements in Microsoft Word.

• **WEB_default_site_server**
  Sets the server URL of the default site server used for server related tasks. You must also set the **Web_default_site_deployed_app_name** preference, which appends the name of the application to be launched to this URL.
  This preference is required to enable the editing of requirements in Microsoft Word.

• **WEB_protocol**
  Defines the protocol used by the web server; either http:// or https://
  This preference is required to enable the editing of requirements in Microsoft Word.

**Enable trace links for requirements**

A trace link is a directional relationship between requirement specifications, functional models, logical models, or physical models. In the relationship, one object is the defining object and one is the complying object. The defining object specifies a condition that a product or a component must fulfill. The complying object must partially or completely fulfill the condition specified by a defining object.

In order to create and use trace links, the trace link functionality must be enabled.

1. In the Teamcenter rich client, choose **Edit→Options**.
2. In the **Options** dialog box, select **Systems Engineering**.
3. Select the **Trace Link Mode** check box.
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.

**Configuring product line engineering**

**Product line engineering general configuration**

Before using product line engineering (PLE), you must configure the following:

- Ensure the **Request Client** and **Request Server** features are installed when you run Teamcenter Environment Manager (TEM).
- Set the required analysis request statuses on the **EnabledDefnStatusList** preference.
- Define deep copy rules used when revising analysis requests in the **Ctr0ValidationContract** template.
- Create a workflow that applies an approved status to a completed attribute definition, so making it available for general use. Optionally, you can add review and signoff tasks that must be performed before the attribute definition is released.
- Set the release status that allows an attribute definition to be consumed on the Teamcenter **PLE_EnableDefnStatusList** preference.
- Enable or disable the editing of the domain bookmark to remove associated components.

When they define an analysis request, users may release it by submitting it to the appropriate workflow. By default, once it is released, you can no longer make changes to the contents of the analysis request. However, you can still remove tracelinks associated with it, effectively removing associated components. Likewise, it is possible to edit the domain bookmark to add or remove components. If you want to change this default behavior, change the **Fnd0PreventTracelinkDelete** constant to **Primary** to disallow such removals.

- Create a workflow to release an analysis request.

**Configuring analysis definition templates**

Each analysis request is based on a template called an analysis definition. You must create an analysis definition for each request type that a user can create.

Each analysis definition is specified in an XML file that contains the following:

- **Section name**
  Specifies the displayed name of the tab.

- **Section group**
  Specifies the group that the section belongs to, one of **View** and **Results**.

- **Object types**
  Specifies one or more object types that can be added to this section.
Chapter 4: Configuration

- **Quantity**
  The number of objects of the relevant type that can be added.

- **Properties**
  The properties that display in the section. The same properties display for all types.

A default analysis configuration file is created whenever an analysis definition object is created. You can replace or remove the configuration file if necessary using any XML editor, and then validate it against the specified XML schema.

An example of the required format follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="Crt0ContractPkgSchema.xsd">
  <view>
    <section displayName="Vehicle Context">
      <objectType type="Fnd0LogicalBlockRevision" quantity="1"/>
      <objectType type="Att0TargetRevision" quantity="1"/>
      <objectType type="PSConnectionRevision" quantity="1"/>
      <property name="object_string"></property>
      <property name="object_type"></property>
      <property name="object_desc"></property>
    </section>
  </view>
  <results>
    <section displayName="System-of-Interest">
      <objectType type="RequirementSpec Revision" quantity="1"/>
      <property name="object_string"></property>
      <property name="object_type"></property>
      <property name="object_desc"></property>
    </section>
    <section displayName="Test Scenario(s)">
      <objectType type="ItemRevision" quantity="2"/>
      <property name="object_string"></property>
      <property name="object_type"></property>
      <property name="object_desc"></property>
    </section>
    <section displayName="Simulation Model Configuration(s)">
      <objectType type="Fnd0LogicalBlockRevision" quantity="*"/>
      <property name="object_string"></property>
      <property name="object_type"></property>
      <property name="object_desc"></property>
    </section>
  </results>
</configuration>
```

Once validated, you should copy the file to the **TC_DATA** directory or other location on each user's system.

When a user creates an analysis request from an analysis definition, the system checks the XML file for correct syntax and data model errors, and then displays an error message if any are identified.

**Configure property formatters**

**About property formatters**

Property formatters are display definitions that dictate how property values are shown in the user interface. For example, the internal database value for a property might be 25.693850392, but the property formatter attached to the property specifies showing only two decimal places, so it displays as 25.69. The value in the database is still the same. Only the value in the user interface is displayed in a more usable way.
Property formatters can be applied to boolean (true/false), double (decimal space), integer (numerical), and string (alphanumeric text) properties.

**Create property formatters**

You can create your own property formatters by defining a preference as follows:  
*object_type.property_name.*FORMATSTRING.*

1. In the Teamcenter rich client, open My Teamcenter, choose Edit→Options, and select the Index link.

2. Select the New link in the Preferences dialog box.

3. In the **Create New Preference** section, specify the name of the preference using the following format:
   
   *object_type.property_name.*FORMATSTRING

   where *object_type* is the type name of the object (for example, UGPartAttr) and *property_name* is the real name of the property to which you want the formatter applied (for example, **weight**).

   For example, creating the **UGPartAttr.weight**.FORMATSTRING preference allows you to define the format for the weight property of **UGPartAttr** object types.

4. In the **Values** section, specify the parameters for the formatting, for example, *#,###.##"KG"*

   For more details on property formatter definitions, see *Property formatter definition examples*.

5. Click Create.

6. Close the Options dialog box.

**Property formatter definition examples**

**Boolean property formatter definition**

The Boolean property formatter allows two choices to the user (for example, True or False).

**Pattern definition**

   "true-value";"false-value"

**Pattern details**

   "Y";"N"

**Pattern examples**

You can type any representative string for the Boolean values.

Following are pattern examples for the Boolean property formatter.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Real value</th>
<th>Formatted value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Y&quot;;&quot;N&quot;</td>
<td>true</td>
<td>Y</td>
</tr>
</tbody>
</table>
Double property formatter definition

The double property formatter specifies the format for a double-precision, floating-point decimal number.

Pattern definition

\[positive;negative;zero\]

Pattern details

\[#,##0.###; (#,##0.###); "Nil"\]

Pattern examples

Following are pattern examples for the double property formatter.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Real value</th>
<th>Formatted value</th>
</tr>
</thead>
<tbody>
<tr>
<td>#,##0.###</td>
<td>.56</td>
<td>0.56</td>
</tr>
<tr>
<td>#,##0.###</td>
<td>-1234.56</td>
<td>-1,234.56</td>
</tr>
<tr>
<td>#,###.000</td>
<td>-1234.56</td>
<td>-1,234.560</td>
</tr>
</tbody>
</table>

Pattern character descriptions:

- 0: Prints a digit. For example, if the value is 8.9, it is to be displayed as 8.90, the format should be entered as #.00.
- #: Prints a digit. Zero is not printed. For example, if the custom format is #.##, and the value is 8.9, the value 8.9 is displayed.
- : Decimal separator.
- , Digit grouping character.
- ; Pattern separator.
- "": Pre-fix and post-fix string.
Pattern | Real value | Formatted value
--- | --- | ---
#,##0.###;(#,##0.###) | -1234.56 | (1,234.56)
#,##0;#,#0 | -1234.56 | -1,234
#,###.###;(#);"Nil" | -1234.56 | (1234)
#,##0.###;(#);"Nil" | 0 | Nil
#.### " dia." | 10.23 | 10.23 dia.
"~ " #.### " ft." | 10.23 | ~ 10.23 ft.
"[" #.### "]" | 10.23 | [10.23]
"$" #.## " ea." | 10.25 | $10.25 ea.
##.## " GHz" | 54.76 | 54.76 GHz

Integer property formatter definition

The integer property formatter specifies the format for a whole number without decimals.

Pattern definition

*positive;negative;zero*

Pattern details

#.##0; (#,##0); "Nil"

<table>
<thead>
<tr>
<th>Positive value pattern</th>
<th>Negative value pattern</th>
<th>Zero value pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>#,##0</td>
<td>(#,##0)</td>
<td>&quot;Nil&quot;</td>
</tr>
</tbody>
</table>

Pattern characters

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Prints a digit. For example, if the value is 12, and it is to be displayed as 012, the format should be entered as 000.</td>
</tr>
<tr>
<td>#</td>
<td>Prints a digit. Zero is not printed. For example, if the custom format is ###, and the value is 12, then the value 12 is displayed.</td>
</tr>
<tr>
<td>,</td>
<td>Digit grouping character.</td>
</tr>
<tr>
<td>;</td>
<td>Pattern separator.</td>
</tr>
<tr>
<td>&quot;&quot;</td>
<td>Pre and post fix string.</td>
</tr>
</tbody>
</table>

Pattern examples

Following are pattern examples for the integer property formatter.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Real value</th>
<th>Formatted value</th>
</tr>
</thead>
<tbody>
<tr>
<td>#,##0</td>
<td>-1234</td>
<td>-1,234</td>
</tr>
<tr>
<td>#,###</td>
<td>-1234</td>
<td>-1,234</td>
</tr>
</tbody>
</table>
**Pattern** | **Real value** | **Formatted value**
--- | --- | ---
#,###0;(#,##0) | -1234 | (1,234)
"$"#,### | -1234 | $-1,234
#,###0;(#);"Nil" | 0 | Nil
#,###0;(#) | 0 | 0
# " MB" | 1024 | 1024 MB

String property formatter definition

The string property formatter specifies the format for a string of characters.

**Pattern definition**

\{d:Positive; Negative; Zero\}|{i:Positive; Negative; Zero}|{s:String}\}

**Pattern details**

\{d:#,###0.##; (#,##0.##);"-"\}|{i:#,##0; (#,##0);"-"\}|{s: "pre " @ " post"}\}

<table>
<thead>
<tr>
<th>Positive value pattern</th>
<th>Negative value pattern</th>
<th>Zero value pattern</th>
<th>Positive value pattern</th>
<th>Negative value pattern</th>
<th>Zero value pattern</th>
<th>String value pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>#,##0.##</td>
<td>(#,##0.##)</td>
<td>&quot;-&quot;</td>
<td>#,##0</td>
<td>(#,##0)</td>
<td></td>
<td>&quot;pre&quot; @ &quot;post&quot;</td>
</tr>
</tbody>
</table>

**Pattern characters**

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Prints a digit. For example, if the value is 12, and it is to be displayed as 012, the format should be entered as 000. This character is applicable for positive, negative and zero value patterns.</td>
</tr>
<tr>
<td>#</td>
<td>Prints a digit. Zero is not printed. For example, if the custom format is ###, and the value is 12, then the value 12 is displayed. This character is applicable for positive, negative and zero value patterns.</td>
</tr>
<tr>
<td>{d: }</td>
<td>Double pattern specifier.</td>
</tr>
<tr>
<td>{i: }</td>
<td>Integer pattern specifier.</td>
</tr>
<tr>
<td>{s: }</td>
<td>String pattern specifier.</td>
</tr>
<tr>
<td>@</td>
<td>Place holder for actual string value. This character is applicable for string value patterns.</td>
</tr>
<tr>
<td>.</td>
<td>Decimal separator. This character is applicable for positive, negative and zero value patterns.</td>
</tr>
</tbody>
</table>
Character | Description
--- | ---
, | Digit grouping character. This character is applicable for positive, negative and zero value patterns.
; | Pattern separator.
"" | Pre and post fix string.

Pattern examples

Following are pattern examples for the string property formatter.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Real value</th>
<th>Formatted value</th>
</tr>
</thead>
<tbody>
<tr>
<td>{d:##0.###;(#,###0.###)}</td>
<td>1234.567</td>
<td>1,234.56</td>
</tr>
<tr>
<td>{d:##0.###;(#,###0.###)}</td>
<td>-1234.567</td>
<td>(1,234.56)</td>
</tr>
<tr>
<td>{d:##0.###;(#,###0.###);&quot;Nil&quot;}</td>
<td>0</td>
<td>Nil</td>
</tr>
<tr>
<td>{d:##0.###;(#,###0.###);&quot;Nil&quot;} {s:000018/A @ }</td>
<td>000018/A</td>
<td>000018/A</td>
</tr>
<tr>
<td>{s:&quot;$ &quot; @ }</td>
<td>1000</td>
<td>$ 1000</td>
</tr>
<tr>
<td>{d:##0} {s:&quot;$ &quot; @ }</td>
<td>1000</td>
<td>1,000</td>
</tr>
<tr>
<td>{d:##0.###;(#)}</td>
<td>-1234.6738</td>
<td>(1234)</td>
</tr>
<tr>
<td>{d: &quot;Dia&quot;:&quot; Dia&quot;;&quot; 0 Dia&quot;}</td>
<td>1000.24</td>
<td>1000 Dia</td>
</tr>
<tr>
<td>{d:;&quot;NIL&quot;} {s: @ &quot; KB&quot;}</td>
<td>4 x 256</td>
<td>4 x 256 KB</td>
</tr>
<tr>
<td>{d:;&quot;NIL&quot;} {s: @ &quot; KB&quot;}</td>
<td>1000.24</td>
<td>1000</td>
</tr>
<tr>
<td>{d:##0.## &quot; ft/sec&quot;}</td>
<td>1000.2345</td>
<td>1,000.23 ft/sec</td>
</tr>
<tr>
<td>{d: &quot;$ &quot; #,###.0 }</td>
<td>1000000.200000</td>
<td>$ 1,000,000.20</td>
</tr>
<tr>
<td>{d: # &quot; GT/s&quot;}</td>
<td>5</td>
<td>5 GT/s</td>
</tr>
</tbody>
</table>

Enabling browser caching

When thumbnail images are displayed in Active Workspace, the image is loaded from the FMS system server using a file read ticket. Each time you display the same thumbnail, a new ticket is created. You can, however, enable browser caching so that the first time an image is loaded it is saved in the cache. This improves performance if the image is loaded again within a specified time period in the same session.

To enable browser caching:

- In Teamcenter, set the **Ticket_Expiration_Resolution** preference to the maximum number of seconds an image could be saved in the cache.

   Essentially, the preference value defines the expiration time resolution of the file read ticket. For example, if you load a thumbnail image at 1:00 p.m., a file read ticket is created. If the value of the preference is set to 7200, the image remains in the browser cache for 7200 seconds (2 hours) after 1:00 p.m. So, any time that image is loaded within the next 2 hours, the same ticket is used. If the image is loaded again after 2 hours, a new ticket is created.
The current default value for this preference is 7200 seconds. Previous versions of Teamcenter used a default value of 1 second.

**Configuring workflows**

**Workflow constants and conditions**

You configure your workflow with a set of Business Modeler IDE constants that have conditions as values.

**Workflow constants**

A set of constants is provided in the form:

```
<prefix><participant-name>AssignableCondition
```

The variable `<prefix>` is the Business Modeler IDE template prefix and `<participant-name>` is an existing participant name.

**Note**

If the participant name also has a template prefix, the prefix appears twice.

For example, if the prefix is `Fnd0` and the participant name is `PROPOSED RESPONSIBLE PARTY`, the constant is `Fnd0ProposedResponsiblePartyAssignableCondition`.

The constants are for item revisions and change item revisions.

**Workflow conditions**

The values of the constants are conditions in the form:

```
is<participant-name>Assignable
```

For example, if the participant name is `PROPOSED RESPONSIBLE PARTY`, the condition is `isProposedResponsiblePartyAssignable`.

This is used while assigning dynamic participants. Teamcenter gets the value of the `<prefix><participant-name>AssignableCondition` constant to get the condition name to evaluate before assigning the participant.

**Search for condition names**

You can search for the constant name given an object type and participant type using pattern matching.

For example, to find a constant associated with an item revision and the `Fnd0MyNewParticipant` participant, search for a constant that ends with `Fnd0MyNewParticipantAssignableCondition`. The actual constant name is `Fnd0Fnd0MyNewParticipantAssignableCondition`.

If there are multiple matches, choose the one which has the same prefix as the prefix of the participant name.
Creating constants and conditions

If you have your own participant types, you must create your own constants and conditions for them.
For example, if your template prefix is **CUS1** and the new participant name is **MyParticipant**:

1. Create a participant named **CUS1MyParticipant**.
2. Create a constant named **CUS1CUS1MyParticipantAssignableCondition** with a value of **isMyParticipantAssignable**.

The participant creation code looks up the constant and corresponding condition and evaluates it.
For more information about creating constants and conditions, see *Business Modeler IDE*.

Assigning participants using workflow handlers

You can use the following workflow handlers when automatically assigning participants:

- **EPM-assign-responsible-party-dynamic-participant**
- **EPM-assign-signoff-dynamic-participant**

The following handlers can be used to get assignees from a property value:

- **EPM-adhoc-signoffs**
  
  **Note**
  
  In Teamcenter 9.1, this handler is named **adhoc-signoffs**.

- **EPM-assign-team-selector**
  
  **Note**
  
  In Teamcenter 9.1, this handler is named **CR-assign-team-selector**.

- **EPM-auto-assign**
  
  **Note**
  
  In Teamcenter 9.1, this handler is named **auto-assign**.

- **EPM-auto-assign-rest**
  
  **Note**
  
  In Teamcenter 9.1, this handler is named **auto-assign-rest**.

- **EPM-fill-in-reviewers**
  
  **Note**
  
  In Teamcenter 9.1, this handler is named **CR-fill-in-reviewers**.
You can use the `user:PROP::property_name`, `resourcepool:PROP::property_name`, or `allmembers:PROP::property_name` values for the `-assignee` argument to get the name of the assignee from a property of the target, reference, or schedule task.

You can find the object type with the `-include_related_type`, `-exclude_related_type`, `-include_type`, `-exclude_type`, `-from_relation`, and `-from_attach` arguments.

For more information, see the full handler description.

**Automatically reassigning dynamic participants**

**Note**
Automatically reassigning dynamic participants is only available with Active Workspace on a Teamcenter 10.1.2.1 or later platform.

When a user reassigns a task to another user, use the `WRKFLW_task_assignee_dynamic_participant_sync` preference to set Active Workspace so it also changes the dynamic participant that is set for the change.

For example, if User 1, the change analyst, forwards a task to User 2, then the change analyst role is reassigned to User 2 for the change, as shown in the figure for a problem report.

In the same way, if User 1 forwards tasks to User 2 while User 1 is out of the office, then all incoming tasks are assigned to User 2 during that time. The change analyst role is reassigned to User 2 in all the changes associated with the task.

For more information about dynamic participants in a change, see *Change Manager* in the Teamcenter help collection.

**Configuring resource pool assignments**

Use the `WRKFLW_auto_subscribe_to_resource_pools` preference to set Active Workspace to always display in a user’s inbox those tasks assigned to resource pools that the user is a member of based on the user’s group or role (referred to as automatic subscription). If you turn off automatic subscription, only tasks assigned to the resource pools to which the user has been manually subscribed appear. The default is automatic subscription.
Currently, users cannot manually subscribe to resource pools through Active Workspace.

**Allowing access to the Inbox of other users**

Currently, allowing access to the inbox of other users is only available with Active Workspace on a Teamcenter 10.1.2.3 or later platform.

Use the `Fnd0AccessInboxPrivilege` Business Modeler IDE condition to set what users can open another user's Inbox. By default, users belonging to the `dba` group are given privileges to open another user's inbox. Other users do not have permission.

Refer to *Business Modeler IDE* in the Teamcenter collection for more information about setting conditions.

**Setting who can reassign tasks**

Setting who can reassign tasks is only available with Active Workspace on a Teamcenter 10.1.2.3 or later platform.

Use the `Fnd0AssignTaskPrivilege` Business Modeler IDE condition to restrict the reassignment of workflow tasks. By default:

- Users belonging to the `dba` group can reassign any task.
- The current user who is either the assignee of the task or an active surrogate can reassign a task.
- For signoff tasks, if the decision has not been set, the current user who is either the assigned group member or the active surrogate can reassign the task.

Refer to *Business Modeler IDE* in the Teamcenter collection for more information about setting conditions.

**Configuring the Inbox**

**Setting up filtering in the Inbox**

You can set the properties to be used to filter the tasks that appear in the Inbox, as shown in the figure for the tasks found when selecting the **Team** tab. The tasks are Workflow business objects of the **EPMTask** type and its subtypes.
To set the filtering, in the Business Modeler IDE, set the following property constants on the property of the Workflow business object on which you want to filter.

- **Awp0InboxCanFilter**
  Indicates that Workflow objects can be filtered on the property.

- **Awp0InboxFilterPriority**
  Indicates the priority of the property that determines its order in the list of filters displayed in the Inbox. The lower the value, the higher its priority and, therefore, the higher its position in the list of filters.

  Siemens PLM Software recommends that you assign values from a range to accommodate additional properties in the future. For example, assign priorities such as 100, 200, and 300, instead of 1, 2, and 3.

By default, the following properties are shown as filters for Workflow business objects:

- **object_type**
  The type of object.

- **due_date**
  The date the object is due.

- **fnd0Assignee**
  The user to whom the task is assigned.

- **fnd0Priority**
  The priority of the task.
• **fnd0WorkflowInitiator**

  The user who initiated the workflow on the task.

Workflow filters can only be set on persistent and compound properties.

<table>
<thead>
<tr>
<th>Properties supported for filtering</th>
<th>Properties not supported for filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Date</td>
<td>• String properties with long string as storage</td>
</tr>
<tr>
<td>• String</td>
<td>• Numeric properties</td>
</tr>
<tr>
<td>• References</td>
<td>• Array properties</td>
</tr>
<tr>
<td>• Logical</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring email notification**

Configure Active Workspace to send users email notifications when there are workflow tasks in their inbox. The email notification contains a link to the task so the user can select it to open the task in their Active Workspace inbox.

Use the following workflow handlers and preferences to configure the notifications.

**Workflow handlers**

Use the following workflow handlers to Insert a link to the workflow process into the notification email.

• **EPM-notify**

  Informs users of a task's status through email.

  Contains a `-url` argument. The value for Active Workspace is `activeworkspace`.

  If the `-url` argument is not specified, the values specified in the `EPM_notify_url_format` preference is used.

• **EPM-notify-report**

  Sends a report through email to all task reviewers.

  If the `-url` argument is not specified, the values specified in the `EPM_notify_url_format` preference is used.

• **EPM-notify-signoffs**

  Informs users of a signoff task's status through email. It does not contain a `-url` argument. Instead, it uses the format specified in the `EPM_notify_url_format` preference.

**Preferences to specify the URL**

For an Active Workspace link to be added in the notification email, either of the following two Active Workspace hosting preferences must be present:
Note

These preference are not included in your Teamcenter installation. You must add it to the database.

- **ActiveWorkspaceHosting.URL**
  Specifies the URL that Teamcenter uses to communicate with Active Workspace for hosted operations, such as search, open item, and others.
  
  URL=http://<host>:<port>/awc

- **ActiveWorkspaceHosting.WorkflowEmail.URL**
  Specifies the URL used that the workflow uses to communicate with Active Workspace for email links. For example:
  
  URL=http://<host>:<port>/awc

In addition, the **EPM_notify_url_format** preference contains the **activeworkspace** value to use to specify that an email notification should contain a link to the task in Active Workspace.

For more information about these workflow handlers and preferences, see *Workflow Designer* and *Preferences and Environment Variables Reference* in the Teamcenter help collection.

## Configuring change management

### Automating the submission of changes to workflow

Use the following Teamcenter rich client preferences to set the default workflow that should start when a user submits a problem report, change notice, or change request in Active Workspace. The default workflow is **ChangeItemRevisionDefaultWorkflowTemplate**, which is a simple process to select a signoff team and then have each participant of the team perform his/her signoff task to approve the change.

<table>
<thead>
<tr>
<th>For a</th>
<th>Use the preference</th>
<th>Its default is</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change notice revision</td>
<td>ChangeNoticeRevision_default_workflow_template</td>
<td>ChangeItemRevisionDefault WorkflowTemplate</td>
</tr>
<tr>
<td>Change request revision</td>
<td>ChangeRequestRevision_default_workflow_template</td>
<td>ChangeItemRevisionDefault WorkflowTemplate</td>
</tr>
<tr>
<td>Problem report revision</td>
<td>ProblemReportRevision_default_workflow_template</td>
<td>ChangeItemRevisionDefault WorkflowTemplate</td>
</tr>
</tbody>
</table>

### Setting the type of changes to be created

Use the following **isCreatable** Business Modeler IDE conditions to filter the types of changes that appear when a user creates a change. You can filter the changes based on the user context (group, role, or user ID) and the current state of the change.

- **isChangeNoticeCreatable**
• isChangeRequestCreatable
• isCm0DevRqstCreatable
• isProblemReportCreatable

Refer to Change Manager and Business Modeler IDE in the Teamcenter collection for more information.

Automating the assignment of participants

To have the participants of a change workflow automatically assigned, use the following workflow handlers:

• EPM-assign-responsible-party-dynamic-participant
• EPM-assign-signoff-dynamic-participant

Refer to Workflow Designer in the Teamcenter collection for more information.

Configuring how changes are derived

When deriving a change from another use the CM_automate_derive_propagation preference to enable the automatic propagation of the relations (such as reference items and problem items) from the source change to the derived change. You configure which relations to propagate using the following preferences. For example, for a problem report enable the propagation of its problem items (CMHasProblemItem) and its reference items (CMReferences).

<table>
<thead>
<tr>
<th>When deriving a change object from a</th>
<th>Set the relations propagated using</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem report</td>
<td>CM_ProblemReportRevision_Relations_To_Propagate</td>
</tr>
<tr>
<td>Deviation request</td>
<td>CM_Cm0DevRqstRevision_Relations_To_Propagate</td>
</tr>
<tr>
<td>Change request</td>
<td>CM_ChangeRequestRevision_Relations_To_Propagate</td>
</tr>
</tbody>
</table>

You configure which change object users can derive from another using the CM_change_derivations preference.

For more information, refer to Preferences and Environment Variables Reference in the Teamcenter help collection.

Defining deep copy rules for creating changes from another

**Note**

Defining deep copy rules is only available with Active Workspace on a Teamcenter 10.1.2.3 or later platform.

Use the Teamcenter Business Modeler IDE deep copy rules to set what objects and attributes are copied when a user creates a copy of a change from another. Deep copy rules define whether
objects belonging to a business object instance can be copied when a user performs a save as or revise operation on that instance. Deep copy rules can be applied to any business object type and are inherited by children business object types.

Using deep copy rules, you can configure whether the following are copied for a change:

- Name, subject, description
- Problem and Impacted Items
- Referenced or related documents

The figure shows the deep copy rules defined by default for a **ChangelItemRevision** in the Deep Copy Rules editor. The rules define that when copying a change, copy the problem, impacted, and reference objects, but do not copy the incorporates and solution items.

Refer to the **Business Modeler IDE** in the Teamcenter collection for more information.

**Note**

Copying changes is not available in the Teamcenter rich client.

### Configuring My Changes

#### Setting up filtering in My Changes

You can set the properties to be used to filter the changes that appear in **My Changes**, as shown in the figure for changes found when selecting the **Submitted** tab. The changes are change business objects of the **ChangelItemRevision** type and its subtypes.
To set the filter, in the Business Modeler IDE, set the following property constants on the property of the change object on which you want to filter.

- **Cm1ChangeCanFilter**
  
  Indicates that change business objects can be filtered on the property.

- **Cm1ChangeFilterPriority**
  
  Indicates the priority of the property that determines its order in the list of filters displayed in My Changes. The lower the value, the higher its priority and, therefore, the higher its position in the list of filters.

Siemens PLM Software recommends that you assign values from a range to accommodate additional properties in the future. For example, assign priorities such as 100, 200, and 300, instead of 1, 2, and 3.

By default, the following properties are shown as filters for Change business objects:

- **creation_date** – Date the change was created.
- **CMMaturity** – Degree of completion of the overall change process (its maturity).
- **object_type** – Type of change.
- **cm0Analyst** – User assigned as the analyst.
- **cm0ChangeSpecialist1** – User assigned as the change specialist.
- **cm0Requestor** – User who created the change.

Change filters can only be set on persistent and compound properties.
Properties supported for filtering | Properties not supported for filtering
---|---
• Date | • String properties with long string as storage
• String | • Numeric properties
• References | • Array properties
• Logical | |

Refer to *Business Modeler IDE* in the Teamcenter collection for more information.

**Configuring the contents of tabs in My Changes**

**Note**

Configuring the contents of tabs is only available with Active Workspace on a Teamcenter 10.1.2.3 or later platform.

You can use the Query Builder in the Teamcenter rich client to configure the queries that are performed when a tab in *My Changes* is selected to redefine the contents that is shown.

Redefining the queries of the tabs is particularly helpful when you have defined custom participants or changes. You can replace your custom participants and changes with those in the default queries so the tabs show your company’s content.

You must set the `CMMMyChangesSublocationQuery` preference to set Active Workspace to use your queries instead of the defaults.

The following show the default queries for each tab in and the results, as well as examples of queries.
### Default queries

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All</strong></td>
<td>Display changes with a Closure of Open and the logged-in user is the requestor, analyst, or change specialist.</td>
<td>Get all ChangeItemRevs where Closure=Open AND (Requestor = Logged-in User OR Analyst = Logged-in User OR Change Specialist1 = Logged-in User)</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong></td>
<td>Display changes with a Closure of Open and the logged-in user is the requestor or analyst:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get all ChangeItemRevs where Closure=Open AND (Requestor = Logged-in User OR Analyst = Logged-in User)</td>
</tr>
<tr>
<td><strong>Saved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Display changes with a Closure of Open and the logged-in user is the requestor, analyst, or change specialist.</td>
<td>Get all ChangeItemRevs where Closure=Open AND (Requestor = Logged-in User OR Analyst = Logged-in User OR Change Specialist1 = Logged-in User) AND ProcessStageList = NULL</td>
</tr>
<tr>
<td><strong>Submitted</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Display changes with a Closure of Open and the logged-in user is the requestor, analyst, or change specialist.</td>
<td>Get all ChangeItemRevs where Closure=Open AND (Requestor = Logged-in User OR Analyst = Logged-in User OR Change Specialist1 = Logged-in User) AND ProcessStageList != NULL</td>
</tr>
<tr>
<td><strong>Approved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Display changes with a Closure of Open and the logged-in user is the requestor, analyst, or change specialist.</td>
<td>Get all ChangeItemRevs where Closure=Open AND (Requestor = Logged-in User OR Analyst = Logged-in User OR Change Specialist1 = Logged-in User) AND Disposition = Approved</td>
</tr>
<tr>
<td><strong>Disapproved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Display all changes with a Closure of Open that belong into the user who is logged in as the analyst or the change specialist.</td>
<td>Get all ChangeItemRevs where Closure=Open AND (Requestor = Logged-in User OR Analyst = Logged-in User OR Change Specialist1 = Logged-in User) AND Disposition = Disapproved</td>
</tr>
</tbody>
</table>
Configuring digital signature

**Note**

- Digital signature is a Teamcenter 10.1.2 feature that is available in Active Workspace.
- Digital signature is supported in Microsoft Internet Explorer 9 or Internet Explorer 10 only. If you use any other browser, you can view digital signatures, but you cannot apply or void a digital signature.

**What is digital signature?**

A *digital signature* is a mathematical stamp on an object used to indicate if that object has been modified after the signature was applied. It also identifies who applied the digital signature. You must use public key infrastructure (PKI) authentication when applying the digital signature.

**Configuring digital signature in Active Workspace**

**Note**

You must have administrative privilege to perform these steps.

1. Install and configure your Teamcenter four-tier server for digital signature as described in Teamcenter *Security Administration*.

2. Patch your environment to the supported Teamcenter 10.1.2. Refer to the general patch instructions in the Teamcenter documentation as well as the readme file for the patch.

   In the Teamcenter documentation:
   - *Windows Server Installation*→*Additional configuration and maintenance*→*Installing Teamcenter patches*
   - *UNIX and Linux Server Installation*→*Additional configuration and maintenance*→*Installing Teamcenter patches*

3. Install Active Workspace and include the digital signature features shown in the Teamcenter Environment Manager (TEM) Features panel:
   - Active Workspace Client
   - Digital Signatures (Client) for Active Workspace
     Enables Active Workspace to support digital signature functionality. This includes applying and voiding digital signatures to Teamcenter objects that are configured to support it and digitally signing data upon workflow task completion.
   - Active Workspace
   - Active Workspace Indexer
   - Digital Signatures (Server) for Active Workspace
     Installs the Active Workspace style sheet to support applying digital signatures on objects.
4. **Deploy the Active Workspace WAR file.**

5. **Configure your system by adding the following code to all style sheets specific to Active Workspace.**

   **Note**

   Generally, these style sheet names begin with the prefix `Awp0` (for example, `Awp0DatasetSummary`). The `Awp0` and `Summary` are standard for each style sheet to be modified. The middle portion denotes the object type to be updated, for this example, `Dataset`.

   ```xml
   <section title="Signatures">
   <objectSet source = "Fnd0DigitalSignatureRel.Fnd0DigitalSignature"
                 sortdirection = "ascending" sortby = "object_string"
                 defaultdisplay = "listDisplay">
       <tableDisplay>
         <property name = "owning_user"/>
         <property name = "fnd0State"/>
         <property name = "creation_date"/>
       </tableDisplay>
       <treeDisplay>
         <property name = "owning_user"/>
         <property name = "fnd0State"/>
         <property name = "creation_date"/>
       </treeDisplay>
       <thumbnailDisplay/>
       <listDisplay/>
       <command actionKey = "addDigitalSignatureAction"
                 commandId = "com.teamcenter.rac.applyDigitalSign"
                 renderingHint = "commandbutton"/>
       <command actionKey = "voidDigitalSignatureAction"
                 commandId = "com.teamcenter.rac.voidDigitalSign"
                 renderingHint = "commandbutton"/>
     </objectSet>
   </section>
   
   6. **Install Microsoft .NET Framework 4 on each client.**

   **Note**

   This is available from Microsoft.

   7. **Install ActiveX on each client.**

   ActiveX is located in your Active Workspace kit in the `additional_applications\Pkcs7install` directory. You must run the installer in this directory.

   When the wizard prompts you to enter the hash algorithm, type the hash algorithm value to be used, such as `SHA256` or `SHA384` or `SHA512` or `SHA1`.

   The default value is `SHA256`. However, if the PKI infrastructure requires any other algorithm, it needs to be configured here and the same needs to be configured in the Teamcenter server in the `tc_profilevars` file. The hash algorithm is stored in the `TC_DS_HASH_ALGORITHMS` environment variable.
For information about applying PKI digital signatures, see *Active Workspace User Assistance*.

### Configuring template files

You can import a sample template file that contains examples (`docmgmt_samples.xml`). You can use these examples as the basis for your dispatcher service configurations and IRDCs.

For more information about configuring template files, a `readme.pdf` file is provided in the following directory:

```
TC_ROOT/sample/document_management
```

### Configuring custom content objects

#### When is configuration needed?

Active content is an implementation-neutral interface for exposing occurrences and product structure in Active Workspace. It encapsulates structure concepts in Teamcenter by providing an interface and a runtime data model. Other models that are based on active content extend this paradigm, for example, requirements management. The application data returned from these modules is adapted to the active content data model before returning to the client.

Additional configuration is necessary if you customize the standard active content data model or any of the modules that are based on it, for example, requirements management or systems modelling.

#### Active content technical overview

Active content provides a number of business objects to represent the occurrences returned by the various structure applications. **Awb0Element** is the root type and is abstract. The following table lists the default active content types that represent occurrence types for BVR structures.
<table>
<thead>
<tr>
<th>Business object</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Awb0Element</td>
<td>The root business object for all occurrence management objects. An element is identified by a name and can belong in a structure with a parent-child relationship. This is an abstract object and should not be mapped with any item revision type.</td>
</tr>
<tr>
<td>Awb0ConditionalElement</td>
<td>A conditional element that represents an occurrence with effectivity and variant conditions. Generic design element (GDE) lines do not have effectivity or variant conditions; they should not be mapped to this type or its subtypes.</td>
</tr>
<tr>
<td>Awb0Connection</td>
<td>A connection may represent an occurrence that does not have any associated geometry information.</td>
</tr>
<tr>
<td>Awb0PositionedElement</td>
<td>If an occurrence represents an item revision type with geometry.</td>
</tr>
<tr>
<td>Awb0DesignElement</td>
<td>Represents an occurrence with geometry. It provides the properties for managing geometry, for example, bounding box and transform. Objects of this type can be visualized in the viewers.</td>
</tr>
<tr>
<td>Awb0Interface</td>
<td>Represents the generic design element.</td>
</tr>
</tbody>
</table>

The active content data model is shown below:
Understanding active content

Active content supports several domains and the capabilities provided by the domains may vary by releases and levels of customization. To accommodate these differences, a feature discovery mechanism is provided. This is implemented as an `awb0SupportedFeature` property on the `Awb0ProductContextInfo` object. This property defines the features supported for a configuration as
a list of runtime business objects. For example, when the `Awb0FullTextSearchFeature` is included in the list, users can perform a full text search on the structure.

**Using the Awb0BOMArchetypeToOccurrence type constant**

The `Awb0Element` business object and its subtypes have a `Awb0BOMArchetypeToOccurrence` type constant. The value of this constant:

- Determines the instance of which particular subtype of `Awb0Element` is created.
- Is a comma-separated list of item revision or GDE subtypes. Using such a list avoids the need to create a separate `Awb0Element` business object for each item revision type.

For example, `Awb0DesignElement` has the value set to `ItemRevision` for this type constant. Consequently, an instance of `Awb0DesignElement` is created if the BOM line represents item revisions.

By creating different object types of `Awb0Element` based on the archetype, the system controls the visible properties by the domain. For example, for a requirement, the geometry specific properties are not relevant and it can be mapped to `Awb0ConditionalElement` or a subtype.

Each subtype of `Awb0Element` may have an associated XRT style sheet containing the details of the properties. You should not create a new subtype to represent each supported custom item revision. Instead, if the custom item revision can be represented by a default type, consider modifying the corresponding constant value.

**Note**

This constant applies only to BVR models.

The default mappings are as follows:

<table>
<thead>
<tr>
<th>Business object</th>
<th>Awb0BOMArchetypeToOccurrence value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Awb0Element</code></td>
<td></td>
</tr>
<tr>
<td><code>Awb0ConditionalElement</code></td>
<td></td>
</tr>
<tr>
<td><code>Awb0Connection</code></td>
<td><code>PSConnectionRevision</code></td>
</tr>
<tr>
<td><code>Awb0PositionedElement</code></td>
<td></td>
</tr>
<tr>
<td><code>Awb0DesignElement</code></td>
<td><code>ItemRevision</code></td>
</tr>
<tr>
<td><code>Awb0Interface</code></td>
<td><code>Interfaces</code></td>
</tr>
</tbody>
</table>

If you need to create a new subtype of `Awb0Element` to represent a custom subtype of `ItemRevision`, note that `ItemRevision` has a discrete set of properties that may not be relevant. Instead, review the table of the default `Awb0Element` subtypes and create a subtype of the `Awb0Element` subtype that best matches your purpose.

For example, if your custom type has custom properties and also effectivity, create a subtype of `Awb0ConditionalElement`. Alternatively, if it also has associated geometry information and must be visualized, create a subtype of `Awb0DesignElement` instead.
Mapping type to model element

The \texttt{Awb0BOMToOccurrence} type constant specifies the subtype of \texttt{BOMLine} that is mapped to a given \texttt{Awb0Element} subtype. The subtype name specified as the value becomes the backing object for the \texttt{Awb0Element} subtype and the properties are fetched from this type. The value is inherited by the subtypes but can be overridden at any level.

For example, the value of this constant for \texttt{Awb0DesignElement} is specified as \texttt{BOMLine}. The value of the properties is then fetched from \texttt{BOMLine}.

Marking archetypes to support structure

The \texttt{Awb0SupportsStructure} global constant controls the types that can be opened in the active content explorer. This constant can take multiple values and each value is the name of a type that supports structure. Only the names of types that are specified in this constant can be opened in the explorer. The structure property is not inherited by subtypes, that is, you must add each subtype separately.

By default, this constant includes \texttt{ItemRevision}, \texttt{DesignRevision} and \texttt{PartRevision} as the values. For BVR structures, use the \texttt{ItemRevision} type as the value.

Mapping properties to occurrence properties

Domain-specific occurrences contain properties relevant to the specific domain. End users understand and interact with these domain-specific properties. These properties are mapped from the \texttt{BOMLine} or \texttt{ModelElement} type onto the occurrence. This mapping is provided by property constants defined in the Business Modeler IDE and the property constants are scoped to the \texttt{Awb0Element} type. Default properties are provided on \texttt{Awb0Element} and its subtypes, but you can add custom properties necessary for your implementation.

All custom properties must be mapped to a property defined on the type specified in the \texttt{Awb0BOMToOccurrence} type constant. The property mapping is then achieved through the \texttt{Awb0BOMToOccurrence} property constant. The value of this property constant is inherited and can be overridden at any level.

For example, the \texttt{awb0BoundingBox} property on the \texttt{Awb0PositionedElement} business object has the value of \texttt{bl\_bounding\_boxes}. It also has the value \texttt{BOMLine} for the \texttt{Awb0BOMToOccurrence} type constant. Consequently, whenever the \texttt{awb0BoundingBox} property is requested on an \texttt{Awb0PositionedElement} object, the value is fetched from the \texttt{bl\_bounding\_boxes} property of the BOM line.

The mapping of some common default properties are listed below:

<table>
<thead>
<tr>
<th>Business object</th>
<th>Property</th>
<th>Awb0BOMToOccurrence value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Awb0Element</td>
<td>awb0Parent</td>
<td>\texttt{bl_parent}</td>
</tr>
<tr>
<td>Awb0Element</td>
<td>awb0Name</td>
<td>\texttt{bl_line_name}</td>
</tr>
<tr>
<td>Awb0Element</td>
<td>awb0ElementId</td>
<td></td>
</tr>
<tr>
<td>Awb0Element</td>
<td>awb0Archetype</td>
<td>\texttt{bl_revision}</td>
</tr>
<tr>
<td>Awb0ConditionalElement</td>
<td>awb0ArchetypeEffFormula</td>
<td>\texttt{bl_revision_effectivity}</td>
</tr>
</tbody>
</table>
Default getter and setter methods are registered for all properties of `Awb0Element` and child business objects that have a mapping defined in the `Awb0BOMToOccurrence` property constant. You must provide a custom getter and setter for all properties that are not already mapped. For example, the `awb0NumberOfChildren` property specifies the number of child elements and does not have a value for the property constant; custom getter and setter methods are registered for it. You can use the same mechanism to register getter and setter methods for properties on custom `Awb0Element` subtypes.

Many configured runtime properties on `BOMLine` are derived from `Item`, `ItemRevision`, or another object. These are always of type `string`, irrespective of the type from which they are derived. Mapping to such a property will result in losing the type information about the property that the system requires for proper filtering of SOLR results. To avoid this, instead of mapping to a configured runtime property, define a new compound property on the backing object and use the relations or reference properties to get the source property. This maintains the type information for the property so that it can be used in mapping.

For example, you may want to get the last modified date stored in the configured runtime property `bl_rev_last_mod_date` on the BOM line. Instead of using the `bl_rev_last_mod_date` property, consider the `awb0RevisionLastModifiedDate` compound property on the BOM line. This uses the `bl_revision` property to access the item revision and you can get the `last_mod_date` property from there. An `awb0ArchetypeRevLastModDate` property is defined on the `Awb0DesignElement` business object and then mapped to the `awb0RevisionLastModifiedDate` property.

**Implementing full text search of structures**

To allow full text search of structures, mark the `Awb0Element` types and their properties for indexing in a similar way to how you identify properties from related objects for object searches. To index a property, set the `Awp0SearchIsIndexed` property constant to `true`, and also set the `Awp0SearchIsIndexed` business object constant of the type on which the property is created to `true`. If filtering is required on the property, set the `Awp0SearchCanFilter` property constant to `true`.

For BVR structures, Active Workspace provides two adapters, one using a BOM window (BOM line) based API and the other an indexed adapter. The indexed adapter uses the configured structure stored in the database for faster access. Only structures that use the index adapter can utilize the full text structure search capabilities.

**Configuring Document Management**

Document management functionality uses item revision definition configuration (IRDC) objects to simplify user interactions with information. Based on your business processes, you define IRDC objects in the Teamcenter Business Modeler IDE that can:
Chapter 4: Configuration

- Automate document creation, rendering of presentation formats, document markup, document batch printing, and document availability.

- Define how item revisions are handled at specific times in the life cycle, such as item creation, checkin, save as, and revise.

For information about configuring IRDC objects, see Business Modeler IDE in the Teamcenter collection.

Configuring print reports

Defining reports with a watermark

You (as a reports administrator with or without DBA privileges) can use Report Builder to create and manage report definitions, and users can use them to generate reports from Active Workspace.

Before you define a report definition, you must create:

1. Query sources to get data from Teamcenter.

2. Closure rules to control the scope of the data.

3. Property sets to get the data into a PLM XML file.

4. Transfer modes containing rules to import data.

The next step is to create a report template (item report) to define the content of the report. You can use an existing style sheet or define a new style sheet to create your template. Each report consists of a report definition and an associated XSL style sheet. While the report definition determines the type of data displayed in the report, the style sheet decides the look and formatting of this report.
For more information about reports, see the sample XML reports in the TC_DATA\crf directory and sample style sheets in the TC_DATA\crf\Resources directory. You can reuse the sample XML reports and sample style sheets to create your own reports.

You can assign templates to specific groups. When users select a single object in Active Workspace, at least one default template is available. If they select multiple objects of the same type, all templates related to the object types are available depending on the user's group permissions. If they select multiple objects of different types, the template of the common parent object type is available. If the immediate common parent template is not available, the system displays the workspace object root template.

In addition, you can optionally define a watermark to reports to identify copyright information and the status of the generated report. For example, you can add a watermark to specific types of reports to display your company name and the report status, such as confidential. Reports with watermarks help protect company proprietary information from unauthorized disclosure. The watermark is defined as a text entity in the style sheet associated with the report type. You can change the text, font size, location, and orientation of the watermark in a report by editing the style sheet.

By default, change notice objects and workspace objects reports have a predefined watermark with the Confidential text entity. You can also use the watermark in other types of reports by editing the style sheets associated with those reports.

Modify report definitions

Before defining a report definition, you must create a:
1. Query source to create and maintain customized searches for objects in the Teamcenter database.

   You use Query Builder to create a query source.

2. Closure rule to specify the structure for processing related items.

3. Property set to specify additional output properties.

4. Transfer mode to combine closure rules and property sets to define the context of the PLM XML import or export operation.

   You use PLM XML/TC XML Export Import Administration to create transfer modes.

The Change Notice Objects Report and WSO Object Report report definitions are provided with default query source, closure rules, property sets, and transfer modes as follows:

<table>
<thead>
<tr>
<th>Rule, set, or mode</th>
<th>Change Notice Objects Report</th>
<th>WSO Object Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closure Rule</td>
<td>CMClosureRuleDefault</td>
<td>WSOClosureRuleDefault</td>
</tr>
<tr>
<td>Property Set</td>
<td>CMPropertySetDefault</td>
<td>WSOPropertySetDefault</td>
</tr>
<tr>
<td>Transfer Mode</td>
<td>CMTransferModeDefault</td>
<td>WSOTransferModeDefault</td>
</tr>
</tbody>
</table>

1. In Report Builder, open the Teamcenter Reports folder.

2. Select the WSO Object Report or the Change Notice Objects Report report definition and click the Report Data tab.

3. Change the information in the boxes as appropriate.

4. For item reports, select one or both options as appropriate.
   - Run in Async to allow users to print the report asynchronously. Clear this option to allow users to print reports synchronously.

   Active Workspace users can continue to work on other tasks while the system generates the report in the background.

   The AsyncService translator uses the Dispatcher framework to independently processes asynchronous requests from the Teamcenter server, using the Teamcenter SOA native C++ framework in the background mode.

   - Use for Print to make this template available for Active Workspace reports.

   [Note]

   This option is not enabled for Change Notice Objects Report report definition by default. Ensure that you select this option for this report definition.

5. To make this report template available to specific users or groups, select users or groups from the Defined Group box and add them to the Selected Group box.
**Tip**
If you do not select a specific user or group, the system automatically assigns the template to all users and groups.

6. Click **Modify** to save your changes.

**Customize the watermark by editing the style sheet associated with the report definition**

You can customize the watermark for the **Change Notice Objects Report** and **WSO Object Report** report definitions by editing the associated style sheet.

- **Change_object_Sample_template.xsl** file to customize the watermark for **Change Notice Objects Report**.
- **WSO_Object_Sample_Template.xsl** file to customize the watermark for **WSO Object Report**.

You can change the content, font size, location, and orientation of the watermark.

1. Open and edit a style sheet associated with a report definition.
   a. In Report Builder, open the **Teamcenter Reports** folder.
   b. Select the report definition for which you want to customize the watermark, and click the **Report Data** tab.
   c. In the **Defined Stylesheets** box, select the style sheet you want to edit, and click the **Open the selected stylesheet** icon.
      Teamcenter exports the style sheet to a temporary directory, for example, **C:\Temp\tc_infodba_build-version\style-sheet-name.xsl**.
   d. Browse to the directory where you exported the style sheet, and open the XSL file in an XML editor.

2. Change the text or the content of the watermark.
   a. Locate the **watermark_print** attribute.
      Example:
      ```xml
      <div id="watermark_print" align="center">
          <p>Confidential&$169;</p>
      </div>
      ```
      The default watermark text is **Confidential**.
   b. Change as appropriate.
      You can include a hardcoded string or add one or more property values of the objects you want to generate for the report.

3. Change the font size of the watermark.
   The default font size is **25pt**.
4. Change the location of the watermark.

You can change the width and height attributes of the watermark. By default, these attributes are set to 100% and 50%, respectively.

a. Locate the width and height attributes in the style type tag.

<styles type="text/css">
  .myOtherTable { background-color:#C2C2C2;border-collapse: collapse;font-size:15px;border:2; }
  .myOtherTable th { background-color:#ECECEC;color:white;
  width:10%; border-bottom:2px dotted #BDB76B;}
  .myOtherTable td, .myOtherTable th { padding:5px;border:2;
  border-bottom:2px; dotted #BDB76B;}
  .myOtherTable td { border-bottom:2px dotted #BDB76B; }
  .myOtherTable th { border-bottom:2px solid #BDB76B; }
  .pagebreak {page-break-before: always;}
</style>

b. Change the width and height attributes as appropriate.

5. Change the orientation of the watermark.

The transform: rotate attribute determines the orientation of the watermark.

The transform attribute applies a transformation and allows you to rotate an element. The attribute varies depending on the type of browser you use. For example, the -webkit-transform: rotate attribute is used for Mozilla Firefox, while the -ms-transform:rotate attribute is used to apply transformation in Internet Explorer.
a. Locate the `transform: rotate` attribute in the `style type` tag.

```css
@media print {
  #watermark_print {
    position:absolute;
    bottom:0;
    width:100%;
    height:60px;
    background:#6cf;
    display:block;
    position:fixed;
    right:0;
    z-index:5;
    color:#B81C1C;
    font-size:25pt;
    height:50%;
    margin:0;
    top:350px;
    left:125px;
    margin:auto;
    margin-right:auto;
    margin-left:auto;
    -webkit-transform: rotate(-40deg); /* FF3.5+ */
    -moz-transform: rotate(-40deg); /* Opera 10.5+ */
    -o-transform: rotate(-40deg); /* Saf3.1+, Chrome */
    -ms-transform: rotate(-40deg);
    transform: rotate(-40deg);
    filter: progid:DXImageTransform.Microsoft.BasicImage(rotation=3); /* IE6,IE7 */
    -ms-filter: "progid:DXImageTransform.Microsoft.BasicImage(rotation=3)"; /* IE8 */
    text-align:center;
  }
}</style>
```

By default, the watermark is set to `transform: rotate(-40deg)`; implying that the watermark text is in an anti-clockwise position at 40 degrees.

b. Change the `transform: rotate` angle as appropriate.

If you do not specify any value, (for example, `transform: rotate;`) the watermark remains in a straight position. However, if you specify a positive value such as `transform: rotate(25deg);` the watermark rotates clockwise to 25 degrees.

6. Import the style sheet to an existing Teamcenter report definition.

   a. In Report Builder, open the Teamcenter Reports folder.

   b. Select the report definition for which you want to customize the watermark, and click the Report Data tab.

   c. In the Report Stylesheets area, click the Import stylesheets button in the Report Stylesheets pane.

   d. In the Dataset Type box, select a style sheet type for the new style sheet, for example, HTML. This option is required if the user wants to save the report as a dataset when the report is run.

   e. Click the Browse on the Location box to locate the style sheet file. The style sheet must be in XSL format.
f. Click **OK**.

   The style sheet is added to the **Defined Stylesheets** pane.

g. Select the new style sheet in the **Defined Stylesheets** pane, and click the + button to move it to the **Selected Stylesheets** pane.

h. Click **Modify** to save your changes.

### Add the watermark to an existing report by editing the style sheet associated with the report

In addition to the default reports with the watermark, you can use the watermark in other types of reports. You can do this by editing the style sheets associated with that report.

The **Change_object_Sample_template.xsl** and **WSO_Object_Sample_Template.xsl** style sheets contain the watermark entity by default. You can open one of these templates and copy the watermark entity to the style sheet associated with the report definition for which you want to define a watermark.

1. Open the XSL style sheet associated with the report for which you want to add a watermark.

2. Open the **Change_object_Sample_template.xsl** or **WSO_Object_Sample_Template.xsl** style sheet.

   These style sheets have the watermark entity within the **style type** tags.

3. Copy the **style type** tags containing the watermark to the style sheet of the report to which you want to add the watermark.

   You can copy the **style type** tags inside the **html** tags after the **<head></head>** tags and within the **<body></body>** tags.

   ![](Tip)

   Some reports have java scripts and the **<script type></script>** tags appear after the **<head></head>** tags and are followed by the **<body></body>** tags.

Tag to copy:

```xml
<style type="text/css">
@media print {
#watermark_print {
position:absolute;
bottom:0;
width:100%;
height:60px;
background:#6cf;
display: block;
position: fixed;
right: 0;
z-index: 5;
color: #B81C1C;
font-size: 25pt;
height: 50%;
margin: 0;
top:350px;
left:125px;
margin: auto;
```
4. Copy the `watermark_print` division tags to the end of the `<html>` tag. This division tag contains the text of the watermark.

Tags to copy:

```html
<div id="watermark_print" align="center">
```

```
</div>
```

<p>After copying (example):</p>

```xml
<html>
  <div id="watermark_print" align="center">
    <p>Confidential®</p>
  </div>
</html>
```

5. Customize the watermark by editing the style sheet associated with the report definition.

6. Import the edited style sheet.

**Create a report definition with a watermark**

Before you can define a report definition, you must create a:

1. **Query source** to create and maintain customized searches for objects in the Teamcenter database.
   
   You use Query Builder to create a query source.

2. **Closure rule** to specify the structure for processing related items.

3. **Property set** to specify additional output properties.

4. **Transfer mode** to combine closure rules and property sets to define the context of the PLM XML import or export operation.
   
   You use **PLM XML/TC XML Export Import Administration** to create transfer modes.

1. In Report Builder, click **File→Create Report**.

2. Select **Item Report** report type, and click **Next**.

3. Click **Assign** to automatically assign a report ID, and then specify a name and description for the report.

4. Specify a query source, closure rule, property set, and then click **Next**.

5. Select a defined style sheet and click **Add** or **Import stylesheets**, and then click **Finish**.

6. **Specify a watermark for this report**.

7. **Customize the watermark by editing the style sheet associated with the report definition**.

8. **Import the edited style sheet**.

**Configuring batch printing**

The **DM_IsBatchPrintingEnabled** preference must be set to **True** in order to use batch printing.
For more information about the **DM_IsBatchPrintingEnabled** preference, see the *Preferences and Environment Variables Reference* in the Teamcenter collection.

For information about creating print configurations, see *Business Modeler IDE* in the Teamcenter collection.

**Note**

The system administrator must configure batch printing prior to setting this preference.
Chapter 5: Administration

Monitor visualization components and processes

Monitor visualization components and processes in Active Workspace

The Active Workspace Viewer Administration page provides information about active visualization components and processes.

1. Log on as a user with administrator privileges, typically one in the dba group with the DBA role, but possibly others in your organization.

2. On the Home page, click Viewer Administration.

The Viewer Administration page appears. The initial view of the page includes a diagram of active visualization components and processes.

![Diagram of active visualization components and processes]
3. For details about an object included in the diagram, select the object, and click **Info**.

   ![Diagram showing a network structure with objects labeled as Assigner, PoolManager, VisProcess, and Clients.](image)

   The **Information** panel appears, providing detailed information on the visualization component or process.

   **Note**

   To maintain a secure environment, sensitive information such as IP addresses and the session ID is not displayed.

   ![Information panel showing detailed data for a VisProcess node.](image)

4. To update the display of information, click **Refresh**.
Monitoring Visualization Server components using JMX

You can monitor the Active Workspace Visualization Server system, including the Visualization Server Manager, Visualization Servers, and the Visualization Pool Assigner, using a freeware Java Management Extensions (JMX) client, such as Oracle Java Mission Control or JConsole. Monitoring these server components with JMX is useful for identifying performance bottlenecks or other problems.

A JMX client installed on the same computer as the Visualization Server components automatically detects all servers running on the machine. The information exposed by the visualization components is presented using MXBeans.

To enable remote access for JMX clients, on the server you must configure authentication (users and passwords) and encryption for the server process. Once remote access is enabled and configured, JMX clients from remote machines can connect to the server.

For information about configuring remote JMX monitoring of server processes, see Monitoring and Management Using JMX Technology in the Oracle Java SE Documentation.

Note

JMX metrics can include the following composite data types with multiple values:

**CurrentMaxTotal**: This object includes these values:
- The current value
- The highest the value has been since startup
- The total value since startup

![Table 1](chart1.png)

An example of CurrentMaxTotal values

**CurrentMaxMin**: This object includes these values:
- The current value
- The highest the value has been since startup
- The smallest the value has been since startup

![Table 2](chart2.png)

Example of CurrentMaxMin values
Visualization Server Manager

Each Visualization Server Manager hosts two MXBeans that contain information about its current state: `<poolName>` and `<poolName>` monitoring. They are located in the Administer-<poolName>-manager folder.

The `<poolName>` MXBean for the Visualization Server Manager provides the following information:

- **CacheConfiguration**
  The configuration parameters used to connect the Visualization Server Manager to the Visualization Pool Assigner.

- **Language**
  The language within which the Visualization Server Manager is running.

- **Load**
  A single ratio that represents how much of the computer’s capacity is currently in use. When this ratio is greater than or equal to 1.0, the system is completely full and new clients are rejected.

- **NumberOfAssignedServers**
  The number of Visualization Server processes in use or recently in use by client users.

- **NumberOfColdServers**
  The number of Visualization Server processes in the process of starting up, although not yet ready for use.

- **NumberOfServers**
  The total number of Visualization Server processes (cold, warm, and assigned).

- **NumberOfWarmServers**
  The number of Visualization Server processes ready for use by new client users.

- **PoolID**
  The name of this Visualization Server Manager.

- **PoolSpecificConfiguration**
  The configuration parameters passed in at startup to this Visualization Server Manager.

- **StartupDate**
  The data and time that this Visualization Server Manager was last started.

The `<poolName>` monitoring MXBean for the Visualization Server Manager provides the following information:

- **accepting**
  Whether or not this server is currently accepting new incoming users.

- **assignedServerCount**
  The number of VisView processes that are currently serving users with visualization functionality.

- **assignedVisViews**
  Specific information about each of the VisView processes that are currently assigned to users.
assignedVisViewsCount
The number of VisView processes that are currently assigned to users.

computerCpuUsageRatio
A ratio indicating how much CPU usage is consumed or unavailable on this computer.

Note
A ratio for Active Workspace MXBeans refers to a current usage value divided by the maximum usage value. For example, a CPU usage of 30% is divided by the maximum of 100% to compute a ratio of 0.3. All ratios in Active Workspace are between 0 and 1, unless the capacity of the visualization system is exceeded, in which case the ratio is greater than 1.

computerMemUsageRatio
A ratio indicating how much system memory is consumed or unavailable on this computer.

computerNetworkUsageRatio
A ratio indicating how much network usage is consumed or unavailable on this computer.

cfg
The configuration parameters passed in at startup to this server.

dateCreated
The date and time that this Visualization Pool Assigner was last started.

gpus
Specific information about each of the GPUs currently used by VisView processes.

hostname
The name or IP address of the computer that this server is hosted on.

languageID
The language that the server is currently running in. The default is English.

loadRatioAbsolute
A ratio indicating how much of the computer’s resources is consumed or unavailable on this computer.

loadRatioRelative
A ratio indicating how much of the computer’s resources is consumed or unavailable on this computer when compared to the maximum allowed resource-consumption-level (default of 0.7) of this server.

maxBandwidthBytesPerSec
The maximum allowed bandwidth (in bytes-per-second) that this server is allowed to consume.

numAssignmentsSinceStartup
The number of models that have used visualization system resources on this server.

numGpus
The number of GPUs that the computer has.
poolName
The alias defined by the administrator to identify this particular Visualization Server Manager.

Prefers
The models preferred by this server.

serverTooFullExceptions
When clients are refused visualization services, this contains the reason.

serverTooFullExceptionsCount
How many clients were refused visualization services.

serves
The models that this server has currently in memory due to requests from users.

totalGpuMemMB
The total amount of system GPU memory on this computer.

upTimeSec
How many seconds have elapsed since this server was last started.

visSysCpuUsageRatio
A ratio indicating the CPU usage of this server on the computer.

visSysGpuUsageRatio
A ratio indicating the GPU usage of this server on the computer.

visSysMemUsageRatio
A ratio indicating the amount of system memory consumption of this server on the computer.

visSysNetworkUsageRatio
A ratio indicating the amount of network usage of this server on the computer.

warmServerCount
The number of VisView processes that do not yet have users but are ready to host visualization services for new users.

warmVisViews
Specific information about the VisView processes that do not yet have users but are ready to host visualization services for new users.

warmVisViewsCount
The number of VisView processes that do not yet have users but are ready to host visualization services for new users.

Visualization Server
Each Visualization Server owned by the Visualization Server Manager hosts one MXBean that contains information about its current state. The MXBeans for the Visualization Servers are called VisView@PID_<processId>@Port_<port>. They are located in a folder called VisServers.

An MXBean for a Visualization Server provides the following information:
ClientConnections
Information about each client user connected to this Visualization Server.

DateCreated
The date and time that this Visualization Server entered a state where it was first made available to client users (warm).

Models
The IDs for the models that this Visualization Server is currently hosting.

MsSinceLastEMM
The number of milliseconds since this Visualization Server last received a message from a client.

Port
The port that this Visualization Server is currently hosting its socket server on for connections from the Visualization Pool Assigner.

ProcessCpuUsageRatio
The average amount of CPU usage that this Visualization Server has consumed on the Visualization Server Manager computer over the last 20 seconds.

ProcessGpu
General information about the GPU that the Visualization Server is using.

ProcessID
Also known as the PID, this is the identifier that the operating system uses to denote this particular Visualization Server.

ProcessMemUsageRatio
The average amount of memory usage that this Visualization Server has consumed on the Visualization Server Manager computer over the last 20 seconds.

ProcessMyGpuMemUsageRatio
The average amount of GPU memory usage that this Visualization Server has consumed (of the particular GPU that this Visualization Server is assigned to) over the last 20 seconds.

ProcessNetworkUsageRatio
The average amount of network usage that this Visualization Server has consumed on the Visualization Server Manager computer over the last 20 seconds.

ProcessTotalBytesTransfered
The number of bytes that have been received and sent by this Visualization Server process (discounts data downloaded from Teamcenter servers).

ServletConnections
Information about each connection from the Visualization Pool Assigner.

TotalNumEMMs
The number of client requests handled by this Visualization Server.

UpTimeSec
The number of seconds that have elapsed since this Visualization Server was created.
Visualization Pool Assigner

Each Visualization Pool Assigner hosts two MXBeans that contain information about its current state: Assigner and Assigner monitoring. The MXBeans are located in the Administrate Assigner manager folder.

**Note**
You must configure the Visualization Pool Assigner to populate some of the JMX metrics with meaningful information. For more information, see *Configure the Visualization Pool Assigner for JMX metrics.*

The Assigner MXBean for the Visualization Pool Assigner provides the following information:

**AssignerSpecificConfiguration**
The configuration parameters passed in at startup to this Visualization Pool Assigner.

**CacheConfiguration**
The configuration parameters used to connect this Visualization Pool Assigner to any other Visualization Pool Assigners in the Visualization Server system, and the configuration parameters used to identify this Visualization Pool Assigner such that other nodes in the Visualization Server system can connect to it.

**Load**
A single ratio that represents how much of the computer’s capacity is currently in use. When this ratio is greater than or equal to 1.0, the system is completely full and new clients are rejected.

**NumberOfPools**
The number of Visualization Server Managers that this Visualization Pool Assigner is currently connected to.

**NumberOfUsers**
The number of client users who are currently connected to Visualization Server processes through this Visualization Pool Assigner.

**StartupDate**
The data and time that this Visualization Pool Assigner was last started.

The Assigner monitoring MXBean for the Visualization Pool Assigner provides the following information:

**clientCount**
The number of users that are connected to this server.

**clients**
Specific information about the clients that have active sessions with this server.

**computerCpuUsageRatio**
A ratio indicating how much CPU usage is consumed or unavailable on this computer.

**computerMaxBandwidthBytesPerSec**
The maximum allowed bandwidth (in bytes-per-second) that this server is allowed to consume.
**computerMemUsageRatio**
A ratio indicating how much system memory is consumed or unavailable on this computer.

**computerNetworkUsageRatio**
A ratio indicating how much network usage is consumed or unavailable on this computer.

**computerTotalMemMB**
The total amount of system memory on this computer.

**config**
The configuration parameters passed in at startup to this server.

**dateCreated**
The date and time that this Visualization Pool Assigner was last started.

**loadRatioAbsolute**
A ratio indicating how much of the computer’s resources is consumed/unavailable on this computer.

**loadRatioRelative**
A ratio indicating how much of the computer’s resources is consumed or unavailable on this computer when compared to the maximum allowed resource-consumption-level (default of 0.7) of this server.

**poolCount**
The number of Visualization Server Managers known to this server.

**poolManagers**
Specific information about the Visualization Server Managers known to this server.

**serverTooFullExceptions**
When clients are refused visualization services, this contains the reason.

**serverTooFullExceptionsCount**
How many clients were refused visualization services.

**upTimeSec**
How many seconds have elapsed since this server was last started.

**visSysCpuUsageRatio**
A ratio indicating the CPU usage of this server on the computer.

**visSysMemUsageRatio**
A ratio indicating the amount of system memory consumption of this server on the computer.

**visSysNetworkUsageRatio**
A ratio indicating the amount of network usage of this server on the computer.

---

**Monitor browser activity with the F12 key**

When you press the F12 key, a window displays the developer tools provided with your web browser. You can use these tools to monitor browser activity when using Active Workspace.
Chapter 5: Administration

**Note**

These tools are not provided by the Active Workspace client. See your web browser documentation for complete information about how to use the tools accessed with the F12 key.

**Tip**

To see the version of Active Workspace that you are using, click the **Console** tab and then click the **Siemens** logo in the upper right of the window.

![Internet Explorer developer tools accessed with the F12 key](image)

**Protecting against cross site request forgery**

To prevent the Active Workspace client from cross-site forgery (CSRF) attacks, select **Internet Options** → **Security** → **Custom level** → **Security Settings** → **Miscellaneous** → **Access data sources across domains**. If **Access data sources across domains** is set to **Enable**, set this to **Disable**. This applies to all security zones in Internet Explorer.

**Note**

If for any reason you need to bypass this Same Origin policy restriction (for certain integrations with Active Workspace), you will need enablement of Cross-Origin Resource Sharing (CORS) in the Teamcenter Active Workspace deployment.

To provide CORS support:
1. Open archive **awc.war** using 7-zip or a similar program.

2. Navigate under **WEB-INF** folder.

3. Edit the **web.xml** file.

4. Scroll down to the following `<init-param>` section under **ProxyServlet**:
   ```xml
   <servlet>
     <servlet-name>ProxyServlet</servlet-name>
     <servlet-class>com.siemens.splm.clientfx.proxies.server.ProxyServlet</servlet-class>
   </servlet>
   ```

5. Save the **web.xml** file, update the Web archive, and redeploy.

**Repin a tile to the home page**

To pin a tile back on the home page, perform the following steps:

1. Log on to the rich client.

2. Search for the **Tile Collection** object owned by the user who unpinned the tile.

3. Select the unpinned tile (for example, **Favorites**).
Searching for a tile collection

4. On the menu bar, choose Edit→Properties on Relation.

5. In the Hidden property, select False and click OK.
   The next time the user logs on to Active Workspace, the tile reappears on the home page.
Utilities

Introduction to utilities

This topic contains a description of the utilities specific to Active Workspace.

For information on working with utilities, refer to the *Utilities Reference* in the Teamcenter collection.
**import_uiconfig**

Use the `import_uiconfig` utility to import column configuration XML files.

**SYNTAX**

```
import_uiconfig [-u=user-id {-p=password | -pf=password-file} -g=group] [-file=file-path] [-for_group=group-name[,group-name]] [-for_role=role-name[,role-name]] [-h]
```

**ARGUMENTS**

- `-u` Specifies the user ID.
  
  Typically `infodba` or another user with administration privileges is used as the value name for the user ID. If `-u` is used without a value, the operator system user name is automatically applied.

  **Note**
  
  If Security Services single sign-on (SSO) is enabled for your server, the `-u` and `-p` arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

- `-p` Specifies the password.
  
  If used without a value, the system assumes a null value. If this argument is not used, the system assumes the `user-ID` value to be the password.

  This argument is mutually exclusive with the `-pf` argument.

- `-pf` Specifies the password file.

- `-g` Specifies the group associated with the user.
  
  If used without a value, the user's default group is assumed.

- `-file` Specifies the path and file name for the input file.

- `-for_group` (Optional) Specifies the group or groups to which user interface configuration objects are imported.

  To specify more than one group, use commas to separate the group names.

- `-for_role` (Optional) Specifies role or roles to which user interface configuration objects are imported.

  To specify more than one role, use commas to separate the role names.
-h  
(Optional) Displays help for this utility.

**Note**

If neither `-for_group` or `-for_role` arguments are included, user interface configuration objects are imported with **Site** scope.

**EXAMPLES**

To import the configuration specified in the XML file for multiple roles, in this case: **Designer** and **engineer**.

```
import_uiconfig -u=<user_id> -p=<password> -g=dba -file=exported.xml
-fo_r_role=Designer,engineer
```

To import the configuration specified in the XML file for the **Engineering** group.

```
import_uiconfig -u=<user_id> -p=<password> -g=dba -file=exported.xml
-fo_r_group=Engineering
```

To import the configuration specified in the XML file for multiple groups, in this case **Engineering**, **system** and "**Validation Administration**".

```
import_uiconfig -u=<user_id> -p=<password> -g=dba -file=exported.xml
-fo_r_group=Engineering,system,"Validation Administration"
```

**Note**

A parameter containing spaces, such as **Validation Administration** in the preceding example, must be enclosed in double quotation marks ("").
**export_uiconfig**

Use this command to export column configuration XML files.

**SYNTAX**

```
export_uiconfig [-u=user-id { -p=password | -pf=password-file }
    -g=group] [-file=file-path] [-for_group=group-name[,group-name]]
    [-for_role=role-name[,role-name]] [-client_scope_URI=clientscope
    URI] [-client=client name] [-h]
```

**ARGUMENTS**

- **-u**
  Specifies the user ID.
  Typically `infodba` or another user with administration privileges is used as the value name for the user ID. If `-u` is used without a value, the operator system user name is automatically applied.

  **Note**
  If Security Services single sign-on (SSO) is enabled for your server, the `-u` and `-p` arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

- **-p**
  Specifies the password.
  If used without a value, the system assumes a null value. If this argument is not used, the system assumes the `user-ID` value to be the password.
  This argument is mutually exclusive with the `-pf` argument.

- **-pf**
  Specifies the password file.

- **-g**
  Specifies the group associated with the user.
  If used without a value, the user's default group is assumed.

- **-file**
  Specifies the path and file name for the output file.

- **-for_group**
  (Optional) Specifies the group or groups to which user interface configuration objects are exported.
  To specify more than one group, use commas to separate the group names.

- **-for_role**
  (Optional) Specifies role or roles to which user interface configuration objects are exported.
  To specify more than one role, use commas to separate the role names.
-client_scope_URI
(Optional) Specifies, for export only, the column configurations and command contributions corresponding to the indicated client scope only.

[Note]

Client scope refers to a sublocation in the client, not group or role.

-client
(Optional) Specifies, for export only, the column configurations and command contributions corresponding to this client only.

-h
(Optional) Displays help for this utility.

EXAMPLES
To export all the site-wide column configurations:

```
export_uiconfig -u=<user_id> -p=<password> -g=dba -file=exported.xml
```

To export all the column configurations specific to the Engineering group:

```
export_uiconfig -u=<user_id> -p=<password> -g=dba -file=exported.xml
    -for_group=Engineering
```

To export the inbox table column configuration specific to the Engineering group:

```
export_uiconfig -u=<user_id> -p=<password> -g=dba -file=exported.xml
    -for_group=Engineering -Client_scope_URI=fnd0Inbox
```
**bomindex_admin**

Adds structured content to the search index.

**SYNTAX**

```
bomindex_admin [-u=user-id [-p=password | -pf=password-file] -g=group] 
-logfile=location_of_logfile -function=[create | delete | list | upgrade] 
-inputfile=location_of_inputfile
```

**ARGUMENTS**

- **-u**
  Specifies the user ID.
  This is generally *infodba* or another user with administration privileges.

  **Note**
  If Security Services single sign-on (SSO) is enabled for your server, the user and password arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

- **-p**
  Specifies the password.
  This argument is mutually exclusive with the **-pf** argument.

- **-pf**
  Specifies the password file.

- **-g**
  Specifies the group associated with the user.
  If used without a value, the user's default group is assumed.

- **-logfile**
  Specifies the location of the log file written by the utilities. You can specify a different location for each utility.

- **-function=function-name**
  Performs the following functions:

  - **create**
    Creates the *BOMIndexAdminData* objects based on the input file.

  - **delete**
    Finds the *BOMIndexAdminData* objects in the input file and marks them as deleted.

  - **list**
    Creates the input file for update or delete operations for existing *BOMIndexAdminData* business objects. The generated file also reports *BOMIndexAdminData* properties such as `window-uid`.

  - **upgrade**
    Upgrades the definition of BOM index tables when the property set is modified.
-inputfile
Specifies the location of a comma-separated variable (CSV) file containing the list of structure objects to index.

The input file line format is as follows:

item-query-string | item-revision-ID | base-revision-rule | effectivity-unit | effectivity-end-item-query-string | effectivity-date (dd-mmm-yyyy hh:mm:ss) | variant-rules

The variant rules are comma delimited, and follow this format:

SVR-name:owning-item-query-string:owning-item-revision-ID

The topline item revision is the default owner.

Following is an example of an input file (bomindex_admin_input.txt):

item_id=HDD-0527 | B | Any Status; Working | 5 | item_id=HDD-0527 | 31-May-2013 00:00:00 | vrule1: item_id=OwnItem1:B, vrule2:, vrule3: item_id=OwnItem3:A

EXAMPLE

Run the following command to create a search index of structures:

bomindex_admin -u=username -p=password -g=dba -logfile=C:\Scratch\log\log1.txt -function=create -inputfile=C:\Scratch\log\bomindex_admin_input.txt
req_bulk_upgrade_docm_to_html

Beginning in Active Workspace 2.3, all requirements are rendered as HTML for scalable navigation in the viewer. Use this command to convert requirements that existed prior to 2.3 so they display in the viewer.

**Note**

This utility must be run after the Active Workspace Requirements Management feature is installed.

**SYNTAX**

```
req_bulk_upgrade_docm_to_html [-u=user-id  -p=password | 
-pf=password-file  -g=group] 
[-failedObjects=file-name] [-path=file-name] [-dumpLogs] [-forceUpdate]
```

**ARGUMENTS**

- **-u** Specifies the user ID to be used for the upgrade. This is generally `infodba` or another user with administration privileges.

- **-p** Specifies the password for the specified user ID.

- **-pf** Specifies the password file. This argument is mutually exclusive with the **-p** argument.

- **-g** Specifies the group associated with the user.

- **-failedObjects** (Optional) Specifies the full path to the `failed_IDS.txt` file that reports the specification element subtypes that were not converted.

- **-dumpLogs** (Optional) Dumps a detailed debug log with more information about the point in the code the utility is failing or which full-text dataset is causing the error.

- **-path** Specifies the full path to the `.Net-HtmlConverter01.exe` utility.

- **-forceUpdate** (Optional) Forces the update and repair of all requirements in database even if they were previously converted.

- **-h** Displays help for this utility.
runTcFTSIndexer

Indexes Teamcenter data into the Solr search engine. Run this command from the FTS_INDEXER_HOME directory, for example, TC_ROOT\TcFTSIndexer\bin.

**SYNTAX**

```
runTcFTSIndexer [-dispatcher] [-maxconnections] -status -stop -service -task=[objdata | structure]:flow-action
```

**ARGUMENTS**

- **-dispatcher**
  Uses the Dispatcher for indexing. Using the Dispatcher provides the capability to use multiple machines as resources for indexing data. To use this argument, the TcFTSIndexer translators must be installed and the Dispatcher Scheduler and Module must be running. If this argument is not used, the indexer runs in stand-alone mode. Following is an example:

  ```
  runTcFTSIndexer -dispatcher -task=objdata:test
  ```

- **-maxconnections**
  Sets a new value for maximum TcServer connections to use at any given time, for example:

  ```
  runTcFtsIndexer -maxconnections=5
  ```

- **-service**
  Starts the indexer as a Java Remote Method Invocation (RMI) service in a console, for example:

  ```
  runTcFtsIndexer -service
  ```

  This argument can also be used with the -task argument to execute a flow when starting the service, for example:

  ```
  runTcFtsIndexer -service -task=objdata:reindex
  ```

- **-status**
  Checks the status of the indexer and reports the flows running in the indexer. For example, the following shows the status for all the flows:

  ```
  runTcFTSIndexer -status
  ```

  This argument can also be used with the -task argument. For example, the following command shows the status of the objdata:reindex flow:

  ```
  runTcFtsIndexer -status -task=objdata:reindex
  ```

- **-stop**
  Stops indexing flows that use an interval.

  Use the following command to stop all flows with intervals:

  ```
  runTcFTSIndexer -stop
  ```

  This argument can also be used with the -task argument. For example, the following command stops the objdata:sync flow with intervals:

  ```
  runTcFtsIndexer -stop -task=objdata:sync
  ```
-task
Runs an indexing task using this format:

```
-task=type:flow-action
```

The type of task can be objdata for object data indexing or structure for structured content indexing. Flow actions are specific to the type of task.

**objdata flow actions**

- **objdata:test**
  Verify whether the environment is set up correctly prior to running the indexer.

- **objdata:clear**
  Clears the existing indexed data and cached files.

- **objdata:index**
  Without clearing the existing index, performs indexing for the specified time period.

- **objdata:reindex**
  Clears the existing indexed data and performs an initial index of data.

- **objdata:recover**
  The action of this option depends on whether it is used after index or reindex.
  After reindex, this option indexes the batch or batches of failed reindex flow time durations as well as any new objects created between the last reindex and the recover start time.
  After index, this option indexes only the batch or batches of failed index flow time durations.

- **objdata:sync**
  Updates the index with changes to data between the previous run and the current run.
  Use this for the index data as changes occur.
  The sync action can have the `-interval=seconds` argument.
  For example, to synchronize object data using the stand-alone indexer every 300 seconds (5 minutes):

  ```bash
  runTcFTSIndexer -task=objdata:sync -interval=300
  ```

  - The value must be greater than 0.
  - The sync action run without the `-interval=seconds` argument runs once.
To update the index with Access Manager delta changes, you must use the pool manager admin console to terminate any unassigned tcserv\r
\r
\r
- **objdata:indexsyncfailures**
  
  Use this flow to index sync failures that required manual intervention.

**structure flow actions**

- **structure:recoverfailures**
  
  Changes all product configurations with failed states to the **ReadyToIndex** state or the **MarkedForDeletion** state. For example:

  ```text
  runTcFTSIndexer -task=structure:recoverfailures
  ```

- **structure:reset**
  
  Resets the given **PRODUCT_CONFIG_UID** setting to the **ReadyToIndex** state or the **MarkedForDeletion** state. For example:

  ```text
  runTcFTSIndexer -task=structure:reset
  ```

- **structure:resetall**
  
  Downloads the latest transform and schema files, resets all active product configurations to the **ReadyToIndex** state, and resets all deleted product configurations back to the **MarkedForDeletion** state. For example:

  ```text
  runTcFTSIndexer -task=structure:resetall
  ```

- **structure:show**
  
  Shows a summary of all configured product configurations, for example:

  ```text
  runTcFTSIndexer -task=structure:show
  ```

- **structure:sync**
  
  Queues synchronization and delete actions for all product configurations, for example:

  ```text
  runTcFTSIndexer -task=structure:sync
  ```

- **structure:syncone product-configuration-UID**
  
  Queues synchronization and delete actions for a single product configuration UID, for example:

  ```text
  runTcFTSIndexer -task=structure:syncone go2RkWxoqd$DyB
  ```

- **structure:test**
  
  Verifies that the environment is set up correctly prior to running the indexer, for example:
runTcFTSIndexer -task=structure:test
smlutility

Use this command to create or delete search indexing views for classification.

For smlutility details not included below, refer to the Utilities Reference in the Teamcenter collection.

SYNTAX

smlutility -create_indexing_views [-u=user-id] [-p=password] [-g=group] [-reportfile=file-name] [-listIds=class-id] [-delimiter=user-specified-delimiter] [-recursive]


ARGUMENTS

-create_indexing_views or -delete_indexing_views
Creates or deletes search indexing views for classification.

-u
Specifies the user ID.

This is generally infodba or another user with administration privileges. If this argument is used without a value, the operating system user name is used.

Note

If Security Services single sign-on (SSO) is enabled for your server, the -u and -p arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p
Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the user-ID value to be the password.

This argument is mutually exclusive with the -pf argument.

-g
Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-reportfile
Specifies the path to the output file. If you do not specify this argument (or if the file cannot be opened), the output is written to stdout.

-listIds
Specifies a list of classes for which search indexing views should be created or deleted.
When creating, if no class ID is specified, indexing views for all classes that do not have indexing views are created. If a class contains a default view, the attributes of the default view are copied to the Search Index view. Otherwise, all the attributes of the class are set as attributes of the search indexing view.

When deleting, if no class ID is specified, indexing views for all classes are deleted.

**-delimiter**
Specifies a user-defined delimiter.

**-recursive**
Specifies that the index view be created (or deleted) for all subclasses.
s2clsocial_delete_utility

Deletes the commentary objects (questions, answers, comments, ratings, and helpful objects) for item revisions. If a commentary object is referenced by a noncommentary object, this utility cannot delete the commentary.

Based on the entered target string, a list of matching item revisions are listed, and you are prompted to delete the associated comments. Enter y=yes, n=no, or q=quit.

**SYNTAX**

s2clsocial_delete_utility [-u=user-id -p=password -g=group]
[-target=item-revision-name] [-h]

**ARGUMENTS**

-u
Specifications the user ID.

This is generally **infodba** or another user with administration privileges. If this argument is used without a value, the operating system user name is used.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Security Services single sign-on (SSO) is enabled for your server, the -u and -p arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.</td>
</tr>
</tbody>
</table>

-p
Specifications the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the user-ID value to be the password.

This argument is mutually exclusive with the -pf argument.

-g
Specifications the group associated with the user.

If used without a value, the user's default group is assumed.

-target
Specifications the item revision name from which to remove comments. A wildcard (*) is valid.

-h
(Optional) Displays help for this utility.

**EXAMPLES**

To delete all commentary associated with an item revision name beginning with hdd-0500-C:

s2clsocial_delete_utility -u=user-id -p=password -g=dba -target=hdd-0500-C*

To delete all commentary associated with an item revision name **hdd-0500-Cover Label**:

s2clsocial_delete_utility -u=user-id -p=password -g=dba -target="hdd-0500-Cover Label"
Chapter 6: Preferences

Introduction to preferences
This topic contains a description of the preferences specific to Active Workspace.
For information on working with preferences, refer to Teamcenter Basics in the Teamcenter collection.

Hosting preferences
Additional preferences enable the hosting of Active Workspace functionality in the rich client or other Teamcenter clients and in NX.

- **ActiveWorkspaceHosting.URL**
  Specifies the URL used by Teamcenter to communicate with Active Workspace for hosted operations such as search and open item.
  This preference takes precedence over the ActiveWorkspaceHosting.NX.URL, ActiveWorkspaceHosting.RAC.URL, and ActiveWorkspaceHosting.WorkflowEmail.URL preferences.

- **ActiveWorkspaceHosting.NX.URL**
  Specifies the URL used by Teamcenter Integration for NX to communicate with Active Workspace for hosted operations.

- **ActiveWorkspaceHosting.RAC.URL**
  Specifies the URL used by the rich client to communicate with Active Workspace for hosted operations.

- **ActiveWorkspaceHosting.WorkflowEmail.URL**
  Specifies the URL used by workflow to communicate with Active Workspace for email links.

- **AWC_NX_AddComponentSupportedTypes**
  Enables addition of the specified object types as components in NX when selected in Active Workspace. Only Item, ItemRevision, and dataset types and subtypes are supported.

- **AWC_NX_OpenSupportedTypes**
  Enables opening the specified object types in NX when selected in Active Workspace. Only Item, ItemRevision, and dataset types and subtypes are supported.

- **AWC_OC_OpenSupportedTypes**
  Enables opening the specified object types in Teamcenter Client for Microsoft Office when selected in Active Workspace. Only MSWord, MSWordX, MSWordTemplateX,
**AWC_RAC_OpenSupportedTypes**

Enables opening the specified object types in the rich client when selected in Active Workspace. Only Item, ItemRevision, Folder, and dataset types and subtypes are supported.

**AWC_VIS_OpenSupportedTypes**

Enables opening the specified object types in Teamcenter lifecycle visualization when selected in Active Workspace. Only Dataset (DirectModel and UGMaster), Item, ItemRevision, BomLine, BomView, and BomViewRevision, types and subtypes are supported.

**Note**

These types are supported only if they have product structure child objects and/or IMAN_Rendering relations.

**TC_Use_ActiveWorkspace_Inbox**

Specifies whether to display the Active Workspace inbox instead of the host environment (rich client) inbox when you click My Worklist in the host environment.

**TC_Use_ActiveWorkspace_Summary**

Specifies whether to display the Active Workspace summary instead of the rich client Summary view when you click Summary in the rich client.

These Active Workspace hosting preferences are detailed in Teamcenter documentation such as Preferences and Environment Variables Reference, Teamcenter Basics, and Teamcenter Integration for NX.

**Note**

By default, Active Workspace hosting preferences have a protection scope of Site. Enabling these preferences enables Active Workspace functionality for all users of the client.

To provide flexibility for access to Active Workspace functionality in other clients, Siemens PLM recommends changing the preference definition to have Group, Role, or User protection scope, as described in the Preferences and Environment Variables Reference.

**AWC_DefaultCreateTypes**

**DESCRIPTION**

Defines the object types to display when creating a new object in the Active Workspace client. The list of available types is displayed in the Create panel.

**Note**

All subtypes associated with the object type(s) defined in the preference are also displayed in the list of available types.
The internal name of any valid object type. For example, **Item** and **Document**.

**Item**

**Folder**

**Site preference.**

---

### AWC_DisplayableChangeTypes

**DESCRIPTION**

Defines the **ChangeltemRevision** types to display in the Active Workspace Change Management.

**VALID VALUES**

Any valid object type.

**Note**

If you are using the Aerospace and Defense template, you must add the following values to view change objects in Active Workspace:

- **Adc0ChangeRqstRevision**
- **Adc0ChangeNoticeRevision**
- **ProblemReportRevision**
- **ChangeRequestRevision**
- **ChangeNoticeRevision**

**SCOPE**

Site preference.
AWC_req_do_html_conversion

**DESCRIPTION**
Performs the HTML conversion of requirements content when requirements are imported and updated in Teamcenter Systems Engineering.

**VALID VALUES**
- true
- false

**DEFAULT VALUES**
true

**SCOPE**
Site preference.

AWC_req_viewer_page_size

**DESCRIPTION**
Defines the number of requirements that can be viewed at a time in the Active Workspace Content—Viewer tab.

**VALID VALUES**
Accepts one or more integers as values.

**DEFAULT VALUES**
3

**DEFAULT PROTECTION SCOPE**
User preference.

AWC_ReqEditor

**DESCRIPTION**
Defines the type of editor that opens when paragraphs, requirements, or requirements specifications are opened for editing.

**Note**
Once you edit and save content in live Word or live Excel, the content always opens as a PDF and you must edit in those applications. Likewise, once you edit and save content in the rich text editor, it always opens in the rich text editor, unless you save it in live Word for live Excel.

**VALID VALUES**
- Plain
  The plain text editor opens when the object is opened for editing.
The rich text editor opens when the object is opened for editing.

Word
The object opens as a PDF, and you must use Live Integration with Microsoft Word, also known as live Word, to edit the content.

Note
These values are case-sensitive. If an incorrect value is entered, the default value is used.

Plain

AWC_search_automaticWildcard

DESCRIPTION
To enhance local metadata searches, automatically adds the asterisk (*) character as a prefix, suffix, or both for search criteria for all string attributes.

VALID VALUES
0 Search criteria are not altered.
1 The asterisk (*) character is added as a suffix.
2 The asterisk (*) character is added as a prefix.
3 The asterisk (*) character is added as both a suffix and a prefix.

Note
All other values have the same effect as adding 0.

EXAMPLES

<table>
<thead>
<tr>
<th>Preference value</th>
<th>Search value</th>
<th>Search results</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>part</td>
<td>All objects whose properties contain the complete word part.</td>
</tr>
<tr>
<td>1</td>
<td>part</td>
<td>All objects whose properties contain words that begin with part, for example, partial and partition.</td>
</tr>
<tr>
<td>2</td>
<td>part</td>
<td>All objects whose properties contain words that end with part, for example, depart and repart.</td>
</tr>
<tr>
<td>Preference value</td>
<td>Search value</td>
<td>Search results</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>3</td>
<td>part</td>
<td>All objects whose properties contain the word <em>part</em> in any position, for example, <em>departed</em> and <em>repartitioned</em>.</td>
</tr>
</tbody>
</table>

The preference is ignored if any of the following occurs:

- The user enters text in double quotation marks (for example, “*part*”).
- The user saves their search. When the user retrieves their saved search, the value of the preference at the time of re-execution is applied.

**Note**

If the user explicitly enters an asterisk in their search text, the wildcard actions are combined. For example, if the preference value is 2 (wildcard as a prefix) and the user enters *part*, the search is performed as if it were *part*.

**SCOPE**

Site preference.

**AWC_visFloorOffset**

**DESCRIPTION**

Specifies the default floor offset for the viewer in Active Workspace.

**VALID VALUES**

A numerical value.

**DEFAULT VALUES**

0.0

**SCOPE**

User preference.

**AWC_visFloorOn**

**DESCRIPTION**

Specifies the default floor visibility for the viewer in Active Workspace.

**VALID VALUES**

- **true**
  Specifies the floor is on.
- **false**
  Specifies the floor is off.

**DEFAULT VALUES**

false
SCOPE
User preference.

AWC_visFloorPlaneOrientation

DESCRIPTION
Specifies the default floor plane orientation for the viewer in Active Workspace.

VALID VALUES
Any numerical value from 0 to 5.

<table>
<thead>
<tr>
<th>Value</th>
<th>Floor plane orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>XY</td>
</tr>
<tr>
<td>1</td>
<td>XZ</td>
</tr>
<tr>
<td>2</td>
<td>YZ</td>
</tr>
<tr>
<td>3</td>
<td>-XY</td>
</tr>
<tr>
<td>4</td>
<td>-XZ</td>
</tr>
<tr>
<td>5</td>
<td>-YZ</td>
</tr>
</tbody>
</table>

DEFAULT VALUES
1

SCOPE
User preference.

AWC_visGridOn

DESCRIPTION
Specifies the default grid visibility for the viewer in Active Workspace.

VALID VALUES
true
Specifies the grid is on when the floor is turned on.
false
Specifies the grid is off when the floor is turned on.

DEFAULT VALUES
true

SCOPE
User preference.

AWC_visMaterial

DESCRIPTION
Specifies the default material for JT files displayed in the Active Workspace viewer.
Any numerical value from 0 to 29.

<table>
<thead>
<tr>
<th>Value</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Shiny metal</td>
</tr>
<tr>
<td>1</td>
<td>Brushed metal</td>
</tr>
<tr>
<td>2</td>
<td>Shiny plastic</td>
</tr>
<tr>
<td>3</td>
<td>Analysis</td>
</tr>
<tr>
<td>4</td>
<td>Flat</td>
</tr>
<tr>
<td>5</td>
<td>Red glossy plastic</td>
</tr>
<tr>
<td>6</td>
<td>Blue glossy plastic</td>
</tr>
<tr>
<td>7</td>
<td>Green glossy plastic</td>
</tr>
<tr>
<td>8</td>
<td>Gray glossy plastic</td>
</tr>
<tr>
<td>9</td>
<td>Black glossy plastic</td>
</tr>
<tr>
<td>10</td>
<td>Brown glossy plastic</td>
</tr>
<tr>
<td>11</td>
<td>Yellow glossy plastic</td>
</tr>
<tr>
<td>12</td>
<td>Teal glossy plastic</td>
</tr>
<tr>
<td>13</td>
<td>White glossy plastic</td>
</tr>
<tr>
<td>14</td>
<td>Clear plastic</td>
</tr>
<tr>
<td>15</td>
<td>Chrome</td>
</tr>
<tr>
<td>16</td>
<td>Copper</td>
</tr>
<tr>
<td>17</td>
<td>Gold</td>
</tr>
<tr>
<td>18</td>
<td>Brass</td>
</tr>
<tr>
<td>19</td>
<td>Steel</td>
</tr>
<tr>
<td>20</td>
<td>Brushed chrome</td>
</tr>
<tr>
<td>21</td>
<td>Brushed aluminum</td>
</tr>
<tr>
<td>22</td>
<td>Brushed titanium</td>
</tr>
<tr>
<td>23</td>
<td>Glass</td>
</tr>
<tr>
<td>24</td>
<td>Smoky glass</td>
</tr>
<tr>
<td>25</td>
<td>Red paint</td>
</tr>
<tr>
<td>26</td>
<td>Gray paint</td>
</tr>
<tr>
<td>27</td>
<td>Black paint</td>
</tr>
<tr>
<td>28</td>
<td>Blue paint</td>
</tr>
<tr>
<td>29</td>
<td>Rubber</td>
</tr>
</tbody>
</table>

**Note**

This is the default value.
**AWC_visNavigationMode**

**DESCRIPTION**
Specifies the default navigation mode for the viewer in Active Workspace.

**VALID VALUES**
- **ROTATE**
  Sets the default navigation mode in the viewer to rotate.
- **PAN**
  Sets the default navigation mode in the viewer to pan.
- **ZOOM**
  Sets the default navigation mode in the viewer to zoom.

**DEFAULT VALUES**
- **ROTATE**

**SCOPE**
User preference.

**AWC_visReflectionOn**

**DESCRIPTION**
Specifies the default reflection visibility for the viewer in Active Workspace.

**VALID VALUES**
- **true**
  Specifies the reflection is on when the floor is turned on.
- **false**
  Specifies the reflection is off when the floor is turned on.

**DEFAULT VALUES**
- **false**

**SCOPE**
User preference.

**AWC_visShading**

**DESCRIPTION**
Specifies the default shading mode for the viewer in Active Workspace.
### AWC_visShadowOn

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>Specifies the default shadow visibility for the viewer in Active Workspace.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VALID VALUES</strong></td>
<td>true</td>
</tr>
<tr>
<td></td>
<td>false</td>
</tr>
<tr>
<td><strong>DEFAULT VALUES</strong></td>
<td>false</td>
</tr>
<tr>
<td><strong>SCOPE</strong></td>
<td>User preference.</td>
</tr>
</tbody>
</table>

### AWC_visStdViewOrientationFront

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>Maps the <strong>Front</strong> view orientation in the Active Workspace viewer to the X, Y, or Z axis.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VALID VALUES</strong></td>
<td>+X</td>
</tr>
<tr>
<td></td>
<td>+Y</td>
</tr>
<tr>
<td></td>
<td>+Z</td>
</tr>
<tr>
<td></td>
<td>−X</td>
</tr>
<tr>
<td></td>
<td>−Y</td>
</tr>
<tr>
<td></td>
<td>−Z</td>
</tr>
</tbody>
</table>

The **Back** view is always the inverse of the **Front** orientation. For example, if **Front** has a value of +Y, then **Back** is −Y.
Use **AWC_visStdViewOrientationLeft** and **AWC_visStdViewOrientationTop** to specify the **Left** and **Top** view orientations.

The **Isometric** and **Trimetric** orientations always have a value of **+Isometric** and **-Isometric**, respectively.

**DEFAULT VALUES**

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>+Y</td>
</tr>
</tbody>
</table>

**SCOPE**

Site preference.

**AWC_visStdViewOrientationLeft**

**DESCRIPTION**

Maps the **Left** view orientation in the Active Workspace viewer to the X, Y, or Z axis.

**VALID VALUES**

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>+X</td>
</tr>
<tr>
<td>+Y</td>
</tr>
<tr>
<td>+Z</td>
</tr>
<tr>
<td>–X</td>
</tr>
<tr>
<td>–Y</td>
</tr>
<tr>
<td>–Z</td>
</tr>
</tbody>
</table>

**AWC_visStdViewOrientationTop**

**DESCRIPTION**

Maps the **Top** view orientation in the Active Workspace viewer to the X, Y, or Z axis.

**VALID VALUES**

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>+X</td>
</tr>
<tr>
<td>+Y</td>
</tr>
<tr>
<td>+Z</td>
</tr>
</tbody>
</table>
–X  X-axis looking in the negative X direction.
–Y  Y-axis looking in the negative Y direction.
–Z  Z-axis looking in the negative Z direction.

The **Bottom** view is always the inverse of the **Top** orientation. For example, if **Top** has a value of –Z, then **Bottom** is +Z.

Use **AWC_visStdViewOrientationFront** and **AWC_visStdViewOrientationLeft** to specify the **Front** and **Left** view orientations.

The **Isometric** and **Trimetric** orientations always have a value of +Isometric and -Isometric, respectively.

### AWC_visTrihedronOn

**DESCRIPTION**

Specifies the default trihedron visibility state for the viewer in Active Workspace.

**VALID VALUES**

true

Specifies the trihedron is on.

false

Specifies the trihedron is off.

**DEFAULT VALUES**

true

**SCOPE**

User preference.

### AWS_Default_Query_Operator

**DESCRIPTION**

Defines the behavior for multi-string searches.

By default, searching for **HDD 0500** returns results with both HDD and 0500 in them. For example, **HDD 0500** or **HDD 050055**. If this preference is set to OR, this same search returns results with either HDD or 0500 in them. For example, **HDD 123** or **ABC 0500**.

**VALID VALUES**

AND

OR
AWS_FullTextSearch_Solr_URL

**DESCRIPTION**
Specifies the URL to the Solr installation for the Active Workspace full-text search functionality.

This preference is used by **TcServer** to query for indexed data and by **TcFtsIndexer** to index data in Solr.

**VALID VALUES**
- http://host-name:port/solr

**DEFAULT VALUE**
- http://localhost:8983/solr

**SCOPE**
Site preference.

AWS_Preferred_Attributes

**DESCRIPTION**
Defines the high-priority attributes for listing search results.

**Note**
Attributes and types specified in this preference must be marked as indexable. For more information, see *Define index data and filters*.

**VALID VALUES**
One or more strings using the following format:

```
type.internal_name_of_attribute
```

**DEFAULT VALUE**
None. This preference must be manually entered into the database. Create new preferences using the **Options** dialog box, accessed from the **Edit** menu in the Teamcenter rich client.

**EXAMPLES**
POM_application_object.owning_user

**SCOPE**
User preference.
**AWS_SearchPreFilter_Property1**

**DESCRIPTION**

Specifies a property on which to prefilter search results.

**VALID VALUES**

The first row contains the property name in the format: `type.property`. The second row and beyond contain the internal values for the property.

You can use keywords as follows:

- Use `$TODAY`, `$THIS_WEEK`, and `$THIS_MONTH` for date properties.
- Use `$ME` for properties that store the values as “user Name ( user ID )”.
- Use `$MY_GROUP` for properties that store the group name.

These keywords are substituted with the correct value when performing the search.

**DEFAULT VALUE**

First row: `POM_application_object.owning_user`

*Note*

This property must be marked for indexing. Only indexed properties can be used as prefilters.

For more information, see *Define index data and filters*.

Second row: *(blank)*

**EXAMPLES**

The following example specifies the `user1` user as an option for the `Owner` prefilter:

```
POM_application_object.owning_user
user1 ( user1id )
```

where `user1` is the user name and `user1id` is the user ID.

The following example specifies the `Engineering` and `dba` groups as options for the `Group ID` prefilter:

```
POM_application_object.owning_group
Engineering
dba
```

**SCOPE**

Site preference.

**AWS_SearchPreFilter_Property2**

**DESCRIPTION**

Specifies a second property on which to prefilter search results.
The first row contains the property name in the format: `type.property`. The second row and beyond contain the internal values for the property.

You can use keywords as follows:

- Use `$TODAY`, `$THIS_WEEK`, and `$THIS_MONTH` for date properties.
- Use `$ME` for properties that store the values as “user Name (user ID)”.
- Use `$MY_GROUP` for properties that store the group name.

These keywords are substituted with the correct value when performing the search.

<table>
<thead>
<tr>
<th>First row:</th>
<th>Second row:</th>
<th>Third row:</th>
<th>Fourth row:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WorkspaceObject.object_type</code></td>
<td><code>DocumentRevision</code></td>
<td><code>ItemRevision</code></td>
<td><code>RequirementRevision</code></td>
</tr>
</tbody>
</table>

**Note**

This property must be marked for indexing. Only indexed properties can be used as prefilters.

For more information, see Define index data and filters.

<table>
<thead>
<tr>
<th>EXAMPLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>The following example specifies requirement revisions, item revisions, document revisions, schedule revisions, and validation requirement revisions as options for the Type prefilter:</td>
</tr>
<tr>
<td><code>WorkspaceObject.object_type</code></td>
</tr>
<tr>
<td><code>RequirementRevision</code></td>
</tr>
<tr>
<td><code>ItemRevision</code></td>
</tr>
<tr>
<td><code>DocumentRevision</code></td>
</tr>
<tr>
<td><code>ScheduleRevision</code></td>
</tr>
<tr>
<td><code>ValidationReqRevision</code></td>
</tr>
</tbody>
</table>

The following example specifies the Last Modified Date prefilter. The default options are `$TODAY`, `$THIS_WEEK`, and `$THIS_MONTH`.

<table>
<thead>
<tr>
<th>SCOPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site preference.</td>
</tr>
</tbody>
</table>
**Awp0FullTextSavedSearch.CellProperties**

**DESCRIPTION**

Specifies properties displayed on tiles for `Awp0FullTextSavedSearch` object types.

- The first two properties are displayed without labels and are formatted as a primary title and subtitle. The remaining properties are displayed in the tile as `name:value`.

- **Date** type properties are always displayed in `dd-mmm-yy hh:mm` format.

**Note**

You do not need to add filter-related properties to this preference. If any filter is active, its value is displayed by default.

**VALID VALUES**

Any property of the `Awp0FullTextSavedSearch` object type.

**DEFAULT VALUE**

```
object_name
creation_date
awp0search_string
```

**SCOPE**

Site preference.

**ChangeNoticeRevision_default_workflow_template**

**DESCRIPTION**

Defines the workflow that starts when a change notice revision is submitted.

**VALID VALUES**

Any valid workflow type.

**DEFAULT VALUES**

`ChangelItemRevisionDefaultWorkflowTemplate`

**SCOPE**

Site preference.

**ChangeRequestRevision_default_workflow_template**

**DESCRIPTION**

Defines the workflow that starts when a change request revision is submitted.

**VALID VALUES**

Any valid workflow type.

**DEFAULT VALUES**

`ChangelItemRevisionDefaultWorkflowTemplate`
**CM_automate_derive_propagation**

**DESCRIPTION**

Enables the automatic propagation of relations from source change objects to the derived change objects when deriving a change in Active Workspace. Specify the relations to be propagated using one of the preferences listed in the table.

For more information about these preferences, see *Workflow Designer* and *Preferences and Environment Variables Reference* in the Teamcenter help collection.

<table>
<thead>
<tr>
<th>When deriving a change object from a</th>
<th>Set the relations propagated using</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR</td>
<td>CM_ProblemReportRevision_Relations_To_Propagate</td>
</tr>
<tr>
<td>Deviation request</td>
<td>CM_Cm0DevRqstRevision_Relations_To_Propagate</td>
</tr>
<tr>
<td>ECR</td>
<td>CM_ChangeRequestRevision_Relations_To_Propagate</td>
</tr>
</tbody>
</table>

**VALID VALUES**

| TRUE | Propagates the relations. |
| FALSE | Does not propagate the relations. |

**DEFAULT VALUES**

| TRUE |

**CMMyChangesSublocationQuery**

**DESCRIPTION**

Registers the saved queries for the contents of the tabs in My Changes.

**VALID VALUES**

Valid values are in the format:

```
tab_name,saved_query_name
```

where `tab_name` is the name of the tab and `saved_query_name` is the name of the saved query.

**DEFAULT VALUES**

| Saved,MyChangesSaved |
| Submitted,MyChangesSubmitted |
| Approved,MyChangesApproved |
**Default_Transient_Server**

**DESCRIPTION**
Specifies the default transient file server location. Environment variable settings override this FCC configuration file setting.

**VALID VALUES**
Accepts a single value. Must be a valid transient file server location. For example:

```
Default_Transient_Server=
http://www.transientserver.com
```

**DEFAULT VALUES**
None.

**DEFAULT PROTECTION SCOPE**
Site preference.
<deviation_type_name>_default_workflow_template

DESCRIPTION

Defines the workflow that starts when a physical part deviation is submitted.

Example

Name: DeviationDoc_default_workflow_template
Value: The DeviationDoc_default_workflow_template has a value of TCM Release Process

VALID VALUES

Any valid workflow type.

DEFAULT VALUES

None

SCOPE

Site preference.
**Disposition_Propagate_To_Child**

**DESCRIPTION**
Determines if the disposition defined during the generation or duplication of the as-maintained structure is propagated to all of the children parts generated.

**VALID VALUES**
- **true**: Propagates the disposition to all child parts of the structure.
- **false**: Applies the disposition only to the top part of the structure.

**DEFAULT VALUE**
true

**SCOPE**
Site preference.

**EPM_notify_url_format**

**DESCRIPTION**
Determines which types of URLs are inserted into Teamcenter's task notification email.

**VALID VALUES**
Accepts one or more of the following values:
- **rich**: Inserts a URL to task object that launches the rich client when clicked.
- **dhtml**: Inserts a URL to task object that launches the thin client when clicked.
- **activeworkspace**: Inserts a URL to task object that launches Active Workspace when clicked.
- **none**: No URL is inserted in the notification email.

**DEFAULT VALUES**
- **rich**
- **dhtml**
- **activeworkspace**

**DEFAULT PROTECTION SCOPE**
All.

**<ExternalSystemObjectName>.CellProperties**

**DESCRIPTION**
For external system objects, defines what properties to show on the tiles for objects in Active Workspace. The first two properties in the list are displayed without labels and
are formatted as a primary title and subtitle. The remaining properties are displayed in the tile as name:value.

For details about this preference, see <object>.CellProperties.

**<ExternalSystemObjectName>.INFORENDERING**

**DESCRIPTION**

Links objects of type <ExternalSystemObjectName> to the XRT style sheet used for the info panel in Active Workspace.

The parameters for this preference are the same as those for the <ExternalSystemObjectName>.SUMMARYRENDERING preference.

For more information, see <ExternalSystemObjectName>.SUMMARYRENDERING.

**<ExternalSystemObjectName>.SUMMARYRENDERING**

**DESCRIPTION**

Links objects of type <ExternalSystemObjectName> to the XRT style sheet used for the summary panel in Active Workspace.

Indicates the XML rendering dataset used to render the properties shown in the Summary tab for a particular business object type, using the form of business-object-type.SUMMARYRENDERING=xml-rendering-style-sheet-dataset. For example, the Item.SUMMARYRENDERING=ItemSummary preference means that for Item business objects, the ItemSummary XML rendering style sheet dataset is used to render the properties on the Summary tab in the user interface.

Teamcenter property displays are based on an XML style sheet. The style sheet must be created and registered by a system administrator.

For more information, see Client Customization in the Teamcenter collection.

The XML file allows users to define a set of properties to display, the display order, and the user interface rendering component to be used.

This preference determines the XML rendering for displaying the defined type’s summary properties. If this preference is defined, the <SummaryXRTDatasetName>.SUMMARY_REGISTEREDDTO preference must also be defined.

**VALID VALUES**

Accepts a single string as a value. It must be a valid dataset name, which contains rendering XML.

**DEFAULT VALUE**

None. This preference must be manually entered into the database. Create new preferences using the Options dialog box, accessed from the Edit menu in the Teamcenter rich client.

**SCOPE**

Site, group, role, or user preference.
**Fms_BootStrap_Urls**

**DESCRIPTION**
Determines which FMS server cache manages file downloads. When searching for an assigned FMS server cache to manage file downloads, the thin client contacts the FSC servers defined in this preference in the order listed. The server responds with the FSC server assigned to the thin client, and all subsequent communication is with that assigned server cache. Environment variable settings override this FCC configuration file setting.

If there is only the thin client and one FSC server in the network, you must select this option. Each Teamcenter network must have at least one server listed in the this preference for thin client use.

For failover purposes, you can include multiple servers. Ensure that you include all of the FSC addresses that support the failover volumes. Also add the FSCs to the `FMS_HOME\fcc.xml` file as priority 0 so that you can get a connection and automatically pick up the server as long as one of the FSCs is up and running in the configuration.

**Note**
This preference can be set using the preference name as an environment variable.

**VALID VALUES**
Accepts multiple strings as values. Each string must be a valid server location. For example:

```
Fms_BootStrap_Urls=
http://abc.com
http://def.com:4040
http://ghi.com
```

**DEFAULT VALUES**
None.

**DEFAULT PROTECTION SCOPE**
Site preference.

**GeolusServer**

**DESCRIPTION**
Specifies the URL for communication between Active Workspace and Geolus.

**VALID VALUES**

```
protocol\llgServer:gPort\llgContext
```

Where:

- *protocol* is **http** or **https**.
Preferences

- **Server** is the machine name or IP address of the machine running the Geolus server. It must be accessible by all Teamcenter clients that need to connect to it.
- **Port** is the port number that the server uses to handle http requests.
- **Context** is the context root of the Geolus server.

**DEFAULT VALUE**

None. This preference must be manually entered into the database. Create new preferences using the **Options** dialog box, accessed from the **Edit** menu in the Teamcenter rich client.

**EXAMPLES**

http://server012:7001/geolus

**SCOPE**

Site, group, or user preference.

**ICS_searchindex_view_visible**

**DESCRIPTION**

Indicates whether the **Search Index View** type is available in the list of Classification View types that users can create.

**VALID VALUES**

true
false

**DEFAULT VALUE**

false

**SCOPE**

Site preference.

**<InfoXRTDatasetName>.INFO_REGISTEREDETO**

**DESCRIPTION**

Links the XRT style sheet for the information panel to objects of type `<ExternalSystemObjectName>` in Active Workspace.

The parameters for this preference are the same as those for the `<SummaryXRTDatasetName>.SUMMARY_REGISTEREDETO` preference.

For more information, see `<SummaryXRTDatasetName>.SUMMARY_REGISTEREDETO`.

**<object>.CellProperties**

**DESCRIPTION**

Defines what properties to show on the tiles for objects in Active Workspace. The first two properties in the list are displayed without labels and are formatted as a primary title and subtitle. The remaining properties are displayed in the tile as `name:value`. 
Any property of the object.

The default values vary by object type.

For example, the default values for item revisions are:

```
object_name
item_id
item_revision
```

The following example specifies the tile display properties for item revisions:

```
<preference name="ItemRevision.CellProperties" type="String" array="true"
disabled="false" protectionScope="Site" envEnabled="false">
  <preference_description>Defines the list of properties that are to be displayed under a cell for objects of type ItemRevision in the Active Workspace client. Valid values are property internal names.</preference_description>
  <context name="Teamcenter">
    <value>object_name</value>
    <value>item_id</value>
    <value>item_revision_id</value>
  </context>
</preference>
```

The values in this example appear as follows in Active Workspace:

![Hard Drive Assembly](image)

Site preference.

```
<object_type>.<property_name>.FORMATSTRING=
```

 Defines the formatting of object property values. This formatting is applied when displaying properties of an object in the Active Workspace client.

Refer to Property formatter definition examples.

None.

Refer to Property formatter definition examples.
**Pdf_viewer_preference**

**DESCRIPTION**
Determines the visibility of the **Print** and **Download** buttons in the PDF viewer in the Requirements Manager application.

**VALID VALUES**
- enablePrint, enableDownload
- disablePrint, disableDownload

**DEFAULT VALUE**
enablePrint, enableDownload

**SCOPE**
Site preference.

**ProblemReportRevision_default_workflow_template**

**DESCRIPTION**
Defines the workflow that starts when a problem report revision is submitted.

**VALID VALUES**
Any valid workflow type.

**DEFAULT VALUES**
ChangelItemRevisionDefaultWorkflowTemplate

**SCOPE**
Site preference.

**RV1_DARB_Tooltip_Max_Row**

**DESCRIPTION**
Defines the maximum number of rows to show in the **Relations** node tooltip in Active Workspace.

**VALID VALUES**
Any integer value.

**DEFAULT VALUE**
4

**SCOPE**
Site preference.
**RV1_DARB_Tooltip_Max_Width**

**DESCRIPTION**
Defines the maximum character length to show in the **Relations** node tooltip in Active Workspace.

**VALID VALUES**
Any integer value.

**DEFAULT VALUE**
50

**SCOPE**
Site preference.

**SS1_DASS_enable**

**DESCRIPTION**
Enables and disables shape search.

**VALID VALUES**
- **true**
  Enables shape search.
- **false**
  Disables shape search

**DEFAULT VALUE**
**true**

**SCOPE**
Site preference.

**SS1_DASS_shape_default**

**DESCRIPTION**
Specifies the default shape similarity for shape search.

**VALID VALUES**
- 1
  Identical
- 2
  Very similar
- 3
  Similar

The range of shape options available and their exact meanings depend on the configuration of your Geolus server.
**SS1_DASS_size_default_max**

**DESCRIPTION**
Specifies the default upper range limit a user can specify when applying a size filter.

**VALID VALUES**
Any integer (percentage) in the range from `SS1_DASS_size_lower_limit` to `SS1_DASS_size_upper_limit`.

**DEFAULT VALUE**
2

**SCOPE**
User preference.

**SS1_DASS_size_default_min**

**DESCRIPTION**
Specifies the default lower range limit a user can specify when applying a size filter.

**VALID VALUES**
Any integer (percentage) in the range from `SS1_DASS_size_lower_limit` to `SS1_DASS_size_upper_limit`.

**DEFAULT VALUE**
130

**SCOPE**
User preference.

**SS1_DASS_size_lower_limit**

**DESCRIPTION**
Specifies the smallest lower range limit a user can specify when applying a size filter.

**VALID VALUES**
Any integer.

**DEFAULT VALUE**
25

**SCOPE**
User preference.
SS1_DASS_size_upper_limit

**DESCRIPTION**

Specifies the highest upper range limit a user can specify when applying a size filter.

**VALID VALUES**

Any integer.

**DEFAULT VALUE**

400

**SCOPE**

Site preference.

<SummaryXRTDatasetName>.SUMMARY_REGISTEREDDTO

**DESCRIPTION**

Indicates the business object type that an XML rendering dataset is registered to for summary style sheet use, using the form of xml-rendering-style-sheet-dataset.SUMMARY_REGISTEREDDTO=business-object-type. For example, the ItemSummary.SUMMARY_REGISTEREDDTO=Item preference means that the ItemSummary XML rendering style sheet dataset is registered as a summary type style sheet to the Item business object type.

Teamcenter property displays are based on an XML rendering style sheet. The style sheet must be created and registered by a system administrator.

For more information, see *Client Customization* in the Teamcenter collection.

The XML file allows users to define a set of properties to display, the display order, and the user interface rendering component to be used. The file can be used by both the rich client and thin client.

This preference is referenced by the style sheet: when an XML style sheet dataset is selected for display in the viewer, the business object type defined in this preference is used. For each <ExternalSystemObjectName>.SUMMARY_RENDERING preference defined, a corresponding <ExternalSystemObjectName>.SUMMARY_REGISTEREDDTO preference must be defined.

**VALID VALUES**

Accepts a single string as a value. It must be the type name defined in the corresponding <ExternalSystemObjectName>.SUMMARY_RENDERING preference.

**DEFAULT VALUE**

None. This preference must be manually entered into the database. Create new preferences using the Options dialog box, accessed from the Edit menu in the Teamcenter rich client.

**SCOPE**

Site, group, role, or user preference.
**Ticket_Expiration_Resolution**

**DESCRIPTION**
Specifies, in seconds, the resolution of the expiry time of the ticket. That is, how long a browser should cache a file.

**VALID VALUES**
Single positive integer.

**DEFAULT VALUES**
7200

**SCOPE**
Site preference.

**WEB_default_site_deployed_app_name**

**DESCRIPTION**
Appends the name of the application to be launched to the server URL defined in the WEB_default_site_server preference.

A value for this preference may be provided during installation.

**VALID VALUES**
Accepts a single string as a value. Must be the name provided for the Teamcenter application during installation.

**DEFAULT VALUES**
tc

**DEFAULT PROTECTION SCOPE**
Site preference.

**WEB_default_site_server**

**DESCRIPTION**
Sets the server URL of the default site server used for server related tasks. You must also set the Web_default_site_deployed_app_name preference, which appends the name of the application to be launched to this URL.

**VALID VALUES**
This preference must be set in the following format:

```
wwwserver_name:8080
```

The server name is the thin client server node name as defined in the browser location field.

**DEFAULT VALUES**
None.
Site preference.

**WEB_protocol**

**DESCRIPTION**

Defines the protocol used by the web server; either http:// or https://

**VALID VALUES**

- **http://**  
  Web server uses the http:// protocol.

- **https://**  
  Web server uses the https:// protocol.

  Use this setting if you are running the Teamcenter Web application and have configured it use a secure socket layer (SSL).

Site preference.

**WRKFLW_auto_subscribe_to_resource_pools**

**DESCRIPTION**

Specifies whether Active Workspace always displays in a user’s Inbox those tasks assigned to resource pools that the user is a member of based on the user’s group or role (referred to as automatic subscription). The default is automatic subscription (true). If you set it to false, only tasks assigned to resource pools to which the user has been manually subscribed are shown.

**Note**

Currently, users cannot manually subscribe to resource pools through Active Workspace.

**VALID VALUES**

- **true**  
  Shows tasks belonging to resource pools based on a user’s role and group (automatic subscription).

- **false**  
  Shows only tasks belonging to resource pools to which the user has manually subscribed.

**DEFAULT VALUES**

true
**SCOPE**

Site preference.

---

**WRKFLW_task_assignee_dynamic_participant_sync**

**DESCRIPTION**

Specifies whether the dynamic participant is set automatically on the change when a user reassigns a task.

**VALID VALUES**

- **true**
  - Reassigns the dynamic participant.

- **false**
  - Does not reassign the dynamic participant.

**DEFAULT VALUES**

- **true**

**SCOPE**

Site preference.
Chapter 7: Business Modeler IDE constants

Global constants

Following are the global constants unique to Active Workspace:

- **Awb0SupportsStructure**
  Specifies the business objects that can have a structure under it. If you want to display a custom business object in the Content tab, add the custom business object to this constant. This constant is added by the Active Content Structure template (activeworkspacebom).

- **Awp0FilterCategoryDisplayCount**
  Specifies the default number of search filter categories to display in the Search Filters panel in the client. If additional categories are available for filtering, a More button appears to display the remaining categories.

- **Awp0FilterValueDisplayCount**
  Specifies the default number of search filter values to display within a search filter category. If additional values are available for filtering in any category, a More button appears to display the remaining values within each category.

- **Awp0IndexableFileTypes**
  Specifies the list of file types that are allowed for text content extraction during search indexing. By changing the value of this constant, you can specify the file types you want to index. The following values are supported:

  - .as
  - .aw
  - .csv
  - .dat
  - .dc
  - .dif
  - .doc
  - .docm
  - .docx
  - .dot
  - .dotm
  - .dotx
  - .eml
  - .epub
  - .fff
  - .htm
  - .html
  - .ip
  - .mdb
Business object constants

Following are the business object constants unique to Active Workspace:

- **Awp0SearchIsIndexed**
  Indicates that the business object will be indexed for searching. This information is propagated through the business object hierarchy. For example, if `ItemRevision` is selected for indexing, all business objects under `ItemRevision` (such as `Part Revision` and `DocumentRevision`) are also indexed.

- **Awp0SearchIsIndexedExt**
  Indicates that external business objects are indexed for searching. By default, the value of this constant is `false`, meaning that external objects are not indexed. The scope for this constant is the `Awp0AWCEnternalSystemObject` business object, which designates objects originating in systems external to Teamcenter.

  To change the value to `true`, open the `Awp0AWCEnternalSystemObject` business object and select the `Awp0SearchIsIndexedExt` business object constant on the **Business Object Constants** tab. Then click **Edit** and select the **Value** check box.

- **Awp0SearchClassifySearchEnabled**
  Enables the searching and filtering of classification data.

- **Awp0SearchIsClassifyDataIndexed**
For the specified business object type and below, specifies that classification data be indexed for searching and filtering.

- **Awp0DatasetTypeToBeIndexedInline**
  Identifies the dataset to be indexed along with the business object. The value has the following format:
  
  `<INHERIT | NO_INHERIT>:RelationName:DatasetType`

  By default, the value `INHERIT:*:FullText` is set for the **SpecElementRevision** object type.

  **Note**
  
  The scope for this constant is for **SpecElementRevision** only. It does not apply to any other object type and is not usable for any other object type.

  This business object constant is strictly for 1:1 relations.

  To avoid behavior inheritance to subtypes, specify `NO_INHERIT`, for example, `NO_INHERIT:*:FullText`.

By default, the indexing of **FullText** datasets is not enabled because they are indexed against the **SpecElementRevision** object and its children. If you choose to enable indexing of **FullText** datasets, users see **FullText** and **SpecElementRevision** objects in search results.

**Property constants**

Following are the property constants unique to Active Workspace:

- **Awp0SearchIsIndexed**
  
  Indicates that the property on the business object will be indexed for searching by the search engine. This information is propagated through the business object hierarchy. For example, if `object_type` on **ItemRevision** is marked for indexing, all business objects under **ItemRevision** (such as **Part Revision** and **DocumentRevision**) also have their `object_type` property indexed. The following constraints apply when indexing properties:

  - Only **Attribute**, **Compound**, and **Reference** properties can be indexed. Indexing of **Runtime** and **Relation** properties is not supported.

  - To index **Compound** properties, they must reference **Attribute** properties from the source object.

  - To index **Reference** properties, the **Awp0SearchRefTypeNames** and **Awp0SearchPropFromRefType** property constants must contain valid values.

  Incorrect values are omitted from indexing; no message appears.

- **Awp0SearchCanFilter**
  
  Indicates that the search results can be filtered on the specific property. It assumes that the property was marked for indexing using the **Awp0SearchIsIndexed** property constant.

- **Awp0SearchFilterPriority**
Indicates the priority of the property that determines its order in the list of filters displayed in the client—the lower the value, the higher the priority. This means that the filter is positioned higher in the list of filters shown in the filters panel. Siemens PLM Software recommends that you assign values from a range in order to accommodate additional properties in the future. For example, assign priorities such as 100, 200, and 300, instead of 1, 2, and 3.

- **Awp0SearchRefTypeNames**

  This constant is only applicable when the property to be indexed is a **Reference** type. Specify a comma-separated list of business object names that the reference property can contain. For example, on the **owning_user** reference property on **ItemRevision**, specify a value of **User**. The following rules apply:

  o  If no value is specified on typed reference properties, the business object that is specified as the **Referenced Type** is used as the type. For example, **release_status_list** would result in **ReleaseStatus**.

  o  On untyped reference properties, if no value is specified, the **POM_object** is used as the type.

  Incorrect values are omitted from indexing; no message appears.

- **Awp0SearchPropFromRefType**

  This constant is only applicable when the property to be indexed is a **Reference** type. Specify a comma-separated list of properties that are on the business objects specified in the **Awp0SearchRefTypeNames** property constant. For example, on the **owning_user** reference property on **ItemRevision**, specify a value of **user_id,user_name**. The following rules apply:

  o  You can only specify **Attribute** and **Compound** properties.

  o  Each property in the list specified for this constant will be matched against each business object in the list specified in the **Awp0SearchRefTypeNames** property constant. Only properties that are valid and applicable on a business object are considered for indexing. In addition, if filtering is enabled on the reference property, only the first property from the list is used.

  Incorrect values are omitted from indexing; no message appears.

- **Awp0InboxCanFilter**

  Indicates that tasks shown in the inbox can be filtered on a specific property of a Workflow business object (**EPMTask** and its subtypes), as shown for tasks found when selecting the **Team** tab.
By default, the following properties are shown as filters for Workflow business objects in the inbox:

- **object_type** – The type of object.
- **due_date** – The date the object is due.
- **fnd0Assignee** – The user to whom the task is assigned.
- **fnd0Priority** – The priority of the task.
- **fnd0WorkflowInitiator** – The user who initiated the workflow on the task.

Learn about setting filtering in the inbox.

- **Awp0InboxFilterPriority**

  Indicates the priority of the property of a Workflow business object (**EPMTask** and its subtypes). It determines the property’s order in the list of filters displayed in the inbox. The lower the value, the higher its priority and, therefore, the higher its position in the list of filters.

  Siemens PLM Software recommends that you assign values from a range to accommodate additional properties in the future. For example, assign priorities such as 100, 200, and 300, instead of 1, 2, and 3.

  Learn about setting filtering in the inbox.

- **Cm1ChangeCanFilter**

  Indicates that the changes in **My Changes** can be filtered on a specific property of the change business object (**ChangeltemRevision** and its subtypes), as shown in the figure for the changes found when clicking the **Submitted** tab.
The default properties of a change object that can be filtered are:

- **creation_date** – The date the change was created.
- **CMMaturity** – The degree of completion of the overall change process (its *Maturity*).
- **object_type** – The type of change.
- **cm0Analyst** – The user assigned as the analyst.
- **cm0ChangeSpecialist1** – The user assigned as the change specialist.
- **cm0Requestor** – The user who created the change.

Learn about setting filtering in *My Changes*.

- **Cm1ChangeFilterPriority**

  Indicates the priority of the property of the change object (*ChangeltemRevision* and its subtypes). It determines the property’s order in the list of filters displayed in *My Changes*. The lower the value, the higher its priority and, therefore, the higher its position in the list of filters.

  Siemens PLM Software recommends that you assign values from a range to accommodate additional properties in the future. For example, assign priorities such as 100, 200, and 300, instead of 1, 2, and 3.

  Learn about setting filtering in *My Changes*. 

---

Image: My Changes interface with filters for Date Created, Maturity, Type, and Analyst.
Chapter 8: Workflow handlers
EPM-adhoc-signoffs

**Note**

In Teamcenter 9.1, this handler is named **adhoc-signoffs**.

**DESCRIPTION**

Determines the behavior of the **Ad-hoc done** check box that displays within the **select-signoff-team** task's interface, allowing the initializing user, address list members and resource pool members to add users to the signoff team in an ad hoc manner. If the task template contains predefined signoff profiles, the ad hoc selections add one-time-only additions to the required signoff team. Alternatively, if the task template contains no predefined signoff profiles, the ad hoc additions comprise the whole of the signoff team.

When this handler is attached to the **select-signoff-team** task, the check box is not selected by default. You can modify this behavior using the **-auto_complete** argument.

**Note**

When this handler is not attached to the **select-signoff-team** task, the check box displays by default as checked, in expectation that ad hoc additions are not required. Users can still clear the check box, add additional signoff team members to the signoff team, and then select the check box again.

Remember that the check box only indicates that the user has completed any ad hoc additions to the signoff team; it does not signify that the required profiles have been added to the signoff team. Even if the user fits into any of the signoff profiles, it is added only as an ad hoc user and not as the signoff profile member.

Using the **-auto_complete** argument with this handler allows the **select-signoff-team** task to complete automatically. If the system's **ad hoc done** query is returned as **true** and any predefined signoff profiles have been selected, the task automatically completes without user interaction. Therefore, the **select-signoff-team** task template can be configured to automatically choose a signoff team and decide whether or not to allow users to modify this predefined signoff team at execution of the task.

This handler's arguments are listed in order of precedence, meaning that the system attempts to find a match for the argument as a user before it tries to find a match as an address list. When a **select-signoff-team** task is created, based on a task template that uses this handler, it parses these arguments and add those signoffs to the task. After that point, the ad hoc signoff functionality allows subsequent modifications to the signoff list. Therefore, what is specified in this handler is only used to initialize this task; during execution of the workflow process, the ad hoc signoff functionality accepts further changes.

By default, this handler is run at workflow process initiation, rather than at the task where it is assigned. It initializes the signoff lists at workflow process initiation, allowing the workflow process initiator to view signoff assignments early in the workflow process and set the assignments as desired. However, this also means that assignments are based on target/assignment data as it exists at the time of initiation. For instance, if you use the **$TARGET_GROUP** keyword argument with this handler and the handler
is run at workflow process initiation, it looks at the group that owns the targets when
the workflow process is initiated, not when the task using this handler is run. When you
use this method, keyword arguments always resolve to the workflow process initiator.

Use the -ce argument to ensure the handler is run when the select-signoff-team task
starts, rather than at workflow process initiation.

SYNTAX

EPM-adhoc-signoffs
[-auto_complete]
[-assignee {user:user | person:person | addresslist:list
| resourcepool:group::role
| allmembers:group::role
| user:PROP::property_name
| resourcepool:PROP::property_name
| allmembers:PROP::property_name
| $PROPOSED_RESPONSIBLE_PARTY | $PROPOSED_REVIEWERS | $USER
| $PROCESS_OWNER | $TARGET_OWNER [type]
| $PROJECT_ADMINISTRATOR
| $PROJECT_TEAM_ADMINISTRATOR
| $PROJECT_AUTHOR | $PROJECT_MEMBER
| $REQUESTOR | $ANALYST
| $CHANGE_SPECIALIST1 | $CHANGE_SPECIALIST2 | $CHANGE_SPECIALIST3
| $CHANGE_REVIEW_BOARD | $CHANGE_IMPLEMENTATION_BOARD]
[-from_include_type=object-type1[,object-type2,...]]
[-from_exclude_type=object-type1[,object-type2,...]]
[-from_attach= target | reference | schedule_task]
[-from_relation=relation-type]
[-from_include_related_type=object-type1[,object-type2,...] ]
[-from_exclude_related_type=object-type1[,object-type2,...]]
[-quorum= quorum-value]
[-ce ] [-clear_signoffs]
[-target_task=multilevel-task-path]

ARGUMENTS

-auto_complete (optional)
(Optional) Allows the task to complete without user interaction. Automatically selects
the Ad-hoc done check box in the select-signoff-team task interface. The task
is assumed to be populated; no select-signoff-team task needs to be performed
through the interface (providing at least one of the signoff profiles have been fulfilled).

When this argument is not used, the system does not automatically select the Ad-hoc
done check box, preventing the select-signoff-team task from completing until the
user manually checks it, typically after ad hoc signoffs have been added. Absence
of the EPM-adhoc-signoffs handler implies the presence of this argument, and the
Ad-hoc done check box is selected and behaves accordingly.

-assignee
(Optional) Assigns signoff members to select-signoff-team or Notify task under
a Route task. It can take more than one value if you specified them using a
comma-separated list. The following value formats are allowed:
- **user**:user
  Adds the user specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter user ID.

- **user**:PROP::property_name
  Adds the user specified by the property name to the signoff member list for the task to which it is attached.

  If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **resourcepool**:PROP::property_name
  Adds the resource pool specified by the property name to the signoff member list for the task to which it is attached.

  If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **allmembers**:PROP::property_name
  Adds all members of a group/role combination that is specified by the property name to the signoff member list.

  If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **person**:person
  Adds the user whose name is specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter person name.

  **Note**

  If the person’s name includes a comma, you must include an escape character (\) to add the correct person. For example, to use **wayne, joan**:

  \-assignee=person:wayne, joan

- **addresslist**:list
  Adds all members of the address list specified to the signoff member list.

- **resourcepool**:group::role
  Results in a single assignment which can be performed by any single member of this group/role.

  You can define resource pools in the form of group::, group::role, or role. Accepts valid Teamcenter resource pool names and these keywords:

  o **$GROUP**
Current user's current group.

- **$ROLE**
  Current user's current role.

- **$TARGET_GROUP[type]**
  Owning group of the first target object of the specified type. The type value is optional. If not specified, the first target is used.

- **$PROCESS_GROUP**
  Owning group of the workflow process.

  - **allmembers:group::role**
    Adds all members of a group/role combination to the signoff member list. You can define role in groups in the form of group::, group::role, or role. Accepts valid Teamcenter resource pool names and these keywords:

  - **$GROUP**
    Current user's current group.

  - **$ROLE**
    Current user's current role.

  - **$TARGET_GROUP[type]**
    Owning group of the first target object of the specified type. The type value is optional. If not specified, the first target is used.

  - **$PROCESS_GROUP**
    Owning group of the workflow process.

  - **$PROPOSED_RESPONSIBLE_PARTY**
    Affects assignments based on the user assigned as the responsible party for the first target object.

  - **$PROPOSED_REVIEWERS**
    Affects assignments based on members assigned as reviewers for the first target object.

  - **$USER**
    Adds the current user to the signoff member list.

  - **$PROCESS_OWNER**
    Adds the workflow process owner to the signoff member list.

  - **$TARGET_OWNER[type]**
Chapter 8: Workflow handlers

Adds the owner of the first target of specified type to the signoff member list. The type value is optional. If not specified, the first target is used.

- **$PROJECT_ADMINISTRATOR, $PROJECT_TEAM_ADMINISTRATOR, $PROJECT_AUTHOR, $PROJECT_MEMBER**
  Dynamically adds the project team members belonging to the role specified in the argument value. The project team is determined by the project team associated with the first target object.

- **$REQUESTOR, $ANALYST, $CHANGE_SPECIALIST1, $CHANGE_SPECIALIST2, $CHANGE_SPECIALIST3, $CHANGE_REVIEW_BOARD, $CHANGE_IMPLEMENTATION_BOARD**
  Dynamically resolves to the user or resource pool associated with the first Change target object in the workflow process. The particular user or resource pool is determined by the role specified in the argument value.

  **Note**
  Change-related keywords apply only to change objects. If the workflow process does not contain a change object as a target, the argument resolves to null.

  Change Manager does not need to be enabled before these keywords take effect, but during installation, Change Management must be selected under Extensions → Enterprise Knowledge Foundation in Teamcenter Environment Manager.

- **-from_include_type=object-type1[;object-type2,...]**
  (Optional) Specifies the object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

- **-from_exclude_type=object-type1[;object-type2,...]**
  (Optional) Specifies the object types to be excluded when getting the property value when it is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

- **-from_attach= target | reference | schedule_task**
  (Optional) Specifies which type of attachment (target, reference, or schedule_task) to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). If this argument is not specified, the default is target.

- **-from_relation**
  (Optional) Specifies the relation of the objects to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). It must be a valid relation.

- For manifestations, use IMAN_manifestation.
- For specifications, use IMAN_specification.
For requirements, use **IMAN_requirement**.

For references, use **IMAN_reference**.

For BOM views, use **PSBOMViewRevision**.

- **-from_include_related_type=object-type1[,object-type2]**
  (Optional) Specifies the related object types to be used to get the property value from when a property is specified in the **-assignee** argument (for example, **-assignee=user:PROP::property_name**). They must be valid object types.

Use this argument when a property is designated and you use the **-from_relation** argument.

This argument should not be used with the **-from_exclude_related_type** argument.

- **-from_exclude_related_type=object-type1[,object-type2]**
  (Optional) Specifies related object types to be excluded when getting the property value when it is specified in the **-assignee** argument (for example, **-assignee=user:PROP::property_name**). They must be valid object types.

Use this argument when a property is designated and you use the **-from_relation** argument.

This argument should not be used with the **-from_include_related_type** argument.

- **-quorum**
  (Optional) Determines the approval quorum for the **perform-signoffs** task. The value can either be a percentage or a number. For example, if it is set to 51% then all the signoff members, 51% of members need to approve for the task to move ahead. If it is set to 5, then 5 members need to approve for the task to move ahead. The value specified here overrides the quorum specified on the **select-signoff-team** task template. If no value is specified, the quorum specified on the **select-signoff-team** task template is used. This argument is ignored if the handler is placed on a **Notify** task.

- **-ce**
  (Optional) Disables the handler from being ran when the workflow process is initiated. Instead, the handler is ran in the conventional manner at the point of handler placement on the task action.

If **-ce** is specified, the **select-signoff-team** task does not auto-complete even if a process assignment list is assigned during process initiation. For the **select-signoff-team** task to autocomplete, you must also use the **-auto_complete** handler argument.

- **-clear_signoffs**
  (Optional) If specified, all existing signoffs are removed from the **select-signoff-team** subtask before the new signoffs are added.

- **-target_task**
  (Optional) Specifies the multilevel task path to which the reviewers are added. The path is from the root task to the **select-signoff-team** subtask with the path levels separated with colons (:). For example:
Change Request Review:QA Review:select-signoff-team

PLACEMENT

Place on the Start action of a select-signoff-team subtask.

This handler is run at workflow process initiation if the -ce argument is not specified. If -ce is specified, the handler is run in a conventional manner at the point of handler placement on the task action.

Place on the Undo action of a select-signoff-team subtask and specify the -ce argument to clear the Ad-hoc done checkbox when the subtask is demoted. In this situation, the next time the subtask reaches the Start action of the select-signoff-team subtask, the user is again prompted to select a signoff team.

RESTRICTIONS

Ignores any invalid arguments without reporting an error.

The keywords always refer to the initiating user because all instances of this handler in a workflow process are ran when the workflow process is initiated, not when tasks are approved.

If the -ce argument is not specified, all instances of this handler are run when the workflow process is initiated and in this case the keywords refer to the initiating user.

EXAMPLES

• This example places the handler on the Undo action of the select-signoff-team subtask. If the select-signoff-team subtask is demoted, the -ce argument clears the Ad-hoc done checkbox. When the workflow process returns to the select-signoff-team subtask, the responsible party is again prompted to select the signoff team because the Ad-hoc done checkbox is clear, indicating the task is not yet complete.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-ce</td>
<td></td>
</tr>
</tbody>
</table>

• This example has a valid user, resource pool, address list and handler-specific keywords as argument values. So Smith, the current logged-on users group/role resource pool, members of the List1 address list, and the members assigned as reviewers are added as signoff attachments to the select-signoff-team task on which this handler is added. The handler is run at the time of workflow process initiation.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:Smith, resourcepool:$GROUP::$ROLE, addresslist:List1, $PROPOSED_REVIEWERS</td>
</tr>
<tr>
<td>-quorum</td>
<td>80%</td>
</tr>
</tbody>
</table>

If the handler with these arguments is specified on the Notify task under the Route task, the signoff attachments are added to the Notify task and used for sending notifications. The quorum is set to 80%, which means that of all the signoff members, 80% need to approve for the task to move ahead.
This example has a valid user, resource pool, address list, and handler-specific keywords as argument values. So Smith, the current logged-on user group/role resource pool, members of List1 address list, and the members assigned as reviewers are added as signoff attachments to the select-signoff-team task on which this handler is added. Because of the -ce option, the handler is run when the task action on which it is attached is run. The quorum is set to 80%, which means that of all the signoff members, 80% need to approve for the task to move ahead.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:Smith, resourcepool:$GROUP::$ROLE, addresslist:List1, $PROPOSED_REVIEWERS</td>
</tr>
<tr>
<td>-quorum</td>
<td>80%</td>
</tr>
<tr>
<td>-ce</td>
<td></td>
</tr>
</tbody>
</table>

If the handler with the above arguments is specified on the Notify task under the Route task, the signoff attachments are added to the Notify task and used for sending notifications.

This example assigns the user whose ID is Smith to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:Smith</td>
</tr>
</tbody>
</table>

This example assigns the owning user ID of the first UGMINTER target found by the system to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:$TARGET_OWNER[UGMASTER]</td>
</tr>
</tbody>
</table>

This example assigns the project team administrator of the project team associated with the first target found by the system to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:$PROJECTTEAM_ADMINISTRATOR</td>
</tr>
</tbody>
</table>

This example assigns all members of the jhList address list to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>addresslist:jhList</td>
</tr>
</tbody>
</table>

This example assigns the manufacturing resource pool (any role within the manufacturing group) to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>resourcepool:manufacturing::</td>
</tr>
</tbody>
</table>
• This example assigns the $PROCESS_GROUP resource pool (any role within the xyz group, where xyz is the owning group of the workflow process) to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>resourcepool:$PROCESS_GROUP::</td>
</tr>
</tbody>
</table>

• This example assigns the $TARGET_GROUP resource pool (any roles within the abc group, where abc is the group of the first item revision target) to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>resourcepool:$TARGET_GROUP::</td>
</tr>
</tbody>
</table>

• This example assigns the engineer role within the manufacturing group resource pool to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>resourcepool:manufacturing::engineer</td>
</tr>
</tbody>
</table>

• This example assigns the current logged-on role within the current logged on group resource pool to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>resourcepool:$GROUP::$ROLE</td>
</tr>
</tbody>
</table>

• This example assigns the engineer role within any group resource pool to the signoff team.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>resourcepool:::engineer</td>
</tr>
</tbody>
</table>

• This example adds user smith and all reviewers of the first target item revision object to the signoff team. The quorum is set to 51%, which means that at least more than half of the signoff members need to approve for the perform-signoffs task to move ahead. Because of the -ce option, the handler is run when the task action on which it is attached is run.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:smith, $PROPOSED_REVIEWERS</td>
</tr>
<tr>
<td>-quorum</td>
<td>51%</td>
</tr>
<tr>
<td>-ce</td>
<td></td>
</tr>
</tbody>
</table>

• This example adds all members of the Engineering group and Engineer role to the signoff team. The members are dynamically evaluated when the select-signoff-team task completes. The quorum is set to 80%, which means that of all the signoff members, 80% need to approve for the task to move ahead.
Because of the -ce option, the handler is run when the task action on which it is attached is run.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>allmembers:Engineering::Engineer</td>
</tr>
<tr>
<td>-quorum</td>
<td>80%</td>
</tr>
<tr>
<td>-ce</td>
<td></td>
</tr>
</tbody>
</table>

This example adds all members of the list1 address list and the Engineering::Engineer resource pool to the signoff team. The quorum is set to 5, which means that of all the signoff members, 5 need to approve for the task to move ahead. Because of the -ce option, the handler is run when the task action on which it is attached is run.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>resourcepool:Engineering::Engineer, addresslist:list1</td>
</tr>
<tr>
<td>-quorum</td>
<td>5</td>
</tr>
<tr>
<td>-ce</td>
<td></td>
</tr>
</tbody>
</table>

This example has a valid user, resource pool, address list, and handler-specific keywords as argument values. So smith, the current logged-on user group/role resource pool, members of the list1 address list, and the members assigned as reviewers are assigned to the signoff team. Because of the -ce option, the handler is run when the task action on which it is attached is run.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:smith,resourcepool:$GROUP::$ROLE, addressList:list1,$PROPOSED_REVIEWERS</td>
</tr>
<tr>
<td>-ce</td>
<td></td>
</tr>
</tbody>
</table>

If the handler with these arguments is specified on the Notify task under the Route task, the signoff attachments are added to the Notify task and used for sending notifications.

This example has a valid user, resource pool, and handler-specific keywords as values. So smith, the current logged on users group/role resource pool, members of the project associated with the first target object, and members assigned as reviewers are added to the signoff team. Because of the -ce option, the handler is run when the task action on which it is attached is run.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:smith,resourcepool:$GROUP::$ROLE, $PROJECT_MEMBER,$PROPOSED_REVIEWERS</td>
</tr>
<tr>
<td>-ce</td>
<td></td>
</tr>
</tbody>
</table>
If the handler with these arguments is specified on the Notify task under the Route task, the signoff attachments are added to the Notify task and used for sending notifications.

• This example has a valid user, resource pool, and handler-specific keywords as values. So smith, the current logged-on user group/role resource pool, and CHANGE_REVIEW_BOARD and ANALYST associated with the first change target object are added to the signoff team. Because of the -ce option, the handler is run when the task action on which it is attached is run.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:smith,resourcepool:$GROUP::$ROLE,</td>
</tr>
<tr>
<td></td>
<td>$CHANGE_REVIEW_BOARD,$ANALYST</td>
</tr>
<tr>
<td>-ce</td>
<td></td>
</tr>
</tbody>
</table>

If the handler with these arguments is specified on the Notify task under the Route task, the signoff attachments are added to the Notify task and used for sending notifications.

• This example removes all existing members of the signoff team and adds PROPOSED_RESPONSIBLE_PARTY. Because of the -ce option, the handler is run when the task action on which it is attached is run. The -auto_complete option allows the task to complete without user interaction by automatically selecting the Ad-hoc done check box in the select-signoff-team subtask interface, and the task does not need to be performed through the interface.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-ce</td>
<td></td>
</tr>
<tr>
<td>-clear_signoffs</td>
<td></td>
</tr>
<tr>
<td>-assignee</td>
<td>$PROPOSED_RESPONSIBLE_PARTY</td>
</tr>
<tr>
<td>-auto_complete</td>
<td></td>
</tr>
</tbody>
</table>

If the handler with these arguments is specified on the Notify task under the Route task, the signoff attachments are added to the Notify task and used for sending notifications.
**EPM-assign-responsible-party-dynamic-participant**

**DESCRIPTION**

Assigns the specified user or resource pool as the dynamic participant to be the responsible party for the target attachment.

**SYNTAX**

```
EPM-assign-responsible-party-dynamic-participant
-name= {PROPOSED_RESPONSIBLE_PARTY
 | ANALYST
 | CHANGE_SPECIALIST1
 | CHANGE_SPECIALIST2
 | CHANGE_SPECIALIST3}
-assignee= [user:user | person:person
 | resourcepool:group::role
 | user:PROP::property_name
 | resourcepool:PROP::property_name
 | $PROPOSED_RESPONSIBLE_PARTY | $USER
 | $PROCESS_OWNER | $TARGET_OWNER [type]
 | $PROJECT_ADMINISTRATOR
 | $PROJECT_TEAM_ADMINISTRATOR
 | $REQUESTOR | $ANALYST
 | $CHANGE_SPECIALIST1
 | $CHANGE_SPECIALIST2
 | $CHANGE_SPECIALIST3]
[-from_include_type=object-type1[,object-type2,...]]
[-from_exclude_type=object-type1[,object-type2,...]]
[-to_include_type=object-type1[,object-type2,...]]
[-to_exclude_type=object-type1[,object-type2,...]]
[-from_attach= target | reference | schedule_task]
[-from_relation=relation-type]
[-from_include_related_type=object-type1[,object-type2,...] |]
[-from_exclude_related_type=object-type1[,object-type2,...]]
[-first_object_only]
[-bypass_condition_check]
```

**ARGUMENTS**

- **-name**
  Specifies the keyword of the dynamic participant that you want to assign participants. Accepts one of the following:

  - **PROPOSED_RESPONSIBLE_PARTY**
  - **ANALYST**
  - **CHANGE_SPECIALIST1**
  - **CHANGE_SPECIALIST2**
  - **CHANGE_SPECIALIST3**
Note

Change-related keywords apply only to change objects. If the workflow process does not contain a change object as a target, the argument resolves to null.

Change Manager does not need to be enabled before these keywords take effect, but during installation, Change Management must be selected under Extensions—Enterprise Knowledge Foundation in Teamcenter Environment Manager.

-assignee
(Optional) Makes the user or resource pool the specified keyword evaluates to the responsible party for the task to which this handler is added. Accepts one of the following in the format specified below:

- **user:** user
  Adds the user specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter user ID.

- **person:** person
  Adds the user whose name is specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter person name.

  **Note**

  If the person's name includes a comma, you must include an escape character (\) to add the correct person. For example, to use wayne, joan:

  -assignee=person:wayne, joan

- **resourcepool:** group::role
  Results in a single assignment which can be performed by any single member of this group/role.

  You can define resource pools in the form of group::, group::role, or role. Accepts valid Teamcenter resource pool names and these keywords:

  - **$GROUP**
    Current user's current group.

  - **$ROLE**
    Current user's current role.

  - **$TARGET_GROUP [type]**
    Owning group of the first target object of the specified type. The type value is optional. If not specified, the first target is used.

  - **$PROCESS_GROUP**
    Owning group of the workflow process.
• **user:PROP::property_name**
  Adds the user specified by the property name to the signoff member list for the task to which it is attached.

  If the property is a multi-value property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

• **resourcepool:PROP::property_name**
  Adds the resource pool specified by the property name to the signoff member list for the task to which it is attached.

  If the property is a multi-value property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

• **$PROPOSED_RESPONSIBLE_PARTY**
  Affects assignments based on the user assigned as the responsible party for the first target object.

• **$USER**
  Adds the current user to the signoff member list.

• **$PROCESS_OWNER**
  Adds the workflow process owner to the signoff member list.

• **$TARGET_OWNER [type]**
  Adds the owner of the first target of specified type to the signoff member list. The type value is optional. If not specified, the first target is used.

• **$PROJECT_ADMINISTRATOR, $PROJECT_TEAM_ADMINISTRATOR**
  Dynamically adds the project team members belonging to the role specified in the argument value. The project team is determined by the project team associated with the first target object.

• **$REQUESTOR, $ANALYST, $CHANGE_SPECIALIST1, $CHANGE_SPECIALIST2, $CHANGE_SPECIALIST3**
  Dynamically resolves to the user or resource pool associated with the first change target object in the workflow process. The particular user or resource pool is determined by the role specified in the argument value.
Note

Change-related keywords apply only to change objects. If the workflow process does not contain a change object as a target, the argument resolves to null.

Change Manager does not need to be enabled before these keywords take effect, but during installation, Change Management must be selected under Extensions→Enterprise Knowledge Foundation in Teamcenter Environment Manager.

-from_include_type=object-type1[,object-type2,...]
(Optional) Specifies the object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

-from_exclude_type=object-type1[,object-type2,...]
(Optional) Specifies the object types to be excluded when getting the property value when it is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

-to_include_type=object-type1[,object-type2,...]
(Optional) Specifies the object types to be used while assigning participants on the target attachment. They must be valid object types.

The -to_include_type and -to_exclude_type arguments are mutually exclusive. If you use one, you cannot use the other.

-to_exclude_type=object-type1[,object-type2,...]
(Optional) Specifies the object types to be excluded while assigning participants on the target attachment. They must be valid object types.

The -to_include_type and -to_exclude_type arguments are mutually exclusive. If you use one, you cannot use the other.

-from_attach= target | reference | schedule_task
(Optional) Specifies which type of attachment (target, reference, or schedule_task) to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). If this argument is not specified, the default is target.

-from_relation
(Optional) Specifies the relation of the objects to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). It must be a valid relation.

- For manifestations, use IMAN_manifestation.
- For specifications, use IMAN_specification.
- For requirements, use IMAN_requirement.
- For references, use IMAN_reference.
• For BOM views, use **PSBOMViewRevision**.

- **from_include_related_type=object-type1[,object-type2]**
  (Optional) Specifies the related object types to be used to get the property value from when a property is specified in the **assignee** argument (for example, **assignee=user:PROP::property_name**). They must be valid object types.

  Use this argument when a property is designated and you use the **from_relation** argument.

- **from_exclude_related_type=object-type1[,object-type2]**
  (Optional) Specifies related object types to be excluded when getting the property value when it is specified in the **assignee** argument (for example, **assignee=user:PROP::property_name**). They must be valid object types.

  Use this argument when a property is designated and you use the **from_relation** argument.

- **first_object_only**
  (Optional) Sets the participants on the first target attachment matching the **to_include_type** and **to_exclude_type** arguments. If this argument is not specified, the participants are set on all target attachments matching the **to_include_type** and **to_exclude_type** arguments.

- **bypass_condition_check**
  (Optional) Bypasses the Business Modeler IDE condition check before assigning participants. If this argument is not specified, the Business Modeler IDE conditions are checked before assigning participants.

**PLACEMENT**

Place on the **Start** action.

**RESTRICTIONS**

Can only be used to assign dynamic participants that resolve to a single user. For example:

**PROPOSED_RESPONSIBLE_PARTY** or **ANALYST**
## EPM-assign-signoff-dynamic-participant

### DESCRIPTION
Assigns the specified user or resource pool as the dynamic participant to the signoff team for the target attachment.

### SYNTAX
```
EPM-assign-signoff-dynamic-participant
-name= {PROPOSED_REVIEWERS
  | CHANGE_REVIEW_BOARD
  | CHANGE_IMPLEMENTATION_BOARD}
-assignee= [user:user | person:person | resourcepool:group::role
  | user:PROP::property_name
  | resourcepool:PROP::property_name
  | $PROPOSED_RESPONSIBLE_PARTY | $USER
  | $PROCESS_OWNER | $TARGET_OWNER [type]
  | $PROJECT_ADMINISTRATOR
  | $PROJECT_TEAM_ADMINISTRATOR
  | $REQUESTOR | $ANALYST
  | $CHANGE_SPECIALIST1
  | $CHANGE_SPECIALIST2
  | $CHANGE_SPECIALIST3]
[-from_include_type=object-type1[,object-type2,...]]
[-from_exclude_type=object-type1[,object-type2,...]]
[-to_include_type=object-type1[,object-type2,...]]
[-to_exclude_type=object-type1[,object-type2,...]]
[-from_attach= target | reference | schedule_task]
[-from_relation=relation-type]
[-from_include_related_type=object-type1[,object-type2,...] |]
[-from_exclude_related_type=object-type1[,object-type2,...]]
[-clear] [-first_object_only]
[-bypass_condition_check]
```

### ARGUMENTS
- **-name**
  Specifies the keyword of the dynamic participant that you want to assign participants to. Accepts one of the following:
  - **PROPOSED_REVIEWERS**
  - **CHANGE_REVIEW_BOARD**
  - **CHANGE_IMPLEMENTATION_BOARD**

  **Note**
  Change-related keywords apply only to change objects. If the workflow process does not contain a change object as a target, the argument resolves to null.

  Change Manager does not need to be enabled before these keywords take effect, but during installation, **Change Management** must be selected under **Extensions → Enterprise Knowledge Foundation** in Teamcenter Environment Manager.
-assignee
(Optional) Makes the user or resource pool the specified keyword evaluates to the responsible party for the task to which this handler is added. Accepts one of the following in the format specified below:

- **user**:user
  Adds the user specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter user ID.

- **person**:person
  Adds the user whose name is specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter person name.
  
  **Note**
  If the person’s name includes a comma, you must include an escape character (\) to add the correct person. For example, to use wayne, joan:
  
  -assignee=person:wayne, joan

- **resourcepool**:group::role
  Results in a single assignment which can be performed by any single member of this group/role.
  You can define resource pools in the form of group::, group::role, or role. Accepts valid Teamcenter resource pool names and these keywords:
  
  o **$GROUP**
    Current user’s current group.
  
  o **$ROLE**
    Current user’s current role.
  
  o **$TARGET_GROUP [type]**
    Owning group of the first target object of the specified type. The type value is optional. If not specified, the first target is used.
  
  o **$PROCESS_GROUP**
    Owning group of the workflow process.

- **user**:PROP::property_name
  Adds the user specified by the property name to the signoff member list for the task to which it is attached.
  If the property is a multi-value property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **resourcepool**:PROP::property_name
Adds the resource pool specified by the property name to the signoff member list for the task to which it is attached.

If the property is a multi-value property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **$PROPOSED_RESPONSIBLE_PARTY**
  Affects assignments based on the user assigned as the responsible party for the first target object.

- **$USER**
  Adds the current user to the signoff member list.

- **$PROCESS_OWNER**
  Adds the workflow process owner to the signoff member list.

- **$TARGET_OWNER [type]**
  Adds the owner of the first target of specified type to the signoff member list. The type value is optional. If not specified, the first target is used.

- **$PROJECT_ADMINISTRATOR, $PROJECT_TEAM_ADMINISTRATOR**
  Dynamically adds the project team members belonging to the role specified in the argument value. The project team is determined by the project team associated with the first target object.

- **$REQUESTOR, $ANALYST, $CHANGE_SPECIALIST1, $CHANGE_SPECIALIST2, $CHANGE_SPECIALIST3**
  Dynamically resolves to the user or resource pool associated with the first change target object in the workflow process. The particular user or resource pool is determined by the role specified in the argument value.

  **Note**
  Change-related keywords apply only to change objects. If the workflow process does not contain a change object as a target, the argument resolves to null.

  Change Manager does not need to be enabled before these keywords take effect, but during installation, Change Management must be selected under Extensions→Enterprise Knowledge Foundation in Teamcenter Environment Manager.

- **from_include_type=object-type1[,object-type2,...]**
  (Optional) Specifies the object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.
-**from_exclude_type**=object-type1[,object-type2,...]
(Optional) Specifies the object types to be excluded when getting the property value when it is specified in the **assignee** argument (for example, **assignee=user:PROP::property_name**). They must be valid object types.

-**to_include_type**=object-type1[,object-type2,...]
(Optional) Specifies the object types to be used while assigning participants on the target attachment. They must be valid object types.

The **to_include_type** and **to_exclude_type** arguments are mutually exclusive. If you use one, you cannot use the other.

-**to_exclude_type**=object-type1[,object-type2,...]
(Optional) Specifies the object types to be excluded while assigning participants on the target attachment. They must be valid object types.

The **to_include_type** and **to_exclude_type** arguments are mutually exclusive. If you use one, you cannot use the other.

-**from_attach**=target | reference | schedule_task
(Optional) Specifies which type of attachment (target, reference, or schedule_task) to get the property value from when a property is specified in the **assignee** argument (for example, **assignee=user:PROP::property_name**). If this argument is not specified, the default is target.

-**from_relation**
(Optional) Specifies the relation of the objects to get the property value from when a property is specified in the **assignee** argument (for example, **assignee=user:PROP::property_name**). It must be a valid relation.

- For manifestations, use **IMAN_manifestation**.
- For specifications, use **IMAN_specification**.
- For requirements, use **IMAN_requirement**.
- For references, use **IMAN_reference**.
- For BOM views, use **PSBOMViewRevision**.

-**from_include_related_type**=object-type1[,object-type2]
(Optional) Specifies the related object types to be used to get the property value from when a property is specified in the **assignee** argument (for example, **assignee=user:PROP::property_name**). They must be valid object types.

Use this argument when a property is designated and you use the **from_relation** argument.

-**from_exclude_related_type**=object-type1[,object-type2]
(Optional) Specifies related object types to be excluded when getting the property value when it is specified in the **assignee** argument (for example, **assignee=user:PROP::property_name**). They must be valid object types.
Use this argument when a property is designated and you use the `-from_relation` argument.

- **-clear**
  (Optional) Removes all previously assigned participants before assigning new participants. If this argument is not specified, new participants are appended to existing participants list.

- **-first_object_only**
  (Optional) Sets the participants on the first target attachment matching the `-to_include_type` and `-to_exclude_type` arguments. If this argument is not specified, the participants are set on all target attachments matching the `-to_include_type` and `-to_exclude_type` arguments.

- **-bypass_condition_check**
  (Optional) Bypasses the Business Modeler IDE condition check before assigning participants. If this argument is not specified, the Business Modeler IDE conditions are enforced before assigning participants.

**Placement**

Place on the **Start** action.

**Restrictions**

Can only be used to assign dynamic participants that resolve to a multiple users. For example:

```
PROPOSED_REVIEWERS or CHANGE_REVIEW_BOARD
```
EPM-assign-team-selector

**Note**

In Teamcenter 9.1, this handler is named **CR-assign-team-selector**.

**DESCRIPTION**

Assigns all select-signoff-team tasks in the entire workflow process to the specified user, person, initiator (owner), or resource pool of the workflow process. Only one argument can be defined; all arguments are mutually exclusive of each other.

**SYNTAX**

```
EPM-assign-team-selector
-assignee= [user:user | person:person | resourcepool:group::role
| user:PROP::property_name
| resourcepool:PROP::property_name
| SPROPOSED_RESPONSIBLE_PARTY | $USER
| $PROCESS_OWNER | $TARGET_OWNER [type]
| $PROJECT_ADMINISTRATOR
| $PROJECT_TEAM_ADMINISTRATOR
| $REQUESTOR | $ANALYST
| $CHANGE_SPECIALIST1
| $CHANGE_SPECIALIST2
| $CHANGE_SPECIALIST3]
[-from_include_type=object-type1[,object-type2,...]]
[-from_exclude_type=object-type1[,object-type2,...]]
[-from_attach= target | reference | schedule_task]
[-from_relation=relation-type]
[-from_include_related_type=object-type1[,object-type2,...] |
-]from_exclude_related_type=object-type1[,object-type2,...]]
[-target_task=multilevel-task-path]
```

**ARGUMENTS**

**-assignee**

Assigns the responsible party for the task to which this handler is added. Accepts one of the following in the format specified:

- **user:user**
  
  Adds the user specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter user ID.

- **person:person**
  
  Adds the user whose name is specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter person name.

**Note**

If the person’s name includes a comma, you must include an escape character (\) to add the correct person. For example, to use **wayne, joan**:

```
-assignee=person:wayne, joan
```
Chapter 8: Workflow handlers

- **resourcepool:group::role**
  Results in a single assignment which can be performed by any single member of this group/role.
  You can define resource pools in the form of group::, group::role, or role. Accepts valid Teamcenter resource pool names and these keywords:
  
  o **$GROUP**
    Current user’s current group.
  
  o **$ROLE**
    Current user’s current role.
  
  o **$TARGET_GROUP [type]**
    Owning group of the first target object of the specified type. The type value is optional. If not specified, the first target is used.
  
  o **$PROCESS_GROUP**
    Owning group of the workflow process.

- **user:PROP::property_name**
  Adds the user specified by the property name to the signoff member list for the task to which it is attached.
  If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **resourcepool:PROP::property_name**
  Adds the resource pool specified by the property name to the signoff member list for the task to which it is attached.
  If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **$PROPOSED_RESPONSIBLE_PARTY**
  Affects assignments based on the user assigned as the responsible party for the first target object.

- **$USER**
  Adds the current user to the signoff member list.

- **$PROCESS_OWNER**
  Adds the workflow process owner to the signoff member list.

- **$TARGET_OWNER [type]**
Adds the owner of the first target of specified type to the signoff member list. The type value is optional. If not specified, the first target is used.

- **$PROJECT_ADMINISTRATOR, $PROJECT_TEAM_ADMINISTRATOR**
  Dynamically adds the project team members belonging to the role specified in the argument value. The project team is determined by the project team associated with the first target object.

- **$REQUESTOR, $ANALYST, $CHANGE_SPECIALIST1, $CHANGE_SPECIALIST2, $CHANGE_SPECIALIST3**
  Dynamically resolves to the user or resource pool associated with the first change target object in the workflow process. The particular user or resource pool is determined by the role specified in the argument value.

  **Note**
  Change-related keywords apply only to change objects. If the workflow process does not contain a change object as a target, the argument resolves to null.

  Change Manager does not need to be enabled before these keywords take effect, but during installation, Change Management must be selected under Extensions→Enterprise Knowledge Foundation in Teamcenter Environment Manager.

- **from_include_type=object-type1[,object-type2,...]**
  (Optional) Specifies the object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

- **from_exclude_type=object-type1[,object-type2,...]**
  (Optional) Specifies the object types to be excluded when getting the property value when it is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

- **from_attach= target | reference | schedule_task**
  (Optional) Specifies which type of attachment (target, reference, or schedule_task) to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). If this argument is not specified, the default is target.

- **from_relation**
  (Optional) Specifies the relation of the objects to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). It must be a valid relation.

  - For manifestations, use **IMAN_manifestation**.

  - For specifications, use **IMAN_specification**.

  - For requirements, use **IMAN_requirement**.
• For references, use IMAN_reference.

• For BOM views, use PSBOMViewRevision.

This argument must be used with the -from_attach argument. A derived object is identified by starting with objects of the specified attachment type indicated by the -from_attach argument and then locating the first secondary object with the specified relation indicated by the -relation argument.

_from_include_related_type=object-type1[,object-type2]
(Optional) Specifies the related object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

Use this argument when a property is designated and you use the -from_relation argument.

This argument should not be used with the -from_exclude_related_type argument.

_from_exclude_related_type=object-type1[,object-type2]
(Optional) Specifies related object types to be excluded when getting the property value when it is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

Use this argument when a property is designated and you use the -from_relation argument.

This argument should not be used with the -from_include_related_type argument.

-target_task
(Optional) Specifies the multilevel task path to which the reviewers are added. The path is from the root task to the select-signoff-team subtask with the path levels separated with colons (:). For example:

Change Request Review::QA Review::select-signoff-team

PLACEMENT

Place on the Start action of the root task.

RESTRICTIONS

None.

EXAMPLES

• This example assigns the user jim all select-signoff-team tasks in that workflow process.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:jim</td>
</tr>
</tbody>
</table>

• This example assigns the person Jim Smith all select-signoff-team tasks in that workflow process.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>person:Jim Smith</td>
</tr>
</tbody>
</table>
- This example assigns the owner of the workflow process all **select-signoff-team** tasks in that workflow process.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$PROCESS_OWNER</td>
</tr>
</tbody>
</table>

- This example assigns the user or resource pool assigned as the responsible party for all **select-signoff-team** tasks in that workflow process.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$PROPOSED_RESPONSIBLE_Party</td>
</tr>
</tbody>
</table>

- This example makes the project administrator of the project associated with the first target the responsible party for all **select-signoff-team** tasks in that workflow process.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$PROJECT_ADMINISTRATOR</td>
</tr>
</tbody>
</table>

- This example makes the user or resource pool associated as **REQUESTOR** with the first change target the responsible party for all **select-signoff-team** tasks in the workflow process.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$REQUESTOR</td>
</tr>
</tbody>
</table>
EPM-auto-assign

Note

In Teamcenter 9.1, this handler is named auto-assign.

DESCRIPTION

Makes the specified user or resource pool the responsible party for the task to which the handler is added. Optionally, you can make the same specified user/resource pool the responsible party for all subtasks of the parent task.

Note

If you use keyword arguments to dynamically generate this assignment, and the system resolve the argument to a user or resource pool, then the argument is ignored.

SYNTAX

EPM-auto-assign [-subtasks]
[-assignee= {user:user | person:person | resourcepool:group::role
| user:PROP::property_name
| resourcepool:PROP::property_name
| $PROPOSED_RESPONSIBLE_PARTY | $USER
| $PROCESS_OWNER | $TARGET_OWNER [type]
| $PROJECT_ADMINISTRATOR
| $PROJECT_TEAM_ADMINISTRATOR
| $REQUESTOR | $ANALYST
| $CHANGE_SPECIALIST1
| $CHANGE_SPECIALIST2
| $CHANGE_SPECIALIST3}]
[-from_include_type=object-type1[,object-type2,...]]
[-from_exclude_type=object-type1[,object-type2,...]]
[-from_attach= target | reference | schedule_task]
[-from_relation=relation-type]
[-from_include_related_type=object-type1[,object-type2,...] | ]
[-from_exclude_related_type=object-type1[,object-type2,...]]
[-target_task=multilevel-task-path]

ARGUMENTS

-subtasks
(Optional) Propagates task assignments to subtasks of the current task (nonrecursively).

-assignee

Assigns as the responsible party for the task to which this handler is added either the specified person, user, resource pool, or the user or resource pool the specified keyword evaluates to.

Accepts one of the following in the format specified:

• user:user
Add the specified user to the signoff member list and as the responsible party for the task to which the handler is attached. Accepts a valid Teamcenter user ID.

- **person:**person
  Adds the person whose name is specified to the signoff member list and as the responsible party for the task to which the handler is attached. Accepts a valid Teamcenter person name.
  
  **Note**
  If the person’s name includes a comma, you must include an escape character (\) to add the correct person. For example, to use wayne, joan:

  ```
  -assignee=person:wayne\, joan
  ```

- **resourcepool:**group::role
  Results in a single assignment which can be performed by any single member of this group/role.
  You can define resource pools in the form of group::, group::role, or role.
  Accepts valid Teamcenter resource pool names and these keywords:
  
  - **$GROUP**
    Current user’s current group.
  
  - **$ROLE**
    Current user’s current role.
  
  - **$TARGET_GROUP[type]**
    Owning group of the first target object of the specified type. The type value is optional. If not specified, the first target is used.
  
  - **$PROCESS_GROUP**
    Owning group of the workflow process.
  
  **Note**
  The $ROLE_IN_GROUP keyword (formerly $ROLEINGROUP) cannot be used. Use resourcepool:$GROUP::$ROLE instead.

- **user:**PROP::property_name
  Adds the user specified by the property name to the signoff member list for the task to which it is attached.
  
  If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **resourcepool:**PROP::property_name
Adds the resource pool specified by the property name to the signoff member list for the task to which it is attached.

If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **$PROPOSED_RESPONSIBLE_PARTY**
  Affects assignments based on the user assigned as the responsible party for the first target object.

- **$USER**
  Adds the current user to the signoff member list and as the responsible party.

- **$PROCESS_OWNER**
  Adds the workflow process owner to the signoff member list and as the responsible party.

- **$TARGET_OWNER [type]**
  Adds the owner of the first target of the specified type to the signoff member list and as the responsible party. The type value is optional. If not specified, the first target is used.

- **$PROJECT_ADMINISTRATOR, $PROJECT_TEAM_ADMINISTRATOR**
  Dynamically adds the project team members belonging to the role specified in the argument value to the signoff member list and as the responsible party. The project team is determined by the project team associated with the first target object.

- **$REQUESTOR, $ANALYST, $CHANGE_SPECIALIST1, $CHANGE_SPECIALIST2, $CHANGE_SPECIALIST3**
  Dynamically resolves to the user or resource pool associated with the first change target object in the workflow process. The particular user or resource pool is determined by the role specified in the argument value.

  **Note**
  Change-related keywords apply only to change objects. If the workflow process does not contain a change object as a target, the argument resolves to null.

  Change Manager does not need to be enabled before these keywords take effect, but during installation, Change Management must be selected under Extensions→Enterprise Knowledge Foundation in Teamcenter Environment Manager.

  **-from_include_type=object-type1[,object-type2,...]**
  (Optional) Specifies the object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.
-from_exclude_type=object-type1[,object-type2,...]  
(Optional) Specifies the object types to be excluded when getting the property value when it is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

-from_attach= target | reference | schedule_task  
(Optional) Specifies which type of attachment (target, reference, or schedule_task) to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). If this argument is not specified, the default is target.

-from_relation  
(Optional) Specifies the relation of the objects to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). It must be a valid relation.

- For manifestations, use IMAN_manifestation.
- For specifications, use IMAN_specification.
- For requirements, use IMAN_requirement.
- For references, use IMAN_reference.
- For BOM views, use PSBOMViewRevision.

This argument must be used with the -from_attach argument. A derived object is identified by starting with objects of the specified attachment type indicated by the -from_attach argument and then locating the first secondary object with the specified relation indicated by the -relation argument.

-from_include_related_type=object-type1[,object-type2]  
(Optional) Specifies the related object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

Use this argument when a property is designated and you use the -from_relation argument.

This argument should not be used with the -from_exclude_related_type argument.

-from_exclude_related_type=object-type1[,object-type2]  
(Optional) Specifies related object types to be excluded when getting the property value when it is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

Use this argument when a property is designated and you use the -from_relation argument.

This argument should not be used with the -from_include_related_type argument.
-target_task
(Optional) Specifies the multilevel task path to which the reviewers are added. The path is from the root task to the subtask with the path levels separated with colons (:) For example:

**Change Request Review:QA Review:perform-signoff**

**PLACEMENT**
Place on the Start action.

**RESTRICTIONS**
None.

**EXAMPLES**

- This example makes **Smith** the responsible party for the task to which this handler is assigned and all of the task's subtasks.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-subtasks</td>
<td></td>
</tr>
<tr>
<td>-assignee</td>
<td>user:Smith</td>
</tr>
</tbody>
</table>

- This example makes the workflow process owner the responsible party for the task to which this handler is assigned.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$PROCESS_OWNER</td>
</tr>
</tbody>
</table>

- This example makes the engineer role within manufacturing group resource pool the responsible party for the task to which this handler is assigned.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>resourcepool:manufacturing::engineer</td>
</tr>
</tbody>
</table>

- This example makes the responsible party group the responsible party for the task to which this handler is assigned.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$PROPOSED_RESPONSIBLE_PARTY</td>
</tr>
</tbody>
</table>

- This example makes the project administrator of the project associated with the first target the responsible party for the task to which this handler is assigned.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$PROJECT_ADMINISTRATOR</td>
</tr>
</tbody>
</table>

- This example makes the user or resource pool associated as **ANALYST** with the first change target the responsible party for the task to which this handler is assigned.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$ANALYST</td>
</tr>
</tbody>
</table>

Workflow handlers

Active Workspace Deployment 8-33
**EPM-auto-assign-rest**

**Note**

In Teamcenter 9.1, this handler is named **auto-assign-rest**.

**DESCRIPTION**

Automatically makes the specified user or resource pool the responsible party for any unassigned subtasks of the parent task to which this handler is added. You specify the user or resource pool by entering a comma-delimited list in the **Arguments** column for this handler.

This handler first assumes that the list contains user IDs and attempts to match the entries (in the order listed) to valid user IDs. The first entry matching a user ID is made the responsible party for any subtasks of the task to which this handler is assigned.

If no entries in the list match a valid user ID, the system attempts to match the entries (in the order listed) to valid resource pool names. The first entry matching a resource pool name (group, group/role, or role) is made the responsible party for any subtasks of the task to which this handler is assigned.

If this handler is attached to the root task with no argument specified, the workflow process initiator is made the responsible party for all tasks in the workflow process.

If this handler is attached to the root task and one or more entries are contained in the list, the first valid user or resource pool is made the responsible party for all tasks in the workflow process.

**SYNTAX**

```
EPM-auto-assign-rest
-assignee= [user:user | person:person | resourcepool:group::role
| user:PROP::property_name
| resourcepool:PROP::property_name
| $PROPOSED_RESPONSIBLE_PARY | $USER
| $PROCESS_OWNER | $TARGET_OWNER [type]
| $PROJECT_ADMINISTRATOR
| $PROJECT_TEAM_ADMINISTRATOR]
| $REQUESTOR | $ANALYST
| $CHANGE_SPECIALIST1
| $CHANGE_SPECIALIST2
| $CHANGE_SPECIALIST3
[-from_include_type=object-type1[,object-type2,...]]
[-from_exclude_type=object-type1[,object-type2,...]]
[-from_attach= target | reference | schedule_task]
[-from_relation=relation-type]
[-from_include_related_type=object-type1[,object-type2,...] |
-from_exclude_related_type=object-type1[,object-type2,...]]
```

**ARGUMENTS**

- **assignee**

  Assigns the responsible party for the task to which this handler is added. Accepts one of the following in the format specified:
• **user**:user
  Adds the user specified to the signoff member list and as the responsible party for the task to which the handler is attached. Accepts a valid Teamcenter user ID.

• **person**:person
  Adds the person whose name is specified to the signoff member list and as the responsible party for the task to which the handler is attached. Accepts a valid Teamcenter person name.

  **Note**
  If the person’s name includes a comma, you must include an escape character (\) to add the correct person. For example, to use wayne, joan:

  - `assignee=person:wayne\, joan`

• **resourcepool**:group::role
  Results in a single assignment which can be performed by any single member of this group/role.

  You can define resource pools in the form of `group::`, `group::role`, or `role`. Accepts valid Teamcenter resource pool names and these keywords:

  o **$GROUP**
    Current user’s current group.

  o **$ROLE**
    Current user’s current role.

  o **$TARGET_GROUP[type]**
    Owning group of the first target object of the specified type. The type value is optional. If not specified, the first target is used.

  o **$PROCESS_GROUP**
    Owning group of the workflow process.

    **Note**
    The `$ROLE_IN_GROUP` keyword (formerly `$ROLEINGROUP`) cannot be used. Use `resourcepool:$GROUP::$ROLE` instead.

• **user**:PROP::`property_name`
  Adds the user specified by the property name to the signoff member list for the task to which it is attached.

  If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

• **resourcepool**:PROP::`property_name`
Adds the resource pool specified by the property name to the signoff member list for the task to which it is attached.

If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **$PROPOSED_RESPONSIBLE_PARTY**
  Affects assignments based on the user assigned as the responsible party for the first target object.

- **$USER**
  Adds the current user to the signoff member list and as the responsible party.

- **$PROCESS_OWNER**
  Adds the workflow process owner to the signoff member list and as the responsible party.

- **$TARGET_OWNER [type]**
  Adds the owner of the first target of the specified type to the signoff member list and as the responsible party. The type value is optional. If not specified, the first target is used.

- **$PROJECT_ADMINISTRATOR, $PROJECT_TEAM_ADMINISTRATOR**
  Dynamically adds the project team members belonging to the role specified in the argument value to the signoff member list and as the responsible party. The project team is determined by the project team associated with the first target object.

- **$REQUESTOR, $ANALYST, $CHANGE_SPECIALIST1, $CHANGE_SPECIALIST2, $CHANGE_SPECIALIST3**
  Dynamically resolves to the user or resource pool associated with the first change target object in the workflow process. The particular user or resource pool is determined by the role specified in the argument value.

  **Note**
  Change-related keywords apply only to change objects. If the workflow process does not contain a change object as a target, the argument resolves to null.

  Change Manager does not need to be enabled before these keywords take effect, but during installation, Change Management must be selected under Extensions→Enterprise Knowledge Foundation in Teamcenter Environment Manager.

- **from_include_type=object-type1[,object-type2,...]**
  (Optional) Specifies the object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.
-from_exclude_type=object-type1,[object-type2,...]
(Optional) Specifies the object types to be excluded when getting the property value when it is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

-from_attach= target | reference | schedule_task
(Optional) Specifies which type of attachment (target, reference, or schedule_task) to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). If this argument is not specified, the default is target.

-from_relation
(Optional) Specifies the relation of the objects to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). It must be a valid relation.
- For manifestations, use IMAN_manifestation.
- For specifications, use IMAN_specification.
- For requirements, use IMAN_requirement.
- For references, use IMAN_reference.
- For BOM views, use PSBOMViewRevision.

This argument must be used with the -from_attach argument. A derived object is identified by starting with objects of the specified attachment type indicated by the -from_attach argument and then locating the first secondary object with the specified relation indicated by the -relation argument.

-from_include_related_type=object-type1,[object-type2]
(Optional) Specifies the related object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

Use this argument when a property is designated and you use the -from_relation argument.

This argument should not be used with the -from_exclude_related_type argument.

-from_exclude_related_type=object-type1,[object-type2]
(Optional) Specifies related object types to be excluded when getting the property value when it is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

Use this argument when a property is designated and you use the -from_relation argument.

This argument should not be used with the -from_include_related_type argument.

PLACEMENT
Place on the Start action. Typically placed on the root task after the EPM-assign-team-selector handler.
RESTRICTIONS
None.

EXAMPLES
• In this example, a five-task workflow process containing the task templates below is initiated by user Jones. The EPM-auto-assign-rest handler is placed on the root task, and the EPM-auto-assign handler is placed on the fourth task, set with the -assignee=$PROCESS_OWNER argument.

The workflow consists of a Do task, Review task, Checklist task, Review task, and Do task.

Because the EPM-auto-assign-rest handler is placed on the root task and Smith is specified with the -assignee argument, Smith is the responsible party for the first three tasks (and their subtasks). Because the EPM-auto-assign -assignee=$PROCESS_OWNER handler is placed on the fourth task, Jones is the responsible party for the fourth task and its subtasks. Smith is the owner of the fifth task.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:Smith</td>
</tr>
</tbody>
</table>

• This example assigns the user or resource pool assigned as the responsible party for the subtasks of the task to which this handler is assigned.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$PROPOSED_RESPONSIBLE_PARTY</td>
</tr>
</tbody>
</table>

• This example assigns the user or resource pool associated as ANALYST with the first change target object the responsible party for the subtasks of the task to which this handler is assigned.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$ANALYST</td>
</tr>
</tbody>
</table>
EPM-fill-in-reviewers

Note

In Teamcenter 9.1, this handler is named **CR-fill-in-reviewers**.

**DESCRIPTION**

Automatically assigns signoff reviewers that meet specified user, group, or role criteria for the specified Review task. This criteria populates the signoff profiles.

This handler compares the assigned user with the profile definition in the corresponding select-signoff-team task. If the assigned user does not match the profile definition, automatic assignment does not occur and the select-signoff-team task must be performed manually.

**SYNTAX**

```
EPM-fill-in-reviewers
-assignee=[user:user | person:person | addresslist:list | resourcepool:group::role | allmembers:group::role | user:PROP::property_name | resourcepool:PROP::property_name | allmembers:PROP::property_name | $PROPOSED_RESPONSIBLE_PARTY | $PROPOSED_REVIEWERS | $USER | $PROCESS_OWNER | $TARGET_OWNER [type] | $PROJECT_ADMINISTRATOR | $PROJECT_TEAM_ADMINISTRATOR | $PROJECT_AUTHOR | $PROJECT_MEMBER | $REQUESTOR | $ANALYST | $CHANGE_SPECIALIST1 | $CHANGE_SPECIALIST2 | $CHANGE_SPECIALIST3 | $CHANGE_REVIEW_BOARD | $CHANGE_IMPLEMENTATION_BOARD | [-from_include_type=object-type1[,object-type2,...]] | [-from_exclude_type=object-type1[,object-type2,...]] | [-from_attach=target | reference | schedule_task] | [-from_relation=relation-type] | [-from_include_related_type=object-type1[,object-type2,...]] | [-from_exclude_related_type=object-type1[,object-type2,...]] | [-add_excess_as_adhoc] | [-target_task=review-task-name | multilevel-task-path]
```

**ARGUMENTS**

- **assignee**

Assigns the specified users, role members, group members, and/or resource pool members to the signoff team.

- **user:user**

  Adds the user specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter user ID.

- **person:person**

  Adds the user whose name is specified to the signoff member list for the task to which it is attached. Accepts a valid Teamcenter person name.
If the person’s name includes a comma, you must include an escape character (\) to add the correct person. For example, to use wayne, joan:

-assignee=person:wayne, joan

- addresslist:list
  Adds all members of the address list specified to the signoff member list.

- resourcepool:group::role
  Results in a single assignment which can be performed by any single member of this group/role.
  
  You can define resource pools in the form of group::, group::role, or role. Accepts valid Teamcenter resource pool names and these keywords:

  o $GROUP
    Current user’s current group.

  o $ROLE
    Current user’s current role.

  o $TARGET_GROUP[type]
    Owning group of the first target object of the specified type. The type value is optional. If not specified, the first target is used.

  o $PROCESS_GROUP
    Owning group of the workflow process.

  Note
  The $ROLE_IN_GROUP keyword (formerly $ROLEINGROUP) cannot be used. Use resourcepool:$GROUP::$ROLE instead.

- allmembers:group::role
  Adds all members of a group/role combination to the signoff member list. You can define role in groups in the form of group::, group::role, or role. Accepts valid Teamcenter resource pool names and these keywords:

  o $GROUP
    Current user’s current group.

  o $ROLE
    Current user’s current role.

  o $TARGET_GROUP[type]
Owning group of the first target object of the specified type. The type value is optional. If not specified, the first target is used.

- **$PROCESS_GROUP**
  Owning group of the workflow process.

- **user:PROP::property_name**
  Adds the user specified by the property name to the signoff member list for the task to which it is attached.
  If the property is a multi-value property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **resourcepool:PROP::property_name**
  Adds the resource pool specified by the property name to the signoff member list for the task to which it is attached.
  If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **allmembers:PROP::property_name**
  Adds all members of a group/role combination that is specified by the property name to the signoff member list.
  If the property is a multivalue property, only the first value is used when only a single user is assigned in the workflow. When more than one user is assigned, all property values are used.

- **$PROPOSED_RESPONSIBLE_P**
  Affects assignments based on the user assigned as the responsible party for the first target object.

- **$PROPOSED_REVIEWERS**
  Affects assignments based on members assigned as reviewers for the first target object.

- **$USER**
  Adds the current user to the signoff member list.
  If $USER is used, and the current user belongs to several groups and roles, the behavior of the $USER keyword depends on the value of the SIGNOFF_fill_in_reviewers preference, as follows:

  - **1**
    Attempts to match the current user’s group/role values with the profile first, default values second, then any other groups/roles of which the current user is a member. This is the default setting.
Chapter 8: Workflow handlers

- 2
  Attempts to match the current user's group/role values first, default values of which the current user is a member second.

- 3
  Attempts to match the current user's group/role values.

- **$PROCESS_OWNER**
  Adds the workflow process owner to the signoff member list.

- **$TARGET_OWNER [type]**
  Adds the owner of the first target of specified type to the signoff member list. The type value is optional. If not specified, the first target is used.

- **$PROJECT_ADMINISTRATOR, $PROJECT_TEAM_ADMINISTRATOR, $PROJECT_AUTHOR, $PROJECT_MEMBER**
  Dynamically adds the project team members belonging to the role specified in the argument value. The project team is determined by the project team associated with the first target object.

- **$REQUESTOR, $ANALYST, $CHANGE_SPECIALIST1, $CHANGE_SPECIALIST2, $CHANGE_SPECIALIST3 $CHANGE_REVIEW_BOARD, $CHANGE_IMPLEMENTATION_BOARD**
  Dynamically resolves to the user or resource pool associated with the first Change target object in the process. The particular user or resource pool is determined by the role specified in the argument value.

  **Note**
  Change-related keywords apply only to change objects. If the process does not contain a change object as a target, the argument resolves to null.

  Change Manager does not need to be enabled before these keywords take effect, but during installation, **Change Management** must be selected under Extensions→Enterprise Knowledge Foundation in Teamcenter Environment Manager.

- **-from_include_type=object-type1[,object-type2,...]**
  (Optional) Specifies the object types to be used to get the property value from when a property is specified in the **-assignee** argument (for example, **-assignee=user:PROP::property_name**). They must be valid object types.

- **-from_exclude_type=object-type1[,object-type2,...]**
  (Optional) Specifies the object types to be excluded when getting the property value when it is specified in the **-assignee** argument (for example, **-assignee=user:PROP::property_name**). They must be valid object types.
-from_attach= target | reference | schedule_task  
(Optional) Specifies which type of attachment (target, reference, or schedule_task) to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). If this argument is not specified, the default is target.

-from_relation  
(Optional) Specifies the relation of the objects to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). It must be a valid relation.

- For manifestations, use IMAN_manifestation.
- For specifications, use IMAN_specification.
- For requirements, use IMAN_requirement.
- For references, use IMAN_reference.
- For BOM views, use PSBOMViewRevision.

This argument must be used with the -from_attach argument. A derived object is identified by starting with objects of the specified attachment type indicated by the -from_attach argument and then locating the first secondary object with the specified relation indicated by the -relation argument.

-from_include_related_type=object-type1[,object-type2]  
(Optional) Specifies the related object types to be used to get the property value from when a property is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

Use this argument when a property is designated and you use the -from_relation argument.

This argument should not be used with the -from_exclude_related_type argument.

-from_exclude_related_type=object-type1[,object-type2]  
(Optional) Specifies related object types to be excluded when getting the property value when it is specified in the -assignee argument (for example, -assignee=user:PROP::property_name). They must be valid object types.

Use this argument when a property is designated and you use the -from_relation argument.

This argument should not be used with the -from_include_related_type argument.

-add_excess_as_adhoc  
(Optional.) Adds the rest of the assignees as ad hoc users if the profile is satisfied.

-target_task  
(Optional) Specifies either the single Review task name or multilevel task path to which the reviewers are added. The path is from the root task to the select-signoff-team subtask with the path levels separated with colons (:). For example:
Change Request Review:QA Review:select-signoff-team

**PLACEMENT**

Place either on the Start action of the relevant select-signoff-team task or on the root task with the -review_task_name argument.

**RESTRICTIONS**

Use only with the select-signoff-team task or on the root task.

**EXAMPLES**

- This example designates the user tom and all members of the engineering group as reviewers for the Review task called Review Task 1.
  
<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:tom, allmembers:engineering::</td>
</tr>
<tr>
<td>-target_task</td>
<td>$ROOT.Review Task 1</td>
</tr>
</tbody>
</table>
  
- This example shows the current user added as a reviewer.
  
<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:$USER</td>
</tr>
<tr>
<td>-target_task</td>
<td>Review Task 1</td>
</tr>
</tbody>
</table>
  
- This example designates members assigned as reviewers for the first target object as reviewers for the Review task called Review Task 1.
  
<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>$PROPOSED_REVIEWERS</td>
</tr>
<tr>
<td>-target_task</td>
<td>Review Task 1</td>
</tr>
</tbody>
</table>
  
- This example designates user tom, all the members of the Engineering group, and the REQUESTOR associated with the first change target object as reviewers for the Review task named Review Task 1.
  
<table>
<thead>
<tr>
<th>Argument</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>-assignee</td>
<td>user:tom, allmembers:engineering::,$REQUESTOR</td>
</tr>
<tr>
<td>-target_task</td>
<td>Review Task 1</td>
</tr>
</tbody>
</table>

If the handler with these arguments is specified on the Notify task under the Route task, the signoff attachments are added to the Notify task and used for sending notifications.
Chapter 9: Active Workspace in other Teamcenter clients

Host Active Workspace in stand-alone visualization

Note
Your administrator must install Lifecycle Visualization 10.1.2 or later on your computer and configure an FCC for the server.

1. In the stand-alone viewer, choose Web→Edit Links.

2. In the Edit Links dialog box, click Add.
   A new link appears in the Link Names list.

3. Type a name for the link, and press Enter.

4. Select the link and then, in Link URL, type the Active Workspace address.

5. Add the following text to the end of the address:
   ?ah=true
   Your full address should now resemble the following:
   http://<your_aw_deployment>?ah=true

Note
Ensure that there is no forward slash / before the question mark ? in the address.

6. Ensure the Display in Info Browser check box is selected after you add ?ah=true to the end of the address.
   If the check box is cleared, an error appears when Active Workspace is launched from the stand-alone viewer.

7. Click OK.
   Your new link is now displayed on the Web menu.

8. To open Active Workspace in the browser embedded in stand-alone visualization, choose Web→<your_aw_link>.

9. Type your user name and password, and click Sign in.

10. Select an item revision, BOM structure, or other object associated with visualization data, click Open , and choose Open in Vis.
11. To load and display parts, do one of the following:
   - In the assembly tree, select the check box for an assembly or part.
   - Right-click the viewing window, and choose All On (Ctrl+break to cancel).

**Host Active Workspace in Adobe applications**

**Install the Active Workspace extension in Adobe Extension Manager CC**

Active Workspace can be hosted within Adobe Illustrator, Photoshop, and InDesign as an extension to the application. Use the Adobe Extension Manager CC to install Active Workspace as an extension to the selected Adobe application.

**Note**

If Adobe Extension Manager is already installed, proceed to step 3 of this procedure.

1. Open the Adobe Creative Cloud application, which was installed when you installed one of the Adobe applications.

2. Scroll to Adobe Extension Manager CC and click **Install**.

3. Open Adobe Extension Manager CC, click **Install**.
4. On the Active Workspace Client Installation CD, navigate to tc_cdrom/additional_applications/adobe/.

5. On the Select Extension to Install dialog, select the Teamcenter.zxp file and click Open.

**Modify the configuration file**

Modify the config.xml file to specify the URL for the Active Workspace installation and the directory to use for downloading and uploading files.

**Caution**

You must complete this procedure (step 4 is optional) to avoid receiving an error when Active Workspace is opened.

1. Copy the configuration file from /Library/Application Support/Adobe/CEP/extensions/com.siemens.plm.Teamcenter/config.xml to ~/Library/Application Support/Teamcenter/config.xml.

2. Navigate to ~/Library/Application Support/Teamcenter/config.xml.

3. Open the config.xml file and modify the <ActiveWorkspaceUrl> tag to reflect the Active Workspace installation location. Be sure that ?ah=true displays at the end of the URL.

   **Example**

   
   ```xml
   <ActiveWorkspaceUrl>http://localhost:8080/awc/?ah=true</ActiveWorkspaceUrl>
   ```
4. Optionally, modify the <WorkingDirectory> tag to reflect the directory to use when downloading or uploading files.

   Example


5. Save and close the config.xml file.
## Chapter 10: Troubleshooting

### General troubleshooting

If the Active Workspace client exhibits unexpected behavior, it is always good practice to clear the browser cache, and try the operation again. This is particularly important when server-side changes are made, such as updating to a new version of Active Workspace.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No server available</strong> error</td>
<td>Tune the pool size using the <code>PROCESS_WARM</code> parameter. For details, see <code>System Administration</code> in the Teamcenter collection.</td>
</tr>
<tr>
<td>Intermittent image loading issues</td>
<td>Perform one of the following:</td>
</tr>
<tr>
<td></td>
<td>• On the server, configure the web application server to exclude the problematic cipher. For example, if you have a jetty server:</td>
</tr>
<tr>
<td></td>
<td>1. In a text editor, open the <code>jetty\etc\jetty-ssl.xml</code> file and add the following lines after the <code>&lt;Set name=&quot;TrustStorePassword&quot;&gt;xxx&lt;/Set&gt;</code> line:</td>
</tr>
<tr>
<td></td>
<td>&lt;!-- avoid IE TLSv1 issue by excluding the problematic cipher --&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;Set name=&quot;ExcludeCipherSuites&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;Array type=&quot;java.lang.String&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;Item&gt;TLS_RSA_WITH_AES_128_CBC_SHA&lt;/Item&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/Array&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/Set&gt;</td>
</tr>
<tr>
<td></td>
<td>2. Save the file.</td>
</tr>
<tr>
<td></td>
<td>3. Restart the Jetty server.</td>
</tr>
<tr>
<td></td>
<td>The steps for other servers will vary.</td>
</tr>
<tr>
<td></td>
<td>• On the client, configure the browser to not use TLS 1.0. For example, in Internet Explorer, perform the following:</td>
</tr>
<tr>
<td></td>
<td>1. Choose <strong>Tools</strong>→<strong>Internet Options</strong> and click the <strong>Advanced</strong> tab.</td>
</tr>
<tr>
<td></td>
<td>2. In the <strong>Security</strong> section, clear the selection of <strong>Use TLS 1.0</strong>.</td>
</tr>
<tr>
<td></td>
<td>3. Click <strong>OK</strong>.</td>
</tr>
<tr>
<td></td>
<td>4. Restart the browser.</td>
</tr>
<tr>
<td></td>
<td>The steps for other browsers are similar to those described.</td>
</tr>
</tbody>
</table>
### Issue

**Upload file size exceeded max limit error during file uploads**

Users working with Active Workspace experience 403 errors when accessing thumbnails, files, or the viewer. (The 403 error may only be visible in the network page of the browser’s developer tools.)

**Possible resolution**

Use the Download Manager to manage large file uploads with Teamcenter.

The Active Workspace administrator should verify that the `tcSOAURL` parameter is set correctly in the `web.xml` file.

1. Open the `web.xml` file in a text editor. The `web.xml` file located in the `awc.war` archive.
2. Search for the following:
   ```xml
   <filter-name>TCLoginVerifier</filter-name>
   ```
3. If necessary, update the value of the `tcSOAURL` parameter so that it is the same as the value used for the `ProxyServlet redirectURL` parameter, which is also specified in the `web.xml` file.
4. Save the file and close the text editor.
5. Redeploy the application.

**Active Workspace does not display the same language (locale) as the Teamcenter server.**

Ensure the following:

1. Set the operating system of the computer running Active Workspace to the correct locale.
2. Set the browser running Active Workspace to the correct locale.
3. Ensure that the `awc.war` file is set to the correct locale.

---

**TcFTSIndexer troubleshooting**

**Obtain TcFTSIndexer troubleshooting logs**

1. Change the logging level to `DEBUG` in the following `%TC_DATA%\logger.properties` file:
   - `logging.rootLogger`
   - `logging.logger.Teamcenter`
   - `logging.logger.Teamcenter.Soa.Communication`
2. Change the logging level to `DEBUG` for the following in the `%TC_ROOT%\TcFTSIndexer\conf\log4j.properties` file:
   - `Log4j.logger.com.siemens.teamcenter.ftsi`
3. Restart Solr/TcServers and rerun the test usecase.
4. Send the **TcFtsIndexer log** and the matching **syslog** and **comlog** files to the Global Technical Access Center (GTAC) team.

You can identify the **syslog** and **comlog** files by matching the local time in the **TcFtsindexer** log where the error occurred with the UTC time within the syslogs. Send all files that fit this criteria.

**Note**

To avoid performance issues, revert the logging levels back to the original state when your debug session is complete.

### Resolving TcFTSIndexer issues

<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
</table>
| Locate errors in TcFTSIndexer | 1. If the **TcFTSIndexer** is still running, open a new shell and run **TcFtsIndexer.bat/sh-status** to generate the summary report in the console.  
Or  
If the **TcFTSIndexer** finished processing, navigate to the end of the **TcFTSIndexer\logs\TcFtsIndexer.log** file.  
• The summary shows the steps in the flow where errors occurred.  
• The report contains a status of each task.  
  Find each **TaskId** that has an error status.  
2. Search for the **TaskId** in the log to locate the point of failure.  
This provides information about the error you can use to resolve the problem.  
3. If you are using the Dispatcher mode of indexing, you can use the same method as above to find the **TaskId** for steps that are not dispatched.  
For the failed steps that are using Dispatcher, information at the point of failure includes the **<TaskId>_log**.  
• Search the module task log directory for the indicated **<TaskId>_m.log**.  
• Open the **<TaskId>_m.log** file and resolve the issues indicated by the errors in the log file. |
<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexing performance</td>
<td>Indexing performance depends on the number of warmed-up tcserver instances and the number of connections to those servers that are available for indexing. Using more servers and connections supports greater parallelization.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>To ensure the optimal number of warmed up servers, Siemens PLM Software recommends that the pool manager that maintains the tcserver be setup on a separate, dedicated machine.</td>
</tr>
<tr>
<td></td>
<td>You can edit the Tc.maxConnections property in the TcFtsindexer\conf\TcFtsIndexer.properties file to specify the maximum number of Tc connections open simultaneously. You can also change this value dynamically:</td>
</tr>
<tr>
<td></td>
<td>1. Open a new Teamcenter command window and navigate to the TC_ROOT\TcFTSIndexer\bin directory.</td>
</tr>
<tr>
<td></td>
<td>2. Run the following runTcFTSIndexer utility command, where the value for connections is the number of connections desired: runTcFTSIndexer -maxConnections=connections</td>
</tr>
<tr>
<td></td>
<td>The connections value should never exceed the number of warmed up servers.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>If you receive WARN - Connection to Tc failed messages, check to ensure the number of Tc.maxConnections has not exceeded the number of warmed servers.</td>
</tr>
<tr>
<td></td>
<td>If you are using Dispatcher with TcFtsIndexer, you can adjust performance as follows:</td>
</tr>
<tr>
<td></td>
<td>• Use the MaximumTasks property in the Dispatcher\Module\conf\transmodule.properties file to specify the total number of translations allowed for the module.</td>
</tr>
<tr>
<td></td>
<td>• Use the maxlimit property in the Dispatcher\Module\conf\translator.xml file to specify the maximum number of instances of a specific translator type to run simultaneously. The aggregate of this property across all translator types should not exceed the MaximumTasks property</td>
</tr>
</tbody>
</table>
### Issue

Login error

### Possible resolution

You may encounter the following error when attempting to run the `runTcFTSIndexer` utility and the environment is configured for SSO:

```
Login Error: The login attempt failed:
either the user ID or the password is invalid.
```

It may occur because the user running the utility is not properly authenticated in the LDAP server. The default user that runs the utility is `infodba`, as defined in the `Tc.user` setting in the `TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer.properties` file.

Ensure that the user running the indexer is authorized in LDAP:

1. If you are using multiple TCCS SSO App IDs, make sure they are configured correctly.

   You can configure multiple application IDs using the Environment Settings for Client Communication System panel in Teamcenter Environment Manager (TEM).

   ![Environment Settings for Client Communication System panel](image)

2. Ensure that the user defined by the `Tc.user` setting in the `TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer.properties` file is a valid user in the LDAP server and the Teamcenter database. Create a user in both if needed, or select an existing valid active user to run the `runTcFTSIndexer` utility.

3. In the console, set an environment variable to the password value.

   ```
   set mytcenv=password
   ```

4. Create an encrypted password file for this user by running the `encryptPass.bat/sh` utility, located in the `TC_ROOT\TcFTSIndexer\bin` directory, with the `-tc` argument and specifying the environment variable name created in the previous step, for example:

   ```
   encryptPass -tc mytcenv
   ```

5. After you create the encrypted password file, remove the environment variable value.

   ```
   set mytcenv=
   ```
<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
</table>
| TcFTSIndexer output that states the search engine is not accessible | The following message is displayed in output after running the `runTcFTSIndexer` utility: 
```
ERROR - The search engine is not accessible or the search engine schema is not correct.
```
The Solr schema needs to be updated. Use the following command: 
```
SOLR_HOME\TcSchemaToSolrSchemaTransform.bat 
TC_DATA\ftsi\solr_schema_files
```

| `tcservers` run out of memory | Reduce the `Tc.maxConnectionUsedCount` value in the `TcFtsIndexer\conf\TcFtsIndexer.properties` file to reduce the number of times a `tcserv`er connection can be reused before logout. This helps to lower the memory consumption per `tcserv`er. |
| `tcserv`er SOLR authentication error | Error: An error has occurred during JSON parsing: Unknown value type. Line 1 character 1.  
To resolve this error, reset the SOLR password as described in Modifying Active Workspace settings. |

### Troubleshooting object data (objdata)

This question-and-answer section assumes you have already obtained troubleshooting logs.

1. **How do I determine the cause of incorrect counts or other incorrect search results for a query in the Active Workspace client?**

   Isolate the objects down to a single problematic object or small group of objects. Change the logging levels to **TRACE** instead of **DEBUG**, then capture all the information from the Solr console and provide as part of the troubleshooting logs.

   You can also examine the results coming back from the server using Firebug or Fiddler.

2. **How do I investigate the cause of performance issues with TcFtsIndexer or while performing a search in the Active Workspace client?**

   In addition to basic logging, you can set an environment variable to detect slow SQL statements from the database.

   - Set `TC_SLOW_SQL=1` to log any SQL statement into the syslog which takes longer than one second.
   - Set `TC_SQL_DEBUG=PT`, to send all SQL statements and their timings into the syslog.

     **Note**

     Collecting these timings can slow `tcserv`er performance.

   - Turn on journalling to provide `jnl` files, and then use the performance journal viewer for initial investigation.
3. **What do I check when I run the "TcFtsIndexer.bat –task=objcdata:test" command and get the "ERROR - java.lang.Exception: Encrypted password does not exist for key IndexingEngine.Passwd" message?**

   Ensure the Solr password is properly set through TEM as described in *Modifying Active Workspace settings* for Active Workspace Indexer.

4. **What do I check when I run the "TcFtsIndexer.bat –task=objcdata:test" command and get the "ERROR - Unexpected Exception processing step: java.lang.RuntimeException: FATAL ERROR: POST to Solr returned an HTTP response code of '401', Unauthorized" message?**

   Ensure the Solr password is properly set through TEM as described in *Modifying Active Workspace settings* for Active Workspace Indexer.

5. **When I have added, modified, or deleted objects in a Teamcenter client, but a search in the Active Workspace client does not show those changes, what can I check?**

   Check the following:
   
   - Ensure that the TcFTSIndexer synchronization service is running.
   - Ensure the database triggers are installed.

6. **Why do I have missing filters or other schema issues?**

   Check the following:
   
   a. Ensure that the data was indexed.
   b. Ensure that Solr was restarted.
   c. Ensure that TcSchemaToSolrSchemaTransform was run to merge the Teamcenter schema with the Solr schema:

   ```
   %SOLR_HOME%\TcSchemaToSolrSchemaTransform.bat %TC_DATA%\ftsi\solr_schema_files
   ```
   
   d. If data model changes were deployed through hot deploy or live update and not TEM, ensure the **bmide_model_tool** was run:

   ```
   bmide_modeltool.bat -u=infodba -p=infodba -g=dba -tool=all -mode=upgrade -target_dir="%TC_DATA%"
   ```
   
   e. Ensure that the necessary data model changes were made in BMIDE.

7. **When I execute a search, why do I see the "Failure in Executing Search. Failed to execute the search request. Undefined field:"TC_0Y0_POM_application_object_0Y0_creation_date"" error?**

   Observed in some upgrade scenarios, this indicates the Solr schema is out of date.

   Run the **bmide_model_tool** and **TcSchemaToSolrSchemaTransform**.

8. **What can I do to verify the schema?**

   a. Examine Solr Schema in Solr Admin Panel and find the Teamcenter schema that starts with **TC_0Y0_**.
b. Run the bmide_extractor utility to extract the datamodel and verify data model changes.

c. Run the preferences utility to extract preferences and verify that Active Workspace preferences are present.

9. **What can I do after installation if indexing fails and generates the "ERROR – Failed to download the FtsIndexerDataset Files. SOA_SERVER_RETURNED_ERROR: The number of instances given is invalid for the operation." message?**

Run the TcSchemaToSolrSchemaTransform to merge the TC_SOLR_SCHEMA with the Solr schema.

**Troubleshooting structure**

<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
</table>
| TcFTSIndexer output indicates a type definition does not exist | The following message is displayed in output after running the runTcFTSIndexer utility:

```
Type definition for type does not exist.
Supported Type [objdata]
```

The **Active Content Structure** features are not installed. Rerun the Teamcenter Environment Manager (TEM) and select all **Active Content Structure** features.

Indexes are in failure states. | Use the -task=structure:recoverfailures argument.  
Examine the log output for details on the failures.

Changes to structured content do not appear immediately in searches. | Changed data is shown immediately in searches when users add, remove, or change elements (occurrences or BOM lines).

However, when users change the underlying objects to which occurrences or BOM lines refer, the changed data is not shown immediately in the content. This includes:

- Revisions to the underlying object.
- Releasing the underlying object.
- Changing effectivity on the release status.
- Changing properties on the underlying object.

Users see these changes in the content the next time data is indexed.

**Note**

The interval between indexing synchronizations is set by the search administrator.
Visualization troubleshooting

<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Visualization Server Manager is installed on a machine with no graphics card.</td>
<td>You must install the Visualization Server Manager on a physical machine with a dedicated graphics card. This is a requirement for visualization to work.</td>
</tr>
</tbody>
</table>

Verification
To check if your machine has a graphics card, do the following:
1. Open the Windows Control Panel.
2. Click Device Manager.
3. Expand Display adapters and see if a graphics card is listed.

Solution
Install the Visualization Server Manager on a physical machine with a GPU.

The FMS_HOME environment variable is not configured properly.

The FMS_HOME environment variable must point to the FMS/FCC on the machine where the Visualization Server Manager is installed. This is not likely to be a problem, but, if troubleshooting, you should ensure the variable is set correctly.

Verification
1. On the machine where the Visualization Server Manager is installed, navigate to the following location:
   
   TC_ROOT/vispoolmanager/jetty/

2. Open the jetty.service.properties file.

3. Ensure the VisPoolProxy.envset.FMS_HOME parameter points to the FMS/FCC used on the machine.
   
   To change the VisPoolProxy.envset.FMS_HOME parameter, refer to the following solution.

Solution
1. In the TC_ROOT/vispoolmanager/jetty/jetty.service.properties file, update the VisPoolProxy.envset.FMS_HOME parameter with the correct path, and save the file.

2. If the Visualization Server Manager is running as a service, stop and restart the service.
<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. If the Visualization Pool Manager is running as a <code>cmd</code> script,</td>
<td>terminate the Java Process Tree for the Visualization Server Manager and then run the <code>cmd</code> script again.</td>
</tr>
<tr>
<td>The FCC for the Visualization Server Manager does not point to the</td>
<td>The FCC installed with the Visualization Server Manager must point to the Teamcenter server that the Active Workspace client is configured to use. This is not likely to be a problem, but, if troubleshooting, you should ensure the correct server is referenced.</td>
</tr>
<tr>
<td>correct Teamcenter server.</td>
<td>Verification</td>
</tr>
<tr>
<td>Verification</td>
<td></td>
</tr>
<tr>
<td>1. On the machine where the Visualization Server Manager is installed,</td>
<td>navigate to the following location:</td>
</tr>
<tr>
<td><code>TC_ROOT/tccs/</code></td>
<td>2. Open the <code>fcc.xml</code> file and ensure that it points to the correct Teamcenter server.</td>
</tr>
<tr>
<td>Verification</td>
<td>If you need to change the <code>fcc.xml</code> file, refer to the following solution.</td>
</tr>
<tr>
<td>Solution</td>
<td></td>
</tr>
<tr>
<td>1. Update the <code>TC_ROOT/tccs/fcc.xml</code> file to point to the correct</td>
<td>Teamcenter server, and save the file.</td>
</tr>
<tr>
<td>Teamcenter server</td>
<td>2. From the command prompt, type the following to stop the currently running FCC:</td>
</tr>
<tr>
<td><code>TC_ROOT/tccs/bin/fccstat -stop</code></td>
<td><code>TC_ROOT/tccs/bin/fccstat -stop</code></td>
</tr>
<tr>
<td>Note</td>
<td>You do not need to manually restart the FCC. The Visualization Server starts the FCC automatically.</td>
</tr>
<tr>
<td>The Visualization Pool Assigner and Visualization Server Manager</td>
<td>In general, it is a good practice to use IP addresses for the hosts for the Visualization Server Manager and Visualization Pool Assigner when entering data in Teamcenter Environment Manager (TEM). However, in a LAN configuration, host names should suffice if you do not have the IP addresses.</td>
</tr>
<tr>
<td>communication protocol configuration is incorrect.</td>
<td>If you are using a VM for the Visualization Pool Assigner, use its IP address for configuring visualization in Active Workspace.</td>
</tr>
<tr>
<td>Verification</td>
<td></td>
</tr>
<tr>
<td>• Check the Visualization Pool Assigner configuration:</td>
<td></td>
</tr>
</tbody>
</table>
### Issue | Possible resolution
--- | ---
1. Using a file archiver application like 7-Zip, open the `awc.war` file. |
2. Using the **Copy To** command from the shortcut menu in 7-Zip, extract the following files:
   - `WEB-INF/web.xml`
   - `WEB-INF/classes/com/teamcenter/thinclient/resources/TreeCache.xml`
3. In the `web.xml` file, make note of the **VisPoolProxy** `<servlet>` entry, which should include the following information:
   ```xml
   <param-name>launchMode</param-name>
   <param-value>Assigner</param-value>
   <param-name>cacheConfigFile</param-name>
   <param-value>/com/teamcenter/thinclient/resources/TreeCache.xml</param-value>
   ```
4. In the `TreeCache.xml` file, make note of the following information:
   o The **ClusterName** value.
   In a subsequent step, you ensure that this is the same as the value used for the Visualization Server Manager.
   o The **ClusterConfig TCP start_port** and **end_port** values.
   These are the local ports that the Visualization Pool Assigner uses for the TreeCache.
   o The **TCPPING initial_hosts** value.
   This is the machine:port for the Visualization Server Manager. The value should point to the host and port where the Visualization Server Manager is running.
   - Check the Visualization Server Manager configuration:
   1. In the Visualization Server Manager installation, open the `TC_ROOT/vispoolmanager/jetty/jetty-service.properties` file.
   2. The value of **VisPoolProxy.launchMode** should be **PoolManager**.
      If this is set to **None**, the Visualization Server Manager was not installed correctly, and you must use TEM to reinstall it.
   3. In the `TC_ROOT/vispoolmanager/jetty/TreeCacheTCP.xml` file, make note of the following:
   o The **ClusterName** value.
<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This should be the same as the value used for the Visualization Pool Assigner.</td>
</tr>
<tr>
<td>o</td>
<td>The <strong>ClusterConfig TCP start_port</strong> and <strong>end_port</strong> values.</td>
</tr>
<tr>
<td></td>
<td>The ports should match the <strong>TCPPING initial_hosts</strong> entry in the <strong>TreeCacheTCP.xml</strong> file used by the Visualization Pool Assigner.</td>
</tr>
<tr>
<td>o</td>
<td>The <strong>TCPPING initial_hosts</strong> value.</td>
</tr>
<tr>
<td></td>
<td>This is the machine:port for the Visualization Pool Assigner. The value should point to the host and port where the Visualization Pool Assigner is running.</td>
</tr>
<tr>
<td></td>
<td>The machine should be the machine where the Visualization Server Manager is installed, and the port should match the <strong>TCP start_port</strong> and <strong>end_port</strong> values noted previously.</td>
</tr>
</tbody>
</table>

**Solution**

If any of these entries do not match, reinstall the Visualization Pool Assigner and the Visualization Server Manager using TEM.

For the Visualization Server Manager, ensure the following:

- The TreeCache **ClusterName** must match that of the Visualization Pool Assigner.

- The TCP Mode Local Service Port is the same as the value used for **TCPPING initial_hosts** in the **TreeCache.xml** file.

- The TreeCache Peers Host and Port is the Visualization Pool Assigner Host name, and the port is that found in the **TreeCache.xml** file.

For the Visualization Pool Assigner, ensure the following:

- The TreeCache Cluster Name matches that of the Visualization Server Manager.

- The TCP Mode Local Service Port is the host/port that matches the Peer/Host and Port input into the Visualization Server Manager.

- The TCP Mode Peer Host/Port is the Local Service Host/Port of the Visualization Server Manager.
Issue
The Visualization Pool Assigner and Visualization Server Manager communication protocol configuration is incorrect.

Possible resolution

Verification

- Check the Visualization Pool Assigner configuration:

  1. Using a file archiver application like 7-Zip, open the awc.war file.

  2. Using the Copy To option from the right-click menu in 7-Zip, extract the WEB-INF/web.xml file.

  3. In the web.xml file, make note of the VisPoolProxy <servlet> entry, which should include the following information:

     [param-name]launchMode[/param-name]
     [param-value]Assigner[/param-value]
     [param-name]localNode[/param-name]
     [param-value]assignerNodeName:port[/param-value]
     [param-name]peerNodes[/param-name]
     [param-value]list of ServerManagerMachineName:port[/param-value]

- Check the Visualization Server Manager configuration:

  1. In the Visualization Server Manager installation, open the TC_ROOT/vispoolmanager/jetty/jettyservice.properties file.

  2. The value of VisPoolProxy.launchMode should be PoolManager.

  3. The value of VisPoolProxy.localNode should be the name and port of the Visualization Server Manager machine.

  4. The value of VisPoolProxy.peerNodes should be a list of the assignerNodeName and port for each Visualization Pool Assigner.

Solution

If any of these entries do not match, please reinstall the Visualization Pool Assigner and the Visualization Server Manager using TEM.

For the Visualization Server Manager, ensure the list of PeerNodes includes the correct assignerNodeName and port for each Visualization Pool Assigner that the Visualization Server Manager can use.

For the Visualization Pool Assigner, ensure the list of PeerNodes includes the correct VisServerManagerMachineName and port for each Visualization Server Manager the assigner can use.
### Issue

Computer alias names used for the Visualization Pool Assigner and Visualization Server Manager machines do not match.

### Possible resolution

#### Verification

On both machines, ensure the C:\Windows\System32\drivers\etc\hosts file contains the appropriate machine alias name.

#### Solution

Update the files with appropriate alias names and then reboot.

---

The Visualization Pool Assigner and Visualization Server Manager machines are using different Java Runtime Environment (JRE) versions.

### Possible resolution

#### Verification

Check the value of each machine’s JAVA_HOME environment variable.

#### Solution

Use identical Java Runtime Environment (JRE) versions for all the server machines.

---

The console for the Visualization Pool Assigner displays messages that no pool managers are found.

### Possible resolution

#### Verification

Check the console for the Visualization Pool Assigner. If there are messages displayed that no PoolManagers are found, then you must stop and restart the Visualization Server Manager.

#### Solution

Stop and restart the Visualization Server Manager.

---

Graphics in the viewer tab display extraneous geometry.

### Possible resolution

This is an indication that the graphics driver on the Visualization Server Manager machine is not up-to-date.

#### Verification

1. On your video card manufacturer’s web site, make note of the latest driver version for your card.

2. Open the Windows Control Panel.

3. Click **Device Manager**.

4. Expand **Display adapters**.

5. Right-click the entry for your display adapter, and choose **Properties**.

6. Click the **Driver** tab.
<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The driver version on your machine is listed. If the installed driver is not the latest available, update it.</td>
<td></td>
</tr>
<tr>
<td>Solution</td>
<td>If the driver for the graphics adapter is not the latest version, update the driver and reboot.</td>
</tr>
<tr>
<td>The Visualization Pool Manager is installed on Windows 7 Professional or lower.</td>
<td>The Visualization Pool Manager is not supported on lower end editions of Windows 7 (Professional or lower).</td>
</tr>
<tr>
<td>Verification</td>
<td>To check your version of Windows, do the following:</td>
</tr>
<tr>
<td>1. Open the Windows Control Panel.</td>
<td>1. Install the Visualization Server Manager on a physical machine running one of the following versions of Windows:</td>
</tr>
<tr>
<td>2. Click System.</td>
<td>• Window 7 Enterprise or Ultimate</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2008 or 2012</td>
</tr>
<tr>
<td>2. In TEM, configure the Visualization Pool Assigner to utilize the new location for the Visualization Server Manager.</td>
<td>2. In TEM, configure the Visualization Pool Assigner to utilize the new location for the Visualization Server Manager.</td>
</tr>
<tr>
<td>You have installed the Visualization Server Manager on a VM that utilizes an older version of OpenGL.</td>
<td>Verification</td>
</tr>
<tr>
<td>Verification</td>
<td>Download and install the following utility:</td>
</tr>
<tr>
<td>You have installed the Visualization Server Manager on a VM that utilizes an older version of OpenGL.</td>
<td><a href="http://www.realtech-vr.com/glview/download.php">http://www.realtech-vr.com/glview/download.php</a></td>
</tr>
<tr>
<td>Verification</td>
<td>This is an OpenGL Extensions viewer that lists the OpenGL versions installed on your machine or VM.</td>
</tr>
<tr>
<td>Solution</td>
<td>Solution</td>
</tr>
<tr>
<td>Solution</td>
<td>If your VM has an earlier version of OpenGL such as OpenGL 1.1, install the Visualization Server Manager on a physical machine with a GPU and OpenGL 3.x or newer.</td>
</tr>
</tbody>
</table>
### Issue

A The visualization servers are busy… message is displayed in Active Workspace.

### Possible resolution

This message indicates a memory problem on the machine where the Visualization Server Manager is installed. If you encounter the issue when attempting to validate the visualization configuration in the client, the Visualization Server Manager machine probably does not have enough memory.

#### Verification

1. Log on as a user with administrator privileges.

2. On the Home page, click **Viewer Administration**.

3. For the machine in question, select **PoolManager**.

4. Click **Info**. The **Information** panel appears, providing detailed information on the PoolManager.

   If the **Memory consumption ratio** is greater than the configured maximum (the value specified for the **VisPoolProxy.maxUsageThreshold** parameter in the Visualization Server Manager **jetty-service.properties** file), the memory configuration of the machine is inadequate.
<table>
<thead>
<tr>
<th>Issue</th>
<th>Possible resolution</th>
</tr>
</thead>
</table>

### Information

- **Node type:** PoolManager
- **Pool name:** VisPool
- **Host name:** <Restricted>
- **CPU usage ratio:** 0.007963636890053749
- **Memory consumption ratio:** 0.08183255791664123

- **Serves:** <Restricted>
- **Maximum bandwidth (in bytes):** 37500000
- **Total memory in MB:** 65490
- **Number of GPU’s:** 4
- **Accepting new clients:** true
- **Total GPU memory:** 16384
- **Number of seconds since started:** 1569861
- **Date started:** Fri, May 09 at 9:30:03 AM EDT
- **Number of assignments start started:** 213
- **Preference:** <Restricted>

### Solution

Do one of the following:

- **(Preferred solution)** Install the Visualization Server Manager on a different machine and use TEM to reconfigure the Visualization Pool Assigner to use the new location of the Visualization Server Manager.

- Add memory to your Visualization Server Manager machine so it meets the minimum memory requirements.

- In the Visualization Server Manager installation, update the `VisPoolProxy.maxUsageThreshold` value in the `jettyserver.properties` file to a larger number between 0.0 and 1.0, and then restart the Visualization Pool Assigner. Use a number larger than the value of the memory consumption ratio seen in the viewer administration location in Active Workspace.

You must also update the Visualization Pool Assigner settings in the `awc.war` file. Add the following parameter to the `VisPoolProxy` `<servlet>` section:

```xml
<init-param>
  <param-name>maxUsageThreshold</param-name>
  <param-value>value described above</param-value>
</init-param>
```
### Issue
Using WebLogic to host the Visualization Pool Assigner, you see a NoClassDefFoundError or a IOException JVM_Bind error.

### Possible resolution
If you are using WebLogic to host the Visualization Pool Assigner and you see a NoClassDefFoundError or a IOException JVM_Bind error, clearing the cache and locks in WebLogic may resolve the problem.

1. Stop WebLogic.
2. Make copies of the currently deployed war files located in the autodeploy directory, and place them in another location.
3. From the domains directory, search for awc.war or tc.war (whichever is applicable in your deployment).
4. Delete the search results.
5. (Optional) Delete the contents of the following directory:
   ```
   ..\oracle\Middleware\user_projects\domains\tc\servers\AdminServer\tmp
   ```
6. Start WebLogic without the war files in the autodeploy directory to ensure WebLogic works correctly before deploying the war files.
7. Return the copies of the war files to the autodeploy directory.
8. Restart WebLogic.

### Issue
Errors on startup in the Visualization Server Manager while it attempts to initialize its server processes.

### Possible resolution
If you see error messages in the Visualization Server Manager such as Monitor was unable to connect to the visualization engine of Cold server too old and, after a period of time elapses, the error messages occur again; then something may be hindering the startup of the visualization servers. Possible causes include slow hardware and/or antivirus software. To work around the problem, open the jettyService.properties file and increase the value of VisPoolProxy.maxColdAgeS to accommodate the Visualization Server startup time on your system.
About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.