



TEAMCENTER

Teamcenter Administration

Teamcenter 2412

Unpublished work. © 2025 Siemens

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

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

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

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

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

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

About Siemens Digital Industries Software

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

Support Center: support.sw.siemens.com

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

Contents

Teamcenter Administration 1-1

Part I: Application Administration

Administration from Active Workspace 2-1

Active Admin

The Active Admin workspace 3-1

Control data access

Understanding groups, roles, users, projects, and data access 4-1

Creating the organizational structure 4-5

Creating and managing groups, roles, and users 4-7

Managing groups, roles, and users in Active Workspace 4-7

Create groups 4-7

Create and add roles to groups 4-8

Create and add users to roles 4-9

Managing users 4-10

Managing roles 4-14

Managing projects 4-15

Understanding project teams 4-15

Configuring an owning project or program 4-16

Configuring project-level security 4-17

Configure project data template folders in Explorer 4-18

Managing users' access to data using Access Manager 4-19

Controlling access to data objects 4-19

Using Access Manager rules 4-20

Managing access control lists 4-23

Importing and exporting Access Manager rules 4-25

Import, Export, and Report Administration Data

What is administration data? 5-1

Tools for managing administration data 5-2

Example tasks and processes for managing site behavior 5-3

Reconcile differences in administration data between Teamcenter sites 5-5

Exporting administration data 5-6

Why would I export administration data? 5-6

Export administration data using the admin_data_export utility 5-6

Export full administration data using TEM 5-7

Export partial administration data using TEM 5-8

Considerations for partial export of administration data 5-10

Importing administration data	5-14
Why would I import administration data?	5-14
Perform a dry run import of administration data using TEM	5-14
Perform a dry run import of administration data using the admin_data_import utility	5-15
Import administration data using TEM	5-16
Import administration data using the admin_data_import utility	5-18
Postprocess imported volume objects	5-19
Administration data import merge options	5-20
Style sheet import behavior	5-21
Modifying an administration data package	5-23
Import a modified administration data package	5-25
Reporting administration data	5-25
What administration data reports are available?	5-25
Administration Data Documentation report	5-26
Administration Data Comparison report	5-27
Administration Data Import (dry run) report	5-30
Administration Data Import (history) report	5-30

Relations configuration

Relations configuration tasks	6-1
Creating new views or updating existing views	6-4
Example of configuring relations expansion	6-5
Localize names that appear in a custom Relations view	6-5
Relation browser config file syntax	6-6

Data sharing

Configuring and managing data sharing	7-1
Configure report layout settings	7-3
Configure Briefcase file sharing	7-3
Configure PLM XML data sharing	7-4
Configure PDX data sharing	7-4
Create mapping rules using Advanced Multi-Schema Exchanger	7-7
Using Advanced Multi-Schema Exchanger to create mapping rules	7-7
Preparing for a mapping project	7-7
Creating and updating mapping projects	7-7
Mapping object types	7-8
Deploying mapping rules	7-10
Manage Multi-Site	7-12
Specifying the scope of Multi-Site remote checkouts and checkins	7-12
Multi-Site Dashboard	7-12
Viewing Multi-Site Dashboard issues	7-13
Configuring Multi-Site Dashboard	7-14
Resolving Multi-Site issues	7-15
Monitor data exchange transactions	7-16
Monitor data exchange transactions	7-16
Configuring data exchange transaction monitoring	7-17

Viewing data exchange transaction records	7-18
Share bulk data	7-19
Bulk loading product data	7-19
Copying product data from your production environment	7-20
Copying product data into your test environment	7-21
 Configuring Teamcenter mail and instant messaging	
Configuring Teamcenter mail	8-1
Configuring Teamcenter to use email servers requiring authentication	8-2
Configuring instant messaging	8-3
 Configuring Teamcenter email polling	
What is Teamcenter email polling?	9-1
Configure email polling types	9-2
Create an email polling rule	9-3
Configure user email account settings	9-5
Configure Dispatcher for email polling	9-7
Start or schedule email polling	9-8
Teamcenter email polling preferences	9-9
 Updating property values in bulk	
Process for updating property values in bulk	10-1
Using the tc_attribute_bulk_update utility arguments to update property values in bulk	10-2
Using an XML input file to update property values in bulk	10-5
Performance statistics	10-8
Best practices for updating property values in bulk	10-9
 Logging	
Monitoring your system	11-1
Configuring the Audit Logs page	11-1
Configuring audit logs	11-1
Activating the Audit Log tab	11-3
Customize audit logs field display	11-3
Audit logs reference	11-3
Customize the audit log display	11-4
Configure user logging	11-6
 Assistant configuration	
Assistant configuration tasks	12-1
Configuring the Assistant panel	12-2
Install the Assistant	12-3
Manually create a database for the Command Prediction Service microservice	12-4

Migrate the Oracle database for the Command Prediction Service microservice	12-7
Migrate the MSSQL database for the Command Prediction Service microservice	12-8
Managing user passwords using the CPS_manage_password utility	12-10
Migrating user data using the CPS_migrate_user_data utility	12-12

Configuring units of measure

Introduction to units of measure	13-1
Configure available units of measure	13-1
Migrate legacy units of measure to the unified measurement system	13-2
Working with the unit of measure definitions file	13-5

Subscription configuration

Configuring subscriptions	14-1
Set up the environment for notifications	14-2
Configuring subscribable properties	14-3
Setting subscription notification preferences	14-4
Configuring subscription to multiple objects	14-5
Configuring My Events	14-5
Configuring news feed retention	14-6
Purging news feed notifications	14-6

Active Collaboration configuration

Installing Active Collaboration features and components	15-1
Modifying discussion default operation	15-1
Keeping users informed about discussion updates with email notifications	15-2
Generating automated messages for users	15-3
Enabling tracked discussions	15-4
Managing user permissions to delete discussions	15-4
Working with snapshots in discussions	15-5

Settings and performance

Manage system settings and performance	16-1
Troubleshooting	16-1
Retrieving Active Workspace client and server versions	16-1
General troubleshooting	16-2
View Active Workspace performance data	16-3
Verify the Active Workspace gateway and other microservices	16-5
Resetting the Active Workspace gateway and microservices	16-6
Monitoring browser activity	16-8
Performance and settings	16-9
Enabling browser caching	16-9
Working with large files in Active Workspace	16-10

Compressing images for loading them quickly	16-10
Configure image resolution	16-10
Preferences	16-12
Why do I need preferences?	16-12
How do preferences work?	16-12
An example of preference hierarchy	16-16
What are environment preferences?	16-20
Working with preferences in Active Workspace	16-20
Displaying items instead of revisions	16-25
Deleting various object types	16-26
Controlling notification timeout	16-27
Defining properties that display in object cells	16-27
Defining the revision rules list	16-28
Where can I get a list of preferences?	16-28
Business Modeler IDE constants	16-29
Global constants	16-29
Business object constants	16-30
Property constants	16-34
Utilities	16-44
Using command-line utilities	16-44
log-level	16-57

Part II: System Administration

Teamcenter system administration 17-1

System logging and journaling

Teamcenter logging	18-1
Logging for business logic servers	18-3
Business logic server (TcServer) logging levels	18-3
Set TcServer startup logging levels	18-4
Swapping logger property files for debugging purposes	18-4
Log files for Teamcenter tiers	18-5
Web tier logs	18-5
Enterprise tier logs	18-5
Resource tier logs	18-8
Dispatcher logs	18-9
Performance journaling	18-10
Finding error codes	18-10

Configuring Teamcenter for performance

Configuring the four-tier architecture for performance	19-1
Techniques for improving four-tier performance	19-1
Managing server memory	19-1
Shared memory	19-8
Sharing a TcServer instance with multiple applications	19-14
Tuning the web tier	19-14

Configuring the rich client for startup performance	19-21
Overview of configuring the rich client for startup performance	19-21
Configuring the FCC file warmer	19-23
Configure TCCS to start when users log on to a Windows operating system	19-26
Setting the PATH and AUX_PATH environment variables for enhanced performance	19-29
Might deleting my rich client cache improve performance?	19-30
What can I do to improve my startup performance?	19-30
POM_timestamp maintenance	19-33
Cleaning the backpointer table after upgrade	19-34
Improving Eclipse expression evaluation	19-35
Optimizing read expression generation	19-36
Rich client timeouts	19-38
Active Workspace timeouts	19-38

Encryption key management

Teamcenter Key Manager	20-1
Install Teamcenter Key Manager	20-1
Install a secondary Teamcenter Key Manager server	20-5
Administering Teamcenter Key Manager	20-7
Work with Key Manager	20-7
Manage policies	20-8
Manage secret keys	20-10
Manage certificates	20-11
Manage a secondary key manager server	20-12
Back up Teamcenter Key Manager data files	20-14
Start and stop Teamcenter Key Manager services	20-14
Start and stop Teamcenter Key Manager daemons	20-14
Install and uninstall Teamcenter Key Manager services	20-16
Grant group access to a Teamcenter Key Manager satellite	20-16

Teamcenter server cryptography

Cryptography and FIPS compliance in Teamcenter server	21-1

Managing secrets using Teamcenter vault

Teamcenter vault overview	22-1
Install and enable the Teamcenter vault	22-1
Start and stop Teamcenter vault services and daemons	22-1
Adjusting vault connection tolerances	22-2
Enable Teamcenter vault high availability	22-3
Updating vault passwords	22-12
Generate new vault keys	22-12
Back up and restore Teamcenter vault data	22-12

Backing up and recovering a Teamcenter environment

Backing up a Teamcenter environment	23-1
Overview of the integrated backup and recovery process	23-2
Oracle Recovery Manager (RMAN)	23-4
Introduction to the Oracle Recovery Manager (RMAN)	23-4
Benefits of RMAN	23-5
Features of RMAN	23-7
ARCHIVELOG mode considerations	23-8
Restoring purged files	23-9
Single file recovery (SFR)	23-9
Single file recovery object model	23-9
Single file recovery in Teamcenter rich client	23-10
Single file recovery query	23-11
Alternative hot backup and recovery procedures	23-11
Back up and restore SQL Server database files	23-11
Back up and restore Teamcenter volumes	23-13
Back up with SQL Server Virtual Device Interface (VDI)	23-13
 Teamcenter licenses	
Licensing overview: seats, features, and bundles	24-1
How license use is calculated	24-3
Federated license servers	24-5
Global and regional licenses	24-6
Teamcenter licensing preferences	24-10
Configure failover license servers	24-11
Managing Teamcenter licenses	24-11
Create a license server reference	24-11
Change the Teamcenter default license server	24-12
Set a default license server for a site	24-13
Assign Teamcenter licenses to users	24-14
Releasing checked out licenses	24-15
Monitor seat license level violations	24-16
Reporting license usage	24-17
License usage information logged in Teamcenter	24-17
Generate Teamcenter license reports	24-18
License usage report	24-21
License seat level report	24-23
Module usage report	24-24
License login (location) report	24-26
 Configuring external management of passwords	
Enabling external password management for Teamcenter	25-1
Configuring Teamcenter clients for Security Services	25-1
Configuring load balancer timeouts	25-3
 Administering Teamcenter client communication system (TCCS)	
Introduction to Teamcenter client communication system (TCCS)	26-1

Tools to configure TCCS	26-3
Configure TCCS for rich client (two-tier and four-tier)	26-3
Creating shared or private TCCS configurations	26-4
Configure TCCS for forward proxy	26-6
Configure TCCS for reverse proxy form-based authentication	26-7
Configure multiple TCCS environments	26-10
Configure TCCS for Kerberos	26-12
Configure TCCS for SSL	26-13
Modify four-tier server connection settings	26-15
Modify FCC parent settings	26-16
Find Security Services settings for use in TCCS	26-16
TCCS configuration files	26-17
Orientation to TCCS configuration files	26-17
tccs.xml file	26-18
fwdproxy_cfg.properties file	26-22
reverseproxy_config.xml file	26-24
TCCS and container applications	26-24
Administering the TCCS container	26-24
Administering the Teamcenter server proxy	26-25
Administering the FCC for TCCS	26-27
Administering TcMEM	26-28
Administering proxy support for clients not integrated with TCCS	26-28
TCCS logging	26-29

Volume Management application

Getting started with Volume Management	27-1
Introduction to Volume Management	27-1
Before you begin	27-2
Volume Management interface	27-3
Configuring Volume Management	27-8
Basic concepts about Volume Management	27-10
Basic tasks for Volume Management	27-13
Monitoring volumes	27-14
Introduction to monitoring volumes	27-14
Retrieving volume information	27-15
Retrieving user information for a volume	27-17
Retrieving volume usage information	27-18
How to monitor volumes	27-21
Reallocate volume resources	27-23
Defining migration policies	27-24
Migration policy options	27-24
Define a VM policy by filters and metadata	27-27
Define a VM policy by watermark levels	27-29
Define a VM policy by moving all files	27-30
Define an HSM policy to purge after migration	27-31
Define an HSM policy to purge at watermark levels	27-33
Define a Volume Management policy based on an existing policy	27-34

Deactivate a migration policy in Volume Management	27-35
Migrating volume data	27-36
Introduction to migrating volume data	27-36
Preview migration results	27-37
Migrate VM policy data using FMS-based migration	27-38
Migrate VM policy data using basic migration	27-39
Migrate HSM policy data using basic migration	27-40
Migrate HSM policy data using API-based migration	27-42
Volume Management migrated file set format	27-43
Volume Management migration reports	27-46
Introduction to generating volume migration reports	27-46
Generate a VM policy report	27-47
Generate an HSM policy report	27-47
Generate a report on all policies	27-48

File Management System

Introduction to File Management System	28-1
Benefits of using FMS	28-2
FMS components	28-4
FMS server cache	28-4
FMS client cache	28-5
FMS configuration files	28-7
Introduction to FMS configuration files	28-7
Editing FMS configuration files	28-8
FMS primary configuration file	28-8
FMS server configuration file	28-10
FMS client configuration file	28-18
Sizing FMS fast cache	28-24
Administering FMS	28-25
Introduction to administering FMS	28-25
Administering FSCs	28-25
Administering FCCs	28-38
Auditing FSCs	28-51
Configuring FMS	28-60
Administering transient volumes	28-103
Administering volumes	28-109
Load balancing FMS	28-131
Using external hardware devices for load balancing	28-134
FMS client maps	28-140
Accessing multiple databases through a single FCC	28-144
Compressing FMS files	28-147
Installing and configuring Data Share Manager	28-150
Routing FSCs between sites	28-161
Accessing remote volumes using aliases (shared network)	28-163
FMS monitoring	28-166
Verifying that files are not corrupted during transmission	28-177
Enable or disable SHA-256 digest algorithm for the ticket digest	28-181
Enable use of FIPS 140-2 cryptography in FMS	28-182

Improving FSC cache performance	28-183
Sample FMS configurations	28-183
About the sample FMS configurations	28-183
Sample LAN configurations	28-184
Remote user WAN configurations	28-195
FSC primary server load balancing	28-205
FCC LAN client failover configuration	28-207
FSC clientmap DNS suffix configuration	28-211
FCC external load balancing configuration	28-214
FSC volume load balancing of FMS data configuration	28-216
FSC volume failover configuration	28-218
FSC remote cache failover configuration	28-219
Alternate FSC remote cache failover configuration	28-224
FSC remote multiple-level cache failover configuration	28-228
FSC remote multiple-level hot cache failover configuration	28-234
FSC group import multisite routing configuration	28-240
FMS shared network configuration	28-243

Server manager

What is the server manager?	29-1
Teamcenter management console	29-1
What is the Teamcenter management console?	29-1
Install the Teamcenter management console	29-3
Configure the web app server for a Teamcenter management console web tier component	29-5
Launch the Teamcenter management console	29-9
Run the Teamcenter management console as a service	29-11
Modify settings of the Teamcenter management console	29-15
Administering pools with the Teamcenter management console	29-16
Configure pool manager monitoring with the Teamcenter management console	29-20
Configure server monitoring with the Teamcenter management console	29-24
Administer logging levels with the Teamcenter management console	29-28
Administer the web tier with the Teamcenter management console	29-31
Administer TcServer instances in the Teamcenter management console	29-32
Inactivate and activate a server pool	29-35
Administer TcServer timeouts in the Teamcenter management console	29-36
Administer embedded LDAP for Teamcenter management console	29-39
Server manager prerequisites	29-55
Server manager properties files	29-56
Overview of server manager property files	29-56
Server manager pool-specific configuration tuning	29-57
Server manager database password	29-63
Server manager monitoring	29-64
Introduction to server manager monitoring	29-64
Pool manager monitoring	29-65
Teamcenter server monitoring	29-72
Server manager logging	29-79

Server manager logging levels	29-79
Configure server manager logging	29-80
Restarting warm servers	29-81
Server Manager Troubleshooting	29-82
Windows desktop heap may cause hang of TcServer processes	29-82
Suse Linux UserTasksMax setting may need tuning	29-83
Server manager failover limitations	29-84
JMX connection error: MBean Server	29-84
Parse errors warning in Oracle trace log	29-85
Running the server manager on a machine separate from the corporate server	29-86

System maintenance

Maintaining Teamcenter servers	30-1
Scheduled maintenance for Teamcenter servers	30-1
Useful administration utilities	30-3
Using process daemons	30-4
Set Oracle-dependent services to automatically restart	30-5
Application process monitoring	30-5
Maintaining the database server	30-6
Maintaining the Oracle database	30-6
Oracle Net implementation	30-31
Maintaining the Microsoft SQL Server database	30-34



1. Teamcenter Administration

This does not include the administration of add-ons or separately licensed pieces, like Aerospace and Defense, or Manufacturing Process Management, or Classification, for example. Those guides have their own administration sections, linked below.

This guide

Application Administration	Administration of the end-user environment for the Teamcenter foundation, including: • Teamcenter mail, instant messaging, email polling • Localization • Preferences and settings • Collaboration, subscriptions
System Administration	Administration of Teamcenter as a piece of software from an operating system and network perspective, including: • License servers, RDBMS servers, FSC services • Backup and recovery • Network communication, file sharing • Encryption key management, cryptography, the secrets vault

Other administration documentation

<i>Access Management Using Projects and Programs</i>
<i>Access Management Using Rules and ACLs</i>
<i>Audit Logs Management</i>
<i>BMIDE for Data Model Design</i>
<i>Briefcase Browser</i>
<i>Data Indexing and Search Configuration</i>
<i>Dispatcher — Deployment and Administration</i>
<i>Multi-Site Collaboration</i>
<i>Organization Management Using Groups, Roles, and Users</i>
<i>Product Data Validation</i>

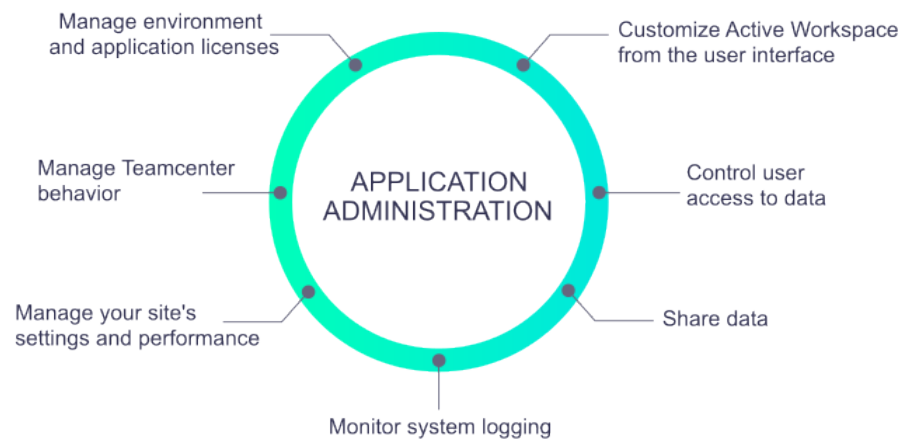
<i>Report Management Using Report Builder</i>
<i>Search and Query Builder on Rich Client</i>
<i>Search Guidemap</i>
<i>Security Administration</i>
<i>Security Services Configuration</i>
<i>Subscription Administration</i>
<i>Tc XML and PLM XML Configuration for Data Import and Export</i>
<i>Teamcenter Data Exchange</i>
<i>Teamcenter Environment Variables</i>
<i>Teamcenter Localization</i>
<i>Teamcenter Preferences</i>
<i>Teamcenter Security</i>
<i>Teamcenter Share</i>
<i>Teamcenter Utilities</i>

Part I: Application Administration

2. Administration from Active Workspace

Using Active Workspace, you can perform routine tasks to:

- Monitor your site and keep it running efficiently
- Provide an environment in which your users perform their tasks quickly.



Where do I go from here?

 Teamcenter Administrator	
Customize Active Workspace from the user interface	Active Admin workspace
Control user access to data	Manage which data your users see
Share data	Share data with external locations
Monitor system logging	Configure and view log files
Manage your site's settings and performance	Manage system settings and performance
Manage Teamcenter behavior	What is administration data? Import, Export, and Report Administration Data
Manage environment and application licenses	Configure available units of measure Update property values in bulk

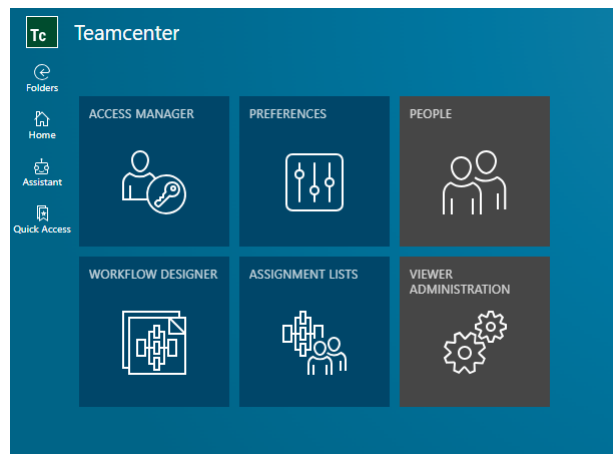
3. Active Admin

The Active Admin workspace

What is the Active Admin workspace?

This workspace is an *exclusive* workspace designed for Active Workspace administrators. Since it is exclusive, there are a limited set of pages and commands that the administrator is allowed to visit. For example, there is no access to initiate a workflow, change management, scheduling, or other end-user functionality.

The workspace provides quick access to the most common administrative functions on the home page tiles.



How do I enable it?

Use Deployment Center or the Teamcenter Environment Manager to install it.

During installation, the workspace definition file, **workspace_TcActiveAdminWorkspace.json**, is added to the **STAGE\src\solution** directory of your Active Workspace development environment, and the workspace is mapped to the dba group.

What applications does it contain?

See the workspace definition file for a complete list of available pages and commands. Following are a few of the applications displayed in the **Active Admin** workspace home page by default:

- **Access Manager**

Control user access to data objects by defining rules and defining access control lists (ACLs).

- **Preferences**

Manage your Teamcenter preferences from within Active Workspace.

- **People**

Manage your organization. Create and modify groups, roles, and users. Remove roles and users.

Note:

You must use the Teamcenter rich client to remove or delete a group or subgroup from the parent group, as this capability is not available using the Active Workspace client.

- **Workflow Designer**

Use a graphical editor to view and design workflows and task templates.

- **Assignment Lists**

Prepare lists of groups or roles to assist your users when they assign users to workflows.

- **Viewer administration**

Help troubleshoot your active visualization installation.

- **Smelters**

Help importing smelters information into Teamcenter for compliance check.

- **Delete declaration data**

Help deleting the declaration documents or imported declaration records related to compliance check from Teamcenter.

4. Control data access

Understanding groups, roles, users, projects, and data access

Groups

A **group** is a named set of users who share one or more specific types of work in common, and who have the same access permissions for data and locations. Groups may be synchronized with the company's user authentication system. Users may be assigned to more than one group. Groups can be created to represent data ownership and to control data access. *Projects* are created with specific groups, users, or roles assigned as team members, privileged team members, and team administrators.

Typically, groups are defined along project lines and not functional lines. However, you can also create groups of third-party organizations such as suppliers.

A group member can be a member of multiple groups. Groups make up the core of the organization structure.

As an administrator, you can:

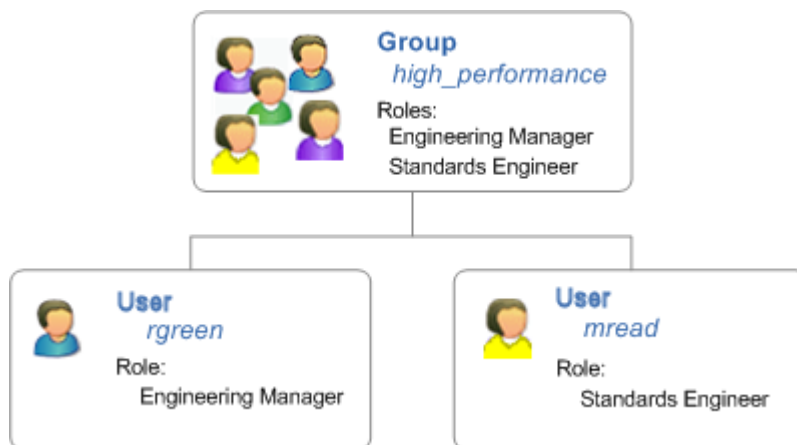
- **Create and modify** groups.

Note:

You must use the Teamcenter rich client to remove or delete a group or subgroup from the parent group, as this capability is not available using the Active Workspace client.

Example:

The **high_performance** group consists of 2 roles, that is, **Engineering Manager** and **Standards Engineer**, and 2 users, namely, **rgreen** and **mread**, who are assigned their respective roles.



- Assign authorized data access privileges to a group.
- Assign default volumes to a group.

A *volume* is a location where files are stored. A volume corresponds to a directory on the operating system. Files stored in volumes are created by CAD applications or other third-party applications. You can assign volumes to groups and define file locations for your organization structure.

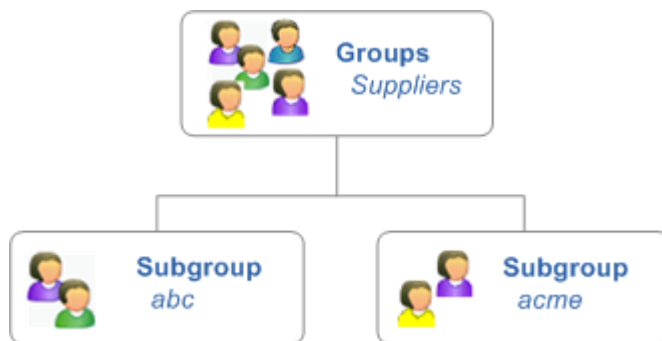
- Manage subgroups within the organization.

A *subgroup* is a group with another group designated as its parent. A subgroup may also be designated as a parent group. The position of subgroups within the organizational hierarchy can be managed by parenting and reparenting groups.

Subgroups can be used to organize users. Subgroups inherit access permissions, volumes, and preferences from their parent.

Example:

Consider a scenario where you wish to restrict contractors from viewing any content in the employee group. In this case, you can create subgroups **abc** and **acme** within a group such that users from these subgroups will not have access to the content from any groups other than their own.



Roles

A **role** defines a specific job function within a group. Roles are attached to groups and users are assigned to one or more roles within a group.

- A role can be assigned to multiple groups.
- Roles add an additional layer of data access control.
- Roles are created along functional lines.

Tip:

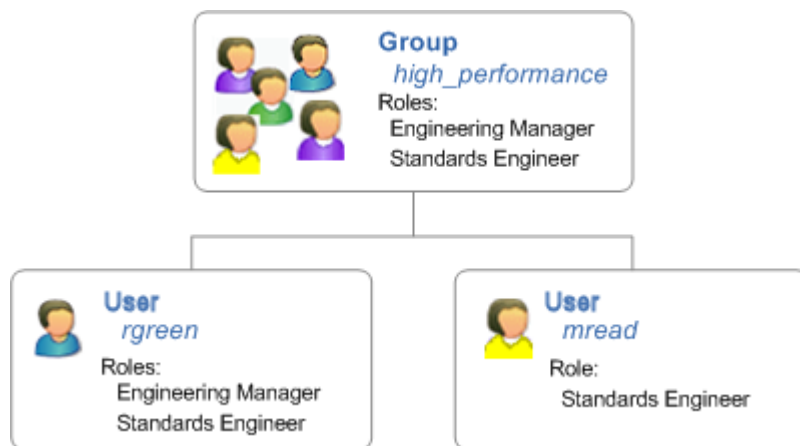
While creating roles, use real-world descriptions, skills, and responsibilities.

As an administrator, you can:

- **Create, modify, and delete** role definitions.
- **Add new or existing roles to groups.**
- Assign a default role within a group.

Example:

Robert Green, a user, is assigned the default role of **Engineering Manager**. In addition to his responsibilities as engineering manager, Robert must also perform standards-related work. Therefore, user **rgreen** is assigned two roles in the **high_performance** group: **Engineering Manager** and **Standards Engineer**.



Users

Users are individuals who interact with Active Workspace. A user is assigned to a default group and takes on a role in the group.

As an administrator, you can:

- **Create, modify, deactivate** user accounts, or **delete users from groups.**
- **Reset user passwords.**
- Assign license bundles, and license servers to a user.

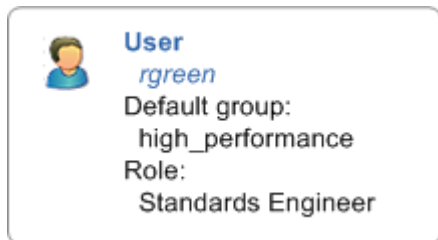
When you assign a license bundle to a specific user, the user assigned to the bundle is assured the availability of all the features in the bundle. You can use license bundling in conjunction with other licensing schemes. Consider a scenario where a user is assigned a license bundle that does not include the Systems Engineering module. When the user launches Systems Engineering, the system confirms if the feature key exists in the license file outside of the license bundle. If the feature key is found, the application can be used.

A *license server* is a process dedicated to tracking license usage by users. It runs on a host machine and port specified by an administrator. An administrator can set up multiple license servers. Each license server can have a different set of users assigned to it. This allows the load balancing of license requests so that a single license server is not overused.

Users can be assigned various roles in the organization. A user can also be part of multiple groups in the organization.

Example:

Robert Green, a user, is assigned the default role of **Standards Engineer** and belongs to the default group **high_performance**.



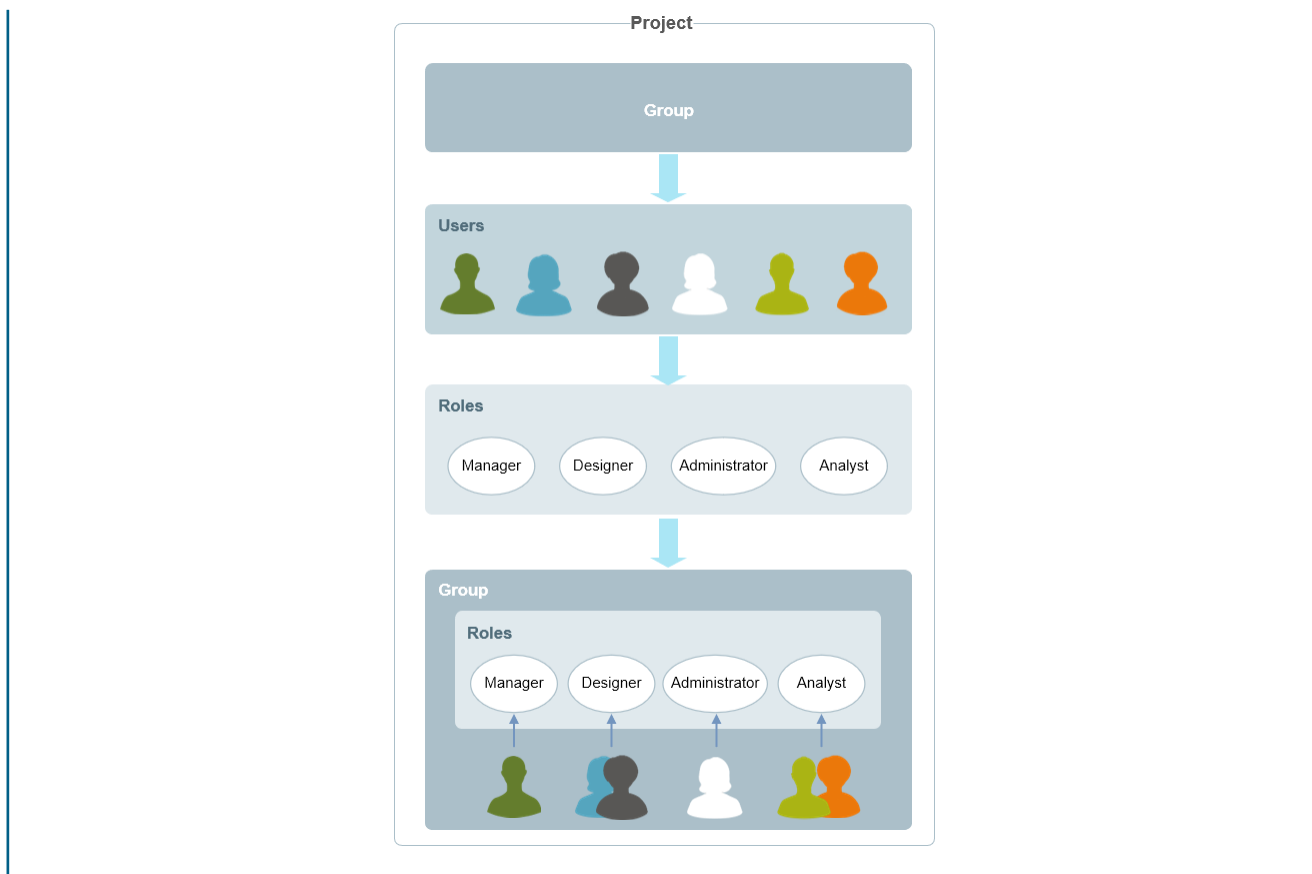
Projects

Projects provide a means to associate data and users from different groups, such as Engineering, Supplier, and Customers, and allow configuration of access rules based on this association. This is an easy way to organize your data and implement the access control based on a business project or program's security requirements.

You can use the **PROJECTS** tile to create projects and programs and manage your project teams by correlating groups of users, potentially at different physical sites, with your product data. However, you must use the rich client to delete projects.

Example:

A project comprises a group of users each having one or more roles. For example, this project consists of an administrator, a project manager, two designers, and two analysts.



There are several measures you can use to control access to your project data:

- When creating a project, you can use the **Project Category** field, which allows you to restrict a user's access to objects based on the category of the project. For example, a user assigned to the **Supplier** project category would be unable to view proprietary information. The default project categories are **Internal**, **Partner**, and **Supplier**.
- Designating team members as *privileged* is one step in the process of granting access to users to allow them to assign data to and remove data from projects and programs.

Controlling data access

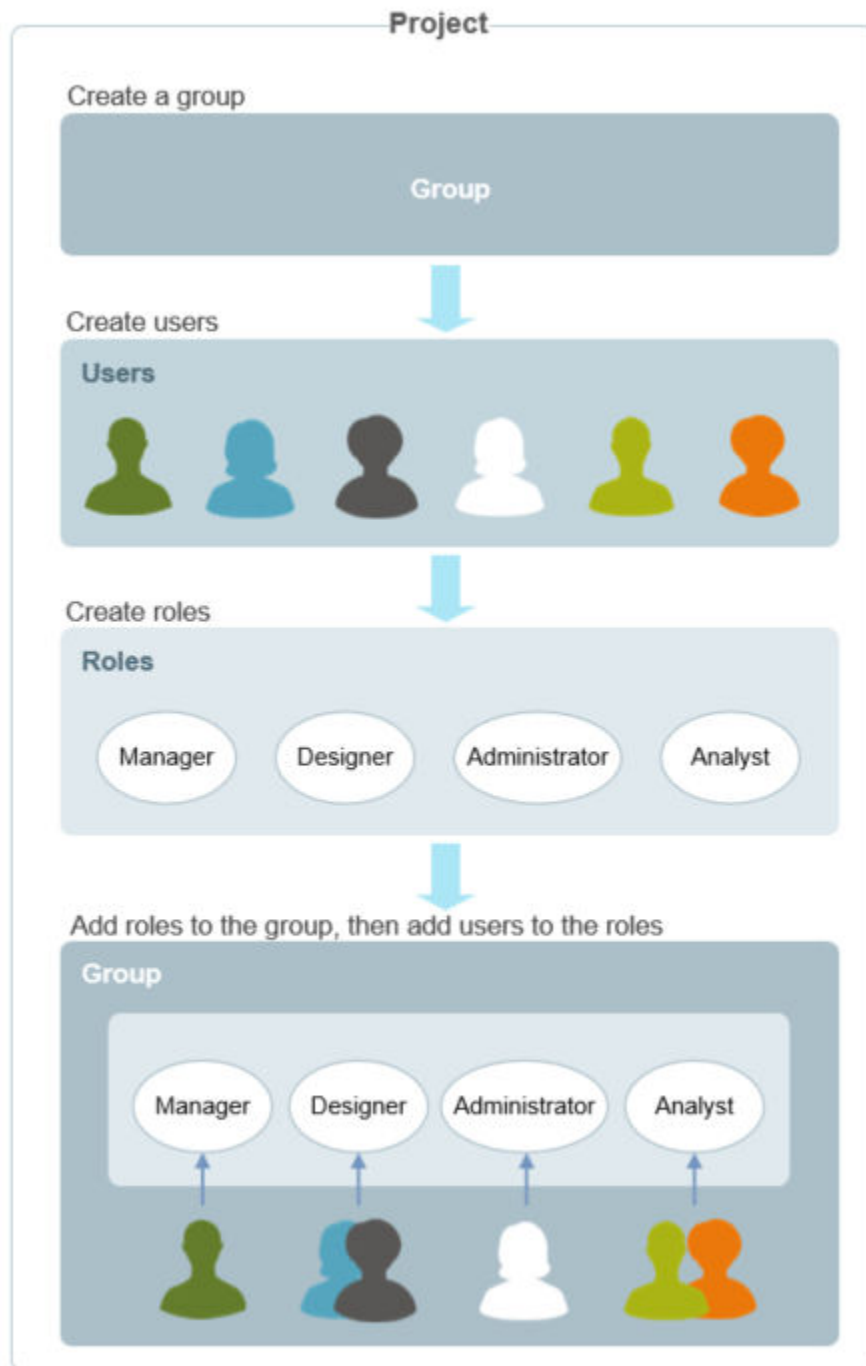
In addition to licensing and authentication, another way to control information access for users, both internal and external, is by configuring Access Manager rules and access control lists (ACLs). These rules and lists control access by user, group, role, organization, and workspace.

Creating the organizational structure

By setting up user management you can control the functionality that is available to users who are mapped to a specific role, thereby controlling the access to restricted data. To do this, you can create groups for specific projects, and add users, or roles assigned as team members to the projects.

Example:

You want to set up the organizational structure for a project that requires an administrator, a project manager, two designers, and two analysts. You first create a project and add a group to the project. You then identify the users or create new users that you require for your project. Next, you search for existing roles or create new roles required for the project. You then add the roles to the group, and add users to the respective roles in the group.



Creating and managing groups, roles, and users

Managing groups, roles, and users in Active Workspace

In Active Workspace, you can use the **PEOPLE** location to create and modify users, roles, and groups and to set up authorized access using login credentials for each user.

The **PEOPLE** tile is visible in both the **Active Admin** and **Default** workspaces.

You can create your own workspace mapped to a special group of non-**dba** group users and add the **PEOPLE** tile to it. This allows users to perform admin work because privileges for **dba** group users are too broad.

As a database administrator, you can create users and user roles specific to your organization. You can then add the users and roles to a specific group to grant them authorized access to the application.

By default, your organization's groups, users and roles are listed on the **Organization** page in Tree view.

- Click **Tree** to change to other views.
- Right-click on objects for quick access to actions you can perform.
- Click **Selection Mode** or Ctrl-click on objects to select more than one object.

Create groups

You can use the **People** location to create groups. To create your group at the organization level, first select **Organization** in the **Tree** view. To create a group within a group, first select that group using any view.

1. Perform one of the following steps:
 - On the **Organization** page, click **Add** ⊕ > **Add Group** ⊕.
 - On the **Organization** page, right-click on **Organization** (**Tree** view only) or another parent group and click **Add Group** ⊕.
 - On the **Groups** page, click **More commands** ... > **New** ✱ > **Add** ⊕.

The **Add** panel appears.

Tip:

To keep the **Add** panel open so that you can continue to add groups without opening the panel each time, click **Pin Panel** ⇄.

- To add an existing group, use **Search** to search for the group.

To create a new group, use **New** to specify the following values.

- **Name**

Specify a name that is not the same as the new group's parent or sibling groups.

- (Optional) **Description**

- (Optional) **Security**

- (Optional) **To Parent**

To create a subgroup for an existing group, select the parent group from the list. By default, the parent is the group selected at the time the new group is created.

- (Optional) **DBA Privilege**

- (Optional) **Default Volume**

- (Optional) **Default Local Volume**

- Click **Add**.

Create and add roles to groups

You can use the **People** location to create and add roles to groups.

- Perform one of the following steps:
 - On the **Organization** page, select one or more groups and click **Add** ⊕ > **Add Role** ⊕.
 - On the **Organization** page, select one or more groups, right-click and click **Add Role** ⊕.
 - On the **Groups** page, select a group and click **Add Role** ⊕ in the **Roles** section.

The **Add** panel appears.

- To add an existing role, use **Search** to search for the role.

To create a new role, use **New** to name and describe the role.

- Click **Add**.

Create and add users to roles

You can use the **People** location to create and add users to roles.

1. On the **Users** page, click **More commands ...** > **New** ✱ > **Add** ⊕.
2. In the **Add** panel, in the **New** section, enter the following:

- **Name**
- **User ID**
- **OS Name**
- **Default Group**
- (Optional) **Default Volume**
- (Optional) **Default Local Volume**
- **Status**

To create an active user, set **Status** = **0**.

- **License Level**

The types of licenses available depends on your license agreement. For descriptions of the available license levels, see your license agreement documentation.

- (Optional) **License Server**
- (Optional) **License Bundle**
- **Visualization Licensing Level**
 - 0 (Base)
 - 1 (Standard)
 - 2 (Professional)
 - 3 (Mockup)
- (Optional) **Geography**
- (Optional) **Nationality**

- (Optional) **Citizenships**

In the **Personal Information** section, specify any of the optional fields:

3. Click **Add**.

Managing users

Edit user information

You can use the **People** location to edit user information.

1. Navigate to the **Users** tab and search for an existing user. Select the user.
2. Click **Edit** .
3. Modify the user information and click **Save** .

Alternately, select the user on the **Organization** tab and click **Edit**  to update user information and other properties.

View user activity logs

If the audit logs page is activated, you can view user activity logs in the **People** location. Audit logs must be enabled for users other than those in the **dba** group. See [Activating the Audit Log tab](#).

1. In the **Roles** page, select the role for which you want to view the user activity logs.
2. Click the **Audit Logs** tab.

A table that shows the **Logged Date**, the **Event Type Name**, and the **Login User ID** is displayed under **Organization Logs**.

The screenshot shows the 'Roles' tab in the Teamcenter interface. A search bar at the top shows '46 results found for "Engineer"'. Below the search bar, a list of roles is displayed, including 'Package Engineer', 'Physical Test Engineer' (selected), and 'Probe Engineer'. The 'Physical Test Engineer' role details are shown on the right, including an 'Overview' and 'Audit Logs' section. The 'Audit Logs' section displays a table of organization logs.

Logged Date	Event Type Name	Login User ID
30-Sep-2023	_Create	dano
30-Sep-2023	_Modify	dano

Viewing user access rights

The **Access** tab enables you to view access rights on objects in Active Workspace. As an administrator, it helps you determine if the correct access privileges have been assigned to the selected user. If a user is assigned multiple groups and roles, you can determine access for that user by selecting a particular group/role combination and clicking **Show Access Rights**.

The **Access** tab contains three sections:

- **User, Group, and Role filters**

Filters the user, group, and role for the current user session context.

You can use these filters to select another user, group, and role combination for which you want to view the associated access rights for the currently selected object. Click **Show Access Rights** to apply these changes.

- **Access Rights**

Lists the operations and privileges granted to the filtered combination of user, group, and role.

- **Associated Rules**

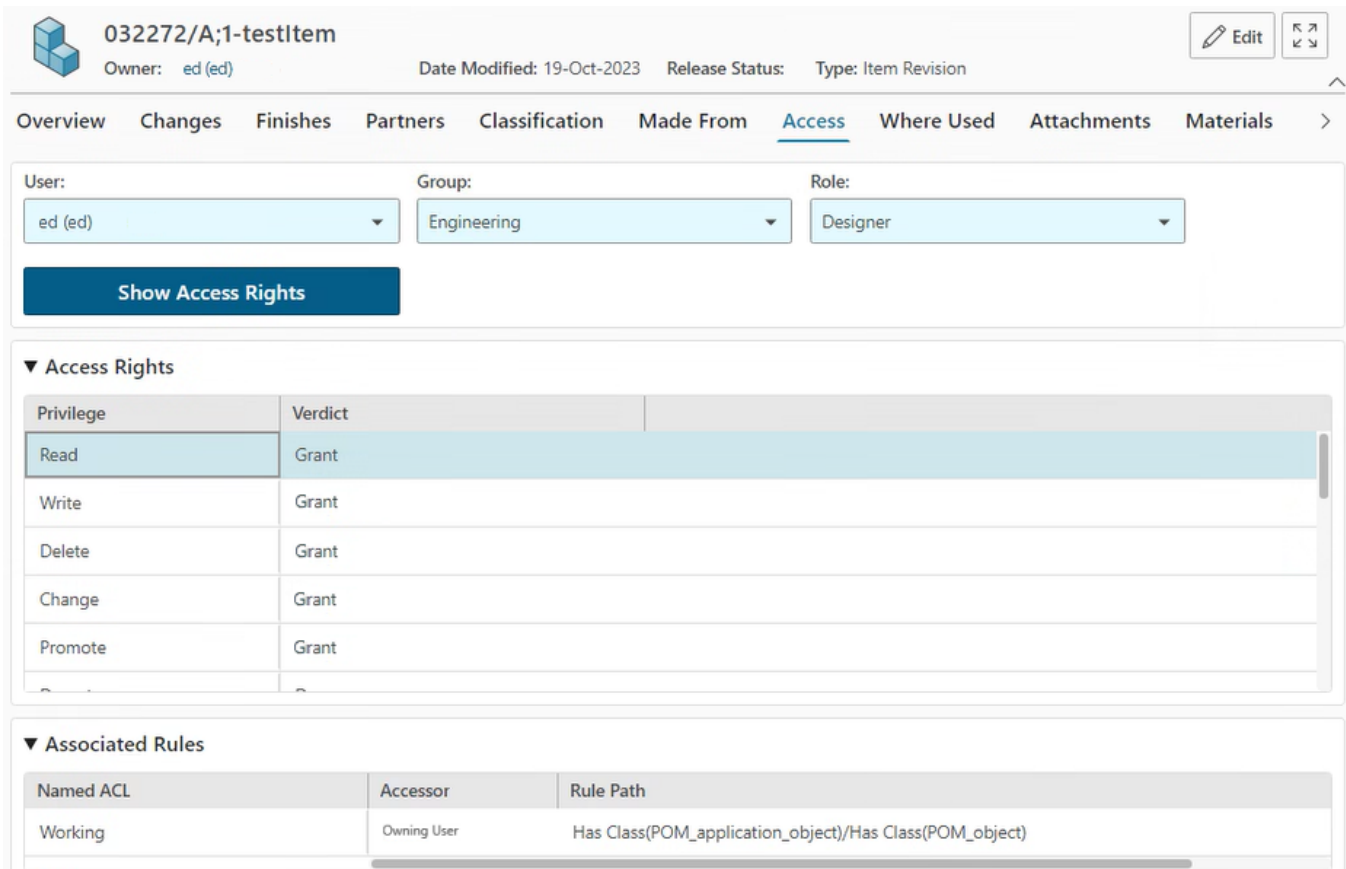
Lists the rules associated with the given object.

Note:

By default, the **Access** tab is not available. To add the **Access** tab, **edit the style sheet registered to the summary view** for the object type to which you wish to add the **Access** tab.

Example:

Ed, a designer in the engineering group, designed the **testItem** object. His access rights are shown in the **Access Rights** section. Ed is granted **Read, Write, Delete**, and other rights because he is the owning user.



032272/A;1-testItem

Owner: ed (ed) Date Modified: 19-Oct-2023 Release Status: Type: Item Revision

Overview Changes Finishes Partners Classification Made From **Access** Where Used Attachments Materials

User: ed (ed) Group: Engineering Role: Designer

Show Access Rights

▼ Access Rights

Privilege	Verdict
Read	Grant
Write	Grant
Delete	Grant
Change	Grant
Promote	Grant

▼ Associated Rules

Named ACL	Accessor	Rule Path
Working	Owning User	Has Class(POM_application_object)/Has Class(POM_object)

Add the Access tab to view user access rights

The **Access** tab enables you to view access rights on objects in Active Workspace.

By default, the **Access** tab is not available. To add the **Access** tab, **edit the style sheet registered to the summary view** for the object type to which you wish to add the **Access** tab.

1. Open the style sheet registered to the summary view for the type of object to which you wish to add the **Access** tab:

- **Item revision**

The default summary style sheet is **Awp0ItemRevSummary**.

- **Document revision**

The default summary style sheet is **Awp0IDocumentRevSummary**.

- **Requirements revision**

The default summary style sheet is **Awp0RequirementRevisionSummary**.

2. Add the following line to the appropriate style sheet.

```
<inject type="dataset" src="Aut0ItemRevSummary" />
```

- a. For an *item revision*, add the line here in the **Awp0ItemRevSummary** style sheet:

```
<inject type="dataset" src="Fsh1FinishesSection"/>
<inject type="dataset" src="Ads1NotesSection"/>
<inject type="dataset" src="Vm1PartnerContracts"/>
<inject type="preference" src="ClassificationStylesheetTab"/>
<inject type="dataset" src="Sm1MadeFromSection"/>
<inject type="dataset" src="Aut0ItemRevSummary"/>
```

- b. For a *document revision*, add the line here in the **Awp0IDocumentRevSummary** style sheet:

```
<inject type="dataset" src="Fnd0ClassificationSummary"/>
<inject type="dataset" src="ProjectListInfo"/>
<inject type="dataset" src="Aut0ItemRevSummary"/>
```

- c. For a *requirements revision*, add the line here in the **Awp0RequirementRevisionSummary** style sheet:

```
<inject type="dataset" src="WorkflowSummary"/>
<inject type="dataset" src="RelationsSummary"/>
<inject type="dataset" src="Ase0SystemRequirementsSubLocation"/>
<inject type="dataset" src="Aut0ItemRevSummary"/>
```

Add or change a user password

You can use the **People** location to change a user password.

1. In the **Users** page, select the user whose password you want to add or change.
2. Click **More commands ...** > **Manage**  > **Change Password** .
3. In the **Change Password** panel, specify and confirm the new password.
4. Click **Change**.

Deactivate users

You can deactivate a specific user ID by modifying the status of the user in the **People** location. This user is retained in the database and can be activated for future use.

Example:

Consider a designer who will be going on an extended leave of absence. Instead of deleting the user from the project group, you can temporarily deactivate the user. Once the user is available, you can set the status to active.

1. On the **Organization** page, search for the user whose status you want to modify and select the user.
2. Click **Edit**  to display the properties and other settings that can be updated for the user.
3. In the **Properties** section, set the **Status** field of the user to **1 Inactive**.
4. Click **Save** .

Delete a user from a group

You can use the **People** location to delete a user from a group.

1. In the **Organization** page, using the tree view, search for the group from which you want to delete a user.
2. Expand the group to show the roles in the group.
3. Expand the role containing the user you wish to remove from the group.
4. Select the user and click **Remove** .

Alternately, select the user, right-click, and choose **Remove** .

Managing roles

Edit a role

You can use the **People** location to edit the properties of a role.

1. On the **Roles** page, search for the role you wish to edit. Click on the role.
2. Click **Edit**  to enable editing of the role's properties.
3. Modify the role name and description.

4. Click **Save** .

Delete a role

You can use the **People** location to delete a role. However, you cannot delete a role that is referenced by another organization object.

1. In the **Organization** page, using the tree view, search for the group from which you want to delete a role.
2. Expand the group to show the roles in the group.
3. Select the role and click **Remove** .

Alternately, select the role, right-click, and chose **Remove** .

Managing projects

Understanding project teams

You can manage your project team members to streamline collaboration, ensure everyone has access to the latest project information, and maintain clear communication.

As the creator of the project or program, you are automatically assigned both **Project Administrator** and **Team Administrator** status.

To allow any user to create a project or program, set the **TC_project_create_validate_condition** preference to **true**. When this preference is set to true, users are able to create projects or programs regardless of authorization rules for **Organization** and **Access Manager** settings for creating a project.

You can manage your project team members in **Projects** or **Explorer**.

Depending on your role, you can perform the following project-related tasks:

Team role	Definition
Project Administrator	Teamcenter user with privileges to administer project teams. Users in the Project Administrator role can: • Modify the properties on the project. • Add and remove team members to projects in which the project administrator is also a member.

Team role	Definition
	- Assign Privileged, Non-privileged, and Team Administrator status to any project team member. - Add objects to their projects or remove objects from their projects. Note: You can designate multiple team administrators for each project. Having multiple team administrators allows you to balance resource management tasks for large projects.
Team Administrator	Project team member with privileges to add and remove project members. Users in the Team Administrator role can: - Add and remove team members to projects in which the team administrator is also a member. - Assign Privileged, Non-privileged, and Team Administrator status to any project team member. - Add objects to their projects or remove objects from their projects. Note: You can designate multiple team administrators for each project. Having multiple team administrators allows you to balance resource management tasks for large projects.
Privileged team members	Project team members with privileges can view their projects and their team members. They can also add objects to their projects or remove objects from their projects.
Non-privileged team members	Project team members without privileges can view their projects and their team members.

Configuring an owning project or program

You can set or change an owning project or program on an object to control access to data.

To view the **Owning** project/program tab, you must set the **AWC_Project_showOwningProgramTab** preference to **true**. An owning project/program can be set on the object only if the **autoAssignToProject** extension is enabled on the object type.

Note:

If you use the Aerospace and Defense template, the **autoAssignToProject** extension is enabled by default.

For more information on owning projects/programs and the **autoAssignToProject** extension, refer to *Access Management Using Projects and Programs* in the Teamcenter documentation.

Once you configure an owning project/program, users can set or change the owning project/program for instances such as:

- No owning project/program is set on the data.
- A user mistakenly assigned data to the wrong project/program.
- Government policies force data to be tagged with a different owning project/program.
- Addressing a partner project/program change request.

Configuring project-level security

You can configure project-level security for selected objects at the item level and apply it to all revisions under the item. To apply the project-level security to all revisions of an object, you must set the following preferences.

- **TC_Security_Apply_To_Visible**

Activates the visibility of the **Apply To** project option. When set to **true** (default), the **Apply To** project option is available.

- **TC_Security_Apply_To_Item_Revision**

Controls the behavior of the **Apply To** project option. When set to **true**, security is applied to item revisions. If set to **false** (default), security is applied to items.

When adding an object to a project, select **This Revision** or **All Revisions** to apply the project-level security.

You can also hide the project section from the **Add** panel for objects. To do this, access your data model configuration using the Business Modeler IDE and set the **Fnd0EnableAssignProjects** constant to **False**.

For more information about modifying business object constants, see *Configure your business data model in BMIDE*.

Configure project data template folders in Explorer

Users can view their assigned projects and project data in the Project Data folder in the Explorer location. The **Project Data Templates** folder appears for administrators in the **Explorer** location.

By default, one **Project Template** is included and is set as the active template. The active template defines the project information displayed for all projects for all users. As an administrator, you can configure custom template folders to define the project information that appears for your users. Only one template at a time can be set as the active template.

Create a project data template

1. Access **Explorer** by clicking the **EXPLORER** tile or **Explorer** in the global navigation bar.
2. Select the **Project Data Templates** folder.
3. Click **Add Project Data Template** .

The **Add Project Data Template** panel opens.

4. In the **Add Project Data Template** panel, enter a **Name** for the template. Optionally, you can enter a **Description** and choose to select the new template as the active template.
5. Click **Add**.

The newly created project data template folder appears under the **Project Data Templates** folder.

6. Select the new template folder.
7. Click **Add Active Folder** .
8. In the **Add Active Folder** panel, enter a **Name** for the folder. Optionally, you can enter a **Description**.
9. Click **Add**.
10. Select the new active folder.
11. On **Rules** tab, click **Edit** .
12. Choose one of the following ways to define the project criteria and filters for the folder.

Advanced	Enter all applicable values, and then click Search to define the criteria and filters for the folder.
Project Reference	Select a project to use as a reference for the criteria of the folder. You can apply additional filters to refine the results. Updating filters does not impact the original reference project. The Project Reference option is a Solr-based query and is updated with the indexing synchronization flow. If you use this option, when users add an object to a project, it may take a few minutes until the synchronization flow is complete, for the added object to be categorized into the applicable object type folder in the project folder.

You can view the applied criteria and filters in the **Overview**.

13. Click **Save** .
14. Add any additional folders to the template as applicable.

If the template is set as the active template, the template criteria and filters are applied to all projects in the **Project Data** folder.

15. If there are object types that you do not want included in the project folders, you can add these object types to the **AWC_Exclude_Types_From_More_Project_Data_Collection** preference.

Set a project data template as the active template

You can set any template as the active template by clicking **Edit**  and selecting **Is Active Template**.

Delete a project data template

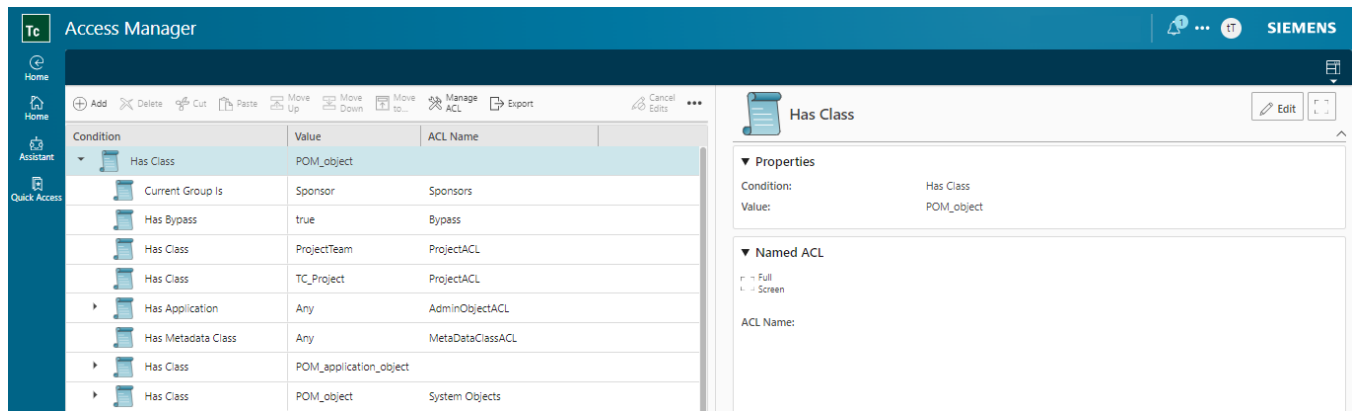
You can delete any template by clicking **More commands**  > **Edit**  > **Delete**.

Managing users' access to data using Access Manager

Controlling access to data objects

In addition to licensing and authentication, another way to control information access for users, both internal and external, is by configuring Access Manager rules and access control lists (ACLs). These rules and lists control access by user, group, role, organization, and workspace.

The **ACCESS MANAGER** tile is available only in the **Active Admin workspace** in Active Workspace for users in the **dba** group and **DBA** role.



Using Access Manager rules

Viewing Access Manager rules

From the **Access Manager** location, select a rule from the AM rule tree, for example, **Has Application**, to view the following sections:

Properties Specifies the condition name and value.

Example:

Condition: Has Application

Value: Any

Named ACL Specifies the access control list (ACL).

Example:

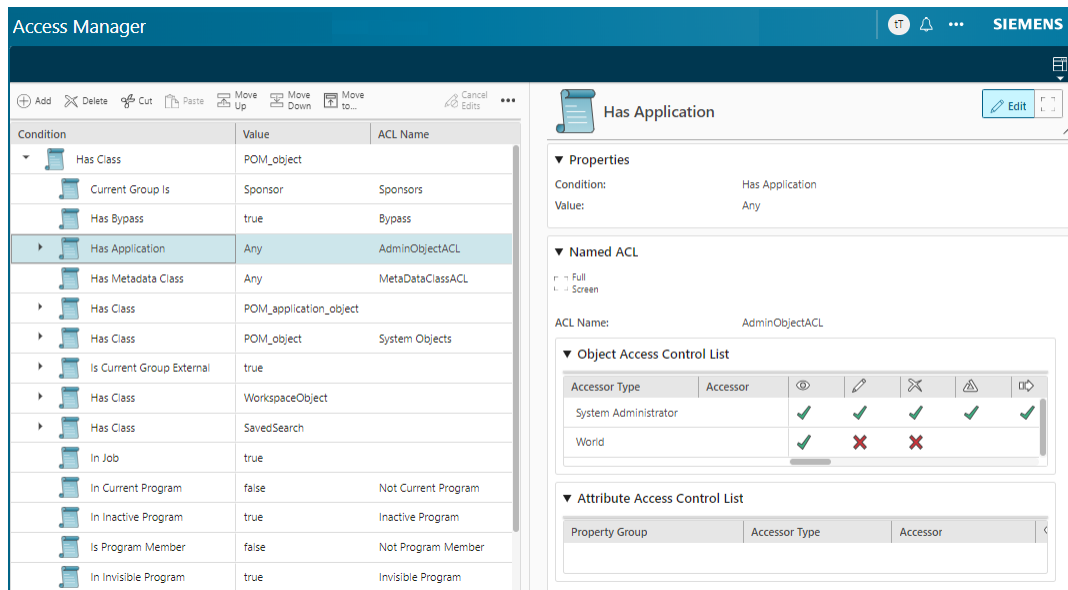
ACL Name: AdminObjectACL

Object Access Control List Specifies the ACL that lists the privileges granted to the specified accessor types.

Example:

World is granted **Read** (👁) access, but it is denied **Write** (✎) and **Delete** (✖) access.

Attribute Access Control List Specifies accessors and accessor types for the specified property groups.



Add an Access Manager rule

You can use the Access Manager location to add rules.

1. Select a node from the AM rule tree, such as **Has Application**.
2. Click **Add** ⊕ in the results panel toolbar.

The **Add Rule** panel is displayed.

3. On the **Add Rule** panel, select the condition, for example, **Has Attribute**.
4. Enter a value.

The value is populated based on the selected condition. However, since the **Has Attribute** condition does not have a value, you can enter any value. For example, **Dataset:object_desc=test**.

5. (Optional) Select an **ACL Name**, for example, **AdminObjectACL**.

To add additional rules, select **Pin Panel** ⇄.

6. Click **Add** to add the new rule under the selected parent node.

By default, the new rule appears as the last node under the parent.

Reorder Access Manager rules

From within the AM rule tree, you can reorder your rules and save your changes.

Reorder action	Icon	Description
Cut		Cut a single or multiple rules.
Paste		Paste the single or multiple rules you have cut.
Selection Mode		Turn on selection mode for making multiple rule selections.
Move Up		Move a selected rule up the AM rule tree.
Move Down		Move a selected rule down the AM rule tree.
Move to Top		Move a selected rule to the top of the AM rule tree.

Example — Move a single AM rule up the AM rule tree

- From the AM rule tree, select a single rule, for example, **In Invisible Program**.

 Add  Delete  Cut  Paste  Move Up  Move Down  Move to...  Manage ACL  Export  Import  Selection Mode  Save  Cancel Edits			
Condition	Value	ACL Name	
 Has Class	WorkspaceObject		
 Has Class	SavedSearch		
 In Job	true		
 In Current Program	false	Not Current Program	
 In Inactive Program	true	Inactive Program	
 Is Program Member	false	Not Program Member	
 In Invisible Program	true	Invisible Program	
 Has Type	EngChange Revision		

- Click **Move Up**  two times to position the rule directly below **In Current Program**.

Add Delete Cut Paste Move Up Move Down Move to... Manage ACL Export Import Selection Mode Save Cancel Edits			
Condition	Value	ACL Name	
Has Class	WorkspaceObject		
Has Class	SavedSearch		
In Job	true		
In Current Program	false	Not Current Program	
In Invisible Program	true	Invisible Program	
In Inactive Program	true	Inactive Program	
Is Program Member	false	Not Program Member	
Has Type	EngChange Revision		
In IC Context	true		

3. Click **Save** .

Edits to the AM rule tree persist for that session until changes are committed to the database using the **Save** command.

Delete an Access Manager rule

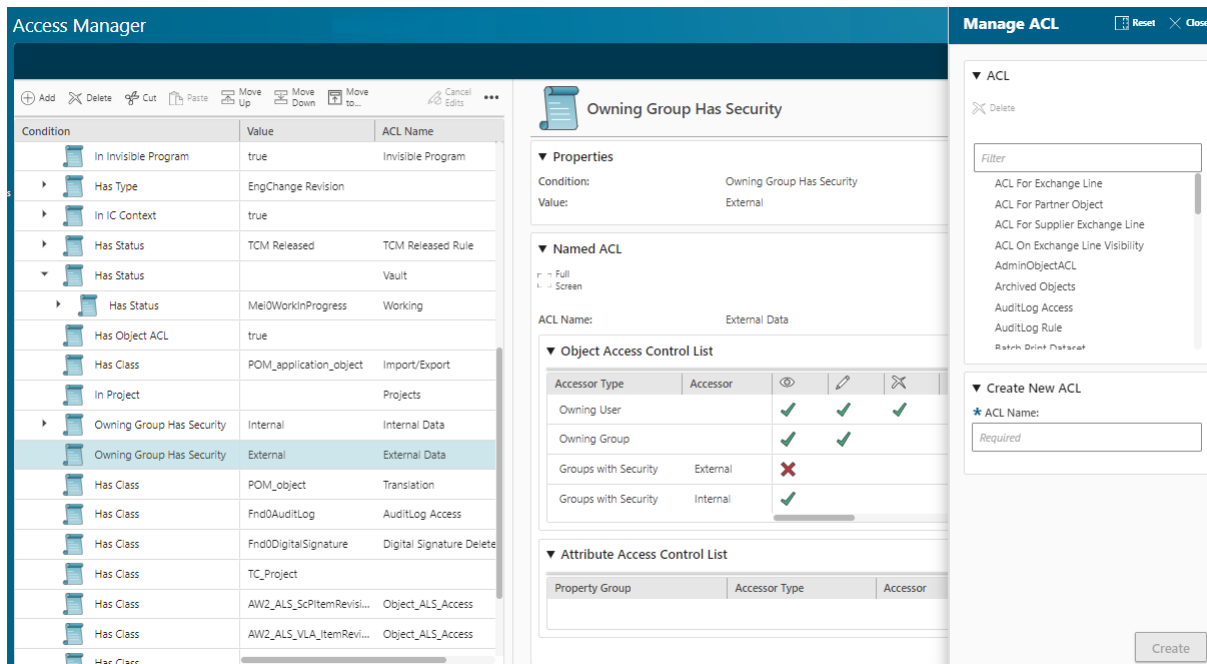
You can use the Access Manager location to delete a rule. However, some rules cannot be deleted.

1. From the AM rule tree, select either a single rule or multiple rules to delete.
2. Click **Delete** .

Managing access control lists

What are access control lists?

Access control lists (ACLs) contain a list of accessors and the privileges granted, denied, or not set for each accessor. Accessors are collections of users who share certain common traits, such as membership in the group that owns the object or membership in the project team. Just as rules have a precedence weighting in the rule tree, accessors have a precedence weighting in the ACL.



Create a new ACL

You can use the Access Manager location to add new access control lists (ACL).

1. Click **Manage ACL** to display the **Manage ACL** panel.
2. From the **Create New ACL** section, enter a unique ACL name in the **ACL Name** field.
3. Click **Create**.

You can confirm the new ACL name in the **ACL** section by filtering on the new name.

Edit an ACL

You can edit the conditions of an access control list (ACL).

1. Select the condition for the ACL you want to edit, for example, **In Project**.
2. From the condition information page, click **Edit** .

You can edit the values in the **Properties** and **Object Access Control List** sections.

3. Perform one or more of the following functions on an ACL.

Function	Icon	Description
Add		Add an access control entry.
Remove		Remove an access control entry.
Allow All		Allow access to all privileges for the selected row.
Deny All		Deny access to all privileges for the selected row.
Reset		Reset the selected row to the last saved values for the privileges.
Clear All		Clear all privileges of the selection.
Selection Mode		Turn on selection mode for making multiple accessor selections.

- Click **Apply** .

Delete an ACL

You can use Access Manager to delete access control lists (ACL). However, certain ACL names cannot be deleted.

- Click **Manage ACL**  to display the **Manage ACL** panel.
- From the **ACL** section, select the existing ACL to delete.
- Click **Delete** .

Importing and exporting Access Manager rules

You can use Access Manager to import and export your rule tree in Active Workspace. This helps you to easily:

- Distribute access rules to other Teamcenter sites.
- Back up your rule tree.
- Restore your local rule tree.

Import an Access Manager rule tree

You can import your Access Manager (AM) rule tree from within Access Manager. Your imported AM rule tree replaces the current one.

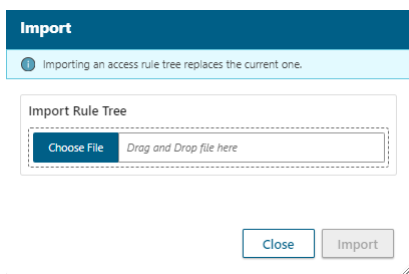
Prerequisites

Before importing a rule tree, you must ensure schema compatibility. To successfully load a new rule tree from a file, the importing site must have the same types, roles, and groups as those referenced in the rule tree file. If there is any incompatibility, the import operation ends and an error message appears. To resolve incompatibility, you can import your organization data and then load your rule tree.

Procedure

1. From the results panel toolbar, click **Import**  to display the **Import** dialog box.

This command is only enabled in non-edit mode.



2. Click **Choose File** to select the XML rule tree file that you want to import.
3. Click **Import**.

A progress indicator is displayed during import.

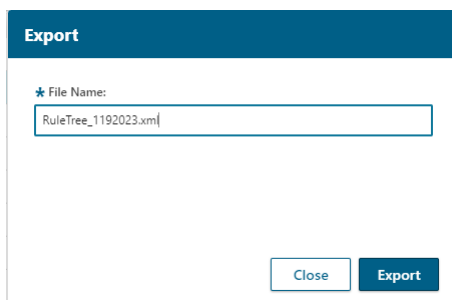
Export an Access Manager rule tree

You can export your Access Manager (AM) rule tree in XML format from within Access Manager.

Procedure

1. From the results panel toolbar, click **Export**  to display the **Export** dialog box.

This command is only enabled in non-edit mode.



2. By default, the rule tree is given a name based on the date. You can change the file name to one that fits your organization's file naming convention.

3. Click **Export**.

The file is downloaded based on your default browser settings, for example, to your **Downloads** folder.

5. Import, Export, and Report Administration Data

What is administration data?

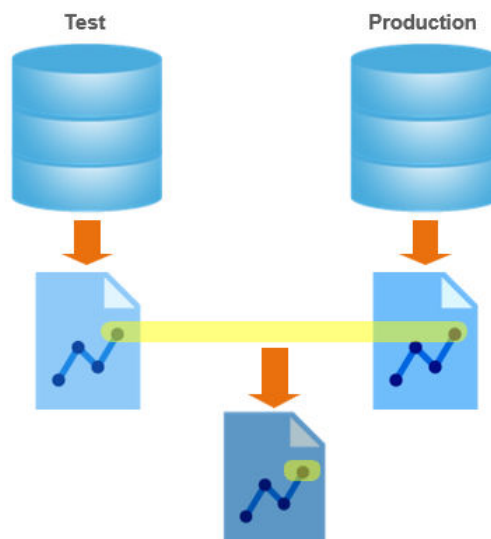
Administration data is data within the Teamcenter site database that defines:

- The behavior of the site.
- Who has access to information in the database.

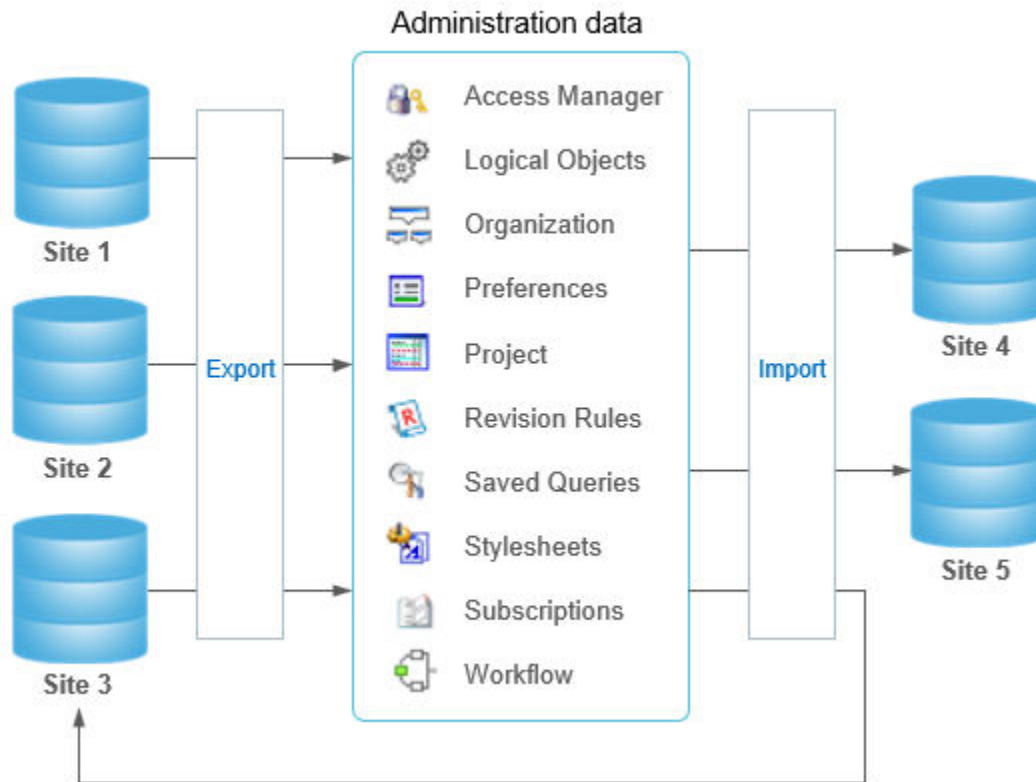
Administration data in the following categories is maintained using a client such as the rich client or Active Workspace. This data is not contained in Business Modeler IDE projects, nor in solution templates:

Access Manager	Preferences	Saved queries	Units of Measure
Logical Objects	Project	Stylesheets	Workflow
Organization	Revision Rules	Subscriptions	

You can use **administration data reporting** capabilities to generate interactive reports of these categories of administration data, which you can use to analyze and troubleshoot a site, and to compare differences in data between environments.



You can use **administration data export** and **administration data import** to copy and paste administration data across Teamcenter environments.



To safeguard functionality and data integrity in a production environment, separate environments are often created for purposes of development, training, patch testing, and troubleshooting. The ability to report and analyze administration data, and the ability to export and import administration data across environments, helps in maintaining these environments.

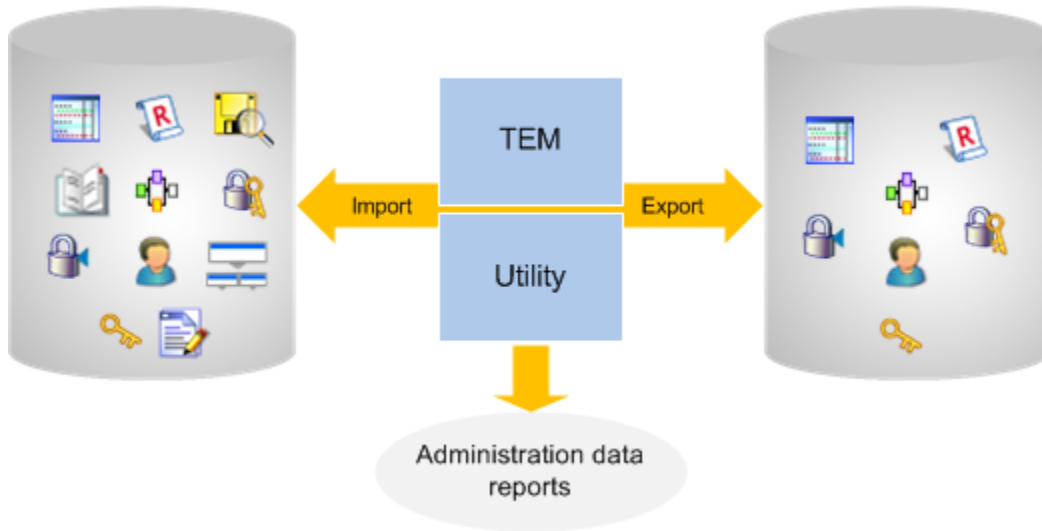
Administration data reporting capabilities have analogous data model element reporting capabilities accessible from within the Business Modeler IDE.

Tools for managing administration data

You can invoke tools for managing administration data from the command line and, to a lesser extent, in the Teamcenter Environment Manager (TEM).

Tool function	Command Line tool	TEM tool
Export administration data from the current site into an export package file.	<code>admin_data_export</code>	Manage Administration Data panel export options.
Import administration data from an administration data export package into the current Teamcenter site.	<code>admin_data_import</code>	Manage Administration Data panel import options.
Generate an interactive report of administration data contained in a site or	<code>generate_admin_data_report</code>	None.

Tool function	Command Line tool	TEM tool
in an administration data export package file.		
Generate an interactive report of differences between the administration data in an administration data export package file and either the current site or another package file.	generate_admin_data_compare_report	None.



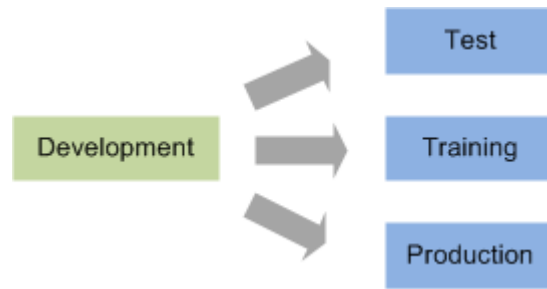
Example tasks and processes for managing site behavior

The following task examples illustrate how you can use **interactive administration data reports** along with administration data export/import utilities to manage your Teamcenter sites.

Task	General process
Analyze how a site is configured and troubleshoot configuration issues.	Generate an interactive report of administration data at the site and review the report in any web browser.
Ensure that one site is configured the same as another.	Export all administration data from one site and import it to another. This capability is useful when you must ensure certain administration data at one site exists at another site, such as a test site and a training site or during site consolidation.

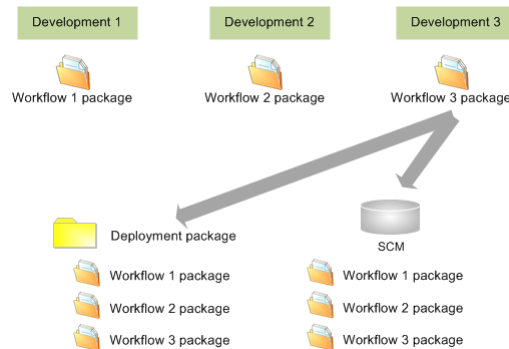
Task

General process



Spread workload for a project, such as adding or modifying multiple workflows, among teams.

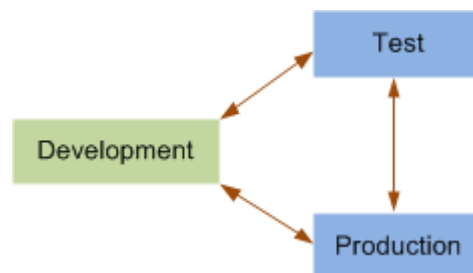
Set up several isolated sites for work group teams to work on particular parts of a development project. A team can make changes in its site and then export the changed (partial) administration data. You can then consolidate the changes by importing the administration data from the multiple partial export packages into one site, such as a sandbox site for final testing.



Determine the cause of differences in site behavior and troubleshoot configuration issues.

Generate a report that compares the administration data from a source site with the administration data in a target site. Then analyze the report of differences in administration data to determine the cause of differences in site behavior.

You can also use the report to provide information to an administrator at a different site.



Determine when a particular change was introduced to a site.

Review a site's **administration data import history** report to track impacts of importing administration data over time. This report is automatically updated upon each successful administration data import to the site.

Reconcile differences in administration data between Teamcenter sites

You may need to determine why two sites behave differently, or may want to ensure that the administration data is the same at both sites to allow consolidation of data. Use the following process to reconcile administration data between two sites.

1. **Export** the administration data from both sites.
2. Generate an **administration data comparison report**.

The report shows the areas that differ between the sites for each type of administration data.

3. View the comparison report and analyze the differences.
4. After you determine the administration data that requires modification at a target site, **import** the data from the source administration data package. Set the appropriate **merge options** during the import to ensure you get the only the desired data modification.

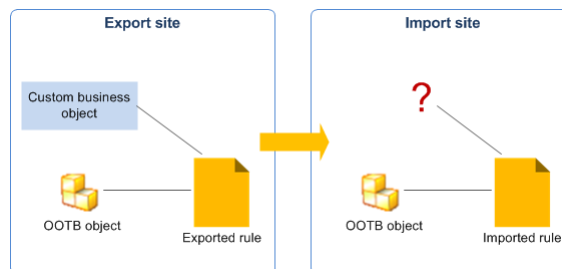
Caution:

When you export and import administration data between sites, it is important that the schemas for the administration data and any related objects involved in the transfer are the same at both sites.

Example:

Given a situation where:

- A custom business object type exists in the schema at an exporting (source) site, but not at the importing (target) site.
- A revision rule included in the administration data export refers to an instance of the custom business object.



Then:

- The administration data import succeeds, but errors occur during activities involving the revision rule due to the nonexistent custom business object.

Exporting administration data

Why would I export administration data?

Export administration data from a site into an administration data export package, which you can use when you want to:

- Configure a target site the same as the source site.
- Export administration data from a test site, so you can import it to a production site.
- Collect information on a site, so you can compare it to another site.
- Export partial administration data from multiple developers into a source code management (SCM) system for compilation into a comprehensive set of administration data.

You can use either the **admin_data_export** command line utility or the Teamcenter Environment Manager (TEM) to export administration data into an administration data package. Procedures in the TEM vary for **partial export** and **full export**.

Note:

It is important to be aware of the **considerations for partial export of administration data**.

Export administration data using the admin_data_export utility

You can use either the **admin_data_export** command line utility or the Teamcenter Environment Manager (TEM) to export administration data into an administration data package. Procedures in the TEM vary for **partial export** and **full export**.

Note:

It is important to be aware of the **considerations for partial export of administration data**.

Export administration data using the admin_data_export utility

At a Teamcenter command prompt, run the **admin_data_export** utility.

Tip:

When the export completes, you may want to use the **generate_admin_data_report** utility to generate a report of the site administration data contained in the package.

Export full administration data using TEM

You can use either the **admin_data_export command line utility** or the Teamcenter Environment Manager (TEM) to export administration data into an administration data package. Procedures in the TEM vary for **partial export** and **full export**.

Export full administration data using TEM

You can select the categories of data to export by using the **Full export** option in Teamcenter Environment Manager (TEM).

1. Start TEM.
 - a. In the **Maintenance** panel, select **Configuration Manager** and click **Next**.
 - b. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration** and click **Next**.
2. Select the configuration for the site where you want to export the administration data file and click **Next**.
3. In the **Feature Maintenance** panel, under **Teamcenter Foundation**, select the **Manage Administration Data** option and click **Next**.
4. In the **Manage Administration Data** panel, select the **Full export** option and click **Next**.
5. In the **Full Administration Data Export** panel, select the administration data package and the categories to export and click **Next**.
 - a. Click the ... button to the right of the **Administration Data Package** box to browse to the location where you want to place the administration data package.
 - b. Select the administration data categories to export and click **Next**.
6. In the **Teamcenter Administrative User** panel, enter the logon information for the Teamcenter administrative user account and click **Next**.
7. In the **Confirmation** panel, review the choices and click **Start**.

On successful export, TEM packages the exported administration data to the specified packaging directory. The export package contains the administration data in TC XML files that are used later for **importing into another Teamcenter environment**.

Tip:

When the export completes, you may want to use the **generate_admin_data_report** utility to generate a report of the site administration data contained in the package.

Export partial administration data using TEM

You can use either the **admin_data_export command line utility** or the Teamcenter Environment Manager (TEM) to export administration data into an administration data package. Procedures in the TEM vary for partial export and **full export**.

Note:

It is important to be aware of **Considerations for partial export of administration data**.

You can export specific administration data components using the **Partial export** option in TEM. You select the specific instances of administration data by category, class, and specific attribute/value criteria.

However, each administrative data type supports only specific elements in partial export, but not all. For example, Access Manager supports only **Named ACL** and **Privilege** elements to be exported partially, but not rule trees.

Export partial administration data using TEM

1. Start TEM.
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. Select the configuration for the site where you want to export the administration data file and click **Next**.
5. In the **Feature Maintenance** panel, under **Teamcenter Foundation** select **Manage Administration Data** and click **Next**.
6. In the **Manage Administration Data** panel, select **Partial export** and click **Next**.
7. Use the **Partial Administration Data Export** panel to specify the administration data to export.

- a. Next to the **Administration Data Package** box click the ... button and browse to the location where you want to place the administration data package.
- b. From the **Category Name** list, select the administration data category to export.
- c. From the **Export** list, select the administration data class to export.
- d. Edit the table contents to refine the administration data that is exported.
 - A. Type the **Attribute** and **Value** pair.

Depending on what you select from the **Export** list, the system may supply an entry for the **Attribute**.

- B. To add rows for additional attribute/value pairs, click **Add** and select the Boolean **Operator** to define how the attributes are added.

To delete an attribute/value pair, select a row from the table and click the **Remove** button.

- C. Click **Next**.

8. In the **Teamcenter Administrative User** panel, enter the logon information for the Teamcenter administrative user account.
9. In the **Confirmation** panel, review the choices and select **Start**.

On successful export, TEM packages the exported administration data to the specified packaging directory. The export package contains the administration data in TC XML files used later for **importing into another Teamcenter environment**.

Tip:

When the export completes, you may want to use the **generate_admin_data_report** utility to generate a report of the site administration data contained in the package.

Examples of partial export attributes and values using TEM

Following are some example attributes and values for partial administration data export using Teamcenter Environment Manager. Depending on your selection from the **Export** list, the system may supply an entry for **Attribute**. These are examples only. The attributes and values listed here may not match what you have in your system.

Category name	Export	Attribute (example)	Value (example)
Access Manager	Named ACL	ACL_Name	System
	Privilege	Privilege_Name	READ
Organization	Discipline	discipline_name	*
	Group	name	Engineering
	License Server	fnd0name	*
	Person	user_name	*
	Role	role_name	Designer
	Site	name	*
	Volume	user_name	*
	Discipline	volume_name	volume
	Discipline	volume_name	volume
Preferences	Preferences by Category	Category Name (supplied)	Change Management
	Preferences by Group	Group Name (supplied)	Engineering
	Preferences by Preference Name	Preference Name (supplied)	HiddenPerspectives
	Preferences by Role	Role Name (supplied)	Designer
	Preferences by User	User Name (supplied)	user0001
Project	Project	project_id	*
Revision Rules	Revision Rule	object_name	*
Saved Queries	Saved Query	query_name	*
Stylesheets	Stylesheets by Preference Name	Name (supplied)	*SUMMARYRENDERING
	Stylesheets by Stylesheet Name	Name (supplied)	Item*
Subscriptions	Subscription	subscriber	*
Workflow	Named ACL	ACL_Name	Vault
	Privilege	Privilege_Name	PROMOTE
	Workflow Templates by Name	Name (supplied)	Quick Release

Considerations for partial export of administration data

General considerations

- To export only part of an administration data category, you must specify input criteria that identify the data you want to export. Prepare your criteria before beginning the export.

- By default, partial exports do not include referenced user information. To understand some implications of not including referenced user information, read and understand the description for the **-session_options** argument option **opt_traverse_ref** in **admin_data_export** before performing the administration data transfer.
- With any partial export, the Access Manager privileges are always exported in order to maintain the data integrity.
- Partial export of an Organization tree is supported bottom-up and not top-down. For example, if you have the following tree and you are exporting the **group1** group partially, the complete tree may not be exported. But if you export the **user1** user partially, the complete tree is exported.

```
group1
  - role1
  - user1
```

Input criteria for a partial export: class name{attribute=value}

To export partial administration data, you must specify input criteria that identify the data you want to export. The specification includes:

- Administration data *class name*.

Note:

If you are specifying a partial export using TEM, the class name is understood by the system when you select an option from the **Export** list.

- The *real name* (not the display name) for an attribute of the named class.

Note:

If you are specifying a partial export using TEM, depending on your selection from the **Export** list, the system may supply the attribute name, in which case the display name is shown.

- The *value* of the named attribute.

What are the classes associated with administration data?

Not all categories of administration data are associated with classes, but many are. The following table lists associated classes for commonly partially-exported categories of administration data.

Administration data type	Associated class
Access Manager	AM_ACL
Organization	
Discipline	Discipline
Group	Group
License Server	Fnd0LicenseServer0
Person	Person
Role	Role
Site	POM_site_config
User	User
Volume	ImanVolume
Project	TC_Project
Revision Rules	RevisionRule
Saved Queries	SavedQueryCriteria
Subscriptions	ImanSubscription

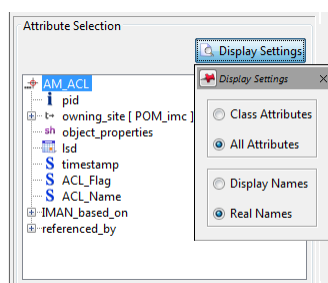
How can I find the real names of attributes?

Following are some tools you can use to look up the real names of attributes:

- Query Builder

Allows you to build queries on classes and their attributes.

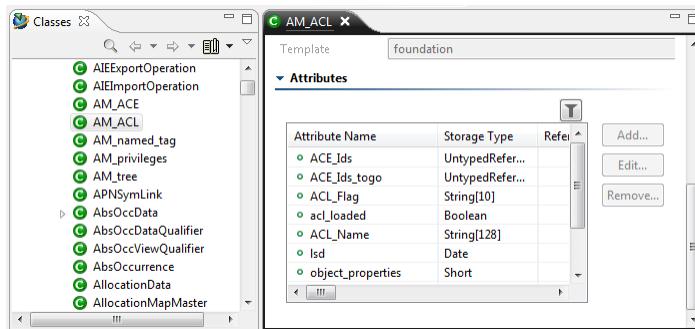
For example, following is a query on the Access Manager class (**ACM_ACL**) showing its attributes:



- Business Modeler IDE

Displays the Teamcenter classes and their attributes in the **Classes** view of the **Advanced** perspective. You can also see the corresponding business objects and their properties in the **Business Objects** view.

Following is a display of the Access Manager class (**ACM_ACL**) in the **Classes** view:



- **taxonomy utility**

Lists the classes in the Teamcenter system and their attributes. To output the list to a file, run the following from a Teamcenter command prompt:

```
taxonomy -f=classes_and_attributes.txt
```

Following is an excerpt of the output for the Access Manager class (**ACM_ACL**):

```
[ 1]<-->[AM_ACL] app [AM]
Class is NOT exportable.

POM class definition for: AM_ACL
  application name: AM
  Number of super classes: 1
    Super class : POM_object
  Number of Attributes: 5

    Attribute 1 Name : ACE_Ids
                  Type : POM_untyped_reference
...
    Attribute 2 Name : ACE_Ids_togo
                  Type : POM_untyped_reference
...
    Attribute 3 Name : ACL_Flag
                  Type : POM_string
...
    Attribute 4 Name : ACL_Name
                  Type : POM_string
...
    Attribute 5 Name : acl_loaded
                  Type : POM_logical
...
  Number of indexes : 0
  Class size in bytes: 171
```

How can I find attribute values?

Generate an **administration data document report**. Review the report to see all the administration data attributes and their values.

Note:

The report shows attribute display names, but not attribute real names.

Importing administration data

Why would I import administration data?

Import administration data previously exported from a source site into a target site when you want to:

- Configure the target site the same as the source site.
- Move administration data configured on a test environment into a production environment.

When importing administration data, you can use merge options to control whether the source or target takes precedence.

Before actually importing data, you can perform a dry run import, and then review the dry run report to analyze the potential impact and resolve any conflicts.

You can perform the dry run import, and perform the actual import, using either the **admin_data_import** utility or **using Teamcenter Environment Manager (TEM)**.

Perform a dry run import of administration data using TEM

Using an administration data package file that is the output of **exporting administration data** from a source site, you can perform a dry run import to a target site to see the impact before actually importing data. You can perform the dry run import **using the admin_data_import utility** or **using TEM**.

Perform the dry run data import using Teamcenter Environment Manager

1. Start TEM.
 - a. In the **Maintenance** panel, select **Configuration Manager** and click **Next**.
 - b. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration** and click **Next**.
2. Select the configuration for the site where you want to import the administration data file and click **Next**.

3. In the **Feature Maintenance** panel, select the **Manage Administration Data** option under **Teamcenter Foundation** and click **Next**.
4. In the **Manage Administration Data** panel, select the **Dry run import** option and click **Next**.
5. In the **Dry Run Import Administration Data** panel, click the ... button in the **Administration Data Package** box and select the administration data package. Click **Next**.
6. In the **Import Merge Options** panel, select the **merge option** for each category and click **Next**.
7. In the **Teamcenter Administrative User** panel, enter the logon information for the Teamcenter administrative user account and click **Next**.
8. In the **Confirmation** panel, review the choices and click **Start**.
9. In the **Install** panel, monitor the **Overall Progress** or select **Show Details** to see the install status.

View the Administration Data Import (dry run) report

To view a dry run report, after the run completes, browse to the following location:

`TC_ROOT\logs\admin_data_import_history\Import_date-and-time\index.html`

The following image shows the summary page of an example partial administration data import dry run report. In an actual report, you can click links to navigate through the report.

Import History

No	Imported Date	Impacts
1	Aug 29 2017 05:50 PM EDT from IMC--1501564889	7

Categories

Import Date: Aug 29 2017 05:50 PM EDT

Summary
[Report Details](#)
[Glossary](#)

Category	Impacts
Organization	7

Administration Data Import
IMC--1501564889 import into IMC--1501564889
Dry Run Summary
Import Date: Aug 29 2017 05:50 PM EDT

Impacts Outstanding

Organization

Merge Option: Override With Source
7 total Impacts

Element	Impacts	Element	Impacts
Organization	4	Persons	1
Groups	0	Sites	0
Roles	0	Volumes	0
Disciplines	0	License Servers	0
Users	2		

Copyright 2017 Siemens Product Lifecycle Management Software Inc. All Rights reserved.
This documentation is proprietary and confidential to Siemens Product Lifecycle Management Software Inc.

Perform a dry run import of administration data using the admin_data_import utility

Using an administration data package file that is the output of **exporting administration data** from a source site, you can perform a dry run import to a target site to see the impact before actually importing data. You can perform the dry run import **using the admin_data_import utility** or **using TEM**.

Perform the dry run data import using the admin_data_import utility

At a Teamcenter command prompt, run the **admin_data_import** command line utility with the **-dryrun** argument.

Note:

Use the **-mergeOption** argument to specify the **merge options** to set for each data type.

View the Administration Data Import (dry run) report

To view the dry run report, after the run completes, browse to the following location:

`TC_ROOT\logs\admin_data_import_history\Import_date-and-time\index.html`

The following image shows the summary page of an example partial administration data import dry run report. In an actual report, you can click links to navigate through the report.

Import History

No	Imported Date	Impacts
1	Aug 29 2017 05:50 PM EDT	7

Categories

Import Date: Aug 29 2017 05:50 PM EDT

[Summary](#)
[Report Details](#)
[Glossary](#)

Category Impacts

Category	Impacts
Organization	7

Administration Data Import

IMC--1501564889 import into IMC--1501564889

Dry Run Summary

Import Date : Aug 29 2017 05:50 PM EDT

Impacts Outstanding

Organization

Merge Option: Override With Source
7 total Impacts

Element	Impacts	Element	Impacts
Organization	4	Persons	1
Groups	0	Sites	0
Roles	0	Volumes	0
Disciplines	0	License Servers	0
Users	2		

Copyright 2017 Siemens Product Lifecycle Management Software Inc. All Rights reserved.
 This documentation is proprietary and confidential to Siemens Product Lifecycle Management Software Inc.

Import administration data using TEM

Using an administration data package file that is the output of exporting administration data from a source site, you can import administration data to a target site. You can perform the import using the **admin_data_import** utility or using TEM.

The type of administration data that you can import is determined by what the import package contains. If multiple categories of data are in the file, you can select the categories that you want to import.

1. Start Teamcenter Environment Manager (TEM).
 - a. In the **Maintenance** panel, select **Configuration Manager** and click **Next**.

- b. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration** and click **Next**.
2. Select the configuration for the site where you want to import the administration data file and click **Next**.
3. In the **Feature Maintenance** panel, select the **Manage Administration Data** option under **Teamcenter Foundation** and click **Next**.
4. In the **Manage Administration Data** panel, select the **Import** option and click **Next**.
5. In the **Import Administration Data** panel, click the ... button to the right of the **Administration Data Package** box to select the administration data package to import. Click **Next**.
6. In the **Import Merge Options** panel, select the **merge option** for each category and click **Next**.
7. In the **Teamcenter Administrative User** panel, enter the logon information for the Teamcenter administrative user account and click **Next**.
8. In the **Confirmation** panel, review the choices and select **Start**.
9. In the **Install** panel, monitor the **Overall Progress** or select **Show Details** to watch the install status.
10. After the import, access the results from the following location:

`TC_ROOT\logs\admin_data_import_history\Import_date-and-time\index.html`

Note:

If your imported administrative package contains volume objects, you must **postprocess the imported volume objects** for them to work properly.

Following is an example import report.

Import administration data using the `admin_data_import` utility

Using an administration data package file that is the output of exporting administration data from a source site, you can import administration data to a target site. You can perform the import using the `admin_data_import` utility or using TEM.

The type of administration data that you can import is determined by what the import package contains. You can specify the administration data categories that you want to import.

Note:

If the administration data contains volume objects, you must **postprocess the imported volume objects** for them to work properly.

1. To see the potential impact before actually importing data, perform a dry run import using either **using the `admin_data_import` utility** or **using TEM**. Review the dry run import report and resolve any conflicts.
2. At a Teamcenter command prompt, run the **`admin_data_import`** command line utility.

Use the **`-mergeOption`** argument to specify the **merge options** for each data type.

3. After the import, access the results from the following location:

```
TC_ROOT\logs\admin_data_import_history\Import_date-and-time\index.html
```


Note:

If your imported administrative package contains volume objects, you must **postprocess the imported volume objects** for them to work properly.

Following is an example import report.

The screenshot displays the 'Administration Data Import' report for IMC--1959296225 imported into IMC--1501564889. The report includes a summary of the import date (Aug 30 2017 03:15 PM EDT) and a table of impacts. The left sidebar shows a 'Categories' list with 'Access Manager' (1 impact), 'Organization' (3 impacts), and 'Draferanar' (0 impacts). The main content area shows details for 'Access Manager' and 'Organization'.

Category	Impacts
Access Manager	1
Organization	3
Draferanar	0

Element	Impacts
Rules	0
Named ACLs	1
Privileges	0

Element	Impacts
Organization	3

Postprocess imported volume objects

If your imported administrative package contains volume objects, the volume's host and FMS configuration must be updated with the importing site data.

1. Extract the **VolumeAttributeUpdateInput.xml** file from the export package.
2. Open the file and update the host name and host path with the importing site's host name and path.
3. Run the **tc_attribute_bulk_update** utility with the updated **VolumeAttributeUpdateInput.xml** file as the **-inputfile** argument value, for example:

```
tc_attribute_bulk_update -u=admin -p=password -g=dba
-inputfile=VolumeAttributeUpdateInput.xml -import
```

4. Run the **update_fms_configuration** utility to update the FSC configuration, for example:

```
update_fms_configuration -u=admin -p=password -g=dba
-volumes=volume-names-in-comma-separated-list -fsc_id=fsc-id
```

Administration data import merge options

When you import administration data, for each category of administration data you must specify how to handle data conflicts between the target (importing) site administration data and the source (exporting) site data in the input package.

Following are the available merge options:

Override With Source

Overrides target objects with source objects. If an object exists in both environments, the attributes on the object are taken from the source environment and updated on the object in the target environment during the import. If an object in the source environment does not exist in the target environment, it is imported to the target environment.

To avoid unexpected behavior that could result from mixing the rule tree contents, this is the only option available for Access Manager administration data.

Keep Target

Keeps the target site's administration data for all conflicts. If an object exists in both environments, the object in the target environment is kept as-is. If an object in the source environment does not exist in the target environment, it is imported to the target environment.

Choose Latest

Chooses objects that were last saved (based on the **Last Saved Date** property value). If an object exists in both environments and the one in the source environment was saved last, the attributes on the object in the source environment are used to update the attributes of the object in the target environment during the import. If the object in the target environment was saved last, the attributes on the object in the source environment are not imported. If an object in the source environment does not exist in the target environment, it is imported to the target environment.

Choose Source for Unresolvable Conflicts

Chooses objects from the source environment if they were edited in the source environment and not in the target environment since the last import. If an object exists in both environments, the attributes on the object in the source environment are used to update the attributes on the object in the target environment during the import. If an object was edited only in the target environment and not the source environment since the last import, the object is left as-is at the time of import. If an object in the source environment does not exist in the target environment, it is imported to the target environment.

This option is not available for workflow administration data.

Choose Target for Unresolvable Conflicts

Chooses objects from the target environment if they were edited in the target environment and not in the source environment since the last import. If an object was edited only in the target environment and not the source environment since the last import, the object is left as-is at the time of import. If an object in the source environment does not exist in the target environment, it is imported to the target environment.

This option is not available for workflow administration data.

How you set import merge options depends on which method you use to perform the import, either with the `admin_data_import` utility or with the Teamcenter Environment Manager (TEM):

- **admin_data_import** utility

Use the **-MergeOptions** argument to specify the merge option to use with each administration data category.

- **Import Merge Options** panel in TEM

When you perform a **dry run import** or a regular **import** of administration data, the **Import Merge Options** panel is displayed. Select the merge option you want to use with each data category.

Tip:

The default values on this panel are the actions that Siemens Digital Industries Software recommends you use in most cases to avoid unintentional changes.

Style sheet import behavior

By default, each style sheet has two preferences set. For example, the summary style sheet has the following:

```
type-name . SUMMARYRENDERING=dataset-name
dataset-name . SUMMARY_REGISTEREDTO=type-name
```

If no preferences or only one preference is defined on a style sheet, then the style sheet is not in use. During import, when the target database is missing one or both preferences for a style sheet, the import is processed as if both preferences are defined in the target database.

The following tables show details of the import actions taken for the two preferences depending on whether the preferences exist. The last column shows the target action when you select the import option **Choose Source** or **Choose Source For Unresolved Conflicts**.

Preference import actions for Item dataset cases

Source preferences	Target preferences	Action	Target preferences after import	Target action with source option ¹
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	First time import, dataset does not exist.	Added at source site.	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Add at target site using import package data.
Dataset does not exist	Item. SUMMARYRENDERING	Added at target site.	No change.	None.

¹ When you select either the **Choose Source** option or **Choose Source for Unresolved Conflicts** option.

Source preferences	Target preferences	Action	Target preferences after import	Target action with source option ²
	ItemSummary. SUMMARY_REGISTEREDTO			
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	First time created at source. First time created at target not by importing.	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Update using source package.
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Change XML file content at source.	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Update.
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Change XML file content at source. Change XML file content at target.	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Update using source package.
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Change XML file content at source. Delete at target.	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Add using source package.
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO		Delete at source.		None.
	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Delete at target.		None.
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Delete at source. Delete at target.	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	None.
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Delete at source. Change at target.	Depends on SyncDelete option.	SyncDelete ON: Delete OFF: Keep target.
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Delete at source.	Depends on SyncDelete option.	SyncDelete ON: Delete OFF: Keep target.

² When you select either the **Choose Source** option or **Choose Source for Unresolved Conflicts** option.

Source preferences	Target preferences	Action	Target preferences after import	Target action with source option ³
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Added at source from source 2. Imported at target from source 1.	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Add using source 2.
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Change at target.		- override_with_source option, update using source package. - choose_source_for_unresolved_conflicts option, keep target.
Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Delete at target.	Item. SUMMARYRENDERING ItemSummary. SUMMARY_REGISTEREDTO	Add using source package.

Preference import actions for Item Revision and Document Revision datasets cases

Source preferences	Target preferences	Action	Target preferences after import	Target action with a source option ⁴
ItemRevision. SUMMARYRENDERING	ItemRevision. SUMMARYRENDERING	Add at source.	ItemRevision. SUMMARYRENDERING	Add at target site using import package data.
ItemRevisionSummary. SUMMARY_REGISTEREDTO	ItemRevisionSummary. SUMMARY_REGISTEREDTO	Add at target.	ItemRevisionSummary. SUMMARY_REGISTEREDTO	DocumentRevision has it own stylesheet.
DocumentRevision. SUMMARYRENDERING	(DocumentRevision inherits ItemRevision's summary style sheet.)		DocumentRevision. SUMMARYRENDERING	
DocumentRevisionSummary. SUMMARY_REGISTEREDTO			DocumentRevisionSummary. SUMMARY_REGISTEREDTO	

Modifying an administration data package

Normally, you do not need to alter the content of the administration data TC XML file. However, after you export an administration data file, you can carefully modify the TC XML content of the file to provide changes for specific sites or groups of sites.

Note:

Because mistakes in the TC XML file can cause corruption and possibly data loss during import, import of a modified file requires an authorization key. After passing the *Manage Administration*

³ When you select either the **Choose Source** option or **Choose Source for Unresolved Conflicts** option.
⁴ **Choose Source** option or **Choose Source for Unresolved Conflicts** option.

Data self-paced course available from Learning Advantage, you can file a support case on Support Center to obtain a key. The course provides the information necessary to avoid potential problems when editing the TC XML content of an administration data file.

Following are rules for editing an administration data TC XML file:

- Ensure the uniqueness of **elemId** attributes across the TC XML file.
- Ensure the value of Boolean type attributes are **Y** (true) or **N** (false).
- Ensure the value of reference type attributes are formatted as:

#<elemId-of-the-GSIdentity-of-the-referenced-object>

- Ensure the value of date type attributes are in GMT format:

year-month-dayHour:minute:SecondZ

For example, for 09 September 2014 at 4:21:04 AM (GMT), **2014-09-23T04:21:04Z**.

- Ensure the timestamp value is present.
- Ensure L10N attributes are list of values formatted as:

locale:M/S:M/A/P/R/I:sequence:[aA-zZ]^32:translation

A locale in the format of **en_US: M/S** is a master or secondary language.

M/A/P/R/I is the translation status.

The sequence number is 4 digits for multiple value attributes.

There are 32 characters reserved.

The rest is the translation, for example:

```
de_DE:S:A:0000::Proxy-Link,
en_US:S:A:0000::Proxy Link,
es_ES:S:A:0000::Enlace proxy,
fr_FR:S:A:0000::Lien proxy,
it_IT:S:A:0000::Collegamento proxy,
pt_BR:S:A:0000::Link de proxy
```

- When editing the attributes that are part of the candidate key, the **GSIdentity** label of the object, and the label of the objects that reference the object, must be updated accordingly.

- Ensure all required attributes for the administration object have values.
- For out-of-the-box administration objects, a user's **home**, **mailbox_folder**, and **newstuff_folder** attributes must not be added. Also, a dataset **rev_chain_anchor** attribute must not be added. This is true even though these are required attributes.
- After editing the file, validate the complete file content against **LLTCXML.xsd** file (low-level TC XML schema) located in the **TC_DATA** directory to ensure the XML is valid.

Import a modified administration data package

Because mistakes in the TC XML file can cause corruption and possibly data loss during import, import of a modified file requires an authorization key. You can obtain the key from Support Center after passing the *Manage Administration Data* self-paced course available from Learning Advantage. The course provides the information necessary to avoid potential problems when editing the TC XML content of an administration data file.

1. In your Teamcenter environment, set the **BULK_LOADER_AUTH_KEY** environment variable to the value of the authorization key.

This enables the bulk load capabilities that are used to import the modified package. If you do not set this variable to a valid value, TEM returns an error message when you attempt to import the modified package.

2. When importing the modified package in Teamcenter Environment Manager (TEM), navigate to the **Manage Administration Data** page and select the **Ignore package validation** ☒ check box.

This enables bulk load functionality that bypasses certain validation checks.

Reporting administration data

What administration data reports are available?

Documentation, Comparison, Import (dry run), and Import (history) administration data reports show detailed Teamcenter administration data for the data categories that are specified when the report is generated. If an object is referenced by other objects, reports include a where-used table that indicates the categories and objects that have references to the object. You can use any web browser to view and interactively navigate through the reports.

Administration data report types

Documentation

Reports administration data for the current Teamcenter environment or for an administration data export package.

For use in effectively administering your environment, generate a documentation report of your environment administration data.

Example *Teamcenter [version] Administration Data Documentation* reports of a default installation of Teamcenter foundation are available on Siemens PLM Support Center under the *Programming and Customization* category. These example reports provide a reference without having to first install Teamcenter. However, they do not contain an accurate representation of your individual environment.

Comparison

Reports differences between administration data in two Teamcenter environments.

Import (dry run)

Reports changes in administration data that will occur due to importing an administration data package to the current Teamcenter environment. Use a dry run import report to assess the effects of importing administration data without changing the existing data.

Import (history)

Reports the administration data that was impacted by an import to the current Teamcenter environment at a particular point in time. A separate report is generated for each import. The report file contains the site name and a timestamp. You can use these reports to determine exactly what and when changes occurred that may be affecting behavior in the environment.

Administration Data Documentation report

You can use interactive Administration Data Documentation reports to:

- Troubleshoot issues with configuration.
- Periodically review a production environment to ensure it complies with business objectives.
- Capture a snapshot of the configuration at a point in time for archiving or audit reviews.
- Train administrators.

Generate an Administration Data Report

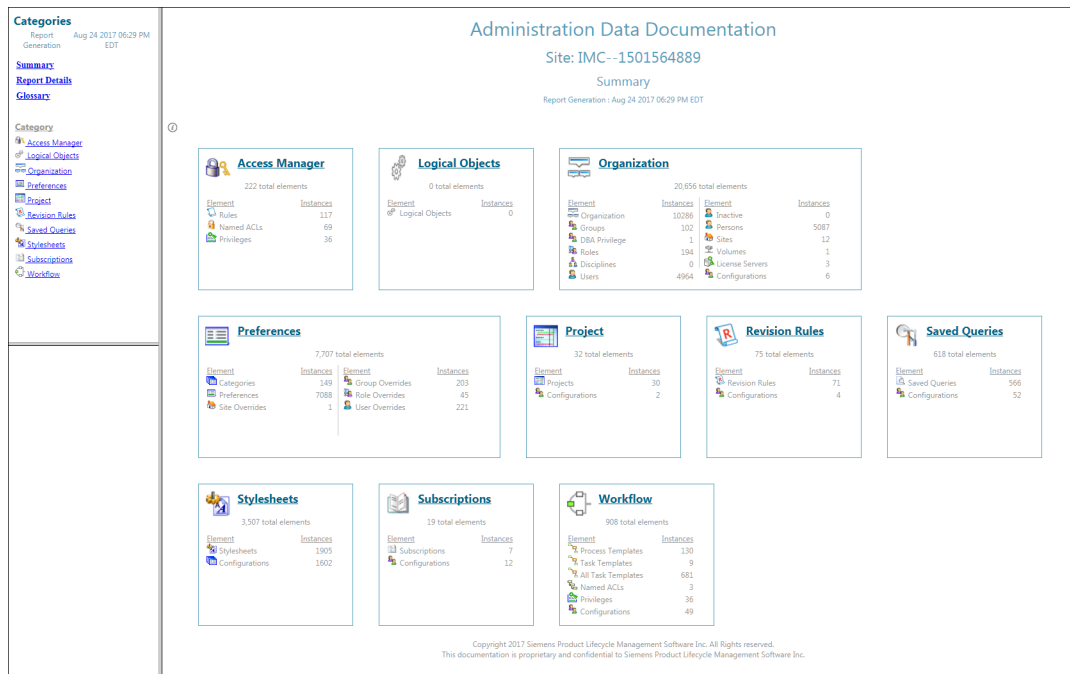
Use the **generate_admin_data_report** utility to generate a report of site administration data.

- You can run the report for the current site or for an exported administration data package.
- You can report all Administration Data categories or only selected categories.

View an Administration Data Report

To view a generated report, browse to the output location and open the *index.html* file.

The following image shows the summary page of an example report. In an actual report, you can click links to navigate through the report to find summaries and details of administration data.



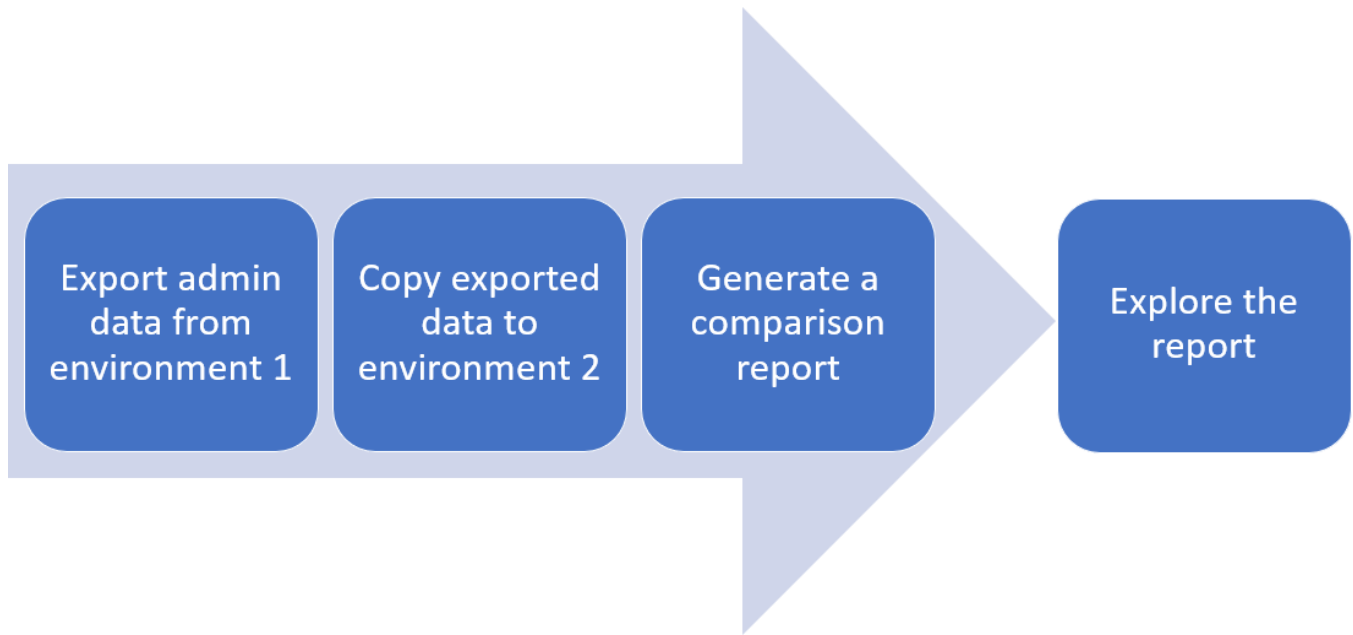
Administration Data Comparison report

You can use Administration Data Comparison reports to:

- Troubleshoot differences in business logic behavior by reviewing configuration differences between two environments.
- Determine whether a new environment is configured the same as a reference environment.
- Prepare for site consolidation by comparing environments to see what is common and what is different.
- Identify differences between a customer environment and an out-of-the-box Teamcenter environment.
- Compare a baseline export of an environment to a later export of the same environment.

Generate an Administration Data Comparison report

To make the comparison, the utility requires an administration data export package from one environment as a source, and either the local environment or another administration data export package as a target. The report highlights differences between the source data and the target data.



If an object is referenced by other objects, the report includes a where-used table that indicates the categories and objects that have references to a selected object in both source and target environments.

The report summary page shows all the administration data types included in the comparison and the number of differences for each element present within the category. The report includes a glossary of administration data categories and the elements available in each of the categories.

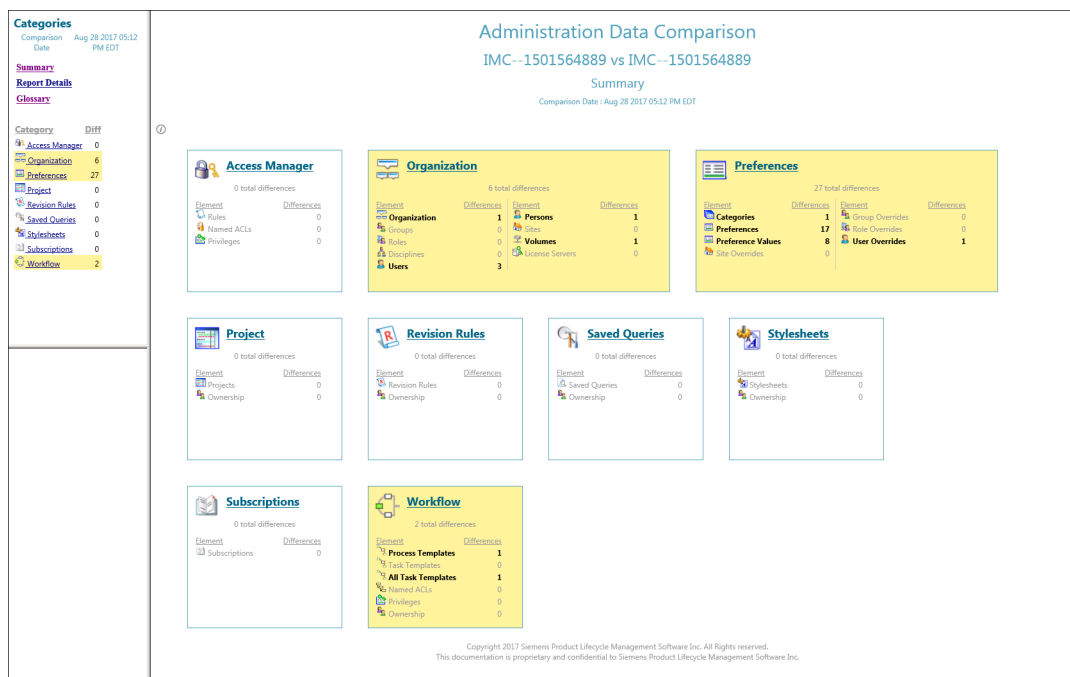
Use the **generate_admin_data_compare_report** utility to generate a report comparing a source environment administration data export package file to either:

- The current environment
- Another administration data export package file.

View an Administration Data Comparison report

To view a generated report, browse to the output location and open the *index.html* file.

The following image shows the summary page of an example comparison report. Differences are highlighted in yellow. In a real report, you can click links to navigate through the report to find summaries and details of administration data.



Example:

As you navigate to details of the report, you can easily identify even very small differences.

Permission difference in Access Manager.

ACE = World		
Accessor Type	World	World
Accessor		
READ	✓ Grant	✓ Grant
WRITE	✓ Grant	✓ Grant
DELETE	✗ Revoke	✓ Grant
CHANGE	✗ Revoke	✗ Revoke

Label difference in a Style Sheet.

14	<rendering>	14	<rendering>
15	<views>properties</views>	15	<views>properties</views>
16		16	
17	<page title="Item Information" titleKey="ItemInformation">	17	<page title="Item Information" titleKey="ItemInformation">
18	<view name="properties">	18	<view name="properties">
19	<label text="This is Item Create" />	19	<label text="This is Item Creation" />
20	<property name="item_id" />	20	<property name="item_id" />
21	<property name="revision:item_revision_id" />	21	<property name="revision:item_revision_id" />
22	<property name="object_name" />	22	<property name="object_name" />
23	<property name="object_desc" />	23	<property name="object_desc" />

Administration Data Import (dry run) report

You can perform a dry run import to a target site in order to generate a report of the potential impact before performing the actual import.

Generate an Administration Data Import (dry run) report

Perform a dry run import using either

- The **admin_data_import** utility with the **-dryrun** argument
- Or the **Dry run import option in Teamcenter Environment Manager (TEM)**.

View an Administration Data Import (dry run) report

To view a dry run report, after the run completes, browse to the following location:

```
TC_ROOT\logs\admin_data_import_history\Import_date-and-time\index.html
```

The following image shows the summary page of an example partial administration data import dry run report. In a real report, you can click links to navigate through the report to find summaries and details of administration data.

Import History

No	Imported Date	Impacts
1	Aug 29 2017 05:50 PM EDT from IMC--1501564889	7

Categories

Import Date: Aug 29 2017 05:50 PM EDT

[Summary](#)
[Report Details](#)
[Glossary](#)

Category	Impacts
Organization	7

Administration Data Import

IMC--1501564889 import into IMC--1501564889

Dry Run Summary

Import Date : Aug 29 2017 05:50 PM EDT

Impacts Outstanding

Organization

Merge Option: Override With Source
7 total Impacts

Element	Impacts	Element	Impacts
Organization	4	Persons	1
Groups	0	Sites	0
Roles	0	Volumes	0
Disciplines	0	License Servers	0
Users	2		

Copyright 2017 Siemens Product Lifecycle Management Software Inc. All Rights reserved.
This documentation is proprietary and confidential to Siemens Product Lifecycle Management Software Inc.

Administration Data Import (history) report

You can use the interactive Administration Data Import (history) report for a site to:

- Track impacts of importing administration data.
- Determine when a particular change was introduced to a site.

Generate an Administration Data Import (history) report

An Administration Data Import (history) report is automatically generated after each successful import of administration data. Imports can be performed using the **admin_data_import** command line utility or Teamcenter Environment Manager (TEM).

View an Administration Data Import (history) report

To view an Administration Data Import (history) report, browse to:

`TC_ROOT\logs\admin_data_import_history\index.html`

The following image shows the summary page for an import report selected from the three available historic reports. For this import, the only imported administration data category was **Organization**. Impacts are highlighted in blue. In a real report, you can click links to navigate through the report to find summaries and details of administration data.

Import History

No Imported	Impacts
Date	
1 Aug 29 2017 05:50 PM EDT	7
2 Aug 30 2017 03:15 PM EDT	7
3 Aug 30 2017 06:15 PM EDT	0

Categories

Import Date: Aug 30 2017 03:15 PM EDT

[Summary](#)
[Report Details](#)
[Glossary](#)

Category	Impa
Organization	

Administration Data Import

IMC--1501564889 import into IMC--1501564889

Summary

Import Date : Aug 30 2017 03:15 PM EDT

Impacts Outstanding

Organization

Merge Option: Override With Source
7 total Impacts

Element	Impacts	Element	Impacts
Organization	4	Persons	1
Groups	0	Sites	0
Roles	0	Volumes	0
Disciplines	0	License Servers	0
Users	2		

Copyright 2017 Siemens Product Lifecycle Management Software Inc. All Rights reserved.
This documentation is proprietary and confidential to Siemens Product Lifecycle Management Software Inc.

6. Relations configuration

Relations configuration tasks

What are relations?

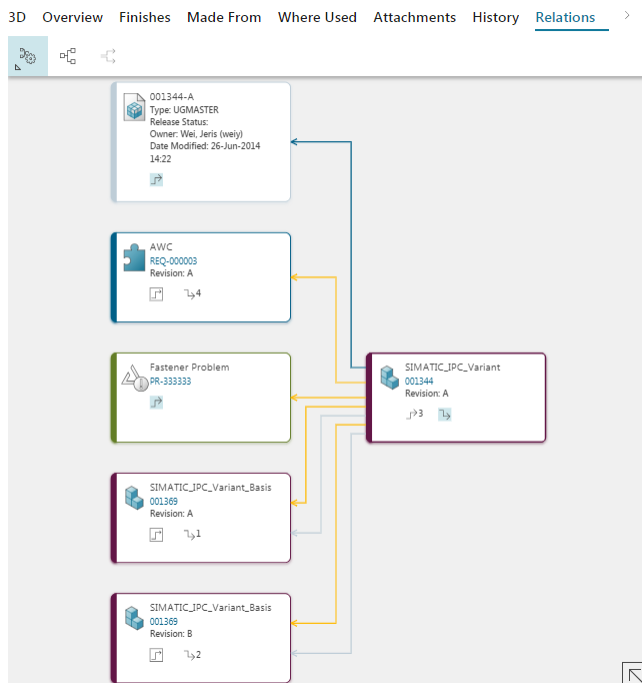
Relations are the associations between two Teamcenter objects. The **Relations** tab in Active Workspace allows you to see all the relationships for a selected object.

Why configure relations?

You may want to change how relations are displayed to end users or change which relations are displayed.

What do relations look like?

Following is an example of the **Relations** tab in Active Workspace.



How can I make changes?

You can configure aspects of relations' appearance and behavior using configuration files. These are stored in the Teamcenter database as datasets, which are located using the following preferences:

- **RV1_DARB_UI_configuration_file_name**

- **RV1_DARB_GraphStyle_file_name**
- **RV1_DARB_PresentationRule_file_name**

1. Find the name of the dataset by getting the value of the config file preference.
2. Find the dataset by name using the advanced search functionality.
3. Make the necessary edits to the configuration file.
4. Save the changes.

Example:

In this example, the **RV1_DARB_UI_configuration_file_name** preference value is **RelationBrowserConf**.

▼ DEFINITION	▼ VALUES
Name:	RV1_DARB_UI_configuration_file_name
Product Area: *	DecisionApps.RelationBrowser
Description: *	Registers the name of dataset which stores an XML file containing the Relation Browser UI configuration.

The **RelationBrowserConf** dataset is located using the **Dataset Name** query.

Advanced Search

Close

Quick

Advanced

Preferred Search...

▼

Clear All

Dataset Name:

List with Summary

Advanced Search

Save Search

☰

1010
0110

RelationBrowserConf

Type: Relations Configuration File

Owner: infodba (infodba)

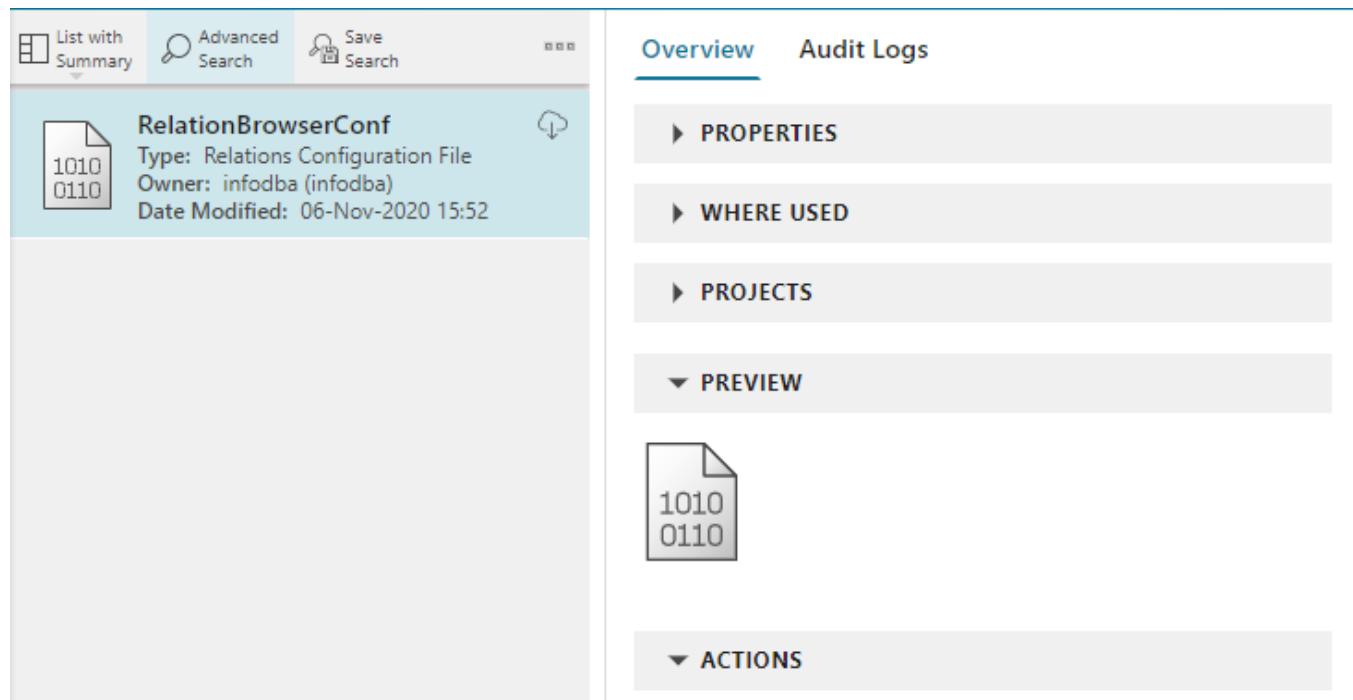
Date Modified: 06-Nov-2020 15:52

How can I edit the configuration files?

To edit the configuration files, you can:

- Download the file, edit it in a text editor, and then upload it back into Active Workspace.
- Edit them in place using the universal viewer.

OOTB, the universal viewer may not recognize the relation configuration file dataset type (**Rv1XML**) as an editable file type and it will only show the generic dataset type icon in the **PREVIEW** section.



To tell the universal viewer how to open this file type, add **Rv1XML.Awp0CodeViewer=Rv1XMLFile** to the preference.

This is because the **Relations Configuration File** dataset type's database name is **Rv1XML** and the named reference for the XML file is **Rv1XMLFile**.

▼ DEFINITION	▼ VALUES
Name: AWC_defaultViewerConfig.VIEWERCONFIG Product Area: * Active Workspace Description: * Defines the default viewer configuration for Active Workspace. Protection Scope: *	Values: SEARCHORDER.DocumentRevision=TC_Attaches,IMAN_ SEARCHORDER.Part Revision=TC_Attaches,IMAN_manif SEARCHORDER.ItemRevision=IMAN_manifestation,Sim SEARCHORDER.Signoff=root_target_attachments Rv1XML.Awp0CodeViewer=Rv1XMLFile

Now the universal viewer knows to open the configuration files with its code viewer.

The screenshot displays the Teamcenter configuration interface. On the left, a file named 'RelationBrowserConf' is shown with a document icon and a download icon. The file details are: Type: Relations Configuration File, Owner: infodba (infodba), and Date Modified: 06-Nov-2020 15:52. On the right, the 'PREVIEW' tab is active, showing the XML content of the file. The XML starts with a declaration and a comment indicating it is a configuration file for the RelationBrowser. The content includes a copyright notice for Siemens and a configuration section for the 'General' view.

```

1 <?xml version="1.0" encoding="UTF-8" standal
2 <!--
3 =====
4 Copyright 2014.
5 Siemens Product Lifecycle Management Softwar
6 All Rights Reserved.
7 =====
8     Filename: RB_UIConfigure.xml
9
10    File used to configure control panel of
11 =====
12 -->
13 <RelationBrowser>
14     <OptionPanel>
15         <view name="General">

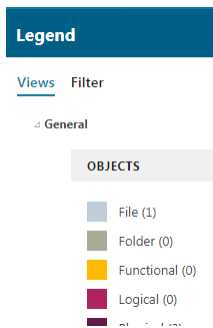
```

What other preferences control the relation browser?

There are many preferences that can help you configure relations for your users. Their names begin with **Rv1_** to make them easy to find. Inspect each of these preferences and update their content if necessary.

Creating new views or updating existing views

A view is a group of objects and relations in the **Legend** panel.



The **RV1_DARB_RelationBrowserViews.Relations** preference lists the available views. You can update this preference to add new views or update existing views.

The first view on this list is the default view. This preference is overridden by any other preference with an object type name appended.

If you append the name of an object type to the preference, the list of available views changes for that class of objects and all of its children. You can create as many of these preferences as you need. For example, the **RV1_DARB_RelationBrowserViews.Relations.ChangeItemRevision** preference controls the list of views (and the default) for change revisions.

Example of configuring relations expansion

You can configure how the system expands the relations forward and backward, and also how many levels. For example, when users open an object in the **Relations** tab, the system may automatically expand the relations forward one level. If your users are not interested in confirming the data attached by outgoing relations but rather confirming incoming relations, you can set the expansion to automatically expand backward. Set the expansion of relations with the following preferences.

- **RV1_DARB_Expand_All_1_Level_Command_Visible**
- **RV1_DARB_Expand_Incoming_Levels**
- **RV1_DARB_Expand_Outcoming_Levels**
- **RV1_DARB_Expand_Selected_1_Level_Command_Visible**

Localize names that appear in a custom Relations view

No additional action is required to localize the names that appear in a custom relations view. If you localize the Active Workspace client in the declarative definition (**messages.json** file) and the Teamcenter TextServer, names in the **Relations** view are localized.

The file name for relation browser keys in the TextServer is **relationshipviewer_text_locale.xml**. Relation browser uses a **k_relations_keyword_** prefix to help promote unique keys throughout the product.

Relation browser config file syntax

Appearance configuration limitations

The appearance configurations are limited to the color and the edge style (relation), the style of the nodes is not configurable.

RV1_DARB_GraphStyle_file_name

The configuration file value stored in this preference uses the following options:

- **PresentationRule**

Defines rules for one type of object. If the object satisfies the conditions specified by the rule, the style will be applied to the object.

This element has the following subelements:

- **type**

Specifies the type of object: **node** for object and **edge** for relation.

Note:

Styling is not supported for **type node**. For the **node** option, set the color from the configuration defined in **RV1_DARB_UI_configuration_file_name** section as shown in this topic.

- **port**

Specifies the port for relation.

- **styleID**

Specifies what style to apply. **styleID** must be the same as defined in the GraphStyle XML file. (The name of the GraphStyle XML file is defined in the **RV1_DARB_GraphStyle_file_name** preference).

- **Conditions**

Specifies a condition value that must be satisfied before the style is applied to the object. The condition value is a logical operator.

This element has the following subelements:

- **Property**

Only used when the **PresentationRule** type is **edge**. **Name** specifies the relation type. **Value** specifies the relation name.

For **Property** elements, only the **equalTo** operator is supported.

■ Type

Only used when **PresentationRule** type is **node**. **Value** specifies the name of the object.

Note:

Styling is not supported for **type node**. For the **node** option, set the color from the configuration defined in **RV1_DARB_UI_configuration_file_name** section as shown in this topic.

■ Conditions Operator

Specifies what conditions must be met before the style is applied. Valid operators are: **and** (all rules must be met) and **or** (at least one of the rules must be met).

Valid operators for **Type** are **equalTo** and **isTypeOf**. When property is **edge**, only the **equalTo** operator is valid.

RV1_DARB_UI_configuration_file_name

The configuration file value stored in this preference uses the following options:

- **view name**

Specifies the name of the view, for example, **General**.

- **ruleName**

Specifies the name of the dataset that implements the traversal logic of the view. The **GenericRule** dataset is the default rule file.

- **group name**

Specifies if this is a list containing objects or relations. These lists are presented in the **Relation Controls** section of the **Relations** browser.

The order of the **objects** filters is important. Your object is displayed using the first filter on the list that successfully matches. The last filter in the **objects** section, **Other**, matches all objects because it is associated with **WorkspaceObject**. It is there to ensure all objects receive some formatting. Do not place any filters after the **Other** filter.

- **filter name**

Specifies the name of the relation or object name.

For example:

```
<filter name="Attach" parameterSet="Attach" color="(64,100,142)"/>  
<filter name="File" parameterSet="Dataset" color="(202,216,234)"/>
```

- **parameterSet**

Specifies the name of the **parameterSet** element in this configuration file. The **parameterSet** element defines the Teamcenter objects and relations that map to the filter name.

- **color**

Specifies the display color of the object or relation.

7. Data sharing

Configuring and managing data sharing

Active Workspace offers users and administrators several ways to share Teamcenter data. Each of these methods of sharing data has capabilities an administrator can manage.

- Briefcase lets users share Teamcenter data with connected and unconnected sites using TC XML and Briefcase files.
- Multi-Site Collaboration lets users share and synchronize data in near real-time between Teamcenter sites across the entire enterprise.
- PLM XML lets users share Teamcenter data with third-party applications using the open PLM XML format.
- Bulk data migration tools let administrators exchange large volumes of data when populating test environments.
- Advanced Multi-Schema Exchanger lets user with Active Architect privileges create mapping rules that transform data when it is shared between Teamcenter sites that use different schemas.

Briefcase

Briefcase (on Windows or Linux), TC XML Import and Export, and Teamcenter Dispatcher Server (on Windows or Linux) must be installed to access Briefcase features using Active Workspace

Each site that is importing or exporting Teamcenter data must have a Briefcase license. Each site using the following optional import and export features must each also have an Advanced Multi-Schema Exchanger license:

- Briefcase export and import dry run and validation capabilities
- Briefcase preview and compare capabilities
- Advanced Multi-Schema Exchanger

Contact your Siemens Digital Industries Software customer service representative for more information on Briefcase licenses.

See [Configure Briefcase file sharing](#) to configure Briefcase capabilities.

Multi-Site Collaboration

Multi-Site Integration (on Windows or Linux) must be installed to access Multi-Site Collaboration features using Active Workspace. Collaborating sites also need to be licensed to use Multi-Site Collaboration.

Configure and monitor Multi-Site Collaboration status.

- **Control the scope of remote Multi-Site Collaboration objects checked out and checked in by users.**
- **Configure** and **use Multi-Site Dashboard** to monitor the status of your Multi-Site federation.

PLM XML

PLMXML Export Import (on Windows or Linux), TC XML Import and Export, and Teamcenter Dispatcher Server (on Windows or Linux) must be installed to access PLM XML features using Active Workspace.

See **Configure PLM XML data sharing** to specify the list of PLM XML object types that can be exported and imported at your site.

PDX

The following features must be installed to configure and access PDX export features using Active Workspace:

- PDX Export (on Windows or Linux)
- Teamcenter Dispatcher Server (on Windows or Linux)
- TC XML Import and Export
- Vendor Management. This feature provides the **TIEPDXOptionSetDefault** that is required to export PDX files.

See **Configure PDX data sharing** to configure the PDX exporting capabilities.

Bulk data migration tools

Briefcase (on Windows or Linux), TC XML Import and Export, and Teamcenter Dispatcher Server (on Windows or Linux) must be installed to access the Active Workspace bulk data migration tools.

See **Bulk loading product data** to extract data from one Teamcenter environment to copy to another.

Advanced Multi-Schema Exchanger

Teamcenter Dispatcher Server (on Windows or Linux), Advanced Multi-Schema Exchanger, and the Advanced Multi-Schema Exchanger Service microservice must be installed to access **Advanced Multi-Schema Exchanger using Active Workspace**.

To open and use Advanced Multi-Schema Exchanger with Active Workspace, the user must be logged on to Teamcenter with privileges to use the Active Architect workspace.

General data sharing

See **Configure report layout settings** to configure the readability of data sharing export, import, and validation reports.

Configure report layout settings

Data sharing export, import, and validation reports are formatted for readability with a monospaced typeface. This layout is controlled by including the following settings in the **AWC_defaultViewerConfig.VIEWERCONFIG** preference value:

```
Text.Awp0TextViewer=Text
XMLRenderingStylesheet.Awp0TextViewer=XMLRendering
```

Ensure the following values, which improve rendering of code, are not included in the **AWC_defaultViewerConfig.VIEWERCONFIG** preference value:

```
Text.Awp0CodeViewer=Text
XMLRenderingStylesheet.Awp0CodeViewer=XMLRendering
```

Configure Briefcase file sharing

Ensure the following **preferences** are set to your organization's needs for exchanging objects using Briefcase files.

Briefcase_checkout_supported_types

The list of types supported for Briefcase checkout.

Briefcase_configured_export_supported_types

The list of types supported for Briefcase configured export.

Briefcase_export_supported_types

The list of types supported for Briefcase export.

Briefcase_ownership_transfer_supported_types

The list of types supported for Briefcase site ownership transfer.

Briefcase_pkg_file_name

The default file name format for exported Briefcase files.

Briefcase_tcmail_notification

Specifies whether an email notification is sent when a Briefcase is created.

Configure PLM XML data sharing

Import and export objects of any type supported by the PLM XML schemas. Set the **AWC_PLMXML_export_supported_types** preference to the list of PLM XML types that can be exported and imported at your site. By default, **AWC_PLMXML_export_supported_types** is set to a list of common PLM XML object types.

Configure PDX data sharing

Use the following procedures to configure PDX export using Active Workspace.

Prerequisites

- PDX export must be installed and licensed as described in [Configuring and managing data sharing](#)
- If you are upgrading from a version earlier than Teamcenter 2312, attach the customized *Tc2PDX.xml* style sheet to the default PDX **TIEPDXExportDefault** transfer mode using the following steps.

Beginning with Teamcenter 2312, new Teamcenter installations have this style sheet attached to the transfer mode by default.

1. In a Teamcenter command shell, use the **plmxml_tm_edit_xsl** utility to attach the style sheet to the transfer mode, for example:

```
plmxml_tm_edit_xsl -u=tc-admin-user -p=password
                  -transfermode=TIEPDXExportDefault
                  -xsl_file=C:\TC\tcdata\data\T2PDX.xml
                  -action=attach
```

2. Use the utility to verify the style sheet is attached correctly, for example:

```
plmxml_tm_edit_xsl -u=tc-admin-user -p=password
                  -transfermode=TIEPDXExportDefault -action=list
```

The utility's output should be similar to the following:

```
Original File Name is <Tc2PDX.xml>
```

Map objects and attributes

When you export Teamcenter data to PDX, the Teamcenter data is first exported as TcXML and then transformed to PDX using XSLT.

Map your Teamcenter objects and their attributes to PDX objects and their attributes. Teamcenter provides a default mapping of standard Teamcenter objects to PDX objects and a default style sheet for transforming standard objects to PDX objects.

1. Identify any custom subtypes or data model changes.
2. If the default PDX transfer option set and transfer mode are not appropriate for scoping your Teamcenter data for export to PDX, define an appropriate transfer option set and transfer mode.
 - a. Copy the default PDX **TIEPDXExportDefault** transfer mode file and rename the copy based on your business processes.
 - b. Update the transfer mode to reference the information that must be exported in a PDX package. For details on how to change the closure rule, filter rule, property set, and revision rule, see *Tc XML and PLM XML Configuration for Data Import and Export*.
 - c. Set the **opt_pdx_export** option in the transfer option set used to export data to PDX to a value of **True**. Only those option sets with this option set to **True** are displayed in the **PDX Export** dialog box. (Transfer option sets delivered with Teamcenter have this option set to **True** by default.)
3. Create a custom style sheet to transform the custom subtypes or data model differences.
 - a. Copy the standard PDX style sheet (*Tc2PDX.xsl* file) located in the *TC_DATA\data* directory of your Teamcenter installation.

Siemens Digital Industries Software recommends renaming the style sheet, for example, *WidgetCorpTc2PDX.xsl*.
 - b. Change the style sheet to transform the information required in your exported PDX package.
 - c. If a style sheet with the same name as your updated style sheet is currently attached to the transfer mode, detach it.

From a Teamcenter command window, enter the following command to determine the name of any rules file attached to the transfer mode.

```
plmxml_tm_edit_xsl -u=user-id -p=password -g=group
                  -action=list -transfermode=transfer_mode_name
```

Where *transfer_mode_name* is the name of the transfer mode you are inspecting. Any rules files attached to the specified transfer mode are listed.

If a rules file with the same name as your rules file is attached to the transfer mode, enter the following command to detach the rules file.

```
plmxml_tm_edit_xsl -u=user-id -p=password -g=group
                  -action=detach -transfermode=transfer_mode_name
                  -xsl_file=rule_file_name
```

Where *rule_file_name* is the name of the rule file identified in the first step of this procedure.

- d. Attach the new style sheet to the transfer mode, for example:

```
plmxml_tm_edit_xsl -u=user-id -p=password -g=group
                  -action=attach -transfermode=transfer_mode_name
                  -xsl_file=rule_file_name
```

Configure the PDX package details

Set the **PDX_pkg_file_name** preference at the exporting site. The following keywords are the only keywords supported by **PDX_pkg_file_name**. The keywords are case sensitive and other values are treated as constants.

<i>PDX-string</i>	<i>PDX-string</i> is a string constant prefixed to the package file name of each exported package. You can use this to identify the package as desired, such as by the originating site.
ItemId	Includes the item ID of the exported object as part of the package file name.
ItemName	Includes the item name of the exported object as part of the package file name.
RevId	Includes the revision ID of the exported object as part of the package file name.
TimeStamp	Includes the time stamp information at the time the object is exported as part of the package file name.

Deploy the transfer option set

Provide your users with the transfer option set name they must select when they are exporting a PDX package, and verify your Teamcenter data exports to PDX as expected.

Modify exportable object types (Optional)

The values of the **AWC_PDX_export_supported_types** preference define the object types that can be exported to PDX at your site. By default, **AWC_PDX_export_supported_types** is set to a list of common object types. Update the values defined with that preference as necessary for your site.

Enable vendor filtering (Optional)

By default, users export vendor information (such as vendor contact and manufacturer parts information) in PDX files. To allow the users to limit or exclude vendor information when exporting, set the transfer option set's **opt_vendor_select** option to a value of **True**. When this option is set as **True**, users can select specific (or no) vendors for which vendor information is exported.

Create mapping rules using Advanced Multi-Schema Exchanger

Using Advanced Multi-Schema Exchanger to create mapping rules

Use Advanced Multi-Schema Exchanger to create mapping rules to transform data when it is transferred between Teamcenter sites using different schemas. You can **create mapping rules** for Item types in your source schema. Create and deploy maps using the following process:

1. **Acquire the source and target schemas describing the data being exchanged.**
2. **Create a mapping project in Active Workspace.**
3. **Create rules that map items and their attributes in the source schema to items and attributes in the target schema.**
4. **Deploy the mapping rules by attaching them to the transfer modes used when transferring the data.**

Preparing for a mapping project

The following items are required for creating mapping rules using Advanced Multi-Schema Exchanger:

Source schema

The source schema describing the data being exchanged. This is a full JSON schema covering all possible items and attributes. See [Creating a JSON schema file to create a JSON schema file from your Teamcenter database](#).

Target schema

The target JSON schema describing the data in the target database.

Creating and updating mapping projects

A collection of Advanced Multi-Schema Exchanger schema mapping rules is called a *project*. Create a mapping project for defining the set of mapping rules required for exchanging data between two sites.

Start Advanced Multi-Schema Exchanger

To open and use Advanced Multi-Schema Exchanger with Active Workspace, you must be logged on to Teamcenter as a user with privileges to use the Active Architect workspace.

- On the Active Architect workspace, click the **Advanced Multi-Schema Exchanger** tile. Advanced Multi-Schema Exchanger opens with the most recent projects listed.

Create a mapping project

- Click **Add** ⊕ above the project list to create a project.

The **Add Schema Map** panel opens.

- Specify a name for the project, select your source and target schema files, and click **Add** to save the project.

Update a mapping project

Update a mapping project when your source or target schemas change or when you wish to rename the project.

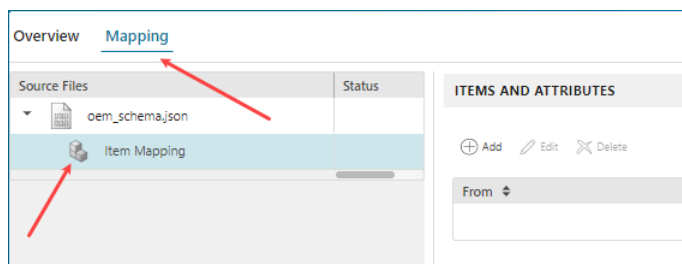
- On the Active Architect workspace, click the **Advanced Multi-Schema Exchanger** tile. Advanced Multi-Schema Exchanger opens with the most recent projects listed.
- Select the project you want to update.

The project's properties are displayed.

- Under **Properties**, click **Edit** ✎.
- Update the name of the project or the schema files as necessary and click **Save Edits** 💾.

Mapping object types

With your project open in **Advanced Multi-Schema Exchanger**, click the **Mapping** tab to show the **Item Mapping** category of the source schema. For example:



Object type mapping

1. Click the **Mapping** tab to show the categories containing the object types that can be mapped.
2. Click **Item Mapping**. A summary of the currently mapped object types and their attributes is displayed in the **Items and Attributes** pane. (A new project has no types mapped.)
3. Select an object type for mapping.

- To create a new object type mapping, click **Add** ⊕ in the **Items and Attributes** pane.

The **Add Mapping** panel opens.

- To edit an existing object type mapping, select the object type in the summary list and click **Edit** .

The **Edit Mapping** panel opens.

4. In the **From** list, select an object in the source schema.
5. In the **To** list, select the target object. (Only valid target objects are listed.)

The source and target objects, and their attributes with default mappings appear in the panel. By default, attributes in the source object are mapped to attributes in the target object when the objects have the same name, length, and type.

For example:

From: To:

From Attribute
Map each attribute from the list below to a target attribute.

Attribute	Data Type	Mapped	Object
L10N_object_desc	POM_string(268)		Item
L10N_object_name	POM_string(176)		Item
a4cm_key1	POM_string(32)		Item
a4cm_key2	POM_string(32)		Item
acl_bits	POM_int		Item
active_seq	POM_int		Item
archive_date	POM_date		Item
archive_info	POM_string(128)		Item
awp0IsSuspect	POM_logical		Item
backup_date	POM_date		Item

To Attribute
Choose a compatible data type to store the attribute in the target schema.

Attribute	Data Type	Mapped	Object
acl_bits	POM_int		Item
active_seq	POM_int		Item
pid	POM_int		Item
revision_limit	POM_int		Item
revision_number	POM_int		Item
acl_bits	POM_int		Item Master
active_seq	POM_int		Item Master
pid	POM_int		Item Master
revision_limit	POM_int		Item Master
revision_number	POM_int		Item Master

Function:

- Map (or remap) a source attribute by selecting it and then selecting the target attribute to which it is to be mapped.

To avoid item ID conflicts, consider adding prefixes and suffixes to the target attributes as part of the mapping. After mapping an attribute, set **Function** to "Prefix" or "Suffix", and specify a value, for example:

Function:

- After mapping an attribute, click **Save** (or **Add** for new mappings) to save the change.

Tip:

Click **Pin Panel**  to keep the panel open as you map several attributes.

To remove the mapping for an object, select the item in the summary list and click **Remove Mapping** .

Deploying mapping rules

Deploying mapping rules attaches the rules to the transfer modes used for transferring the data. Use the following steps to deploy your mapping rules.

To deploy mapping rules, you must be logged on to as a Teamcenter administrator with DBA privileges.

1. Ensure that you have **mapping rules for all object types**.
2. Select the project in the Advance Multi-Schema Exchanger list of projects. Click **Deploy**  in the **Deployment** section.
3. Select the transfer mode set you are using to transfer the data.
4. Click **Deploy**.

You receive a report alert when the deployment is complete. Click on the alert to review the details of the deployment. All recent alerts are available from the **Subscription** tile. Check the **SCHEMA_MAPPING** message type to see all deployment reports.

Once deployed, the mapping rules are attached to the transfer mode. You can now use the transfer mode to share data.

Detach rules from a transfer mode

Detach rules from a transfer mode as follows:

1. From a Teamcenter command window, enter the following command to determine the name of the rule file attached to the transfer mode:

```
plmxml_tm_edit_xsl -u=Tc-admin-user -p=password
-g=group -action=list -transfermode=transfer_mode_name
```

Where:

Tc-admin-user, password, group

Logon credentials of a Teamcenter administrator with DBA privileges

transfer_mode_name

The name of the transfer mode with the attached rules

2. Use the following command to detach the rule file from the transfer mode:

```
plmxml_tm_edit_xsl -u=Tc-admin-user -p=password
-g=group -action=detach -transfermode=transfer_mode_name
-xsl_file=rule_file_name
```

Where:

transfer_mode_name

The name of the transfer mode with the attached rules

rule_file_name

The rule name identified in the first step

Manage Multi-Site

Specifying the scope of Multi-Site remote checkouts and checkins

Use the **MultiSiteCICORule** closure rule to control the scope of objects checked out and checked in by users when they check out and check in remote objects managed by Multi-Site Collaboration. The closure rule defines primary objects and the related secondary objects to be checked out or checked in as well.

Update the closure rule definition to add other related objects to check out and check in automatically with objects. For example, to automatically check out related item revision datasets, add a line to the closure rule definition with the following values:

Clause Item	Value
Primary Object Class Type	CLASS
Primary Object	ItemRevision
Secondary Object Class Type	CLASS
Secondary Object	Dataset
Relation Type	RELATIONP2S
Related Property or Object	IMAN_specification
Action Type	PROCESS
Conditional Clause	
Predicate	

Multi-Site Dashboard

Multi-Site Dashboard provides a way to view the issues **in your Multi-Site federation** through charts, graphs, and detailed object reports. Analyze and resolve these issues to reduce the number of errors and time spent by users in attempting to transfer data or to perform other business tasks.

The data reported in Multi-Site Dashboard is gathered when **running the data_report utility**. The dashboard identifies the following issues for the entire federation and for individual sites in the federation.

- ID issues such as:
 - Duplicate Item IDs, Item Revision IDs, or Keys (including multi-field keys)
 - Item IDs, ItemRevision IDs, or keys that are not synchronized with the owning site.
- Item ownership issues such as:

- Items with multiple owning sites
- Items with no owning site
- Replicas with an owning site different than the primary item
- Object ownership issues such as:
 - Objects with ownership different than the item
 - Items and item revisions with inconsistent ownership
 - Objects with ownership inconsistent with their parent

See [Resolving Multi-Site issues](#) for recommendations on resolving these Multi-Site Collaboration data issues.

Viewing Multi-Site Dashboard issues

When you log on as a Teamcenter administrator, the Multi-Site Dashboard tile is available on the Active Admin workspace home page.

Use the following guidelines for viewing Multi-Site Dashboard issues.

Viewing issues

When first opening Multi-Site Dashboard, the status from the most recently run report is displayed, summarizing all issues across your entire Multi-Site federation. Hover over a chart to view the number of issues of each type.

The lower half of Multi-Site Dashboard breaks down the different types of issues across the federation. Change the value of **Chart by** to toggle between viewing issues by type or by site.

Return to the overall summary of issues across your Multi-Site federation at any time by clearing the selection for all sites in the left pane.

Viewing objects with issues

Click a site name in the left pane to see a breakdown of the types of issues found at that site. To see a further breakdown of an issue type, along with a listing of all objects with that issue, click one of the charts.

Viewing report history

Click **History** to see the federation or selected site trends for each error type over previous runs of the **data_report** utility. Select a different value for **Runs** to change the number of runs displayed. Hover over charted data points for summaries from each run.

Configuring Multi-Site Dashboard

Multi-Site Dashboard presents the status for specified sites in your Multi-Site federation based on data gathered when running the **data_report** utility. Configure and gather Multi-Site Dashboard data as follows.

Define sites from which Multi-Site data is gathered

The Multi-Site sites from which data is gathered and reported must be specified with the **MS_dashboard_Supported_Sites** option.

On the site from which reports are generated, add **MS_Dashboard_Supported_Sites** to the list of Teamcenter site names for which you want data reported by Multi-Site Dashboard. This is also the list of sites that are used with the **data_report** utility.

Run data reports

From a command line, use the **data_report** utility to gather and generate reports presented with Multi-Site Dashboard. You can run this report manually or use your operating system to schedule it to run at regular intervals.

- Run **data_report** on a site with access to the Teamcenter object database.
- Log on with administrator credentials when running **data_report**.

Following are some examples of using the **data_report** utility (not including required login information). See **data_report** for additional details.

To gather data and generate reports for all sites and the overall federation (as defined with **MS_Dashboard_Supported_Sites**):

```
data_report t
```

To gather data for a specific site:

```
data_report -site=site01 -f=collect_data
```

To generate reports for all sites and the Multi-Site federation using data previously collected:

```
data_report -f=generate_reports
```

Resolving Multi-Site issues

Following are guidelines for resolving issues identified by Multi-Site Dashboard.

Item ownership

Problem	Solution
Item with multiple owning sites	Decide which site is to be the owning site. To determine this, run the item_report utility to compare the items on the different sites. Convert the appropriate item from a primary item to a replica using ensure_site_consistency with the -f=recovery option. Using -f=recovery will recover the SST dataset if the dataset exists. If -f=recovery fails to convert the item to a replica, retry the command using -f=offline_recovery in place of -f=recovery. - If you do not have a TC XML meta file, use the form: <pre>ensure_site_consistency -f=recovery -mode=auto -real_owning_site=master_site</pre> - If you have a TC XML meta file, use the option -mode=min or -mode=full with -dir=meta_file_directory.
Item with no owning site	Decide which site is to be the owning site. To determine this, run the item_report utility to compare the items on the different sites. Convert the appropriate item from a replica to a primary item using ensure_site_consistency with the -f=recovery option. Using -f=recovery will recover the SST dataset if the dataset exists. If -f=recovery fails to convert the item to a primary item, retry the command using -f=offline_recovery in place of -f=recovery. - If you do not have a TC XML meta file, use the form: <pre>ensure_site_consistency -f=recovery -mode=auto -remote_site=replica_site</pre> - If you have a TC XML meta file, use the option -mode=min or -mode=full with -dir=meta_file_directory.
Inconsistent ownership replica	From the owning site, run data_share to resend the data.

Duplicate IDs

Problem	Solution
Duplicate Item ID or Key	Avoid the ID conflict during import by using the item_rename utility to rename the target site item.
Duplicate ItemRevision ID or Key	Avoid the ID conflict during import by using the item_rename utility to rename the target site item.
Inconsistent Item ID or Key	From the owning site, run data_share to resend the data.
Inconsistent ItemRevision ID or Key	From the owning site, run data_share to resend the data.

Object ownership

Problem	Solution
Object ownership error	Add the problem object UID to a UID list file named <i>uid.txt</i> and run the tcxml_ownership_recovery utility with the following form: <pre>tcxml_ownership_recovery -inputuidfile=uid.txt -action=flip -targetsite=owning_site_id</pre>

Monitor data exchange transactions

Monitor data exchange transactions

You can use **Data Exchange Transactions** to view and analyze your organization's history of Multi-Site and Briefcase transactions. Filter transactions based on completion status, exchange type, source and target sites involved, and other factors. With these transaction records, you can:

- Drill down through charts and graphs to quickly diagnose data exchange failures
- Use the broad view of your data exchange activities to better identify patterns of failure across multiple sites
- Analyze patterns and trends in data exchange usage to inform ongoing process improvements

The following types of transactions are recorded and reported.

Multi-Site

- Data shared and retrieved between sites

- Remote export and import operations made using the rich client
- Exports and imports made using the **data_share** command
- Transactions made using the **data_sync** command

Briefcase

All Briefcase import and export transactions involving the site as a source or a target are recorded and reported, whether they are run from the Teamcenter rich client, Active Workspace, or the command line.

Bulk data exchanges and PLM XML transactions are not recorded.

Configuring data exchange transaction monitoring

Configure the following aspects of gathering and displaying your organization's data exchange transaction records.

Specify transactions to monitor

You can configure your site to record Multi-Site, Briefcase, or both types of data exchange transactions originating at a site. Use the **TXN_Supported_Application_Types** preference to specify which data sharing transactions are recorded and reported. Set the preference to a value of **Briefcase**, **Multisite**, or both values depending on the types of data sharing transactions you want to record.

Specify a central site for monitoring transactions

By default, all transactions started from a site are recorded and can be viewed only when logged onto that site. Alternatively, several sites can send their transaction records to a common site for consolidated viewing of the organization's data exchange transactions at that common site.

To use a common site to view transaction records, on each of the remote sites set the **TXN_Central_Site** preference to the **Site Name** value of the common site.

Change the period of reported transaction records

By default, a site's transaction records for the previous 30 days are reported. Change the default number of days to report by updating the value of the **TXN_history_range** preference.

Clear data from the transaction history

Transaction records remain in a site's history until they are manually removed. To delete records from a site's history, log onto the site, open a command window, and run the **transaction_monitor** utility. (See **transaction_monitor** for more details on the utility.) For example:

Delete records in a date range

Use a command of the following form to delete all transaction records up to a certain date:

```
transaction_monitor -u=youruserid -p=yourpw -f=delete -from="2010-01-01
01:00:00" -to="2021-09-30 24:59:59"
```

Delete records of successful transactions in a date range

Use a command of the following form to delete all successful transaction records that occurred in a certain time period:

```
transaction_monitor -u=youruserid -p=yourpw -f=delete
-final_status=Success -from="2021-09-30 16:20:00" -to="2021-10-15
13:00:00"
```

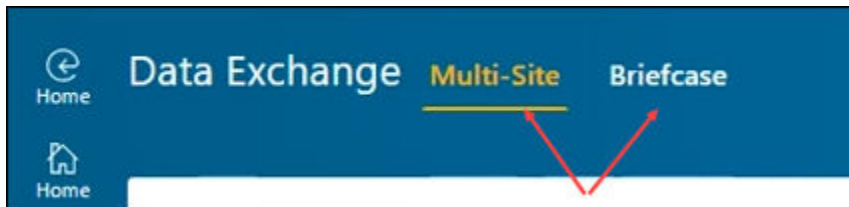
Viewing data exchange transaction records

When you log on as a Teamcenter administrator, the **Data Exchange Transactions** tile is available on the Active Admin workspace home page.

By default, records for only those data transactions originating at your site are displayed. Your site may be configured to **view transactions originating at other sites**, too.

View transaction records

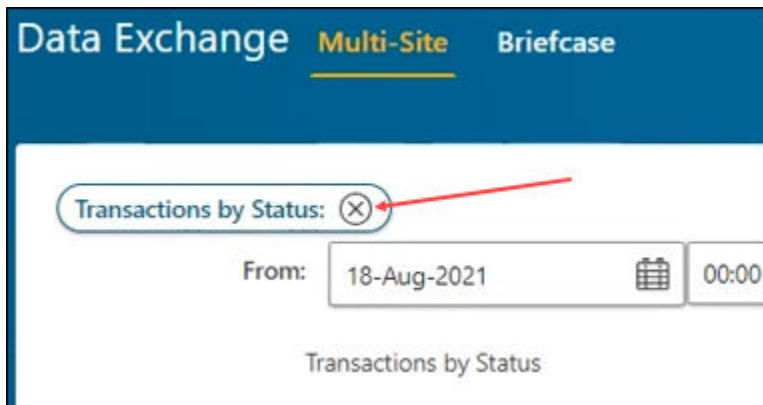
Click the **Multi-Site** or **Briefcase** tab to view transaction records for your site's recorded transactions. (Depending on **your site's configuration**, the tabs available may vary.)



Several graphs summarizing data exchanges originating at your site are displayed. Summaries are organized by status, type, and involved sites.

Below the graphs, the transaction records for the site are listed. Open any of the records in the list to view additional details about the record, such as the source and target sites, transaction type, objects involved in the transaction, any reported errors, date and time of the transaction, and locations of related log files.

Click one or more bar graphs or chart segments to refine the list of transaction records to those selections. Remove filters at any time by closing the selected filter.



Data Exchange **Multi-Site** Briefcase

Transactions by Status: (X)

From: 18-Aug-2021 00:00

Transactions by Status

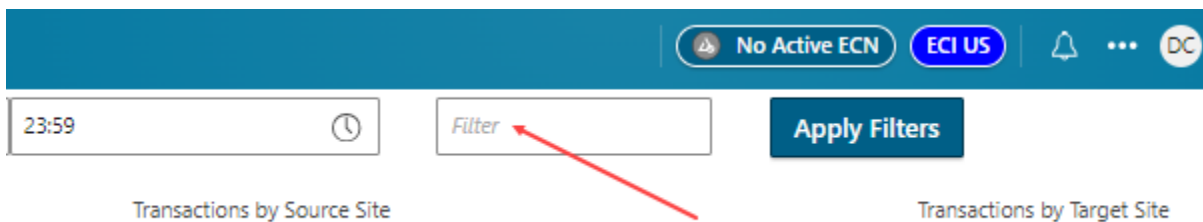
Filter transaction records by date range

Specify a different date range from which to view transaction records by entering **From** and **To** values and clicking **Apply Filters**. The record list is updated to show transactions from the new date range.

By default, the transaction records for the previous 30 days are reported.

Filter transaction records by ID

Enter a value in the **Filter** field and click **Apply Filters** to refine the reported records to only those objects with IDs containing the value and to records with Briefcase file names containing the value.



No Active ECN ECI US

23:59 Filter Apply Filters

Transactions by Source Site Transactions by Target Site

Share bulk data

Bulk loading product data

Active Workspace provides tools to extract data from one Teamcenter environment to copy to another. You would typically use these tools when testing Teamcenter upgrades and customizations in test environments.

When creating a test environment, you may need to duplicate large amounts of data from a production environment in the test environment. Doing so delivers a test environment with an applicable, broad set of data with which to test customizations and updated software.

Requirements

You must be logged on as a user with administration privileges to use the bulk extract and copy tools.

To bulk copy product data into a site, the site must be designated as a test site. You can designate a site as a test site when installing it. You can also use the **install** utility with the **-mark_as_test_env** argument to convert a site created as a production environment to a test environment.

Copying product data from your production environment

Use the following steps to copy data from a production environment to a Briefcase file that can be loaded into a test environment.

1. Log on with Teamcenter administrative privileges and select one or more items, assemblies, or folders as root objects to copy.
2. Click **More commands ...** > **Share**  > **Bulk Extract** to display the **Bulk Extract** panel.
3. Select the transfer option set to use when copying the objects. (Only unconfigured transfer option sets are available.)
4. Accept the default file Briefcase file name or update it as necessary.
5. Click **Override Options** and review the default settings for extracting the data. Update the options as necessary and click **Override**.
6. Click **Extract** to begin copying the objects to the Briefcase file. You receive an alert when it is complete.

Review the report and transfer the Briefcase file

You receive a report alert when the Briefcase file is created. Click on the alert to view the report. Access all recent alerts from the **Subscription** tile.

- The **Properties** section of the report includes details such as a list of the user extracting the data and the transfer option set used. Under **Related Objects**, click on the export log entry to view additional details, including a complete list of the objects extracted.
- The **Target Object** section of the report contains a link to the Briefcase file containing the extracted objects. Select the Briefcase file and click **View**  > **Load Briefcase** to review the contents of the file.

Download the file and save it to a location accessible by the test site.

Copying product data into your test environment

Use the following steps to load data copied from a Teamcenter production environment into a test environment.

To bulk copy product data into a site, the site must be designated as a test site. You can designate a site as a test site when installing it. You can also use the **install** utility with the **-mark_as_test_env** argument to convert a site created as a production environment to a test environment.

1. Log on with Teamcenter administrative privileges and select the folder into which the root objects will be copied.
2. Click **More commands ...** > **New** ✨ > **Bulk Copy** to display the **Bulk Copy** panel.
3. Use **Choose File** to locate the Briefcase file containing the data to copy.
4. Select the transfer option set to use when importing the file. Override any options as necessary.
5. Click **Bulk Copy** to import the objects in the Briefcase file. You receive an alert when the import run is complete.
6. Review the selected folder for the imported objects.

Note:

If you are using Active Workspace with a version of Teamcenter earlier than 13.2 when copying the data from the Teamcenter production environment or to the test environment, the root objects will not be copied to the selected folder. You must search the Teamcenter database for the copied data.

Review the import report

You receive a report alert when the copying completes. Click on the alert to view the report. (Access all recent alerts from the **Subscription** tile.)

The folder into which you copied the data is listed under **Related Objects** in the **Properties** section of the report. Under **Target Object**, download the import log, which includes details such as a list of the objects copied and the transfer option set used.

8. Configuring Teamcenter mail and instant messaging

Configuring Teamcenter mail

Administrators can configure Teamcenter mail to:

- Specify the mail gateway server, port, and character set used to send mail.
- Allow e-mail addresses to exceed 32 characters in length.
- Notify users of important schedule dates, milestones, and task completion.
- Enable and disable the capability to send operating system e-mail from within Teamcenter.
- Specify the interval at which Teamcenter checks for new mail.
- Work with external e-mail for subscription notices.
- Receive email notifications when there is a new discussion activity within Teamcenter.

Change email server authentication settings

Email server authentication settings are initially configured using Teamcenter Environment Manager manager.

1. Run Teamcenter Environment Manager from your Teamcenter home directory.
2. Select **Configuration Manager** and choose to perform maintenance on your existing Teamcenter Foundation configuration.
3. On the **Foundation Settings** panel, choose the **Email server settings** tab and specify a new password and change related security settings.

Configure external e-mail for subscription and workflow notification

To use external (operating system) e-mail for subscription and workflow notification, you must:

- Set mail gateway preferences:
 - **TC_subscription=ON** to enable display and use of the **Tools→Subscribe** menu command.
 - **Mail_server_name=a_valid_SMTP_mail_server**

- **Mail_OSMail_activated=true** to enable operating system email from Teamcenter.

To view notification e-mail sending records, set **TC_audit_manager=ON**, **TC_audit_manager_version=3**, and **SCM_notification_history=true**.

- For the person objects associated to users to be notified, ensure the **E-Mail** address fields are set correctly in the Organization application.
- Start the **subscriptionmgrd** subscription monitor process daemon.

Configure email notifications for discussion activities

To receive email notifications for discussion activities, you must:

- Set the email preferences at your site.
- Set **Ac0EnableEmailDiscussionNotifications** to **true**. This preference is set to **false** by default.
- Ensure that your users have their email IDs specified.

Configuring Teamcenter to use email servers requiring authentication

You can **use TEM to configure Teamcenter to access an email server that requires authentication**. The following preferences get set based on the values supplied in TEM, and can be updated if your email server requirements change.

Mail_server_authentication_activated

Set to **true** to connect to a server requiring authentication.

Mail_server_authentication_id

The e-mail ID or the user ID to use for e-mail authentication.

Mail_server_authentication_passwd_location

The full path and file name of the encrypted password to use for e-mail authentication.

Mail_server_connection_security

The type of secure connection to establish when connecting to the mail server.

None	(The default value.) No security connection protocol is used when connecting with the mail server.
SSL/TLS	Establish an SSL/TLS connection when connecting to the mail server.
STARTTLS	Establish an initial insecure connection and later convert to a TLS connection with the mail server.

Mail_server_ssl_protocol

The protocol and version to use when connecting with the mail server.

TLSv1	(The default value.) Transport Layer Security version 1.
TLSv1_1	Transport Layer Security version 1.1.
TLSv1_2	Transport Layer Security version 1.2.
SSLv2	Secure Sockets Layer version 2.
SSLv23	Secure Sockets Layer version 2.3.
SSLv3	Secure Sockets Layer version 3.

Configuring instant messaging

Note:

Teamcenter rich client instant messaging with Microsoft Office Communicator should continue to work. However, Siemens Digital Industries Software no longer has an environment in which to validate the instant messaging functionality.

Teamcenter users can view the current status of the owning and last modified users, and, from within a Teamcenter application, can initiate communication using Microsoft Office Communicator.

This capability:

- Is supported only on Windows systems with Microsoft Office Communicator.
- Lets you access all available Microsoft Office Communicator features.
- Is available in Teamcenter and is enabled and controlled by preference settings.
 - The **OCS_use_presence_display** preference specifies whether the functionality is integrated with Teamcenter.
 - The **OCS_use_email_property** preference specifies whether to use the e-mail ID from the e-mail ID field of the user **Person** object.
 - The **OCS_company_email_domain** preference value specifies a domain name if the **OCS_use_email_property** preference value is **false**.
- For the rich client, integration is implemented in:
 - My Teamcenter **Summary** view
 - My Teamcenter **Viewer** view

- Workflow **perform-signoffs** panes
- Systems Engineering
- Structure Manager
- Schedule Manager

Note:

Microsoft Office Communicator must be installed and running on your computer to use the Teamcenter integration with Microsoft Office Communicator.

- If you try to use this feature when Microsoft Office Communicator is not installed and running on your computer, Teamcenter displays the symbol color corresponding to the Microsoft Office Communicator **Presence Unknown** status.
- If you try to use the Teamcenter integration with Microsoft Office Communicator to contact a user for whom the system cannot find an e-mail ID, the system displays a message listing possible reasons for the inability to continue the communication.

9. Configuring Teamcenter email polling

What is Teamcenter email polling?

Teamcenter email polling is functionality that checks an email account for unread messages that match criteria specified in a polling rule. For messages that meet the polling criteria, Teamcenter performs a specified type of response.

You can use email polling to bring email content from third parties into Teamcenter. You can also extend email polling to perform additional actions on email messages and attachments.

Actions performed during an email polling cycle include the following:

1. The business user defines an email polling rule and then starts or schedules email polling.
2. Teamcenter logs in to the user's e-mail account and checks the Inbox folder for unread messages.
3. Teamcenter tests unread messages for matches to criteria (subject and token files) specified in the business user's rules.
4. If an unread message matches the specified criteria, then Teamcenter performs actions according to the response type specified in the rule. The default response type performs the following actions:
 - Downloads the message body and attachments to a specified folder on the Teamcenter server.
 - Moves the message to a specified archive folder in the mail system.
 - Starts a review workflow of the downloaded message in Teamcenter.

An administrator or developer can create a subtype of the basic email-polling business object and define additional actions for a response. For example, attachments could be scanned for viruses.

5. From the review workflow task, the business user reviews the message and attachment files and chooses whether to approve or reject the message and attachments.

For this decision	Teamcenter does this with the file (default action)
Approve	No further action.
Reject	Purges the downloaded file.

Email polling can be started or scheduled either of these ways:

- By the business user from within the rich client, if standard Teamcenter dispatcher services have been configured. Multiple rules can be scheduled to run.
- By an administrator at the command line or in a cron job using the email_polling utility.

Configure email polling types

System administrators or customizers configure email polling types and can customize the actions performed by a type. The types are referenced by email polling rules.

Prerequisites

- Teamcenter is installed and configured to work correctly.
- BMIDE is installed and is working property.
- The system administrator or customizer performing the configuration:
 - has sufficient privileges (dba) to create and install templates or make changes to the Teamcenter database.
 - has write access to all Teamcenter installation folders.
- If scheduled polling using Teamcenter Dispatcher is to be used, when installing or updating Dispatcher, in the **Select Translators** panel, **Email Polling**→**Email Polling** is selected.

Configuration steps

1. In BMIDE, for the e-mail polling type that you want to configure:
 - a. Create or open a template project.
 - b. In the classic LOV list **Fnd0EmailResponseTypes**, create a list value to identify the polling type that you want to define.

Only the *value* parameter of the list value is used for polling; its *description* and *condition* parameters are not used.
 - c. If in your application you want to persist more data than just e-mail body text and attachments, or to impart additional behavior, then create a subtype of the object **Fnd0EmailResponseRecord**.

Add your custom actions to your new object subtype.
 - d. Save and deploy the template.

2. In the rich client:
 - a. Log on using an account with dba privileges.
 - b. Choose **Edit→Options**.
 - c. Find the preference **Email_polling_download_dir**, and set the preference value to the desired location for downloaded attachments in the server's file system. This location is used by all email polling types.

The specified location must exist in the server's file system; the polling functionality does not create the folder.

- d. For each value that you added to the classic LOV list **Fnd0EmailResponseTypes**, do the following:
 - A. Create a new preference named *<LOV value>_Response_Record_Object*.
 - B. Set the preference value to **Fnd0EmailResponseRecord** or, if you created a subtype of the object **Fnd0EmailResponseRecord** and you want to associate the LOV value with that object subtype, set the preference value to the name of the object subtype.

Create an email polling rule

Business users create email polling rules to configure checks for incoming email messages and attachments, and to specify the polling type to apply to qualified messages. Multiple rules can use the same polling type.

1. In My Teamcenter, choose **File→New→Other→Email Polling Rule**.

The screenshot shows a 'New Business Object' dialog box with the title 'New Business Object'. Below the title bar is a section titled 'Object Create Information' with the subtitle 'Define business object create information'. The main area is titled 'Email Polling Rule' and has a 'General' tab. The fields are as follows:

- Description:** A text area with a vertical scrollbar.
- Name:** A text field with an asterisk indicating it is required.
- Polling Archive Folder:** A text field.
- Polling Inbox Folder:** A text field.
- Polling Subject:** A text field with a vertical scrollbar.
- Polling Token File Names:** A text area with a vertical scrollbar and a small icon at the bottom right.
- Response Type:** A dropdown menu.
- Rule ID:** A text field with an asterisk indicating it is required.
- Relation:** A dropdown menu set to 'References'.
- Open On Create:** A checked checkbox.

At the bottom are four buttons: '< Back', 'Next >', 'Finish', and 'Close'.

- In the **Email Polling Rule** dialog box, enter the following information.

Field	Description
Description	A brief description of the email polling rule and its usage.
Name	A name for the email polling rule.
Polling Archive folder	The destination user email folder for archived email messages. The folder must exist. The polling functionality does not create the folder.
Polling Inbox folder	The email user folder containing the messages to test.

Field	Description
Polling Subject	The words at the beginning of the email message subject that qualify it for action by the rule.
Polling Token File Names	The names of files provided to responders for required attachment to response emails. Required attachments may include digitally signed request-identification-documents or pre-encrypted binary files. Token files are a means of enhancing security. Leave blank if token files are not used.
Response Type	The type of email response to which this rule applies. This is a value contained in the application template classic LOV (list of values) Fnd0EmailResponseTypes as configured by the system administrator/customizer.
Rule ID	The ID for the newly created rule. The ID is used to identify the rule when starting email polling.

3. Click **Finish**.

Configure user email account settings

To enable Teamcenter polling functionality to communicate with an email server, user email account information must be specified before starting or scheduling email polling.

1. In My Teamcenter, choose **Tools→Email Polling→Configure Email Polling**.

The screenshot shows a 'Configure Email Polling' dialog box. It has a title bar with a blue icon and a red close button. The main area contains the following fields:

- Address of the email server being polled:
- Polling user email ID:
- Polling user password:
- Server port number:
- SMTP port number:
- SMTP server address:
- Polling protocol for the server:

At the bottom, there are two buttons: 'OK' and 'Cancel'.

2. In the **Configure Email Polling** dialog box, enter the following information:

Field	Description
Address of the email server being polled	The address of the email server for your email account. Example: mycasa001.net.acme.com
Polling user email ID	Your email account address. Example: john.smith@acme.com
Polling user password	The password for your email account.
Server port number	The email server port number for messages sent to your account (incoming). Example: 993
SMTP port number	The email server port number for uploading messages that you want to send from your account (outgoing). Example: 465
SMTP server address	The host name of the SMTP (Simple Mail Transfer Protocol) server for outgoing mail. Example: smtp.acme.com
Polling protocol for the server	The protocol for connecting with the email server, either POP3 or IMAP.

3. Click **OK**.

Configure Dispatcher for email polling

Note:

Dispatcher settings for email polling may have been set by Teamcenter Environment Manager (TEM). In that case, confirm the following configuration.

1. To activate the **EmailPollingService** service, set the **isactive** attribute to **true** in the *DISP_ROOT\Module\conf\translator.xml* file.

```
<EmailPollingService provider="SIEMENS" service="emailpolling" isactive="true">
```

Note:

DISP_ROOT is the dispatcher root directory provided in Teamcenter Environment Manager (TEM).

2. In the *DISP_ROOT\Module\Translators\emailpolling\emailpolling.bat* file, set the following variables to your installation locations:

- *TC_ROOT*
- *TC_DATA*
- *JRE_HOME*

3. In the *TC_DATA\EmailPolling.conf* file, remove the comments before these settings:

- **EmailPolling_JAVA_XMS=16m**
- **EmailPolling_JAVA_XMX=128m**

4. Run the **genregxml.bat** utility.

- Open a Teamcenter command prompt and type the following command:

```
TC_ROOT\portal\registry\genregxml.bat
```

5. For each Teamcenter user who performs email polling actions, set the **E-Mail Address** property.
 - a. Using an account with dba privileges, log on to the rich client and open the Organization application.
 - b. Expand the **Persons** node and select the person who performs email polling actions.
 - c. Set the **E-Mail Address** property.

Start or schedule email polling

Prerequisites

- **User email account information is configured**
- **Dispatcher is configured for email polling**

Procedure

1. In My Teamcenter, choose **Tools>Email Polling>Start Email Polling**.

2. In the **Start Email Polling** dialog box, enter the following information:

Field	Description
Start now!	Select one of these choices.
Schedule	• Start now! runs a single on-demand poll. • Schedule enables scheduling periodic polling.
Start Time	The date and time to start scheduled polling.
End Time	The date and time to end scheduled polling.
Interval	The interval of time to repeat the poll.
Rules ID	The ID of the email polling rule to use.

3. Click **OK**.

You can administer scheduled polling requests using the Dispatcher request administration console.

Teamcenter email polling preferences

For basic email polling functionality, you do not need to change default values for the following preferences.

EMLPOLLING_default_polling_config_id

Defines the identifier for email polling configuration to be used in the email polling operation.

Default value: **def_polling_config_id**

Fnd0EMLPollingRevFrm.JAVARENDERING

Defines the Java class to render the property panel of a form with type **Review Email Polling Revision Form**.

Default value:

com.teamcenter.rac.emailpolling.commands.rendering.EmailPollingRevisionForm

Fnd0_review_email_attachments_workflow

Defines the create workflow for email polling functionality.

Default value: **Review Mail Attachments**

10. Updating property values in bulk

Process for updating property values in bulk

You can update the values of object properties in your Teamcenter database in bulk using the **tc_attribute_bulk_update** utility. This utility replaces the previous method for updating property values in bulk, which entailed using the **attribute_export** utility to export the property values and the **tcxml_import** utility to import the updated property values into the database.

Tip:

You can also use live updates to update property values.

Use the **tc_attribute_bulk_update** utility to:

- Query for the objects that satisfy the conditions specified through condition property name/property value pairs.
- Provide new values for the properties to be updated on the found objects.
- Export the data to a TC XML file.
- Import the data from the TC XML file into the database.

The scope of your bulk updates can range from changing a few values on a few properties (simple) to changing many values on all properties throughout the database (complex). The scope of your planned update determines which format to use to provide query criteria and replacement values to the **tc_attribute_bulk_update** utility.

- **Simple updates**

Use this method when updating a few values on a small number of properties for a few object types.

Example:

The designers at Manufacturing, Inc. create two new colors for their spinning widgets and discontinued another color. Accordingly, their Teamcenter administrator must add the **slide_green** and **twisting_turquoise** values to the **color** property on the **widget_spin** object and remove the **boring_brown** value.

Specify the condition property name/property value pairs and update property name/property value pairs for the utility. Optionally, you can also specify an operator for condition property name/property value pairs.

-cond_prop

```
-cond_value
-update_prop
-update_value
-cond_operator
```

- **Complex updates**

Use this method when changes to your business practices require sweeping changes to existing property values throughout the database.

Example:

At Acme Company, every workspace object includes **Division** as a required property. Valid **Division** values are the different business divisions of Acme. A reorganization changes the list of existing business divisions significantly and creates new subsectors. Acme's Teamcenter administrator must update the **Division** values on every workspace object in the database.

Create an XML input file to specify property name/property value pairs to query for the objects to be updated, and then specify property name/property value pairs to update the found objects.

Using the `tc_attribute_bulk_update` utility arguments to update property values in bulk

When performing a simple update using the `tc_attribute_bulk_update` utility (updating a few values on a small number of properties on a few different object types), use the command line arguments to provide the query/update information.

You must use a combination of the `-type` argument and condition property name/property value pairs, along with update property name/property value pairs to:

- Query for the objects that satisfy the conditions specified through condition property name/property value pairs.
- Provide new values for the properties to be updated on the found objects.
- Export the data to a TC XML file.
- Query arguments

These arguments are the equivalent of the **WHERE** clause in an SQL phrase. They identify the property type, property names, and property values to be queried.

```
-type
-cond_prop
-cond_value
-cond_operator (optional)
```

The two condition property name/property value arguments must always be used in conjunction with the update property name/property value arguments.

- The **-type** argument specifies the object type to be updated, such as **item**, **dataset**, or **StructureContext**.

This argument accepts only a single value. You can use multiple instances of this argument to specify multiple object types. Each instance must be paired with the **-cond_prop** and **-cond_value** arguments.

- The **-cond_prop** argument specifies the internal name of the property to be queried for, as opposed to the display name.

This argument accepts multiple values in a comma-separated list. Each value must be a valid property on a Teamcenter object. For example:

-cond_prop=object_name,last_mod_date

You can use multiple instances of this argument, but it must always be paired with the **-cond_value** argument.

- The **-cond_value** argument specifies the current value of the property specified by the **-cond_prop** argument.

This argument accepts multiple values in a comma-separated list. Each value must be a valid property on a Teamcenter object. For example:

-cond_value=TextData_es_ES,"01-DEC-2011 00:00"

You can use multiple instances of this argument, but it must always be paired with the **-cond_prop** argument.

- You can use the **-cond_operator** argument to specify an operator to be used with the **-cond_prop** and **-cond_value** arguments.

This argument must be placed after the **-cond_prop** and **-cond_value** arguments and accepts the following values:

EQ	Equal
NE	Not equal
GT	Greater than
GE	Greater than and equal
LT	Lesser than
LE	Lesser than and equal

For example:

```
-cond_prop=object_name -cond_value="My obj #1"
-cond_prop="last_mod_date"
-cond_value="01-DEC-2011 00:00" -cond_operator="LE"
-cond_prop="last_mod_date"
-cond_value="01-DEC-2010 00:00" -cond_operator="GE"
```

- Update arguments

These arguments are the equivalent of the **UPDATE** clause in an SQL phrase. They identify the property names and values to be updated.

```
-update_prop
-update_value
```

Update property name/property value pairs must always be used in conjunction with the condition property name/property value pairs.

- The **-update_prop** argument specifies the internal name of the property to be updated (as opposed to the display name), for example, **object_desc**, **char VLA**, and so on.

This argument accepts multiple values in a comma-separated list. Each value must be a valid property on a Teamcenter object. For example:

```
-update_prop=object_name,object_desc
```

You can use multiple instances of this argument. Each instance of this argument should be paired with the **-update_value** argument.

- The **-update_value** argument specifies the new value for the property specified by the **-update_prop** argument.

This argument accepts multiple values in a comma-separated list. For example:

```
-update_value=folder,"Home folder"
```

You can use multiple instances of this argument. Pair each instance of this argument with the **-update_prop** argument.

The following examples illustrate the use of the condition property name/property value pairs and update property name/property value pairs with the **tc_attribute_bulk_update** utility:

- To update **prop3** to **value3** and **prop4** to **value4** for **type1** objects (when **prop1** equals **value1** and **prop2** is greater than **value2**) in batches of **400**, enter a command with the following form on a single line:

```
tc_attribute_bulk_update -u=tc-admin-user -pf=d:\password.txt -g=group
-type=type1 -cond_prop=prop1 -cond_value="value1"
-cond_prop=prop2 -cond_value="value2" -cond_operator="GT"
-update_prop=prop3 -update_value="value3"
-update_prop=prop4 -update_value="value4"
-batchsize=400
```

- To update an object description to **blue Desc** and the **int_VLA** to **10,3,0,99** for all datasets where the **object_name** property is **TextData_es_ES** and the **last_mod_date** property is **01-DEC-2011 00:00** or greater, enter the following command on a single line:

```
tc_attribute_bulk_update -u=tc-admin-user -p=password -g=group
-type=Dataset -update_prop=object_desc -update_value="blue Desc"
-update_prop="int_VLA" -update_value="10,3,0,99"
-cond_prop=object_name -cond_value="TextData_es_ES"
-cond_prop="last_mod_date" -cond_value="01-DEC-2011 00:00"
-cond_operator="GE"
```

Using an XML input file to update property values in bulk

When performing a complex update using the **tc_attribute_bulk_update** utility (updating many values on many properties throughout the database), use an XML input file to provide the query/update information with the **-inputfile** argument. The input file is a more efficient format for listing lengthy update instructions than using the command line arguments.

You must use a combination of condition property name/property value pairs and update property name/property value pairs to:

- Query for the objects that satisfy the conditions specified through condition property name/property value pairs.
- Provide new values for the properties to be updated on the found objects.

Structure the file as a series of **UpdateSets** entries. Each update set must contain type, condition, and update components. The following sample XML file illustrates the required sequence of the XML components, first **type**, then **where**, and then **update**.

```
<?xml version="1.0" encoding="utf-8"?>
<BulkUpdate>
  <UpdateSets>
    <UpdateSet>
      <type name = "item" />
      <where>
        <cond_prop attrName = "int_single" attrValue = "50" cond_operator = "<=" />
        <cond_prop attrName = "string_attr" attrValue = "Wed, Mon, Sat, Tue" />
        <cond_prop attrName = "last_mod_date" cond_operator = ">" attrValue = "01-
DEC-2011 00:00"/>
      </where>
    </UpdateSet>
  </UpdateSets>
</BulkUpdate>
```

```

    </where>
    <update>
      <update_prop attrName = "object_desc" attrValue = "red Desc"/>
      <update_prop attrName = "intVLA" attrValue = "10,20,0,9,7"/>
      <update_prop attrName = "some_VLA_property" attrValue =
"value1,value2,value3"/>
    </update>
  </UpdateSet>
</UpdateSets>
</BulkUpdate>

```

Note:

There is no required sequence within the **cond_prop** component for the **attrName**, **attrValue**, and **cond_operator** components.

- Type component

This component specifies the object type to be updated, such as **item**, **dataset**, or **StructureContext**.

This component accepts only a single value.

- Condition components

These components are placed in the **<where>** section of the update set. They identify the property names and values to be queried.

```

cond_prop
cond_value
cond_operator (optional)

```

- The **cond_prop** component specifies the internal name of the property to be queried for, as opposed to the display name.

This component accepts multiple values in a comma-separated list. Each value must be a valid property on a Teamcenter object. For example:

```
cond_prop=object_name,last_mod_date
```

You can use multiple instances of this component, but it must always be paired with the **cond_value** component.

- The **cond_value** component specifies the current value of the property specified by the **cond_prop** component.

This component accepts multiple values in a comma-separated list. Each value must be a valid property on a Teamcenter object. For example:

```
cond_value=TextData_es_ES,"01-DEC-2011 00:00"
```


You can use multiple instances of this component, but it must always be paired with the **cond_prop** component.

- You can use the **cond_operator** component to specify an operator to be used with the **cond_prop** and **cond_value** components.

This component accepts the following values:

=	Equal
!=	Not equal
>	Greater than
>=	Greater than and equal
<	Lesser than
<=	Lesser than and equal

For example:

```
<where>
  <cond_prop attrName =
    "int_single" attrValue = "50" cond_operator = "<=" />
  <cond_prop attrName = "string_attr" attrValue = "Wed, Mon, Sat,
Tue" />
  <cond_prop attrName =
    "last_mod_date" cond_operator = ">" attrValue = "01-
DEC-2011" />
</where>
```

- Update components

These components identify the property names and values to be updated.

update_prop
update_value

- The **update_prop** component specifies the internal name of the property to be updated (as opposed to the display name), for example, **object_desc**, **char VLA**, and so on.

This component accepts multiple values in a comma-separated list. Each value must be a valid property on a Teamcenter object. For example:

update_prop=object_name,object_desc

You can use multiple instances of this component. Each instance of this component should be paired with the **update_value** component.

- The **update_value** component specifies the new value for the property specified by the **update_prop** component.

This argument accepts multiple values in a comma-separated list. For example:

update_value=folder,"Home folder"

You can use multiple instances of this component. Each instance of this component should be paired with the **update_prop** component.

The input file supports the following persistent properties:

- Single-value properties
- Array properties
 - Fixed length array
 - Variable length array (VLA)

Performance statistics

You can estimate the duration of your bulk update gathered on the following performance statistics, based on custom **ADSPart** objects.

Custom ADSParts	attribute_export in seconds	tcxml_import in seconds	Total time in seconds
169	11.261	13.115	24.376
602	28.001	27.569	55.57
1,356	52.338	73.519	125.857
9,570	318.393	687.489	1,005.882
29,241	954.282	1,522.68	2,476.962

The statistics are based on the following setup:

- Oracle database running on **machineA**.
- Teamcenter server running on **machineB**.

- CPU speed: 3 GHz
- CPU type: Intel Pentium 4 CPU 3.00GHz
- Memory: 2 GB
- Both **machineA** and **machineB** are on a LAN network.
- The **UGII_CHECKING_LEVEL** environment variable is set to **0**.

Best practices for updating property values in bulk

Siemens Digital Industries Software recommends observing the following best practices:

- Do not perform this operation when users are accessing Teamcenter data. If objects specified for update are locked by other processes, the update process is impacted.
- To determine all the current values of attributes you plan to update, with the **tc_attribute_bulk_update** utility, use the **-untransformed** switch with either the **-inputfile** argument containing only condition property/value entries (no update entries) or the **-cond_prop** and **-cond_value** arguments.
- To validate the schema of the XML input file, use the **-performSchemaValidation** switch.

If the schema is invalid, the following information is added to the log file:

Error: attributeUpdateSchemaValidator: XML Exception during schema file validation.

If the schema is not found, the following information is added to the log file:

Error: AttributeUpdateSchemaValidator: Unable to find schema file for validation.

- To confirm the updates were made as expected, specify an output directory with the **-outdir** argument so that you can review the output log.
- Because extensive updates are time-consuming, Siemens Digital Industries Software recommends you determine the following before performing a complex bulk update of property values:
 - Affected targets

Run the **tc_attribute_bulk_update** utility with the **-queryonly** switch to determine the number of target objects affected by the proposed update. When you specify this switch, the utility does not perform the update, it merely reports the number of affected objects.

This switch outputs the number of target objects affected by the specified update parameters to a log file. If the number of objects is less than 100, the object UUIDs are included in the log file.

Use this switch in conjunction with either the input file or the condition and update arguments to determine how many objects are affected by the specified update operation. You can use the resulting information to determine batch size and to estimate the duration of the update operation.

- Duration

To estimate the duration of various update operations, see the performance tables. If the estimated time is considerable, there are three methods for managing the operation duration.

- Run the **tc_attribute_bulk_update** utility with the **-batchsize** argument to specify the number of objects to update in each batch operation per TC XML file.

The default batch size is 500. This is also the maximum batch size.

- Run the **tc_attribute_bulk_update** utility with the **-islandsize** argument to specify the number of islands to update in each operation. Islands tie logically related objects together. The data in low-level TC XML is grouped into an island by closure rules.

The default size is 100.

- Manage the number of objects processed per update by running multiple updates constrained by dates. The updates can be run in parallel. For example:

- ◇ First update

```
-cond_prop="last_mod_date" -cond_value="01-DEC-2010 00:00"
-cond_operator="LT"
```

- ◇ Second update

```
-cond_prop="last_mod_date" -cond_value="01-DEC-2011 00:00"
-cond_operator="LE"
-cond_prop="last_mod_date" -cond_value="01-DEC-2010 00:00"
-cond_operator="GE"
```

- ◇ Third update

```
-cond_prop="last_mod_date" -cond_value="01-DEC-2011 00:00"
-cond_operator="GT"
```

11. Logging

Monitoring your system

You can use several types of logging to monitor the activities of Teamcenter processes and objects.

Audit logs

Audit logs track the activities performed on selected data model objects. This provides a tracking mechanism for any changes made to those objects for historical record. To use audit logs, they must be **enabled and configured**.

Core Teamcenter logs

These process log files contain diagnostic information for the core foundation of Teamcenter which is described in the Teamcenter *System Administration* topic *Teamcenter Logging*.

Configuring the Audit Logs page

Configuring audit logs

What are audit log types?

The following audit log types hold audit records based on logical groupings of object type and event type combinations:

- General logs
- License export logs
- Organization logs
- Security logs
- Schedule logs
- Structure logs
- Workflow logs

What is an audit log dataset?

An audit log dataset is a stylesheet configuration representing applicable audit log types for a context object. The **Audit Logs** tab in Active Workspace provides a segregated view of audit logs in different sections. As an administrator, you can create and configure audit log datasets.

What do audit logs look like?

As a **DBA** user, you can view audit logs using the **Audit Logs** tab in Active Workspace.

The screenshot shows the Siemens Teamcenter Active Workspace interface. The top navigation bar includes 'Search', 'Results', 'Shape', 'Saved', and 'Advanced'. A search bar contains 'awc_audit' and shows '1016 results found for "awc_audit"'. The left sidebar lists several audit log datasets, including 'AWC_Audit_Basic' and 'AWC_Audit_000541'. The main content area displays the 'Audit Logs' tab for the object 'AWC_Audit_Basic/A;33'. It contains two tables: 'Workflow Logs' and 'General Logs'.

Logged Date	Event Type Name	Performer	Object Name	Object Type	Job Name
15-Jan-2017	Complete	am3tester	Add Status Task (TCM R...	EPMAddStatusTask	AWC_Audit_Basic/A;33
15-Jan-2017	Complete	am3tester	TCM Release Process	EPMTask	AWC_Audit_Basic/A;33
15-Jan-2017	End	am3tester	AWC_Audit_Basic/A;33	Job	AWC_Audit_Basic/A;33
15-Jan-2017	Process Initiated	am3tester	AWC_Audit_Basic/A;33	Job	AWC_Audit_Basic/A;33
15-Jan-2017	Add Attachment	am3tester	TCM Release Process	EPMTask	AWC_Audit_Basic/A;33
15-Jan-2017	Assign	am3tester	TCM Release Process	EPMTask	AWC_Audit_Basic/A;33
15-Jan-2017	Start	am3tester	TCM Release Process	EPMTask	AWC_Audit_Basic/A;33
15-Jan-2017	Start	am3tester	AWC_Audit_Basic/A;33	Job	AWC_Audit_Basic/A;33

Logged Date	Event Type Name	User ID	Change ID	Reason	Group Name
15-Jan-2017	_Modify	am3tester			Engineering
15-Jan-2017	_Modify	am3tester			Engineering

What must I install to enable the audit log feature?

To enable the audit log feature, you must install the **Audit** feature during installation.

- If you are using Teamcenter Environment Manager (TEM), select **Audit** on the **Features** panel.
- If you are using Deployment Center, select **Audit** on the **Applications** tab.

What can I configure?

You can configure the following aspects of audit logs:

- **Activate the Audit Log page.**

- **Customize which audit log fields appear to users.**
- **Customize which audit logs appear to users.**

Activating the Audit Log tab

The **AWC_show_audit_logs** preference controls the display of the Audit Log tab in Active Workspace. By default, the value is set to **true** for the **dba** group. As a **DBA** user, you can set the preference for specific groups, roles, and users.

Customize audit logs field display

You can customize which fields are displayed in each audit log. For example, by default the **General Logs** audit log displays the following fields:

- **Logged Date**
- **Event Type Name**
- **User ID**
- **Change ID**
- **Reason**
- **Group Name**
- **Role Name**
- **Secondary Object Display Name**
- **Secondary Object Type**

To remove a field from the display, use the **Arrange** panel to reorder, hide, or display columns.

Audit logs reference

Audit logs track what information has changed and who has changed the information.

You can view the following audit logs using the **Audit Logs** tab:

- **Audit - General Report**
- **Audit - General Sponsored Authentication Report**

- **Audit - File Access Read-Write Report**
- **Audit - File Access Report**
- **Audit - File Access Sponsored Authentication Report**
- **Audit - Security Report**
- **Audit - Schedule Report**
- **Audit - Organization Report**
- **Audit - Digital Signature Report**
- **Audit - License Change Report**
- **Audit - License Export Report**
- **Audit - License Export Sponsored Authentication Report**
- **Audit - License Change Sponsored Authentication Report**
- **Audit - Organization Sponsored Authentication Report**
- **Audit - Structure Sponsored Authentication Report**
- **Audit - Workflow Detailed Report**
- **Audit - Workflow Summary Report**
- **Audit - Workflow Attachment Report**
- **Audit - Workflow Signoff Report**

Customize the audit log display

By default, the following four audit logs are viewable in Active Workspace for **Item**, **ItemRevision**, and its subtypes:

- **Workflow Logs**
- **General Logs**
- **License Export Logs**
- **Structure Logs**

You can customize which audit logs are displayed to users by adding or removing audit logs to customize your XRT pages.

Following is a table that shows the audit dataset for the associated object type.

Object type	Audit dataset
Item/ItemRev and its subtype	AuditLogForItemRev
Workspace object	AuditLogAllForWSO
BOMLine	AuditLogForBOM
Form/Folder/WSO	AuditLogGNForWSO
Dataset	AuditLogForDataset
User/Group/Project	AuditLogForUserContext
GroupMember/Person/ Role/Site/Volume/ TCCalendar	AuditLogForOrg
Schedule, ScheduleTask	AuditLogForSchedule
SchDeliverable/SchTaskDeliverable/ ScheduleMember	AuditLogForScheduleMgmt
EPMJob/EPMTask/ PerformSignoffTask	AuditLogForWorkflow
Scp0Regulation/ Scp0SubstanceCmplResult/ Scp0Exemption/ Mat1Substance	AuditLogForSubscmpl

To customize which audit logs can be viewed:

1. In **XRT Editor**, open the XRT page that you want to modify, for example, **Awp0DocumentRevSummary**, and add the audit log to your custom XRT page by inserting an **inject** statement for the audit log you want to add.
2. Save the file with a new name.

Note:

Siemens Digital Industries Software recommends you rename your edited file before saving changes to retain the default file.

Configure user logging

Users can enable logging when experiencing performance, memory, or other issues. You can configure the following aspects of this logging.

Configure client log notifications

Specify the email addresses to be notified when log files are created using the **TC_reactive_logging_notification_list** preference.

Allow users to download log files

Allow the logging user to download log files by setting **TC_reactive_logging_file_download** to **true**. (The default value is **false**.)

12. Assistant configuration

Assistant configuration tasks

What is the Assistant?

The Assistant suggests the next possible actions to perform and provides the relevant data required to perform them. These suggestions are based on the context, history, and usage frequency of actions performed by other previous users belonging to the same role and group.

What can I configure?

You can **configure the Assistant panel** using the provided site and user preferences.

What do I need to do before configuring?

Before you can configure the **Assistant** panel, you must **install the features and the microservice**. Install the following from the **Features** panel of Teamcenter Environment Manager (TEM):

- **Assistant** (server)

Installs the server-side definitions for the Assistant.

Select the **Active Workspace**→**Server**→**Active Workspace Assistant** feature in the corporate server.

- **Assistant** (client)

Installs the user interface elements for the Assistant.

Select **Active Workspace**→**Client**→**Active Workspace Assistant**.

- **Command Prediction Services** (microservice)

Installs the microservice for the Assistant.

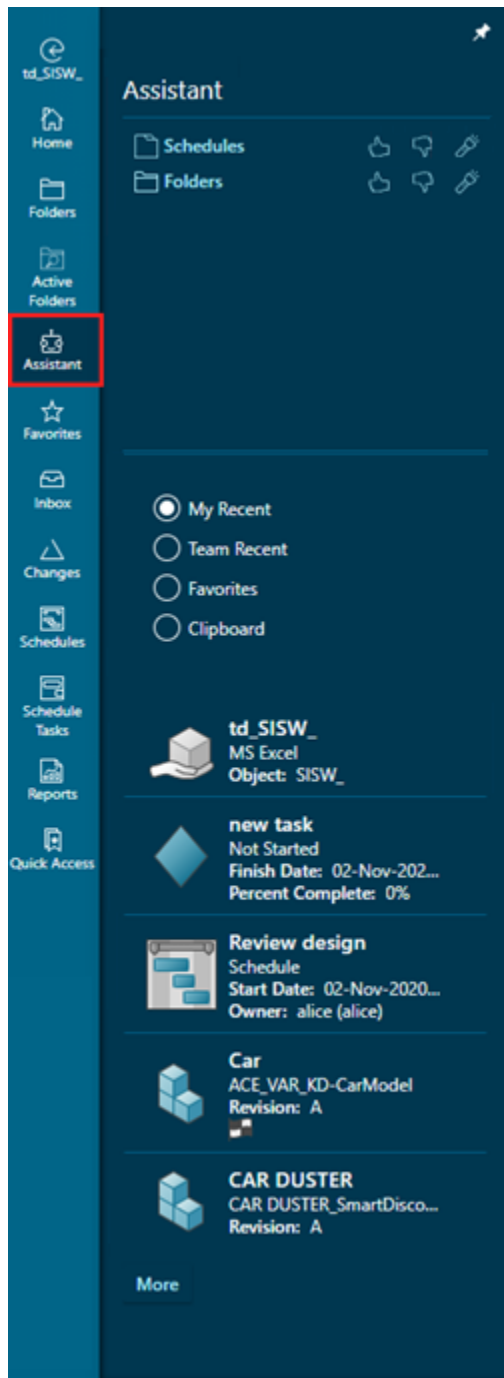
Select **Microservices**→**Command Prediction Services**.

Where can I find out more about the Assistant?

See The Teamcenter Assistant in the help.

What does the Assistant panel look like?

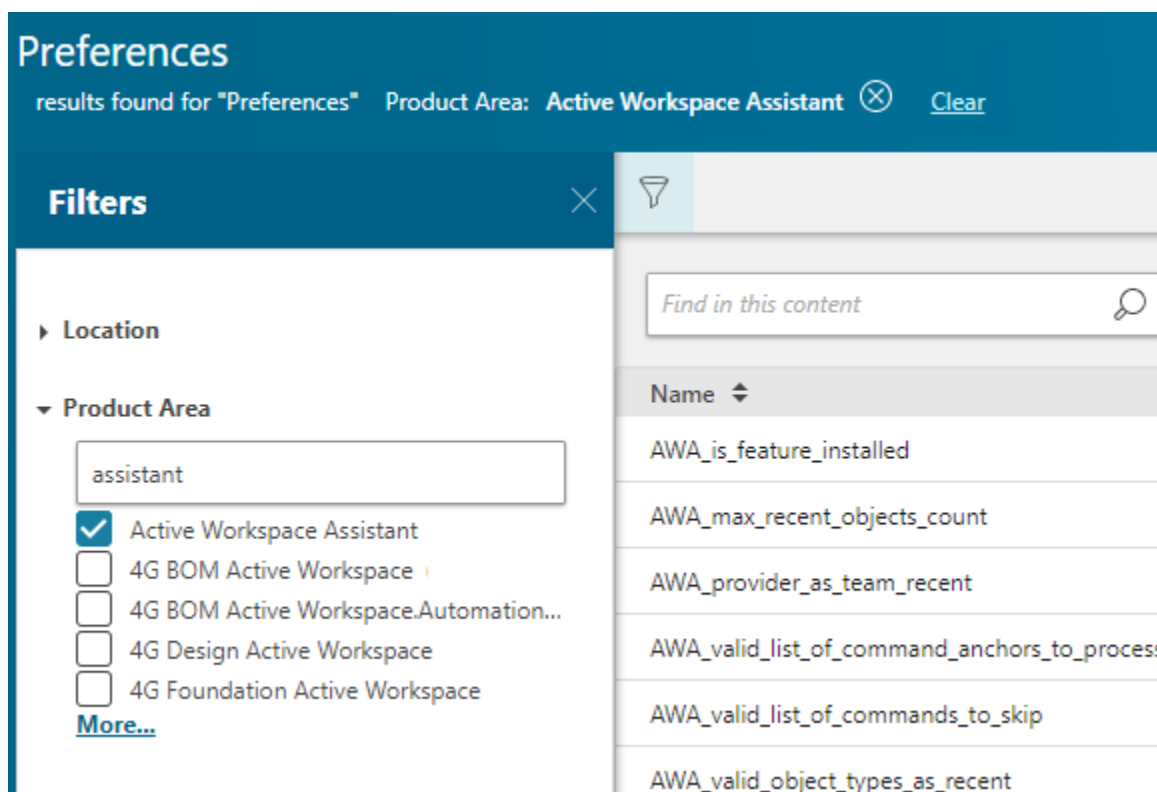
Following is an example of the **Assistant** panel in Active Workspace.



Configuring the Assistant panel

You can manage the settings for the **Assistant** panel using the provided site and user preferences. The availability of options such as **Like**, **Dislike**, **Show**, or **Tutor Mode** can also be configured using preferences.

Find the list and description of these preferences in the **Active Workspace Assistant** product area.



Where can I get a list of preferences?

Install the Assistant

The **Assistant** uses the **Command Prediction Service** microservice, which stores data in a dedicated database that you configure during installation.

This procedure assumes that Teamcenter and Active Workspace, including the microservice framework, are installed.

1. Ensure your environment includes the Teamcenter Foundation, Active Workspace, and Teamcenter Microservice Framework software.
 - Deployment Center
 - In the **Software** step, select the software kits.
 - TEM
 - In the **Media Locations** panel, specify locations of the software kits. If you use a minor release version of Teamcenter, make sure you include the major *and* minor release software kits.
2. Add the Assistant to your environment:

- Deployment Center
 - In the **Applications** step, select the **Active Workspace Assistant** application.
 - In the **Components** step, select the **Command Prediction Service Configuration** component.
- TEM
 - In your server *and* client configurations, add the corresponding server and client **Active Workspace Assistant** feature.
 - In your microservice framework configuration, add the **Command Prediction Service** microservice.

For information about the TEM panels, in TEM, see the **Help** button

3. Complete the installation as appropriate for your installation tool.

Note:

- When you configure the database for the **Command Prediction Service** microservice, make sure you have the database system user credentials, and make sure you do not use the same tablespace as the Teamcenter database. The PredictiveUI user, database, and tablespace must be different from the Teamcenter and Server Manager database.

If you choose to use an existing database for the **Command Prediction Service** microservice, make sure that you **manually create the database** before the installation.

- Due to the architectural changes in the Active Workspace client in Active Workspace 6.2 release, the command usage data required for the Teamcenter Assistant is not available and it has to be retrained. Active Workspace users will not be able to leverage the data that was created in releases prior to Active Workspace 6.2.

Manually create a database for the Command Prediction Service microservice

If you choose to use an existing database for the Command Prediction Service microservice, you must create the database using template scripts prior to install.

These scripts can be modified to create the database user, the tablespace, and the schema that includes the tables and indexes. Apart from changing tokens to appropriate values, and minor syntax changes, no other changes to the scripts are advisable.

Manually create an Oracle database for the Command Prediction Service microservice

1. Copy the template scripts from
`<OS>\tc\microservices\commandprediction_service.zip\tc\db_scripts\<database>.`

2. Rename them as .sql files.

Following SQL scripts must be used for on-premise Oracle database:

- `oracle_create_predui.sql.template` - Use this script to create the base schema database.
- `oracle_update_predui.sql.template` - Use this script to update the base schema database.

Following SQL scripts must be used for cloud hosted Oracle database:

- `oracle_rds_create_predui.sql.template` - Use this script to create the base schema database.
- `oracle_update_predui.sql.template` - Use this script to update the base schema database.

3. Replace the tokens to appropriate values.

The following table describes the database parameters to be replaced in the template. Additionally, the comments in the template file also indicate the values that must be replaced.

Parameter	Example value	Description
<code>@TABLESPACE_NAME@</code>	<code>predictivedb</code>	Name of the tablespace.
<code>@DB_PATH@</code>	<code>D:\oracle\data</code>	Path to the directory in which the data file will reside.
<code>@DB_USER@</code>	<code>predictivedbuser</code>	Database log in name for the Teamcenter database.
<code>@DB_PWD@</code>	<code>predictivedbpw</code>	The password for the database user.

4. Save the modifications to the scripts and execute them.

If syntax errors occur while executing the database scripts, the following additional changes are required:

- On the last line in the SQL script, delete one semicolon after END statement.
- Make sure that the last line of the SQL script ends with the `/` character. If not, you must insert the `/` character on a new line after the last line of the script

The last two lines of the SQL script must appear as shown below:

```
;END;
```

```
/
```

Note:

For a complete database setup, it is necessary to execute both the *create* and *update* scripts after they are modified.

Manually create an MSSQL database for the Command Prediction Service microservice

1. Copy the appropriate template scripts from
`<OS>\tc\microservices\commandprediction_service.zip\tc\db_scripts\<database>`.
2. Rename them as .sql files.

The following SQL scripts must be used for the SQL Server database:

- `mssql_create_predui.sql.template` - Use this script to create the base schema for the SQL Server database.
- `mssql_update_predui.sql.template` - Use this script to update the base schema database.

3. Replace the tokens to appropriate values.

For more information, see [Create an SQL Server database \(Windows\)](#) or [Create an SQL Server database \(Linux\)](#).

4. Save the modifications to the scripts and execute them.
 - To execute the script on the command line, enter the following:

```
> sqlcmd -i mssql_create_predui.sql
```

```
> sqlcmd -i mssql_update_predui.sql
```

For detailed information and help about the `sqlcmd` utility and its arguments, run the utility as follows:

```
sqlcmd -h
```

- To execute the scripts using the SQL Server Management Studio, do the following:
 - a. Open the `mssql_create_predui.sql` file in Microsoft SQL Server Management Studio.
 - b. In the **SQL Editor** toolbar, click **Execute**, or choose **Query→Execute** to start creating the database.

- c. Open the *mssql_update_predui.sql* file in Microsoft SQL Server Management Studio.
- d. In the **SQL Editor** toolbar, click **Execute**, or choose **Query→Execute** to start updating the base schema database.

Note:

For a complete database setup, it is necessary to execute both the *create* and *update* scripts after they are modified.

Migrate the Oracle database for the Command Prediction Service microservice

When you upgrade from Teamcenter 13.x / 14.x to Teamcenter 2412, you must run scripts to migrate the existing data, so that the Teamcenter Assistant can store the history of clicks performed by users and predict the next command successfully.

Note:

These migration scripts migrate the command data for Teamcenter Assistant. The scripts do not migrate the actual commands.

Prerequisites

- You must migrate the commands and then run the migration scripts.
- Changes required for custom commands in updated UI should be completed
- Create the following lists of custom group commands that you want to migrate:
 - A list of custom group commands that will be visible when you click ... or **More** in the work area toolbar or primary toolbar.
 - A list of custom group commands that will be visible directly on the work area toolbar or primary toolbar.

Procedure

1. Copy the template scripts from
`<OS>\tc\microservices\commandprediction_service.zip\tc\db_scripts\tc2312_migration\oracle.`
2. To migrate the data generated for commands that are included with the default Teamcenter installation, run the *assistant_tc2312_migration_script_oracle.sql* script as follows:

```
sqlplus -s username/password@database
@assistant_tc2312_migration_script_oracle.sql
```

3. To migrate the data generated for custom group commands created by the user that were originally present on the primary toolbar on the right, do the following:
 - a. Modify the tokens in the *assistant_tc2312_custom_migration_script_oracle.sql* as follows:

Token	Example value	Description
@@LIST_OF_ANCHORS_FOR_WORKAREA_TOOLBAR_MORE@@	'Group1','Group2', 'Group3'	List of custom command groups that will be migrated to "More" group in workarea toolbar.
@@LIST_OF_ANCHORS_FOR_WORKAREA_TOOLBAR_MORE@@	'Group1','Group2', 'Group3'	List of custom command groups that will be migrated to group in workarea toolbar.

Note:

You do not need to run this script for custom commands, this script is needed only for the custom group commands.

- b. Save the modifications and run the script use the following command:

```
sqlplus -s username/password@database
@assistant_tc2312_custom_migration_script_oracle.sql
```

Migrate the MSSQL database for the Command Prediction Service microservice

When you upgrade from Teamcenter 13.x / 14.x to Teamcenter 2412, you must run scripts to migrate the existing data, so that the Teamcenter Assistant can store the history of clicks performed by users and predict the next command successfully.

Note:

These migration scripts migrate the command data for Teamcenter Assistant. The scripts do not migrate the actual commands.

Prerequisites

- You must migrate the commands and then run the migration scripts.
- Changes required for custom commands in updated UI should be completed
- Create the following lists of custom group commands that you want to migrate:
 - A list of custom group commands that will be visible when you click ... or **More** in the work area toolbar or primary toolbar.
 - A list of custom group commands that will be visible directly on the work area toolbar or primary toolbar.

Procedure

1. Copy the template scripts from
`<OS>\tc\microservices\commandprediction_service.zip\tc\db_scripts\tc2312_migration\mssql.`
2. To migrate the data generated for commands that are included with the default Teamcenter installation, run the *assistant_tc2312_migration_script_mssql.sql* script using one of the following methods:
 - To run the script on the command line, use the following command:


```
sqlcmd -S database -U user -P password -i
assistant_tc2312_migration_script_mssql.sql
```

OR
 - To run the script using the SQL Server Management Studio, open the *assistant_tc2312_migration_script_mssql.sql* file in Microsoft SQL Server Management Studio, and in the **SQL Editor** toolbar, click **Query** → **Execute**.
3. To migrate the data generated for custom group commands created by the user that were originally present on the primary toolbar on the right, do the following:
 - a. Modify the tokens in the *assistant_tc2312_custom_migration_script_oracle.sql* as follows:

Token	Example value	Description
@@LIST_OF_ANCHORS_FOR_WORKAREA_TOOLBAR_MORE@@	'Group1','Group2', 'Group3'	List of custom command groups that will be migrated to the More group in work area toolbar.
@@LIST_OF_ANCHORS_FOR_WORKAREA_TOOLBAR_MORE@@	'Group1','Group2', 'Group3'	List of custom command groups that will be migrated to a group in work area toolbar.

Note:

You do not need to run this script for custom commands, this script is needed only for the custom group commands.

b. Save the modifications and run the script using one of the following methods:

- To run the script on the command line, use the following command:

```
sqlcmd -S database -U user -P password -i
assistant_tc2312_custom_migration_script_mssql.sql
```

OR

- To run the script using the SQL Server Management Studio, open the *assistant_tc2312_custom_migration_script_mssql.sql* file in Microsoft SQL Server Management Studio, and in the **SQL Editor** toolbar, click **Query** → **Execute**.

Managing user passwords using the CPS_manage_password utility

The **CPS_manage_password** password management utility is used for generating an encoded database user password for the Command Prediction Service database. Changing the database user password for Command Prediction Microservice database requires updating the same in the microservice to be able to connect to the database. Failure to update the password results in database connection failure. This utility encodes the given input string, which can be replaced in the *pred.pwd* file.

The Command Prediction Microservice reads the database user password from the *pred.pwd* file for connecting to the Command Prediction Service database.

CPS_manage_password

The following options are available:

- **-h | -help** - Displays help for the utility.
- **-encode** - Encodes input string.
- **-input=** - Input string.

Manage user passwords using the CPS_manage_password utility on Windows

1. Traverse to the location `<microservice_root>\commandprediction-<service_version>` and execute the following command:

```
CPS_manage_password -encode -input=<input_string>
```

2. Copy the generated string with the above command and update the *pred.pwd* file to replace the existing string with newly generated string.
3. Restart the **Teamcenter Process Manager** to enable the microservice to read the updated password from the file.

Manage user passwords using the CPS_manage_password utility on Linux

1. Get the container ID of the running container for the Command Prediction Service.

```
docker container ls
```

2. Generate a new encoded password string by executing the utility inside the container.

```
docker exec -it <container_id> /app/CPS_manage_password -encode  
-input=<input_password_string_to_encode>
```

Retain the generated string for further updates in *pred.pwd*.

3. Stop the running container.

```
docker stop <container_id>
```

4. Get the Command Prediction Service ID.

```
docker service ls
```

5. Remove the Command Prediction Service.

```
docker service rm <service_id>
```

6. Get the secret ID for *pred.pwd*.

```
docker secret ls
```

7. Remove the secret ID for *pred.pwd*.

```
docker secret rm <secret_id>
```

8. Update the newly generated password string in *pred.pwd* located at *<microservice_container_root>/secrets*.

9. Go to the location *<microservice_container_root>* and execute the following command to update and redeploy the service.

```
docker stack deploy -c commandprediction.yml <stack_name>
```

Ensure that the new docker secret is created.

Migrating user data using the **CPS_migrate_user_data** utility

The **CPS_migrate_user_data** utility migrates MD5 hashed data for users, groups, and roles to SHA256 hashed data in the Command Prediction Service database. Before running this utility, you must use the **admin_data_export** utility export the Teamcenter user, role, and group data to a file.

```
CPS_migrate_user_data [OPTION]
```

The following options are available:

-dbplat - Database platform used for Command Prediction Services such as MSSQL, ORACLE, or POSTGRESQL.

-username - Database username.

-password - Database password.

-hostname - Database hostname.

-port - Database port.

-namedinstance - For MSSQL platform, connect with named instance instead of database port.

-instance - Command Prediction Service database instance name. In the case of the Oracle database, the instance is SID, while in the case of PostgreSQL and MSSQL databases, the instance is the name of the database

-inputfile - Full file path of the *Organization.XML* file generated using the Teamcenter **admin_data_export** utility.

-batchsize - Number of user profiles to be updated during the user data migration run. If not provided, the utility migrates the data of all the users.

Migrate user data using the CPS_migrate_user_data utility

1. Run the **admin_data_export** utility to export Teamcenter user, group and role data.

For Windows

- a. Open the Teamcenter command prompt.
- b. Run the following command:

```
admin_data_export -u=<tc_user> -p=<password> -g=dba
-adminDataTypes=Organization -outputPackage=<export_file.zip>
```

For Linux

- a. Open the terminal at <TC_ROOT>/bin.
- b. Set TC_ROOT, TC_DATA and run <TC_DATA>/tc_profilevars for the terminal.
- c. Run the following command:

```
/admin_data_export.sh -u=<tc_user> -p=<password> -g=dba
-adminDataTypes=Organization -outputPackage=<export_file.zip>
```

2. Extract the newly created ZIP file and locate the *Organization.XML* file in the <extracted_folder>\ADMINISTRATION_DATA\Organization folder.
3. Run the **CPS_migrate_user_data** utility to migrate the data.

For Windows

- a. Open the Command prompt at the <Microservice_Root>\CommandPrediction-<version> folder.
- b. Run the following command:

```
CPS_migrate_user_data -dbplat=<db_platform>
-hostname=<db_hostname> -username=<db_user>
```

```
-password=<db_password> -port=<db_port>
-instance=<db_instance> -inputfile=<path_to_organization_xml>
```

The log file is created in the <Microservice_Root>\CommandPrediction-<version>\Logs\UserDataMigration folder.

For Linux

- a. Get the running container ID for the Command Prediction Service.

```
docker container ls
```

- b. Copy the *Organization.XML* file into the container.

```
docker cp <path_to_oraganization_XML_on_host>
<command_prediction_container_id>:/app/
```

- c. Run the following command:

```
ocker exec -it <command_prediction_container_id> ./
CPS_migrate_user_data -dbplat=<db_platform>
-hostname=<db_hostname> -username=<db_user>
-password=<db_password> -port=<db_port>
-instance=<db_instance> -inputfile=/app/Organization.xml
```

The log file is created in a docker container in the /tmp/Logs/UserDataMigration folder.

13. Configuring units of measure

Introduction to units of measure

A *unit of measure* is a quantifiable measurement category such as length, volume, mass, angle, pressure, and so on. A unit of measure may be needed to define an accurate bill of materials (BOM).

If unit of measure is not specified for a part, the unit *each* is implied, meaning the **quantity** parameter value specifies the number of discrete component parts.

When adding or modifying a part, a user can choose from a list of predefined values for the **Unit of Measure** parameter. An installer or system administrator defines units of measure so that parts can be expressed in standardized units across an entire Teamcenter site. A unit of measure can be a compound such as microjoule per kilogram kelvin.

Once a unit of measure and a quantity are specified for a part, if a user sets a preferred unit of measure for a category, in the Parameters application the unit and value are displayed in the preferred unit of measure.

As of Teamcenter 2406, available unit of measure values are defined using the Unit Management System. For more information, see [Configure units of measure](#).

Configure available units of measure

Administrators can configure the units of measure that are available across Teamcenter. This unit management system (UMS) based on classified units is first available in Teamcenter version 2406. A list of nearly 1400 **units of measure** is provided in the Teamcenter installation kit. Out of the box, only a subset of the units are installed, depending on the applications installed. Installers and administrators can configure the units to add, and can add new units.

1. In the Teamcenter kit or in the `TC_ROOT\TD` folder, find and open `unit_definitions.csv`.

Application columns begin with column Q (**BASE**). Unit rows for which an application column has the value **1** will be considered by the `ums_import_unit_definitions` utility.

Note:

During initial installation, if an application is installed in the environment, unit rows for which the application column has the value **1** are imported to the database.

2. Edit the file to ensure that the unit definitions you want to administer have the value **1** for an application.

In the event that no class definition exists for a unit you want in your environment, create a new row.

- Using the edited *unit_definitions.csv*, run the `ums_import_unit_definitions` utility.

Migrate legacy units of measure to the unified measurement system

As of version 2406, Teamcenter has an updated units of measure capability. Following upgrade of a Teamcenter installation from a version prior to 2406, an administrator can migrate legacy BMIDE units of measure to the new unit management system (UMS) using the **ums_mapping** utility. The utility links legacy units to unit definitions in the UMS. If necessary, the linking and mapping can later be reverted.

Configure units of measure relationships

Use the following steps to create a mapping configuration file that identifies the relationship between legacy units of measure and unit definitions in the unified measurement system.

- In a Teamcenter command window, run the **ums_mapping** utility with the **-analyze -report_file** option. For example,

```
ums_mapping -u=username -p=password -analyze
-report_file=c:\temp\mapping_config.csv
```

The utility examines the Teamcenter database for existing UMS unit definitions. The utility then creates mappings between the legacy units of measure to existing unit of measure definitions in the UMS. These mappings are created based on unit names and symbols. The utility generates the *mapping_config.csv* file showing these mappings along with units for which it could not create mappings between the systems. The file contains the following information for each unit.

Column	Definition
Is Approved	Remove the # character in this column to process this unit when using -map , -unMap , or -createMap .
UOM Name	The name of the unit in the legacy system.
UOM Symbol	The symbol of the unit in the legacy system.
In Use	A value of Yes indicates that this unit is used by an object in the legacy system.
Action	The utility action performed to create the file.
Status	The unit's mapping state at the time the file was created.
Details	A description of the unit's state and any recommended actions.
Unit Definition ID	The ID of the unit in the UMS.
Unit Definition Name	The name of the unit in the UMS.
Unit Definition Symbol	The symbol of the unit in the UMS.

In the command window, the utility also summarizes the unit mappings between units in the legacy and unified measurement systems.

2. Using an ASCII editor or Microsoft Excel, review the mappings in *mapping_config.csv*. Approve proposed matches by removing the "#" symbol from the first column of that line of the file. Units have the following mapping states.

Existing-mapped	Units with this state already exist in the UMS. No further action is needed for these.
Candidate-match	The utility proposes a mapping for the unit of measure. Legacy units with this status completely or partially match unit definitions in the UMS.
Multiple-match	The legacy unit of measure is a candidate to match more than one unit definition in the UMS. Determine which definition is the correct mapping for the unit and update the .csv file accordingly.
	A legacy unit of measure can map to only one unit definition in the UMS. Multiple legacy units of measure can map to a single unit definition in the UMS.
Non-matching	The utility could not determine a candidate mapping for the unit of measure or the UMS unit definition.
	If there is no matching UMS unit for a unit of measure, manually review the UMS unit definitions in the .csv file for an appropriate definition. Update the column information to map the unit of measure to this unit definition. If you cannot find a match, use the ums_import_unit_definitions utility to create an appropriate unit definition in the UMS and re-run this process to approve the mapping of the unit of measure to this new unit definition in the UMS.
	If there is no matching unit for a UMS unit definition, follow the steps in <i>Create units of measure</i> later in this topic.
NA	This unit of measure is specific to Classification and is not mapped. No further action is needed for these.

3. Once you have reviewed, revised, and approved the units of measure mappings, perform a dry run of linking the approved units of measure to unit definitions in the UMS. Use a command of the following form

```
ums_mapping -u=username -p=password
-analyze -input_file=c:\temp\mapping_config.csv
-report_file=c:\temp\analyze_mapping_config.csv
```

The utility generates a .csv report file that shows the results that would have occurred in and actual linking of the units approved in the mapping configuration file. Review these results to ensure the linking will occur as you expect. Address any failures or other issues by updating the configuration file and performing additional dry runs.

Add the unit of measure relationships to the Teamcenter database

Once you have confirmed the units of measure are mapped properly in the mapping configuration file, you are ready to add the relationships to the Teamcenter database. Do so using a command of the following form. (You must be a Teamcenter administrator who is a member of a group with DBA privileges to run this command.)

```
ums_mapping -map -input_file=c:\temp\mapping_config.csv
-report_file=c:\temp\mapping_report.csv
```

This example updates the Teamcenter database with the relationships defined in *mapping_config.csv*. Approved unit relationships in the file (lines with no "#" symbol in the first column) are added to the Teamcenter database. The utility generates the report *mapping_report.csv* listing the updates and their completion status.

Remove unit of measure relationships from the Teamcenter database

If you want to remove one or more relationships from the Teamcenter database, you can do so using the **ums_mapping** utility. You must be a Teamcenter administrator who is a member of a group with DBA privileges to remove these relationships.

1. To remove a limited number of mapping relationships, update the mapping configuration file (*mapping_config.csv* in the earlier example) by removing the "#" symbol from the first column of each unit for which you want to remove the mapping relationship.
2. In a Teamcenter command window, run the **ums_mapping** utility with the **-unmap** option. For example,

```
ums_mapping -u=admin-username -p=admin-password
-g=dba -unmap -input_file=c:\temp\mapping_config.csv
-report_file=c:\temp\mapping_report.csv
```

The utility generates the report *mapping_report.csv* listing the removed unit relationships.

In cases where you want to remove several existing units of measure relationships from the Teamcenter database, consider using the **ums_mapping** utility with the **-all** option. Doing so removes all existing units of measure relationships from the Teamcenter database. After removing all relationships, restart migrating legacy units of measure as described earlier.

To remove all existing units of measure relationships from the Teamcenter database, use a command of the following form.

```
ums_mapping -u=admin-username -p=admin-password -g=dba -unmap -all
-report_file=c:\temp\mapping_report.csv
```

The utility generates the report *mapping_report.csv* listing the removed unit relationships.

Create units of measure

If the UMS contains definitions with no related unit of measure, create units of measure for the definitions using the **ums_mapping** utility with the **-create_uom** option. Doing so creates new units of measure with relationships to the unit definitions previously with no related definitions. These relationships are automatically added to the Teamcenter database.

In a Teamcenter command window, enter a command of the following form:

```
ums_mapping -u=admin-username -p=admin-password -g=dba -create_uom
-report_file=c:\temp\unit_report.csv
```

The utility analyzes the UMS, creates new units for definitions without related units, and generates the report *unit_report.csv* listing the created units.

Working with the unit of measure definitions file

The unit of measure definitions file specifies the attributes of each unit of measure in the Teamcenter unit management system (UMS), combining aspects of Teamcenter units of measure and the Classification unit management system. During Teamcenter installation and upgrade, the definitions in this file are used to create the units of measure in the UMS. Teamcenter administrators can also use this file with the **ums_import_unit_definitions** utility to manage units of measure.

The unit of measure definitions file supplied with Teamcenter is *<TC_DATA>\unit_definitions.csv*. Each line in the file defines a particular unit of measure. Columns define the attributes for each unit of measure.

Warning:

When editing the **unit_definitions.csv**, ensure that you do not change existing **Conversion multiplication factor** and **Conversion addition factor** values or enter inaccurate values for these attributes when creating new unit definitions. Inaccurate values for these attributes will cause compromised data when converting between measurement systems.

Make sure you back up your .csv file before making any changes to the file.

As an authoring aid, an annotated .xls version of this .csv file is also available in the *<TC_DATA>* directory. Instead of directly editing **unit_definitions.csv**, consider editing **unit_definitions.xls**. When your edits are complete, save **unit_definitions.xls** as a comma-delimited .csv file, remove the first line (the topmost header row) of the new .csv file, and resave the file. You can then use this .csv file to create and manage the units of measure.

Following is a description of the *unit_definitions.csv* file contents.

Classification Properties

Classification properties define the attributes of a unit of measure and how it relates to other units of measure of the same type.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Classification Properties													
2	ID	Master	Quantity	Unit name	Symbol	System of Measurement	Base unit	Conversion multiplication factor	Conversion addition factor	Ignore for Optimization	Number of Decimal Places	NX Unit ID	NX Display	ECLASS IRDI

ID

A unique identifier for the unit of measure. ID is referred to as the "Object ID" attribute in Classification.

Master

A unit of measure can be associated with only one UMS object (a Classification unit definition). When multiple units have the same dimensionality (see *Dimensionality* later in this topic), only one of these units can be designated as the unit associated with the UMS object. Designate this unit with a value of **Y** in this column. Set this column value for the other units using the same dimensionality to **N**.

Quantity

The type of measurement, for example, length, mass, temperature, and so on. Quantity is referred to as the "Measure" attribute in Classification.

Unit name

The name of the unit as displayed in Teamcenter.

Symbol

The scientific symbol representing the unit of measure. Symbols must be unique and used only once in the file. Symbol is referred to as the "Unit display name" attribute in Classification.

System of Measurement

The measurement system the unit belongs to. If the unit of measure is a metric unit, identify it as such with a value of **m**. Identify all other units with a value of **nm** (non-metric).

Base unit

The unit of measure within a quantity that is used to calculate the values of other units of measure in that quantity. Identify the base unit of measure with a value of **Y**. Identify all other units of measure in the same quantity with a value of **N**.

Conversion multiplication factor

The value by which the quantity's base unit must be multiplied to derive this unit of measure's value.

Warning:

Ensure that you do not change existing **Conversion multiplication factor** and **Conversion addition factor** values or enter inaccurate values for these attributes when creating new unit definitions. Inaccurate values for these attributes will cause compromised data when converting between measurement systems.

Conversion addition factor

Any value that must be added after applying the conversion multiplication factor. Conversion addition factors typically apply only to temperature units of measure.

Warning:

Ensure that you do not change existing **Conversion multiplication factor** and **Conversion addition factor** values or enter inaccurate values for these attributes when creating new unit definitions. Inaccurate values for these attributes will cause compromised data when converting between measurement systems.

Ignore for Optimization

Specifies whether this unit of measure should not have its value optimized in Classification. Set this value to **Y** to exclude this unit of measure from optimization. Set this value to **N** to include this unit of measure for optimization.

Number of Decimal Places

The number of digits displayed to the right of the decimal when using the Classification application.

NX Unit ID

The ID value for this unit of measure in NX.

NX Display

Specifies that the unit is part of the list of units shown in NX when creating a part.

ECLASS IRDI

The ECLASS International Registration Data Identifier (IRDI) value for this unit of measure.

Display Set

Identifies units as members of specific display set (on the MBSE **Parameters** tab). Set this value to **Y** to include this unit a particular display set. Set this value to **N** to exclude this unit from a particular display set. When multiple units have the same dimensionality, only 1 of the units can be included in a particular display set.

O	P	Q
Display Set		
Metric Display Set		
Metric(SI) Display Set		
Non-Metric Display Set		

The following display sets are provided with Teamcenter. You can edit these sets and create new display sets.

Metric Display Set

Metric(SI) Display Set

Non-Metric Display Set

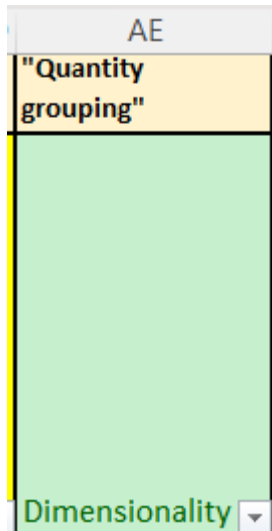
Base install and application specific units

When installing Teamcenter, certain basic units and units related to specific detected applications are installed. These units are identified with a value of 1 in their respective columns. The default units identified for installation are a balance between necessity and volume of units. Enabling all units may be overwhelming to users in their everyday work. Edit these values as necessary for your site during installation.

R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
Base install, and application specific units												
Application:Base	Application:CLS	Application:NX	Application:NX_ALL	Application:NxORIG	Application:frnfoundation	Application:hrm	Application:CPG	Application:retail	Application:MedicalDevice	Application:TcQualityCenter	Application:TcX	Application:EDA

Quantity grouping

Dimensionality is a comma-separated list of values representing the physical properties of the unit.



A *dimension* is a property that you can measure. A *unit* is a way to assign a value to the dimension. For example, length is a dimension, but you measure it in units of feet or meters. You can measure each property with a variety of units, such as meter and inch, but each property can have only one dimension, such as length.

The International System of Units (SI) defines the following seven base properties.

Base Property	Metric Unit	English Unit
Length	Meter (m)	Inch (in)
Mass	Kilogram (kg)	Pound (lb)
Time	Second (s)	Second (s)
Electric Current	Ampere (A)	Ampere (A)
Temperature	Kelvin (K)	Degree Fahrenheit (°F)
Luminous Intensity	Candela (cd)	Candela (cd)
Amount of Substance	Mole (mol)	Mole (mol)

Using **Dimensionality**, you can express most of the physical properties as a combination of the base properties. For example, you can express Force as a combination of mass, length, and time as follows:

$$\text{Force} = \text{Mass} \times \text{Length} / \text{Time}^2$$

Therefore, the dimensionality of Force is ML/T^2 .

In the **Dimensionality** column, the dimensionality is represented by nine integers, each representing the exponent of one of the following basic properties:

Length

Time

Temperature

Electric charge

Angle

Delta temperature

Luminous intensity

Amount of substance

For example, the dimensionality of Force is represented as **1,1,-2,0,0,0,0,0,0**.

ums_import_unit_definitions

Creates, updates, and configures the display of Teamcenter units of measure.

ums_import_unit_definitions runs automatically during Teamcenter installation and can subsequently be run by a Teamcenter administrator to manage units of measure. Siemens Digital Industries Software strongly recommends backing up any current unit definitions file before modifying the file.

Units of measure are created and managed based on unit of measure definitions specified in a comma-delimited .csv file. The initial unit of measure definitions file supplied with Teamcenter is `<TC_DATA>\unit_definitions.csv`. See Working with the unit definitions file for a description of this file.

After running **ums_import_unit_definitions**, ensure users log out of Teamcenter start new sessions to use the updated units of measure.

Use the **ums_mapping** utility to map and add legacy units of measure to the Unit Management System.

SYNTAX

```
ums_import_unit_definitions
[-u=user-id {-p=password | -pf=password-file} -g=group]
-file=unit_def_file
-applications={app_name1,app_name_2,... | all}
[-allowOverwrite]
[[-createUOM] [-updateUOM]] | [-hideUOM] | [-showUOM]
[-dryRun]
[-h]
```

ARGUMENTS

-u

Specifies the user ID. The user must have Teamcenter administration privileges and be a member of a group with **DBA** privileges.

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.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

For more information about managing password files, see [Manage password files](#).

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user. This group must have **DBA** privileges.

If used without a value, the user's default group is assumed.

-file

The comma-delimited .csv file (*unit_def_file*) containing unit of measure definitions.

-allowOverwrite

Optional. Overwrite existing units of measure based on the definitions in *unit_def_file*.

-applications

Restrict updates to one or more applications (as named in *unit_def_file*) containing unit of measure definitions. *app_name* is the string following **Application:** in *unit_def_file*. For example, set *app_name* to **NX** for the column **Application:NX**.

Separate multiple application names with commas. Set **-applications** to **all** to process all applications defined in *unit_def_file*.

-createUOM

Optional. Create the units of measure defined in *unit_def_file*. If **-allowOverwrite** is not specified, only new units of measure are created. If **-allowOverwrite** is specified, existing units of measure are overwritten based on the definitions in *unit_def_file* as well. **-createUOM** cannot be used with **-hideUOM** or **-showUOM**.

If **-createUOM**, **-updateUOM**, **-hideUOM**, or **-showUOM** are not specified, **-createUOM** is assumed.

-updateUOM

Optional. Update existing units of measure with the definitions in *unit_def_file*. Units of measure defined in *unit_def_file* that do not currently exist will not be created. **-updateUOM** cannot be used with **-hideUOM** or **-showUOM**.

-hideUOM

Optional. Suppress existing units of measure defined in *unit_def_file* from appearing in Teamcenter and application lists of values. These units of measure are not removed from Teamcenter. **-hideUOM** cannot be used with **-createUOM**, **-updateUOM**, or **-showUOM**.

-showUOM

Optional. Show units of measure defined in *unit_def_file* that have previously been hidden using **-hideUOM**. **-showUOM** cannot be used with **-createUOM**, **-updateUOM**, or **-hideUOM**.

-dryRun

Optional. Run **ums_import_unit_definitions** with the specified parameters without actually making changes to the units of measure. The results of running the utility are displayed in the command window, but no units of measure changes are made.

Siemens Digital Industries Software strongly recommends using **-dryRun** to validate changes before actually creating or updating your UMS definitions.

-h

Displays help for this utility.

ENVIRONMENT

As specified in Manually configure the Teamcenter environment.

FILES

As specified in Log files produced by Teamcenter.

RESTRICTIONS

None.

EXAMPLES

- Create units of measure for all units of measure defined in a .csv file. All existing units of measure are updated with any updated definitions.

```
ums_import_unit_definitions -file=c:\temp\unit_defs.csv
                             -applications=all
```

- For units of measure that are members of the Base application, update existing units of measure with changed definitions.

```
ums_import_unit_definitions -file=c:\temp\unit_defs.csv
                             -applications=Base -allowOverwrite -updateUOM
```

- Suppress units of measure in the Base application from appearing in lists of values.

```
ums_import_unit_definitions -file=c:\temp\unit_defs.csv
                             -applications=Base -hideUOM
```

- Display units of measure in the Base application in lists of values.

```
ums_import_unit_definitions -file=unit_definitions.csv
                             -applications=Base -showUOM
```

ums_mapping

As of version 2406, Teamcenter employs an updated unit management system (UMS). Following an upgrade of a Teamcenter installation from a version prior to 2406, use this utility to map legacy BMIDE units of measure to definitions in the UMS.

SYNTAX

```
ums_mapping [-u=user-id {-p=password | -pf=password-file} -g=group]
[-analyze {-report_file= | -input_file=}config_file] |
[-map -input_file=config_file -report_file=map_report_file] |
[-unmap {-input_file=config_file | -all} -report_file=unmap_report_file]
[-create_uom -report_file=create_uom_report_file]
[-h]
```

ARGUMENTS

-u

Specifies the user ID.

To use the **-map**, **-unmap**, and **-create_uom** arguments, this is a user with Teamcenter administration privileges.

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.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

For more information about managing password files, see

For more information about managing password files, see [Manage password files](#).

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

To use the **-map**, **-unmap**, and **-create_uom** arguments, this is a user who is a member of a group with DBA privileges.

If used without a value, the user's default group is assumed.

-analyze

Evaluates the legacy existing units of measurement system.

Use **-analyze** with **-report_file** to examine the Teamcenter database for existing units of measure and their definitions. The utility attempts to map legacy units of measure to units of measure in the UMS. This mapping is performed based on unit names and symbols.

The utility generates the .csv file specified by **-report_file=config_file** showing these mappings along with units that could not be mapped between the systems. Confirm and update the mappings in this file as described in Migrate legacy units of measure.

The .csv file contains the following mapping information:

Column	Definition
Is Approved	Remove the # character in this column to process this unit when using -map , -unMap , or -createMap .
UOM Name	The name of the unit in the legacy system.
UOM Symbol	The symbol of the unit in the legacy system.
In Use	A value of Yes indicates that this unit is used by an object in the legacy system.
Action	The utility action performed to create the file.
Status	The unit's mapping state at the time the file was created.
Details	A description of the unit's state and any recommended actions.
Unit Definition ID	The ID of the unit in the UMS.
Unit Definition Name	The name of the unit in the UMS.
Unit Definition Symbol	The symbol of the unit in the UMS.

After confirming and completing the mappings in the .csv file, use **-analyze** with **-input_file** to generate a dry-run report showing what the results will be if you use the .csv file to map the approved mappings.

-map

Updates the Teamcenter database references between the legacy unit of measure line items with the UMS line items defined in the configuration file specified with **-input_file**.

Approved units in the file (units with no "#" symbol in the first column) are updated in the Teamcenter database. The utility generates the report specified with **-report_file** listing the updates and their completion status.

-unmap

Removes Teamcenter database relationships between legacy units of measure and the UMS.

Using **-input_file** with **-unmap** removes specific unit of measure relationships. Relationships for approved units in the file (units with no "#" symbol in the first column) are removed from the Teamcenter database.

Using **-all** with **-unmap** removes all existing units of measure relationships from the Teamcenter database. You could then restart migrating legacy units of measure as described in Migrate legacy units of measure to the unit management system.

The results of either removal is detailed in the file specified by **-report_file**.

-create_uom

Analyzes the UMS and creates new units for definitions without related units. Results are listed in the report specified by **-report_file**.

-h

Displays help for this utility.

ENVIRONMENT

As specified in Manually configure the Teamcenter environment.

FILES

As specified in Log files produced by Teamcenter.

RESTRICTIONS

None.

EXAMPLES

- Analyze the Teamcenter database for existing UMS unit definitions. Create a mapping configuration file containing mappings between the legacy units of measure to existing unit of measure definitions in the UMS.

```
ums_mapping -u=username -p=password -analyze
-report_file=c:\temp\mapping_config.csv
```

- Perform a dry run of linking approved units of measure to unit definitions in the UMS.


```
ums_mapping -u=username -p=password -analyze
-report_file=c:\temp\mapping_config.csv
```

- Add unit of measure relationships to the Teamcenter database.

```
ums_mapping -u=admin-username -p=admin-password
-g=dba -map -input_file=c:\temp\mapping_config.csv
-report_file=c:\temp\mapping_report.csv
```

The utility generates the report *mapping_report.csv* listing the updates and their completion status.

- Remove unit of measure relationships from the Teamcenter database

```
ums_mapping -u=admin-username -p=admin-password
-g=dba -unmap -input_file=c:\temp\mapping_config.csv
-report_file=c:\temp\mapping_report.csv
```

The utility generates the report *mapping_report.csv* listing the removed unit relationships.

- Create new units of measure and add the new units to the Teamcenter database.

```
ums_mapping -u=admin-username -p=admin-password -g=dba -create_uom
-report_file=c:\temp\unit_report.csv
```

The utility generates the report *unit_report.csv* listing the created units.

14. Subscription configuration

Configuring subscriptions

What are subscriptions?

Subscriptions are objects or events on objects being followed using the **Follow**  command. When objects are changed in Active Workspace, a number appears on **Alerts**  when subscription notifications are received. Users click **Alerts**  to view notifications.

What do I need to do before configuring?

Before you can configure subscriptions:

- If you are using Teamcenter Environment Manager (TEM), you must install the following features from the **Features** panel.

- **Subscription** (client)

Installs the user interface elements for viewing subscription notifications in Active Workspace.

Select **Active Workspace**→**Client**→**Subscription**.

- **Subscription** (server)

Installs the server-side definitions for subscriptions.

Select **Active Workspace**→**Server Extensions**→**Subscription**.

- If you are using Deployment Center, you must select **Subscription** on the **Applications** tab.
- **Set up the environment for notifications.**

What can I configure?

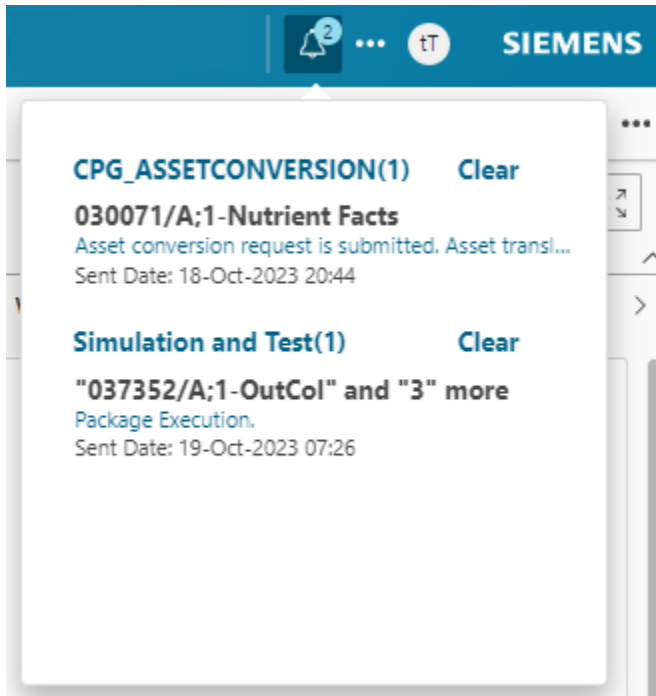
You can configure the following aspects of subscriptions for your users:

- **Subscribable properties.**
- **Email and news feed notifications.**
- **Number of objects to which a user subscribes.**
- **Number of events to which a user follows.**

- **Number of days news feed notifications are retained.**
- **Purge of old news feed notifications.**

What do subscription notifications look like?

Following is an example of notifications.



Set up the environment for notifications

To enable notifications, you must set a value for the **TC_MESSAGING_MUX_URL** environment variable to indicate the source of the information. The value of the **TC_MESSAGING_MUX_URL** environment variable must depend on whether your environment uses a shared **TC_DATA** directory or unique **TC_DATA** directories:

```
TC_MESSAGING MUX_URL=protocol://host:mux_port
```

If the deployment shares the same **TC_DATA** directory for all server managers or if you have only one server manager in the environment, the **TC_MESSAGING_MUX_URL** host must be set to **localhost**.

If the deployment does *not* share the same **TC_DATA** directory, meaning each server manager has its own **TC_DATA**, the **TC_MESSAGING_MUX_URL** host must be set to the host on which the server manager is installed.

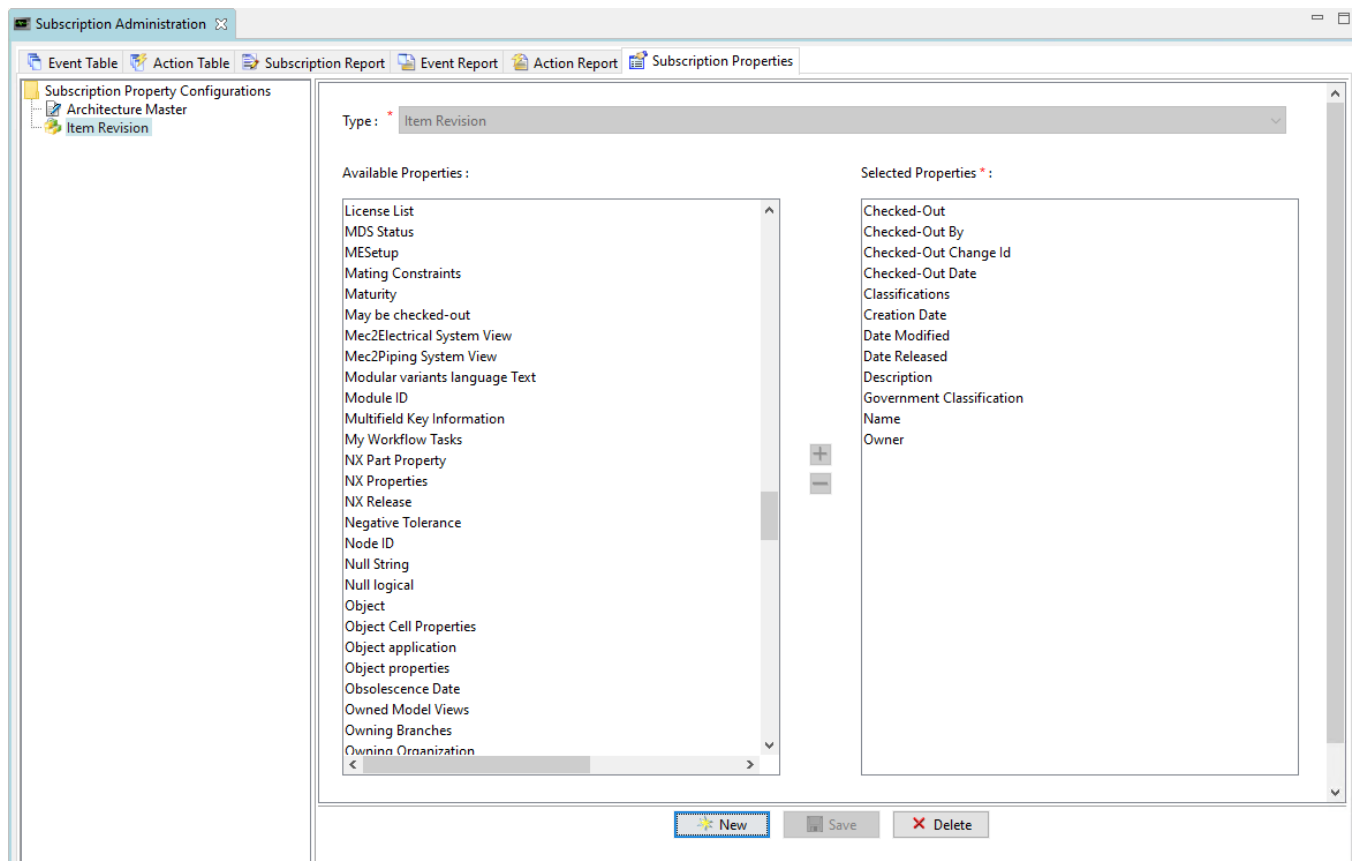
Configuring subscribable properties

If you want users to only subscribe to certain properties on subscribable objects, such as an **Item** or **ItemRevision**, you can use the **Subscription Properties** tab in **Subscription Administration** in the Teamcenter rich client to configure the properties on which users can subscribe.

For example, using the **Subscription Properties** tab, you can configure several **ItemRevision** properties to make them available to users.

When configuring subscribable properties, consider the following:

- If no configuration is defined on a subscribable object, all properties are available to Active Workspace users for defining subscription criteria.
- A subtype inherits the configuration of the parent type unless it has its own configuration.
- If a property is turned off from subscription, but existing subscriptions utilize that property, then these subscriptions continue to have the property in the criteria until the subscriber manually removes it. Though existing criteria continue to apply, the Active Workspace user cannot define a new subscription using that property.



Setting subscription notification preferences

Setting email and news feed preferences

Set the following preferences to control notification behavior in Active Workspace:

- **SCM_notification_mode**

Specifies whether you want notifications to be delivered by email or news feed or both.

Value	Description
1	Deliver notifications by email.
2	Deliver notifications by news feed.
3	Deliver notifications by both email and news feed.

- **AWS_Notifications_Polling_Interval**

Specifies (in minutes) how frequently the system is polled for notifications, site-wide. The default setting is **5**. If this preference is set to **0**, the **Alert**  navigation button is disabled.

The number of notifications appears next to the **Alert**  when:

- Print notifications are received.
- News feeds to which you have subscribed are received.
- There are existing unread notifications.

Setting periodic digest preferences

In addition to receiving notification by email or news feed, users can also select to receive daily and weekly digests that contain subscription notifications by selecting **Set Daily and Weekly Digests (Collate all notifications)** from the profile page.

▼ SUBSCRIPTIONS

Notification Method:

News Feed ▼

☐ Send Daily and Weekly Digests(Collate all notifications)

The periodic digest collates all the daily notifications into a single email and it collates all notifications from the week into a weekly digest that is sent as a single email.

Set the following preferences to control periodic digest notification:

- **SCM_notification_digest**

Specified whether notifications are delivered as a digest. If this preference is set to **1** (default), digest notifications are disabled at the site level. Once the user enables the digest from the profile page, the digest is enabled for that user.

- **SCM_notification_digest_file_size_limit**

Specifies the digest notification file size limit in megabytes (MB). The default value is **1 MB**.

When the size of the collated digest notification exceeds the specified size limit, the digest is sent in multiple parts.

This applies to both email and news feed notification methods.

- **SCM_execution_day**

Specifies the day the weekly digest is triggered. The default is **Sunday**.

- **SCM_execution_time**

Specifies the time the digest is triggered. The default is **15:00**.

- **WEB_default_site_server**

When set, the digest contains the link for the subscription. Use the format *localhost:7001*; for example, *http://10.123.54.46:7001/awc*.

Configuring subscription to multiple objects

Since users can subscribe to multiple objects, you can use the **AWC_followMultiObject_max** preference to control the number of objects to which a user subscribes at a time. By default, this preference is set to **5**.

Configuring My Events

Users can use **My Events** to follow multiple events on an object. Because it is important to control the number of events to which a user follows at a time to prevent users from creating many subscriptions, use the following site preferences:

- **AWC_followMultiEventConfig_max**

Controls the maximum number of events a user can select on an object to follow. By default, this preference is set to **5**.

- **AWC_followMultiEventConfiguredEventTypes**

Controls default configured event types for multiple events. Valid values are **__Attained_Release_Status**, **__Attach**, and **__Item_Rev_Create**.

Configuring news feed retention

Use the **SCM_newsfeed_purge_threshold** site preference to configure the number of days user news feed notifications are retained before being purged. The default setting is **0** days, which retains the news feed messages always. If you remove the preference value, the **Retain News Feed (In Days)** field still displays on the **Profile** page.

Users can view their news feed retention time by going to the **SUBSCRIPTIONS** area of their **Profile** page. From there, they can configure their threshold value. Otherwise, the purge is based on the site configuration.

Purging news feed notifications

You can purge old news feed notifications using the **clear_old_newsfeed_messages** command-line utility. To process only read messages, enter the following:

```
clear_old_newsfeed_messages -u=Tc-admin-user -p=password -g=group  
-process_only_read_messages
```

This utility purges messages based on the **purge threshold** that you set using the **SCM_newsfeed_purge_threshold** preference.

15. Active Collaboration configuration

Users can use Active Collaboration to discuss objects such as items, parts, and documents. Active Collaboration supports threaded discussions, embedded images, 3D product snapshots, notifications, and role-based permissions for managing discussions. Administrators configure Active Collaboration features to enable users to effectively communicate, share visual information, and control access based on their roles and responsibilities within their organization.

Installing Active Collaboration features and components

You can install Active Collaboration using either Teamcenter Environment Manager (TEM) or Deployment Center.

Install these Active Collaboration features and components on either TEM or Deployment Center:

From the Features panel of TEM:	From the Applications panel of Deployment Center:
• Active Collaboration Client Select Active Workspace→Client→Active Collaboration Client to install the user interface elements for viewing discussions. • Microservices Select Teamcenter GraphQL Service. • Active Collaboration (server) Select Active Workspace→Server Extensions→Active Collaboration to install the server-side definitions supporting discussions.	• Active Collaboration Select Teamcenter→Teamcenter→Active Collaboration to install the Active Collaboration feature. • Microservice Node Confirm that the Microservice Node component is configured.

Note:

If you have existing questions, replies, and comments from the Active Collaboration component in Active Workspace 5.0 or earlier or from the Active Collaboration for Retail component in Active Workspace 6.0 or earlier, you may perform a one-time migration of them to Discussions using the `ac0_migrate_s2cldata` utility.

Modifying discussion default operation

You can modify the following default discussion behaviors by **changing their related preferences**.

Enabling tracked discussions

Enabling tracked discussions allows users to identify and view particular discussions based on priority and status. A **Tracked** tab is added to the **Discussions** location and an unchecked **Tracked** check box is available on each discussion. The **Tracked** tab lists any discussions for which the user has clicked **Tracked** and set priorities and status. Users can sort and filter their messages based on priority and status on the **Tracked** tab.

To activate tracked discussions for users, set the **Ac0EnableTrackedDiscussions** preference to **true**. The default value of this preference is **false**.

Automatically adding object owners as discussion participants

By default, an object's owner is not automatically added to every discussion on the object. This approach is often appropriate for organizations in which objects and structures are typically created by individuals not actually responsible for those objects. Organizations in which individuals typically create objects they are responsible for may wish to have the owner automatically added to every discussion on an object. Owners can manually be removed as participants in any individual discussion at any point in time.

To automatically add object owners to discussions on objects they own, set the **Ac0AddOwnerAsDefaultParticipant** preference to **true**.

Setting all messages as private

By default, all discussions are available for viewing by all users who can access the related objects. Participants can manually restrict viewing of individual discussions to only the participants of the discussion. Organizations that prefer to limit the viewing of all discussions can specify that discussions be viewable by only participants in the discussions by default. Be aware that participants in a discussion can remove the privacy restriction by unchecking **Private Message** on the discussion.

To have discussions viewable by only their participants by default, set **Ac0DefaultIsPrivate** to **true**.

Keeping users informed about discussion updates with email notifications

You can configure Active Collaboration to notify users by email when updates have been made to discussions that they are associated with.

Use the following procedures to configure Active Collaboration email notifications.

Prerequisites

Teamcenter mail must be configured for users to receive email notifications when discussions are updated. For example, your organization's mail server may require the following preferences to be set:

- **Mail_server_name**

- **Mail_server_port**
- Optionally, set **Mail_OS_from_address**.
- You may also set additional email preferences for your site configuration requirements.

Notifying users about discussion activities with email notifications

Email notifications can help users keep track of discussion activities. To enable discussions-related email notifications for users, set the **Ac0EnableEmailDiscussionNotifications** preference to **true**. When users select **Notify me by email for discussion activity** on the profile page, they receive email notifications when updates are made to discussions they follow and to discussions on objects they follow. The default value is **false**.

To have **Notify me by email for discussion activity** selected for all users by default, set the **Ac0EnableUserEmailDiscussionNotifications** preference to **true**.

Reducing discussion email notifications

Reducing discussion-related email notifications minimizes distractions and reduces clutter. To help reduce discussions-related email notifications for users, set the **Ac0ThrottleUserEmailDiscussionNotifications** preference to **true**. When users select **Reduce the number of notifications for the same discussion** on the profile page, they receive an initial email for a change to a discussion, but do not receive additional email notifications for that discussion for the time specified by **Ac0EmailDiscussionNotificationsInterval**. The time interval resets once the user views the related object or discussion.

To have **Reduce the number of notifications for the same discussion** selected for all users by default, set the

Specifying time intervals for reducing discussions-related email notifications

You can specify the number of hours to mute email notifications for users who have elected to reduce the number of discussions-related email notifications they receive. To specify the time interval in which users do not receive email notifications, set the **Ac0EmailDiscussionNotificationsInterval** preference to the number of hours that email notifications will be muted. The default value is **8**.

Generating automated messages for users

Active Collaboration can automatically generate messages on **Discussion** panels. Receiving automated messages can be helpful for users when discussions are updated on the **Discussions** panel.

To generate automatic messages for users, set the **Ac0AutoMessagesEnable** preference to **true**. The default value is **true**. Automated messages are generated when a participant is added to or removed from the discussion, a tracked discussion's status is changed, or there are changes to privacy.

Enabling tracked discussions

Tracking discussions can help users monitor the status, priority, and progress. Users can also find and review the content of the discussion and ensure important discussions are connected to relevant project or action items. To activate **Tracked** discussions for users, set the **Ac0EnableTrackedDiscussions** preference to **true**. The default value of this preference is **false**. Users can see and filter by **Status** and **Priority** in a **Tracked** discussion.

Procedure

1. Select **Preferences**.
2. Search for the **Ac0EnableTrackedDiscussions** preference.
3. Click **Edit**.
4. Specify the value of the **Ac0EnableTrackedDiscussions** preference to **true**. The default value of this preference is **false**.
5. Click **Save**.
6. Select **Tracked** on a particular discussion to track it.

This enables the **Status** and **Priority** fields. Users can change the status and priority of a discussion.

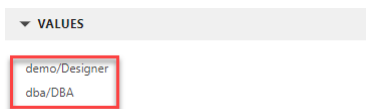
Managing user permissions to delete discussions

Deleting discussions helps keep users' **Discussions** panels free of outdated discussions, reduces clutter, and protects the privacy of participants who want their private information to be removed. You can grant permissions to groups and roles to delete discussions that are no longer needed. By default users logged in with the group/role of dba/DBA have the ability to delete discussions.

Procedure

1. Select **Preferences**.
2. Search for the **Ac0DeleteDiscussionGroupRole** preference.
3. Click **Edit**.
4. In the **Values** section, specify a group and role to grant delete permissions using the form *group_id/role_id*.

You can specify multiple combinations, for example:



The default value is **dba/DBA**.

5. Click **Save**.

Working with snapshots in discussions

Sharing 3D snapshots in discussions

Users can share 3D snapshots in discussions to present visual aids to other discussion participants, identify issues that are not obvious in a written description, and speed up the decision making process without writing a long comment. To enable users to share 3D snapshots in discussions, set the **Ac0SnapshotDiscussionsEnabled** preference to **true**.

Users can create discussions using previously captured product snapshots. When users create a new discussion from the **3D** tab on an assembly where visualization is installed, they can use **Capture Snapshot** .

Enabling users to use snapshots in discussions

Users can share specific views and configurations of models directly within discussion threads if snapshots are enabled for use in discussions. Using snapshots in discussions provides visual context that enhances communication and understanding between discussion participants.

The presence and configuration of an access control list (ACL) is necessary for enabling users to leverage snapshots within discussions. If visualization is installed and users want to share 3D snapshots, then the **SnapshotDiscussionsAccessor** ACL must be configured in the Access Manager. The **SnapshotDiscussionsAccessor** ACL enforces participant checks and gives permissions to those participants to view a snapshot shared in a discussion.

1. From the Access Manager, select the **Has Class (WorkspaceObject)** rule.
2. Select the **Has Type (Vis_Session) > HasProperty (Vis_Session:fnd0ShareLevel=Private) SnapshotDiscussionsAccessor** ACL. Click **Edit** . Add  the **SnapshotDiscussionsAccessor** ACL in this rule if the ACL does not exist.

If you create the **SnapshotDiscussionsAccessor** ACL for the **Has Type (Vis_Session)** or the **Has Class (Fnd0AppSession) > Has Attribute (Fnd0AppSession:fnd0ShareLevel=Private)** rule, then the ACL is created for both rules.

3. Select the **Has Class (Fnd0AppSession) > Has Attribute (Fnd0AppSession:fnd0ShareLevel=Private) SnapshotDiscussionsAccessor** ACL. Ensure that the **SnapshotDiscussionsAccessor** ACL exists for this rule.
4. Ensure that the **Discussion Participant** has read (👁) access.

If not, click **Edit**. Grant the **Discussion Participant** read access.

16. Settings and performance

Manage system settings and performance

There are many utilities and settings that help you maintain the health and performance of your site.

Troubleshooting

Perform many basic tasks including retrieving software release versions, resetting your gateway server, monitoring the browser activity, and so on.

Client performance

Discover settings and concepts that may help improve client performance.

Preferences

Learn about how Teamcenter and Active Workspace store various settings as preferences.

Data model settings

Learn about various constant types that are part of the data model.

Server-side utilities

Gain an overview of certain Teamcenter server command-line utilities to help monitor and manage your site.

Troubleshooting

Retrieving Active Workspace client and server versions

Information about the running Active Workspace client and server, as well as the Teamcenter server version, site ID, and database ID to which they are connected is available when you are logged in.

To retrieve version information, click **Help**  > **About**.

Your results will vary, but following is an example of the results.

```
active-workspace@5.0.0 (Active Workspace Client (Staging Environment))
afx@4.1.0-361 (Siemens Web Framework)
Client Build: Wed May 06 2020 08:35:37
Server Build: aw5.0.0.13x.2020050601;...
Server Version: P.13.0.0.20200429.00
```

Site: IMC--1821067151 (-1821067151)

Database: tc

User Session Logfile: tcserver.exe41f085b3.syslog

General troubleshooting

Note:

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.

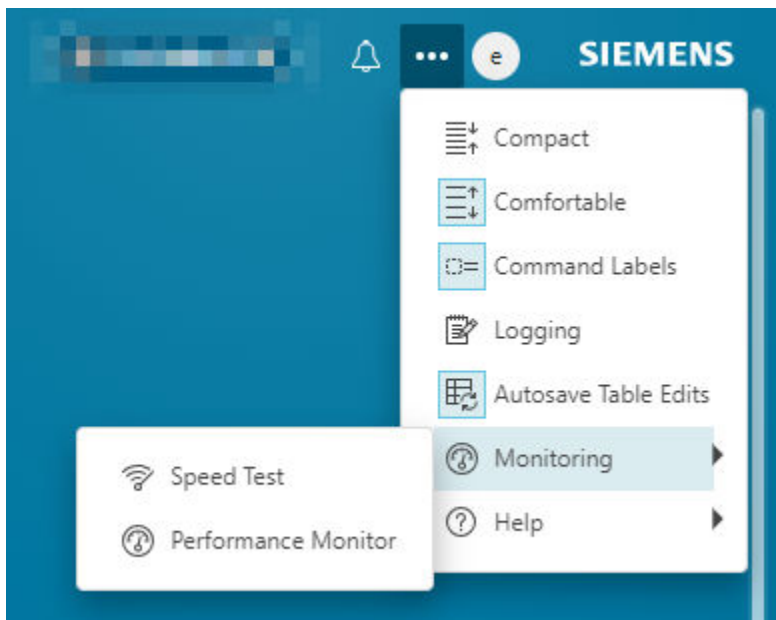
Issue	Possible resolution
No server available error	Tune the tcserver pool size using the PROCESS_WARM parameter. For details, see <i>Teamcenter Administration</i> in the Teamcenter collection.
Intermittent image loading issues	Perform one of the following: - On the server, configure the web application server to exclude the problematic cipher. For example, if you have a jetty server: - In a text editor, open the jetty\etc\jetty-ssl.xml file and add the following lines after the <Set name="TrustStorePassword"xxx</Set> line: <pre> <!-- avoid IE TLSv1 issue by excluding the problematic cipher --> <Set name="ExcludeCipherSuites"> <Array type="java.lang.String"> <Item>TLS_RSA_WITH_AES_128_CBC_SHA</Item> </Array> </Set> </pre> - Save the file. - Restart the Jetty server. The steps for other servers will vary. - On the client, configure the browser to not use TLS 1.0.
Active Workspace does not display the same language (locale) as the Teamcenter server.	Ensure the following: - Set the operating system of the computer running Active Workspace to the correct locale. - Set the browser running Active Workspace to the correct locale. - Ensure that the web application file is set to the correct locale.

View Active Workspace performance data

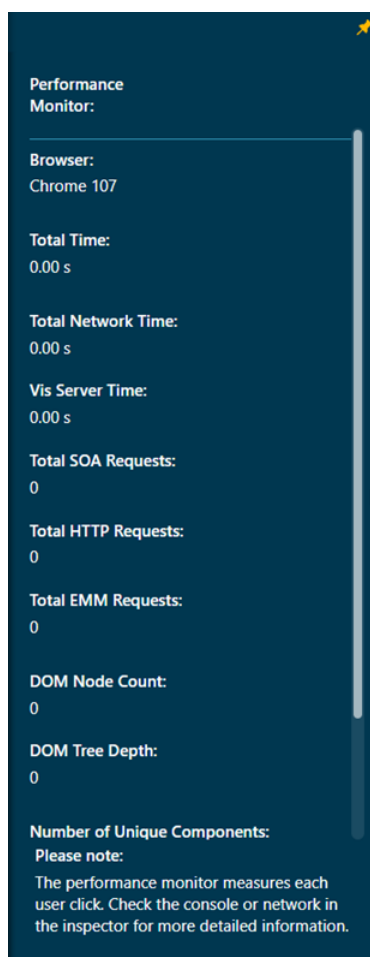
Performance monitor

Use the **Performance Monitor**  command to view performance and optional telemetry data.

When you use the **Performance Monitor**, it opens a panel and reports to the browser console about memory usage, overhead times, DOM node count, and so on. Additional information may be available depending on which components you have installed in your Teamcenter environment. Close the panel to stop tracking the statistics.



Select the **Performance Monitor**  command.

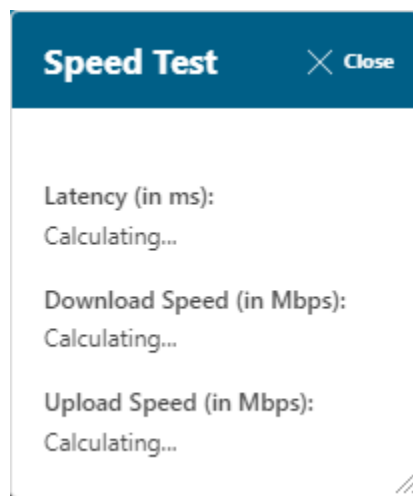


Value	Description
Browser	Displays the current browser and version.
Total Time	Marks the point at which the application is both visually rendered and capable of reliably responding to user input. It is the time from the user doing an action on the page until the page is rendered and becomes idle for 200ms.
Total Network Time	Time from browser when sends the request until it gets the response. Includes tcserver time plus the wire transfer time.
Vis Server Time	Time to process a request from the Vis Server
Total SOA Requests	The number of SOA calls made.
Total HTTP Requests	The number of HTTP calls made.
Total EMM Requests	The number of EMM calls made.
DOM Node Count	The number of DOM Nodes on the page.
DOM Tree Depth	Maximum DOM Tree depth of the page.

Value	Description
Number of Unique Components	The number of unique components on the page.
Number of Component Renders	The number of times components on the page were rendered.

Speed test

Additionally, you can run the **Speed Test**  to see your bandwidth and latency to the server.



What else should I know?

In the detailed view provided in the console output, all values that are preceded by an asterisk will be reported to Siemens Digital Industries Software, if analytics is enabled.

Example:

```
*ScriptingTime: "xxx.xxms"
```

Verify the Active Workspace gateway and other microservices

Use the **ping** functionality to check the various components of the Active Workspace gateway architecture, and verify connectivity.

```
http://hostname:3000/ping
```

Active Workspace Gateway Ping

Routes

#0 ServiceDispatcher	OK
File Repository Service	OK
/tc	OK
/tc/micro	OK
/vis	OK
fms.bootstrapFSCURLs	OK

Gateway

uptime	2 days 12 hours 16 minutes 19 seconds

You can disable this functionality by changing the **pingEnabled** setting to **false** in the gateway *config.json* file.

AW ROOT/microservices/gateway-x.x.x/config.json

```
"pingEnabled": false
```

Tip:

You must **restart the gateway** to apply the change.

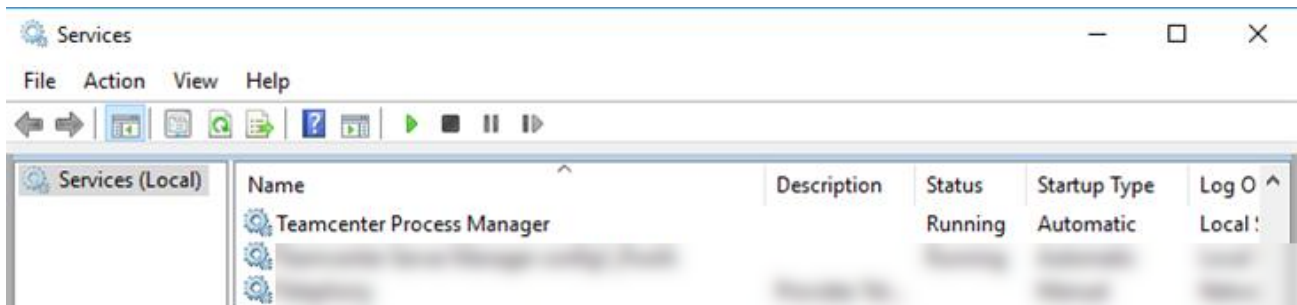
Resetting the Active Workspace gateway and microservices

When you make changes to the configuration files for the **Active Workspace gateway** and microservices, you must restart them for your changes to be recognized.

Make sure the FMS server cache (FSC) service is running before you start the Active Workspace gateway.

Windows

On Windows, all of the gateway and microservices on a given machine are managed by a single multi-threaded Windows service. To implement your configuration changes, restart the service using the **Services** control panel:



Or from the command line:

```
net stop "Teamcenter Process Manager" && net start "Teamcenter Process Manager"
```

Caution:

You *must* use Microsoft **Services** to start the service. Do not run the service script directly.

Linux

On Linux, the gateway and microservices are managed by **Docker**. Remove and recreate the gateway services and config. The other services will restart in turn.

- Locate the **gateway** service and remove it.

```
TC_ROOT/container> docker service ls | grep gateway
wfvrc598nkbz    config1_MS_STACK_gateway
replicated     1/1           localhost:5000/teamcenter/afx-gateway:1.9.0
TC_ROOT/container> docker service rm wfvrc598nkbz
wfvrc598nkbz
TC_ROOT/container>
```

- Locate the **config.json** config and remove it.

```
TC_ROOT/container> docker config ls | grep config.json
e9ldu5xqhhwsya7t4i22skr9u    config1_MS_STACK_config.json
TC_ROOT/container> docker config rm e9ldu5xqhhwsya7t4i22skr9u
e9ldu5xqhhwsya7t4i22skr9u
TC_ROOT/container>
```

- Deploy the **gateway.yml** Compose file.

```
TC_ROOT/container> docker stack deploy -c gateway.yml config1_MS_STACK
Creating config config1_MS_STACK_config.json
Creating service config1_MS_STACK_gateway
TC_ROOT/container>
```

Start up race conditions

It is recommended to start the gateway after any GraphQL services have had a chance to stabilize. The gateway will make several attempts before giving up, but in some cases this will not be enough. You can adjust the number of attempts made in the gateway's *config.json* file. Each attempt is 15 seconds.

```
"graphql": {
  "endpoints": [
    "darsi",
    "tcgql"
  ],
  "attempts": 5
},
```

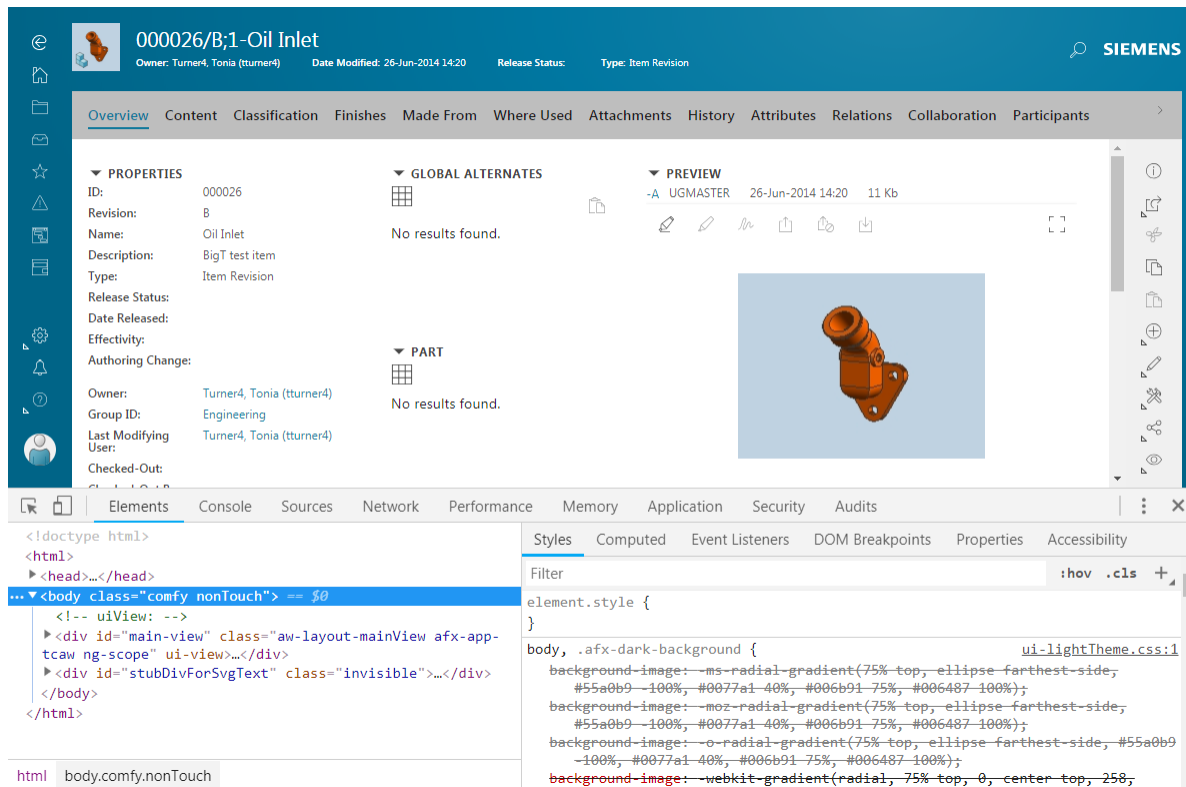
This problem is more prevalent on Linux and Docker and especially if all of your gateway services are installed on the same machine.

Monitoring browser activity

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.

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.



Performance and settings

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.

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.

Working with large files in Active Workspace

During normal use, the Teamcenter web tier maintains a user's Teamcenter session because the Active Workspace gateway is sending regular user requests. But during a file upload, there is no explicit communication from the gateway to the web tier. When transferring very large files, it is possible that it takes more time than the Teamcenter session timeout setting, and the web tier will time out during the operation.

To avoid this issue, increase the web tier's timeout setting to the time it typically takes to transfer files of that size.

One thing to note is that even without any user activity, the Active Workspace gateway still sends a keep-alive ping to the web tier every 2 hours. Because of this, even if you need more than two hours for file transfers, you can set your web tier timeout to just over 2 hours and the automatic gateway ping will keep the session alive indefinitely.

Using Deployment Center, navigate to your environment's components and select the web tier. Click **Show all parameters**  to display the values for the web tier's timeout setting.

 Teamcenter Web Tier (.Net)  Configuration Session Time Out 1440	 Teamcenter Web Tier (Java EE)  Context Parameters Connection Timeout 30000

After making any changes, you must redeploy and run the updated scripts on the appropriate systems.

Compressing images for loading them quickly

Image files are used in Active Workspace for tiles, preview images, thumbnails, breadcrumbs, and so on. Image resolution is the clarity with which you can view the image with distinct boundaries. The resolution of the image depends on the number of pixels; more pixels correspond to more clarity, but also increases the size of the image. Large images take a lot of time for rendering and viewing.

To render images not only quickly but also with high clarity in Active Workspace, you can compress them and reduce their sizes without distorting the quality. You can manage the quality, sharpness, color, and accuracy of the images with lower resolutions. You can generate low, medium, and high resolutions of the original uploaded image while maintaining their aspect ratio. You can also define custom resolutions for the images.

Configure image resolution

You can compress images in Active Workspace used for tiles, preview images, thumbnails, breadcrumbs, and so on. This reduces their size without distorting the quality. Following are the prerequisites for configuring image resolution in Active Workspace:

- Teamcenter Visualization with **Mockup** and **Convert & Print** features.
- Dispatcher Server and Dispatcher Client components under Teamcenter Enterprise Knowledge Foundation are installed using Teamcenter Environment Manager.
- Image translator installed using the Teamcenter Environment Manager.

To compress images:

1. Enable the image compression feature using the **TC__image_compression_enabled** preference in Teamcenter rich client. The default value for this feature is set to **false**.
2. Configure the resolution values in the **TC_image_compression_types** preference in Teamcenter rich client.

The out-of-the-box (OOTB) values are:

- **64px::Low**
- **300px::Medium**
- **600px::High**

You can also define custom values for images, such as **1200px::LARGE** or **2800px::EXTRALARGE**.

These values are for the height of the translated image, and the appropriate width is automatically adjusted by the image translator based on the aspect ratio of the original image.

3. To specify the default image to be used for scaling across Active Workspace application, set the value for the **AWC_default_image_resolution** preference. The default OOTB value is **Medium**.
4. To customize the image for the **Overview** tab:

In the **tc_xrt_Preview** tag, specify the value for the default image:

```
<section titleKey="tc_xrt_Preview">
  <section titleKey="tc_xrt_Preview">
    <image resolution="<user_input>" source="thumbnail"/>
  </section>
</section>
```

- If no image resolution is defined, the system resolves to a high resolution image.
- If the resolution preference value is **Medium**, the system resolves to medium resolution.
- If image resolution is an undefined or invalid resolution type, the system resolves to high resolution.

- If the image resolution is a custom value as defined in the **TC_image_compression_types** preference, it resolves to the specified custom resolution value. For example, if you specify **2800px::EXTRALARGE** as the image resolution, the image resolves to the custom value **EXTRALARGE**.

Note:

The values for image resolution are not case sensitive.

Preferences

Why do I need preferences?

You can use Teamcenter preferences to control various aspects of Teamcenter's behavior and appearance.

Following are only a few examples of what preferences control:

- Whether or not live updates are allowed.
- Password requirements when not using LDAP.
- Which XML rendering template (XRT) to use.
- Which query to use as the default quick access query.

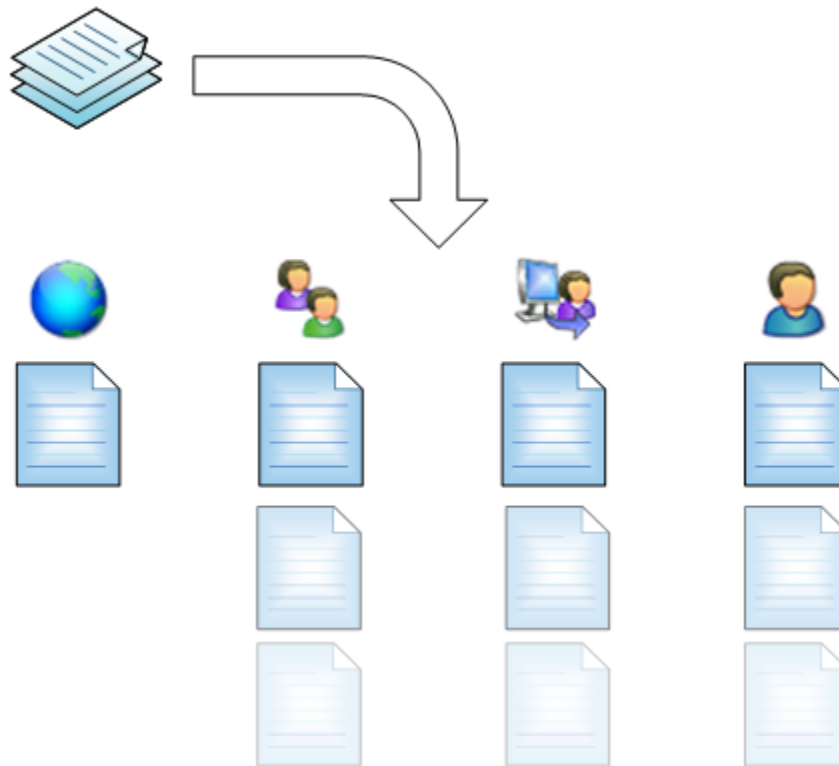
Siemens Digital Industries Software recommends browsing through the list of preferences to see which ones might be useful to you. Each preference's definition will document its use.

How do preferences work?

At their core, preferences are simply a way to store information. They are similar to environment variables, except that they operate with several layers of permissions.

Overview

Each preference consists of two major components, a definition and instances.



A preference *definition* along with all of its preference *instances* are together considered to be a preference.



Definition

The preference definition is like a blueprint. It defines the nature of the preference and is used to create the instances at the various locations. Even though it may define a default value, the definition itself is never retrieved or read as a preference. If there are no instances of this preference, there is no value.



Site

A preference instance created at the site location applies to everyone logged in to Teamcenter unless overridden. There can be only one site instance.



Group

Any preference instances created at the group location apply only to users who are currently logged in as that group, and they supersede site preferences. There can be one group instance created for each group.



Role

Any preference instances created at the role location apply only to users who are currently logged in as that role (regardless of group), and they supersede site and group preferences. There can be one role instance created for each role.



User

Any preference instances created at the user location apply only to that user, and they supersede site, group, and role preferences. There can be one user instance created for each user.

Preference definition

You use the preference definition to create the overall limits and restrictions on the preference as well as setting the default value. Think of this as an abstract template from which the preference itself will be instantiated. Following are the fields used to define a preference definition:

Name	The name of the preference. Naming patterns help organize the preferences and give an idea of what they do even before you read the description. See the list of existing preferences for examples.
Protection Scope	Determines where and by whom it can be instantiated.
Type	Specify the preference value type.
Multiple	Specify if this preference can hold multiple values.
Description	Explain the use of the preference. What does it control? What format is expected for the values? Etc.
Value	Specify the default value that an instance will contain when initially created.
Environment	Retrieve the value from an OS environment variable of the same name.
Category	Organize related preferences based on their category. There are many existing categories you can use, or you can create your own.

Preference instance

You create a preference instance from its definition. When you create a new instance of a preference it must belong to a location. This location specifies when it is active and its priority in the hierarchy. You cannot create a preference instance if the protection scope does not allow it.

When referring to preference instances, it is common to shorten the phrase. For example, the *preference instance in the Engineering group location* is commonly referred to as the *Engineering group preference*.

When you create a new preference, you specify two things:

Location	Locations are where the preference instances reside. You can create preference instances at the following locations: • User • Role • Group • Site / System
Value	You can keep the default value from the definition or specify a new one.

Preference locations

- **User**

This assigns the instance to a specific user. These are commonly the preferences that Teamcenter uses to track things like column widths in the rich client, or most recently searched text, for example.

Although you can control your active preferences like style sheet registration down to the user level, it is normally recommended that you keep those kinds of settings to the Group level or higher. It makes things easier when people move in and out of groups and roles.

- **Role**

You can control the behavior based on a user's role. This is handy for things such as style sheets. Keep the consumer's page simple while being able to provide the information the author or approver needs.

- **Group**

Similar to the **Role** location, you can control the behavior at the next step up, at the group level.

- **Site / System**

Preferences created at these locations apply to everyone. This is typically where you instantiate preferences that control system-wide behavior or default behavior that can be overridden at the group, role, or user level.

Site preferences only allow a single instance, but a dba can change the protection scope of a site preference to something else.

System preferences do not allow their protection scope to be changed, *even by a dba*. In all other ways, they behave like a site preference.

Caution:

An existing non-system preference may be changed into a system preference by a dba, but once it has been changed, it *cannot* be changed back. If you want to change it, it *must* be deleted and re-created.

Customer-facing preferences

You control an aspect of the UI or behavior directly by making changes to the preference. Examples of these preferences are configuring default paste relations, which style sheets are used in a given situation, or how the Dispatcher handles certain file types.

Internal preferences

Teamcenter uses preferences extensively to remember application parameters, like column width. Even though you can see and possibly modify the values of these preferences, it is not advised to do so.

An example of preference hierarchy

Everything in this example is based on a single preference, one which registers a style sheet to a business object for the summary view. It could be any preference as all preferences behave the same way. Since this preference definition's protection scope is **User**, you can create instances at the **Site**, **Group**, **Role**, and **User** location. This means you can control its value based on your users' current group, role, or even user name.

Example: I want the summary view's property layout for item revisions to depend on my users' login information

Following are the details of this example.

- You have three groups: Engineering, Manufacturing, and Testing.
Each group has three roles: Manager, Designer, and Viewer.
- You want a default style sheet that everyone will use unless otherwise specified.
- Your technical users need an extended set of properties.
- Your managers need a page of workflow information.
- Your designers need classification information.
- You have users that just need a simplified layout for viewing.
- You have Conner. Conner is a power-user.
Conner needs a special layout regardless of which group or role he's in.

Style sheet datasets

Five style sheet datasets are considered.

ItemRevSummary

Configured to be the default style sheet for the Item Revision summary page. This applies to everyone unless overridden.

IRSumTech

Configured to provides the extra properties for the Engineering and Manufacturing groups, but not for any other groups.

IRSumMgr

Configured to display workflow information for the Manager role, regardless of group.

IRSumDes

Configured to show the classification trace for the Designer role, regardless of group.

ConnersIRSum

Configured for Conner. Conner has his own requirements

Preference instances

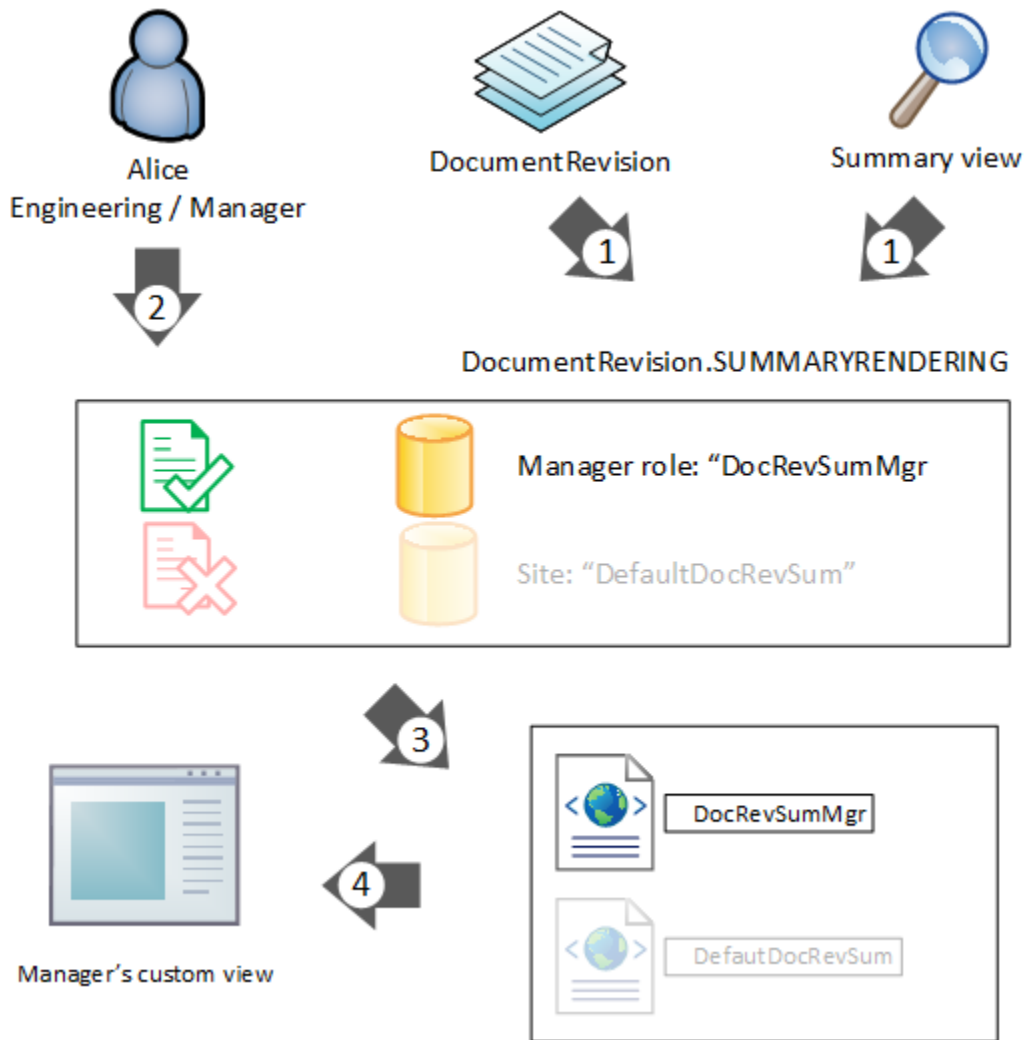
Assign the style sheets to the various groups and roles, and even users if desired, by creating each preference instance with the value pointing to the respective style sheet. In this example, there are 6 preference instances created.

User preferences	Conner: ConnersIRSum
Role preferences	Manager: IRSUMMgr Designer: IRSUMDes
Group preferences	Engineering: IRSUMTech Manufacturing: IRSUMTech
Site preference	<i>value:</i> ItemRevSum

The Viewer role and the Tester group have no preference instances created for their location.

How does Teamcenter choose which preference to use?

In this example, Alice selects a **DocumentRevision** business object and uses the **Summary** tab. When she does this, Teamcenter performs a few steps to determine which style sheet to use.



1. Based on the object type and the view location, the system knows the name of the preference instances to retrieve.

In this example, **DocumentRevision.SUMMARYRENDERING**.

There are two instances: one at the **Site** location, and one at the Manager **Role** location.

2. Based on the user's current session information, Teamcenter chooses the appropriate preference instance.

Less specific locations are overridden by more specific locations.

3. The value of the chosen preference instance is read, providing the name of the style sheet to retrieve.
4. Teamcenter uses the style sheet to render the view.

Result

Your users see a different set of information based on what group or role they are in because the client uses different style sheets.

User - Group / Role	Preference instance build-up	Resulting style sheet
Alice — Engineering / Manager	Alice: <i>none</i> Manager: IRLSumMgr Engineering: IRLSumTech Site: ItemRevSum	IRLSumMgr
Ted — Manufacturing / Manager	Ted: <i>none</i> Manager: IRLSumMgr Manufacturing: IRLSumTech Site: ItemRevSum	IRLSumMgr
Sue — Testing / Manager	Sue: <i>none</i> Manager: IRLSumMgr Testing: <i>none</i> Site: ItemRevSum	IRLSumMgr
Bob — Engineering / Designer	Bob: <i>none</i> Designer: IRLSumDes Engineering: IRLSumTech Site: ItemRevSum	IRLSumDes
Carol — Engineering / Viewer	Carol: <i>none</i> Viewer: <i>none</i> Engineering: IRLSumTech Site: ItemRevSum	IRLSumTech
Pat — Testing / Viewer	Pat: <i>none</i> Viewer: <i>none</i> Testing: <i>none</i> Site: ItemRevSum	ItemRevSum
Conner — Engineering / Manager	Conner: ConnersIRLSum Manager: IRLSumMgr Engineering: IRLSumTech Site: ItemRevSum	ConnersIRLSum
Conner — Testing / Viewer	Conner: ConnersIRLSum Viewer: <i>none</i> Testing: <i>none</i> Site: ItemRevSum	ConnersIRLSum

- Alice sees the style sheet for Managers because she does not have a user preference set to supersede it. The site preference is overridden by the Engineering group preference, which is overridden by the Manager role preference. Ted has the same result; the Manufacturing group preference is overridden by the Manager preference. Sue doesn't have a group preference, but she still gets the Manager role preference.
- Bob sees the style sheet for Designers because of his role, similar to the preceding example.

- Carol sees the tech style sheet because there is no role preference for Viewers.
- Pat's group and role do not have preferences associated with them, and neither does she have a user preference, so she gets the default style sheet defined by the site preference.
- Conner gets Conner's style sheet regardless of which group or role he's in, since a user preference supersedes all others.

What are environment preferences?

You can define a preference to retrieve its value from an environment variable in the operating system.

If you want to pass multiple values from the environment to the preference, you must configure the following:

- Set the preference's **Multiple** setting to **multiple**.
- Use the appropriate separator in the environment variable. The environment variable is read from the operating system on which the **tcserver** process is running.

Windows Semicolon — For example, MyEnvPref=Value1;Value1;Value3

Linux Comma — For example, MyEnvPref=Value1,Value1,Value3

The environment variable is only read by the **tcserver** process when the value is first requested, so any changes made to the environment variable after that will not be reflected in the Teamcenter preference until after the next time the **tcserver** process is started.

Remember, the environment variable is read from the environment where the **tcserver** process is running, which is not necessarily the environment where the client is running.

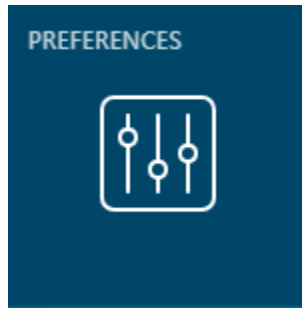
Working with preferences in Active Workspace

You can work with all Teamcenter preferences from within the Active Workspace client using the **Preferences** page.

Preference Management is part of the **Active Admin** installation option for Teamcenter. Once installed, you can get to this page by either:

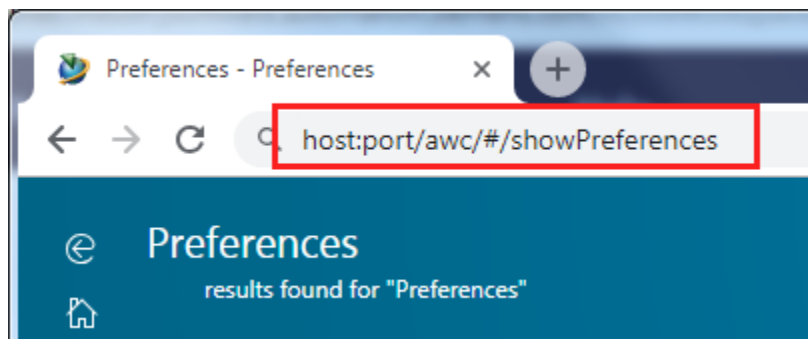
- Using the **PREFERENCES** tile from your home page.

By default, this tile appears in the home page of the **TcActiveAdminWorkspace** workspace.



- Navigate directly to the page with the `host:port/awc/#/showPreferences` URL.

This option is only available if your current workspace allows it.



What can I do with the preferences location?

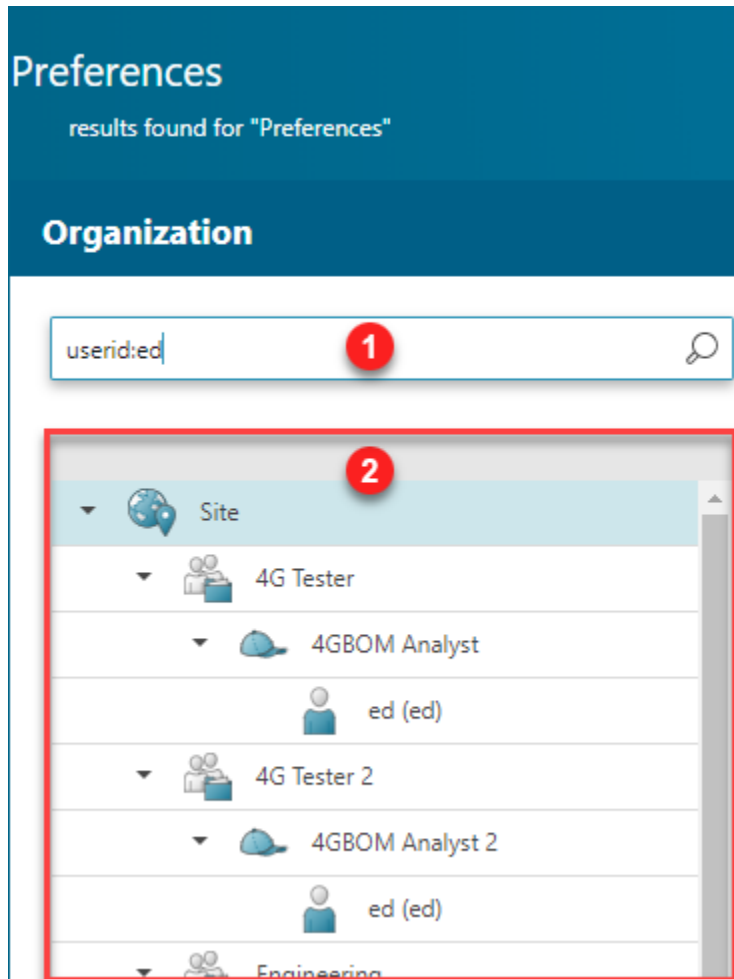
Your ability to work with preferences is determined by whether you are currently logged in to a group with administrator privileges.

Administrator?	Permissions.
Administrator	Search and modify preferences Create and override site, system, group, role, and user preference instances Delete preference instances and definitions
Group Administrator	Search and modify preferences Create and override group, role, and user preference instances within your group
User	Search existing preferences Create and override your own user preference instances

The organization panel

Use the **Organization** panel to select in which session context you wish to work.

If you do not have a choice of working in other session contexts, (you have no administration privileges), then you will not see the **Organization** panel, and you can only work in your current session context.



1. (optional) Filter the organization list.

If you simply use text, the system will match within group, role, user name, or user id. If you use quotation marks, it will search for exact matches, if you don't it will append a wildcard to the end of your text. If you want to include spaces or commas in your search, you must use quotes.

You can narrow down the search by using the following prefixes:

- group:
- role:
- username:
- userid:

You can specify more than one of these by putting a space between them.

Example:

To search for the word, **design**, put **design** in the field.

To search for the specific user id, **ed**, put **userId:"ed"** in the field.

To search for all users with user ids beginning with **ed**, put **userId:ed** in the field.

To search for the specific user named **Smith, Bob** in roles beginning with **design**, put **username:"Smith, Bob" role:design** in the field.

2. Select which session context in which you wish to work.

If you select the site, you can only work with site locations.

If you select a group, you can work with that group's location overrides.

If you select a role, you can work with location overrides for that role and its group.

If you select a user, you can work with location overrides for that user, role, and group.

The preference list will be populated with all valid preference locations for the session context that you have selected, and you are able to:

- Modify the values of existing preference locations in the session context.
- Create new preference locations for the session context.
- Override preferences for this session context, if the preference scope allows it.

The preference list

The screenshot shows the Teamcenter preference management interface. On the left, a table lists various preferences. On the right, the 'DEFINITION' and 'VALUES' for a selected preference are displayed.

Name	Location
AWC_CPPackageltemRevision.SUMMARYRENDERI...	Site
AWC_CPTradeltemRevision.SUMMARYRENDERING	Site
AWC_Cdm0DataReqItemRevision.SUMMARYREND...	Site
AWC_ItemRevision.SUMMARYRENDER...	Site
AWC_Pdm1ProblemItemRevision.SUMMARYREND...	Site
AWC_mockTCAWAdminWorkspace.ItemRevision.S...	Site
AWC_mockTCAWAuthorWorkspace.ItemRevision.S...	Site
AWC_mockTCAWConsumerWorkspace.ItemRevisio...	Site
CPPackageltemRevision.SUMMARYRENDERING	Site
CPTradeltemRevision.SUMMARYRENDERING	Site
Cdm0DataReqItemRevision.SUMMARYRENDERING	Site

DEFINITION

Name: AWC_ItemRevision.SUMMARYRENDERING

Product Area: * Active Workspace

Description: * Registration of summary stylesheet for Item Revision objects in Active Workspace Client - This Preference Created by Mohan Elankuppan

Protection Scope: * User

Environment: * Disabled

Location: Site

Type: String

Multiple Values: No

VALUES

Awp0ItemRevSummary

Edit

Select your working context in the **Organization** panel, if available.

1. (optional) Filter the preference list by category.
2. (optional) Filter listed preferences.
3. Select a preference.
4. View preference information.
5. (optional) Edit the value at this preference location.

If you do not have permission, you will not see this button.

Override a preference

To override a preference, you must create a new instance of the preference at a higher-precedence **location**. Each preference defines its own **scope**, which is the highest precedence location allowed.

For example, If a preference's scope is **Site**, then it cannot be overridden, but if its scope is **User** then it can be overridden at every level.

If a preference instance's location is **Site**, it will be overridden by any other location instance but if its location is **User** then it overrides any other location for that specific user.

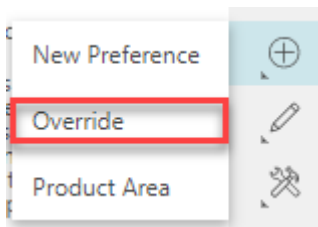
Tip:

If a preference in the list has a **location** of **None**, then that means it is the preference *definition* and there are no current location instances.

Following are the levels of precedence for locations.

	scope	location
User	Can override at any location.	Overrides all other location values.
Role	Can override at group and role locations.	Will override Site and Group location values.
Group	Can only override at group locations.	Will override Site location values.
Site	Can not override.	Is overridden by any other location

1. (optional) Filter the preference list.
2. Select the preference you want to override.
3. Open the **New** command stack ⊕, and then choose **Override**.



4. In the **Add Override** panel, choose the location for the override (if allowed), set the new value, and then choose **Add**.

Displaying items instead of revisions

Caution:

This is a non-standard configuration and requires careful planning. Work with your Siemens Digital Industries Software representative if you need this functionality.

Active Workspace is designed for the user to work almost exclusively with objects of the **ItemRevision** type (and its children).

If you have a *special use case* and you need to display item types, then you need to configure the following:

- Allow the user to create an Item.

Modify the **AWC_DefaultCreateTypes** preference to include an item type. That item type and all subtypes will be considered creatable.

- Change the **Fnd0ItemRevPasteOnTargetUponCreate** business object constant for the new type. This will automatically paste the item type into the user's folder when it is created.
- Configure indexing to work with Items instead of revisions.

In addition, you should also suppress the ItemRevision type from the UI.

Caution:

Siemens Digital Industries Software does not recommend that you present both items and revisions to your users.

Deleting various object types

By default, the **Delete** command does not appear for every object type your users can see. If you want to add additional object types to the list, then you must change the following preference:

AWS_allowedTypesForDelete

This is a multiple-value preference that accepts a list of business object types for which the **Delete** command appears in the Active Workspace interface.

Caution:

Even when the **Delete** command is visible, there are still two conditions that must be met for your users to delete an object:

- They must have the delete permission for the object.
- The object must not be referenced by other objects.

Special behavior for folders

You can control what happens when your users attempt to delete an object with folder references by changing the following preference:

TC_auto_delete_folder_references

- **true (default)**

The default value is **true**, which ignores folder references when checking for references to other objects, and if no *other* references are found, then the folder references are automatically removed, and then the object is deleted without complaint.

- **false**

Changing this to **false** prevents the object from being deleted from the database and presents an error message if it has any references to any other objects, including folders.

This preference applies when deleting a folder that contains objects, and when an object is contained in a folder.

Controlling notification timeout

You can control the notification panel timeout using a preference.

AWC_notification_timeout

The value is the number of seconds to wait before closing the notification. If the value is negative, the window will not close automatically.

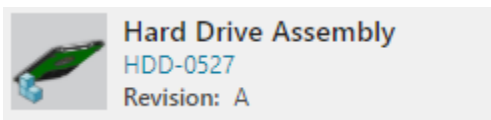
Defining properties that display in object cells

To define the properties that display on the cells for objects in Active Workspace list view, use the *business-object.CellProperties* preference. The first two properties in the list of properties in the preference are displayed without labels and are formatted as a primary title and subtitle. The remaining properties are displayed in the cell as *name:value*.

The default values vary by object type. For example, following are the default values of the **ItemRevision.CellProperties** preference:

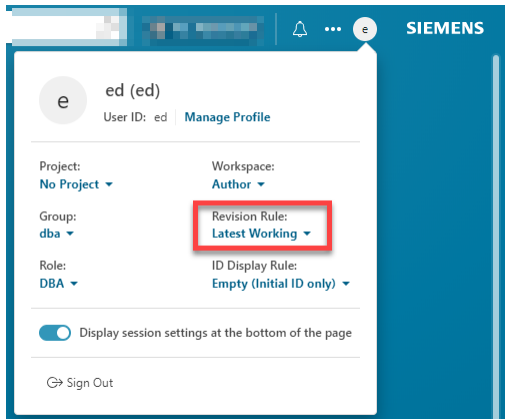
```
object_name
item_id
item_revision_id
```

The values in this example appear as follows in Active Workspace.



Defining the revision rules list

Revision rules determine the state of objects you view in the user interface. Users click the revision rule to display the list of all available revision rules. To set a different revision rule, a user selects another revision rule from the list.



By default, the list of available revision rules is obtained from Teamcenter. However, as an administrator, you may want to provide different revision rules for Active Workspace than are used in the rich client. For example, you may want to have Active Workspace default to **Latest Released** whereas you want the rich client to still default to **Latest Working**.

To set a different list of revision rules for Active Workspace, add revision rules to the **AWC_Rev_Rule_List** preference. Whenever a custom revision rule is created, you must add it to this preference for it to appear in the revision rules list. By default, the preference is empty, meaning that the revision rules list in Active Workspace defaults to the revision rules from the rich client.

To set the revision rule that is selected by default, add it to the **AWC_Rev_Rule_Selected** preference. The revision rule in this preference must match a revision rule in the **AWC_Rev_Rule_List** preference.

If the revision rule in the **AWC_Rev_Rule_Selected** preference is removed from the **AWC_Rev_Rule_List** preference, you must change the revision rule in the **AWC_Rev_Rule_Selected** preference to one in the **AWC_Rev_Rule_List** preference.

By default, the **AWC_Rev_Rule_Selected** preference is empty, meaning that the first revision rule in the **AWC_Rev_Rule_List** preference is the one that is selected by default in the user interface.

Where can I get a list of preferences?

There are several sources from which to retrieve a list of preferences and their definitions.

Administration data report

You can find the **Administration Data Report** in the **References for Administrators and Customizers** in the Teamcenter documentation area in Support Center. In this report, you will find a complete list of all preferences shipped with Teamcenter. When you install additional features, like

Dispatcher, NX Integration, 4th Generation Design, and so on, additional preferences will be added to your site. To get the most accurate and up-to-date listing of preferences contained in your site, you must create your own Administration Data Documentation report.

Rich client

You can use the various tabs of the rich client's **Edit→Options** menu to interact directly with preferences, including a report of which preferences have changed since installation.

Raw XML export

You can produce an XML file containing preference information using the **preferences_manager** utility.

Active Workspace client

You can interact directly with preferences using the **showPreferences** location.

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.

- **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. The default value is **5**.

The threshold to display the box to search filter values is twice the value of **Awp0FilterValueDisplayCount**.

- **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. (This global constant is used if no value is set for the **AW_Indexable_File_Extensions** preference.)

The following values are supported:

<code>.as</code>	<code>.dotm</code>	<code>.msg</code>	<code>.various</code>	<code>.xltm</code>	<code>.jp2</code>
<code>.aw</code>	<code>.dotx</code>	<code>.ods</code>	<code>.vdx</code>	<code>.xltx</code>	<code>.dft</code>
<code>.csv</code>	<code>.eml</code>	<code>.pdf</code>	<code>.wo</code>	<code>.xlw</code>	<code>.prt</code>
<code>.dat</code>	<code>.epub</code>	<code>.ppt</code>	<code>.wpd</code>	<code>.xlsx</code>	
<code>.dc</code>	<code>.fff</code>	<code>.pptx</code>	<code>.xml</code>	<code>.zip</code>	
<code>.dif</code>	<code>.htm</code>	<code>.rtf</code>	<code>.xla</code>	<code>.7z</code>	
<code>.doc</code>	<code>.html</code>	<code>.stc</code>	<code>.xlam</code>	<code>.png</code>	
<code>.docm</code>	<code>.ip</code>	<code>.sxc</code>	<code>.xls</code>	<code>.jpg</code>	
<code>.docx</code>	<code>.mdb</code>	<code>.tar</code>	<code>.xlsm</code>	<code>.tiff</code>	
<code>.dot</code>	<code>.mif</code>	<code>.txt</code>	<code>.xlt</code>	<code>.gif</code>	

For example, to enable indexing on compressed files, add any combination of `.zip`, `.7z`, or `.tar` to the list.

Read permission is required for indexing a file. If any file is password protected, the file will be skipped and no warning will be generated.

- **Awp0StoreDatasetContent**

Global search results can display the location of search terms inside file content attached to items. Search results that match content from attached dataset files display as snippets of text matching the search terms. The Indexing Engine (Solr) requires that you set this global constant In Business Modeler IDE to true to ensure that dataset fields are stored in the Solr schema. You must also configure the preferences to enable snippets, merge the schemas, and reindex your data, including datasets.

Business object constants

Following are the business object constants unique to Active Workspace:

- **Awb0AvailableFor**

Lists the business object types for which a feature should be made available. The values are comma separated. This constant honors type hierarchy.

When this constant is used with the **Ase0ArchitectureFeature** business object, the constant controls visibility of the **Architecture** tab for business object types. The **Architecture** tab is made visible for all listed business object types and their all subtypes. The types should be those that represent the top line in structures. This constant supports comma-delimited values of business object types, for example:

Functionality
 Fnd0LogicalBlock
 RequirementSpec
 Requirement
 Paragraph
 Fnd0SystemModel

Note:

The **Architecture** tab is not visible for custom business objects. Determine the business object types that require the **Architecture** tab. From these types, determine those that are the top line in structures. Add only those types to the value of the **Awb0AvailableFor** business object constant on the **Ase0ArchitectureFeature** business object.

You can edit the display name configured for the **Ase0ArchitectureFeature** business object in Business Modeler IDE to suit your business processes. By default, the display name of the tab is **Architecture**.

- **Awb0BOMArchetypeToOccurrence**

Determines the instance of which particular subtype of the **Awb0Element** business object is created. This business object constant 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.

- **Awp0SearchIsIndexed**

Indicates that the business object will be indexed for searching when this constant is set to **true**. 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.

Note:

Do not set this constant to **true** for the **WorkspaceObject** business object. This results in indexing errors.

- **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 **Awp0AWCExternalSystemObject** business object, which designates objects originating in systems external to Teamcenter.

To change the value to **true**, open the **Awp0AWCExternalSystemObject** 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.

- **Awp0SearchDatasetIndexingBehavior**

Defines the behavior of inline indexing for dataset file contents. The scope for this constant is **WorkspaceObject** types and their subtypes. Specify one of the following:

Inline	Dataset content is indexed inline with the parent business object. When a search matches dataset content, the search results returns the parent business object instead of the dataset. The datasets are defined using Awp0DatasetTypeToBeIndexedInline .
Relation	Dataset content is indexed as part of the dataset but it maintains the reference to the parent business object. When a search matches dataset content, the search results return the dataset as well as the parent business object.

The default value is **Inline** except for **DocumentRevision**, where the default is **Relation**.

- **Awp0DatasetTypeToBeIndexedInline**

Identifies the datasets to be indexed along with the business object. The scope for this constant is **WorkspaceObject** types and their subtypes.

Set this property only for types that are also marked for indexing and **Awp0SearchDatasetIndexingBehavior** is set to **Inline**.

The format is:

```
<INHERIT | NO_INHERIT> : RelationName : DatasetTypes
```

The keyword specifies the behavior to apply to the rule:

INHERIT	Applies the rule to the specified type and all its subtypes. For example, index PDX dataset content related to TC_Attaches for ItemRevision and its subtypes:

```
INHERIT:TC_Attaches:PDF
```

NO_INHERIT	Applies the rule only to the type where the rule is defined. A rule applied to a parent is not inherited by child types. Specifying NO_INHERIT can help improve performance.

```
NO_INHERIT:IMAN_Specification:Text
```

You can specify one to many relationships between *RelationName* and *DatasetTypes*.

RelationName is a relation name or the wildcard character *. Specify one or more clauses separated by commas.

INHERIT: TC_Attaches: PDF, INHERIT: IMAN_Specification: Text

INHERIT: *: PDF

DatasetTypes is a dataset type. Specify one or more values separated by a tilde ~.

INHERIT:TC_Attaches: PDF~MSWordX

Specifying only wildcards is not valid (for example, do not specify **INHERIT: *: ***).

No default value is specified with the exception that **SpecElementRevision** is set to **INHERIT: *: FullText**.

As a best practice, do not specify all relations (*) for inline indexing and then subsequently try to limit inheritance by setting a subtype clause to index only a specific relation to a specific type. You can avoid inheritance by:

- Using **NO_INHERIT** to limit the scope of indexing for a specific type.

For example, if all **ItemRevision** PDFs for any relation are being indexed inline, do not write a qualifying **INHERIT** rule for a subtype. For example:

If an **INHERIT** rule for **ItemRevision** is defined as **INHERIT: *: PDF**,

And, an **INHERIT** rule for an **ItemRevision** subtype indexes only PDFs associated with the **TC_Attaches** relation,

Then the indexing behavior at the subtype level might not behave as expected, because you already specified all **ItemRevision** subtypes to index all PDFs,

- Configuring the subtype to override the parent. For example, to index PDX content for all the relations of the subtype, set **INHERIT: *: PDX** .
- Setting **Awp0DatasetTypeToBeIndexedInline** to an empty string for the subtype avoids all inheritance from the parent type.
- Setting **Awp0SearchIsIndexed** to **false** to turn off indexing for the type.

By default, the indexing of **FullText** datasets is not enabled because they are indexed inline for **SpecElementRevision** and its subtypes. If you choose to enable indexing of **FullText** datasets, users see **FullText** and **SpecElementRevision** objects in search results.

Property constants

The following property constants are unique to Active Workspace:

- **Awp0FilterPropFromRefType**

Applicable only when the property to be indexed is a reference type or a compound property whose source property is a form data file property.

You can use the **awp0MasterFormStorageClass** compound property, available by default on all Master forms for **ItemRevision** and its subtypes, to index and filter the properties of the form.

Specify a comma-separated list of properties that are a subset of the properties listed in the **Awp0SearchPropFromRefType** property constant. For example, you can set the following constants for the property constant reference type you want to filter:

Awp0SearchIsIndexed

Set the Boolean value to **true** to search on the property.

Awp0SearchCanFilter

Set the Boolean value to **true** to filter on the property.

Awp0SearchPropFromRefType

Enter the list of properties to index.

Awp0FilterPropFromRefType

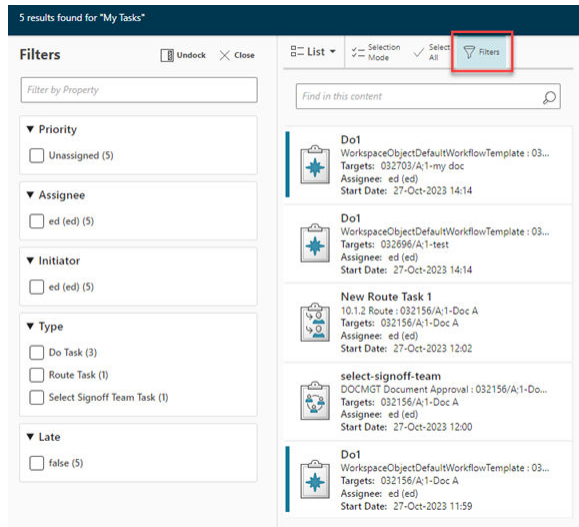
Enter the list of properties to filter.

Awp0SearchRefTypesNames

Enter the referenced object type names.

- **Awp0InboxCanFilter**

Indicates that tasks shown in the inbox can be filtered on a specific property of a workflow business object (**EPMTask** and its subtypes). The following example shows 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.
- **Awp0InboxFilterPriority**

Sets 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 Digital Industries 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.

- **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. For the filters to show up correctly in the user interface, this property constant should be set for the property on its source business object.

In a Multi-Site Collaboration environment, apply **Awp0SearchCanFilter** to published record objects so that they can be indexed using the **POM_owning_site** property.

- **Awp0SearchFilterPriority**

Sets 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 Digital Industries 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.

For the filters to show up correctly in the user interface, this property constant should be set for the property on its source business object.

When a **Table** type property is marked for filtering, all valid properties of the referenced **TableRow** type are available as filters. All the table row properties get the same filter priority, so they are displayed together, vertically listed in the filter pane.

- **Awp0SearchIsIndexed**

Indicates that the property on the business object will be indexed for searching by the indexing 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, table, reference, and publication record properties can be indexed. Indexing of runtime and relation properties is not supported.
- For multi-site searching the **POM_owning_site** property can be indexed out of the box. Apply the **Awp0SearchIsIndexed** to published objects to enable indexing.
- To index compound properties, they must reference attribute properties from the source object.

Note:

If the compound property value comes from a form, the compound property must use the form storage class in the property definition rather than the form itself, or indexing fails.

- To index reference properties, the **Awp0SearchRefTypeNames** and **Awp0SearchPropFromRefType** property constants must contain valid values.
- Table type property indexing is only supported for properties that reference **Fnd0TableRow** and its subtypes. Indexing is not supported for **Fnd0NameValue** and its subtypes.

When a table type property is marked for indexing, all the valid properties of the referenced table row type are indexed. You do not need to mark individual table row properties.

Incorrect values are omitted from indexing; no message appears.

- **Awp0SearchPropFromRefType**

Applicable when the property to be indexed is a reference type or a compound property whose source property is a form data file property.

You can use the **awp0MasterFormStorageClass** compound property, available by default on all Master forms for **ItemRevision** and subtypes, to index and filter the properties of the form.

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:

- You can only specify attribute and compound properties.

Note:

If the compound property value comes from a form, the compound property must use the form storage class in the property definition rather than the form itself, or indexing fails.

- Each property in the list specified for **Awp0SearchPropFromRefType** is matched against each business object in the list specified for 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.
- The first property in the list must be a string property.

Incorrect values are omitted from indexing; no message appears.

- **Awp0SearchRefTypeNames**

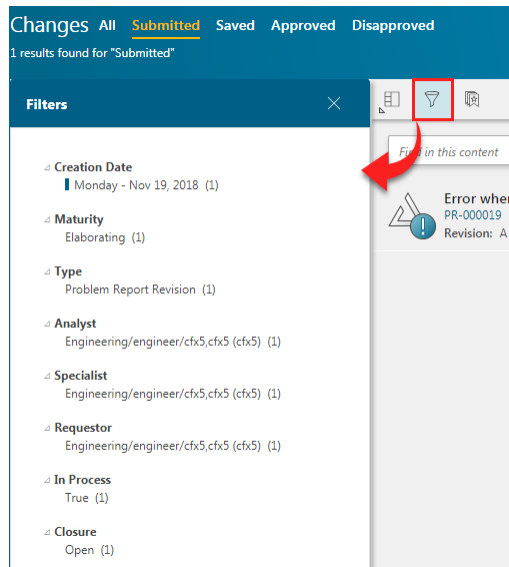
Applicable only when the property to be indexed is a reference type or a compound property whose source property is a form data file property. Specify a comma-separated list of business object names that the reference property can contain. For example, on the **owning_user** reference property on **ItemRevision**, specify a value of **User**. The following rules apply:

- 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** results in **ReleaseStatus**.
- 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.

- **Cm1ChangeCanFilter**

Indicates that the changes in the **Changes** page can be filtered on a specific property of the change business object (**ChangeltemRevision** and its subtypes). The following example shows 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.
- **Cm1ChangeFilterPriority**

Sets 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 the **Changes** page. The lower the value, the higher its priority and, therefore, the higher its position in the list of filters.

Siemens Digital Industries 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.

Note:

Compound properties presume that the security level of the source property of the source object is read access. This is because the compound property belongs to the target object related to the source object through the compound property.

For example, Object B has a compound property that is related to its source property on Object A. When Object B and its compound property are indexed, the indexer presumes that Object A has a security level of read access to everyone. That means that anyone with read access to the compound property of Object B could also find the source property on Object A, regardless of its security level.

Enabling searching and filtering on referenced objects or forms

Enable searching and filtering for a referenced object using a property value defined on the referenced object. You need to add or update the property constants for the property of the referenced object.

In the first example, an **ItemRevision** object has a referenced **Item** object, where the **description** contains the value **Example Text**.

In the Business Modeler IDE, find the template containing your object definitions, and find the **ItemRevision** object. On the property that references the **Item** object, add or update the following property constants:

Awp0SearchIsIndexed

Set the Boolean value to **true** to index this property.

Awp0SearchRefTypesNames

Set the referenced object type name, **Item** in this example.

Awp0SearchPropFromRefType

Set the list of properties on the referenced object type specified in **Awp0SearchRefTypesNames**. In this example, the property is **description**.

To filter the search results by the preceding property values, add or update the following property constants:

Awp0SearchCanFilter

Set the Boolean value to **true** to filter on the property.

Awp0FilterPropFromRefType

Set the list of properties on the referenced object. In the example, the property is **description**.

Awp0SearchFilterPriority

This property constant is optional. You can set a numeric value for priority. The lower the value, the higher the priority.

Example: Search properties of a related Master or custom form

Enable searching and filtering using the properties of a related Master or custom form. You need to add or update the property constants for the property of the related form. By default, the **awp0MasterFormStorageClass** compound property is configured for **ItemRevision**. The compound property references the data file on the Master form for **ItemRevision**.

1. An **ItemRevision** has a subtype **Awp0TestItemRevision** with a Master form **Awp0TestItemRevisionMaster**:

Business Object : Awp0TestItemRevision

Main Properties Operations Display Rules Deep Copy Rules GRM Rules Operation Descriptor

Details

Project: aws2

Name: Awp0TestItemRevision

Display Name: Test Item Revision

Storage Class: Awp0TestItemRevision

Parent: ItemRevision

Item: Awp0TestItem

Form: Awp0TestItemRevisionMaster

Icon: Default

2. The Master form has two properties:

Form : Awp0TestItemRevisionMaster

Main Properties Operations Display Rules Deep Copy Rules GRM Rules Operation Descriptor

Enter Search Text Here

Property Name	Type	Storage Type	Inherited	Source	COTS
acl_bits	Attribute	Integer	✓	POM_application_object	✓
active_seq	Attribute	Integer	✓	WorkspaceObject	✓
archive_date	Attribute	Date	✓	POM_application_object	✓
awp0CellProperties	Runtime	String[400]	✓	BusinessObject	
awp0MasterFormProp1	Runtime	String[128]		Awp0TestItemRevisionMaster	
awp0MasterFormProp2	Runtime	String[128]		Awp0TestItemRevisionMaster	
awp0ThumbnailImageTicket	Runtime	String[400]	✓	BusinessObject	
backup_date	Attribute	Date	✓	POM_application_object	✓

3. **Awp0TestItemRevMasterS** is the storage class name for the form.

Form : Awp0TestItemRevisionMaster

Main	Properties	Operations	Display Rules	Deep Copy Rules	GRM Rules	Operation Descriptor
Details						
Project:	aws2					
Name	Awp0TestItemRevisionMaster					
Display Name	Test Item Revision Master					
Storage Class	 Form					
Parent	 ItemRevision Master					
Icon	Default					
Type	Persistent					
	☐ Is Abstract? ☒ Is Exportable? ☒ Allow creating instances of Secondary Business Objects ☐ Store as lightweight object					
Description:	* Test Item					
Master Form Owner	 Awp0TestItemRevision					
Form Storage Class	 Awp0TestItemRevMasterS					
	☐ COTS?					
Template	aws2					

4. **awp0MasterFormStorageClass** is a compound property that references the data file of the Master form.

Compound Property Page
Manage a CompoundProperty

Project:

Name: *

Display Name: *

Description: *

☐ ReadOnly

Path: *

▼  ItemRevision.IMAN_master_form_rev

 Form.data_file

5. Configure the Master form storage class property **awp0MasterFormStorageClass** with the property constants that enable indexing, searching, and filtering.

Business Object : Awp0TestItemRevision

Main Properties Operations Display Rules Deep Copy Rules GRM Rules Operation Descriptor

Enter Search Text Here

Property Name	Type	Storage Type	Inherited	Source
awp0MasterFormStorageClass	Compound	UntypedReference	✓	ItemRevision
awp0RequiredParticipants	Runtime	String[128]	✓	ItemRevision
CAEAnalysis	Configured Runtime	UntypedReference	✓	ItemRevision

Property Constants Naming Rule Attaches LOV Attaches Property Renderer Attaches Property Formatter Attachments

Property Constants of awp0MasterFormStorageClass

Name	Value	Overridden	Allow More
Awp0FilterPropFromRefType	awp0MasterFormProp1	✓	✓
Awp0SearchCanFilter	true	✓	✓
Awp0SearchFilterPriority			✓
Awp0SearchIsIndexed	true	✓	✓
Awp0SearchIsStored	false		✓
Awp0SearchPropFromRefType	awp0MasterFormProp1,awp0MasterFormProp2	✓	✓
Awp0SearchRefTypeNames	Awp0TestItemRevMasterS	✓	✓

Example: Search properties of an Item Master form

Enable searching and filtering of an Item Master form using a compound property. You need to create a compound property on a persistent attribute of the Item Master **data_file** storage class. The compound property references the data file for the Master form of the **Item**. You need to create a compound property for every attribute you want to search.

In the example, the compound property **a2ItemMasterAttr1** is created on **user_data1**, which is a persistent attribute on the Item Master **data_file** storage class.

The screenshot shows a configuration window for a new compound property. The fields are as follows:

- Name:** * a2ItemMasterAttr1
- Display Name:** * Item Master Attr1
- Description:** (empty text box with up/down arrows)
- ReadOnly:** ☐ (unchecked)
- Path:** * A tree view showing the hierarchy:
 - ItemRevision.items_tag
 - Item.IMAN_master_form
 - Item Master.data_file
 - ItemMaster.user_data_1 (selected)

Be sure to set the index, search and filter property constants on the new compound property.

Utilities

Using command-line utilities

In order to perform some administrative activities, you must run command-line utilities. Even if it's not the only option, sometimes using command-line utilities can also make some administrative tasks easier.

To run command-line utilities, you must have access to the Teamcenter platform command-line environment.

For information about working with command-line utilities, refer to the *Teamcenter Utilities* in the Teamcenter documentation.

ac0_migrate_s2cldata

Migrates previous Active Collaboration questions and comments (**S2clSocial** objects) to the Active Collaboration discussions feature that uses **Ac0ActiveCollaboration** objects.

Siemens Digital Industries Software recommends that you first run this utility with the **-dryrun** argument before performing an actual migration.

Note:

If you have existing questions, replies, and comments from the Active Collaboration component in Active Workspace 5.0 or earlier or the Active Collaboration for Retail component in Active Workspace 6.0 or earlier, you can perform a one-time migration using this utility to migrate your existing questions and comments to the new Active Collaboration discussions feature. You can also view the information in the Teamcenter Administration guide to learn how to configure Active Collaboration.

Caution:

Perform this migration only once to prevent duplication of data.

SYNTAX

```
ac0_migrate_s2cldata -u=admin-user-id [{-p=password | -pf=password-file}] [-g=group]
-report=report-file-path
[-dryrun]
[-h]
```

ARGUMENTS

-u

Specifies the user ID.

A user with administration privileges is used as the value name for the user ID. If **-u** is used without a value, the operating 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, which must be **dba**.

-report

Specifies the path where the migration report file is stored.

-dryrun

Simulates a run of the **ac0_migrate_s2cldata** utility without actually migrating the data. The results of the simulated migration are logged in the file specified by **-report**. Review the report for any potential issues before actually migrating the data.

-h

Displays help for this utility.

EXAMPLE

Migrate the existing **S2clSocial** objects to **Ac0ActiveCollaboration** objects:

```
ac0_migrate_s2cldata -u=admin -p=pwd -g=dba  
-report=C:\temp\admin_data\active_collaboration_objects.xml
```

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 a 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.

-pf

Specifies the password file that holds the cleartext or encrypted password. For enhanced security, use a password file instead of the password. If the **-pf** argument is not used, the system uses the given password.

-p

Specifies 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.

-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 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 | subscribers | closure-rules
```

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 |
MMV | clouserule1
```

- **effectivity-unit**

If you have multiple effectivity units, their numbers must be comma-separated. Also, you must repeat the effectivity end item query string for each effectivity unit, for example:

```
| 5,10,12 | item_id=HDD-0527,item_id=HDD-0527,item_id=HDD-0527 |
```

The maximum number of effectivity units you can specify is 960.

- **variant-rules**

The variant rules (also known as saved 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 maximum number of saved variant rules you can specify is 960.

- **subscribers**

(Optional) You may specify:

- **VDS**

Specifies that this product configuration is indexed for viewing using the Visualization Data Server. This makes the structure available faster in visualization. **VDS** requires deployment of the Visualization Data Server.

- **closure-rules**

(Optional) If a closure rule is applied for a configuration, content in the structure (excluded by the closure rule) does not appear in **where used** query results for top-level contexts.

- **structure-type**

(**Defunct**) The type of structure that the product represents. **OCC** is the only valid value.

- **owning-user**

(**Optional**) If Visualization for MMV is enabled for the product, specify the user who owns the MMV delta collection dataset.

- **MMVIdxAccessControllers**

(**Optional**) Specify the users that control access to MMV index files.

HOW TO SPECIFY OWNING USER AND MMVIDXACCESSCONTROLLERS

Each VDS site can be configured to run as a different user. **BOMIndexAdminData** table entries are returned according to the permission specified for the configured VDS user. This helps you implement export compliance using these attributes.

Example:

Site owners configured for each VDS site:

- **USA** is **VDSadmin**
- **Mexico** is **mex_VDSadmin**
- **Canada** is **can_VDSadmin**
- **China** is **chi_VDSadmin**

BOMIndexAdminData (BIAD) entries are defined as follows:

BIAD	Product	Revision Rule	Structure Type	MMV Owning User	MMV Access Users
Biad1	Ship	Latest Working	BVR	VDSadmin	mex_VDSadmin can_VDSadmin
Biad2	Truck	Latest Working	BVR	VDSadmin	mex_VDSadmin can_VDSadmin chi_VDSadmin
Biad3	Car	Latest Working	BVR	VDSadmin	chi_VDSadmin

To summarize the access for each VDS site:

- **USA** site user **VDSadmin**
Access to **Biad1**, **Biad2**, and **Biad3** because it is specified as the MMV owning user.
- **Mexico** site user **mex_VDSadmin**
Access to **Biad1** and **Biad2** only. No access to **Biad3**.
- **Canada** site user **can_VDSadmin** user
Access to **Biad1** and **Biad2** only. No access to **Biad3**.
- **China** site user **chi_VDSadmin** user
Access to **Biad2** and **Biad3** only. No access to **Biad1**.

EXAMPLE

The following command creates 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
```

OVERRIDING EFFECTIVITY

When you want to specify override effectivity, do not specify it in the input file containing the product configurations to index. The override effectivity in the input file is ignored during index generation, causing a discrepancy between the indexed BOM and the BOM in use.

To set override effectivity during index generation, add the effectivity data to a Revision Rule.

For example, the Revision Rule might contain:

Z_ACE_DateOverride_Rule23_10Jan2020

which includes entries for the effective date **10-Jan-2020 00:00:00**.

The corresponding input file entry for **bomindex_admin** would have this corresponding effectivity override entry:

```
item_id=ACE_KK_EC01 | A | Z_ACE_DateOverride_Rule23_10Jan2020 | | | | |
```

data_report

Gathers data from participating Multi-Site sites in the federation and generates reports on data inconsistencies. Reports are generated for individual sites and for the overall Multi-Site federation. This data is presented in the Active Workspace Multi-Site Assistant. (When working with the **data_report** utility, sites in the federation are the sites defined by **MS_Dashboard_Supported_Sites**.)

SYNTAX

```
data_report -u=user-ID {-p=password | -pf=password-file} -g=group
[-site=site-name]
[-dir=report-directory]
[-format=report-format]
[-sql_file=sql-file-name]
[-include=included-report-categories]
[-exclude=excluded-report-categories]
[-date_format=date-filter-type [-from=start-date] [-to=end-date]]
[-object_type=type]
[-f={collect_data | generate_report}]
[-h]
```

ARGUMENTS

-u

Specifies the user ID. The user must have administrative privileges.

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. This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

-site

Specifies the name of a specific site for which data is gathered and a report is generated. If no site is specified, reports are generated for all sites and the overall Multi-Site federation (as defined by **MS_Dashboard_Supported_Sites**).

-dir=

Specifies directory in which reports are saved. By default, reports are saved in the location specified by **TC_TMP_DIR**.

-format

Specifies the format for reports. Valid values are **csv** (the default) and **json**.

-sql_file

Specifies a JSON file containing custom SQL data queries or predefined filters.

-include

Specifies the categories of issues to include in reports. Valid values are **itemOwnershipReports**, **duplicateIDReports**, and **objectOwnershipReports**. By default, all listed categories are included. Multiple values must be separated by commas.

The **objectOwnershipReports** analysis uses the external closure rules defined in **MSA_ObjectOwnershipCR**. As the **objectOwnershipReports** analysis may take significant time, consider first running **data_report** excluding **objectOwnershipReports**, and then later running only the **objectOwnershipReports**.

-exclude

Specifies the categories of issues to be excluded from reports. Valid values are **itemOwnershipReports**, **duplicateIDReports**, and **objectOwnershipReports**. Multiple values must be separated by commas.

-date_format

When specifying a reporting date range, **-date_format** specifies the date type. Valid values are **creation** (the date the object was created) and **last_modified** (the date the object was last modified).

-from

Used with **-date_format** to specify, inclusively, the start date and time for reporting. The format of the date is "YYYY-MM-DD HH:MM:SS" and must be inside the double quotes due to the space between the year and the hour.

-to

(Optional) Used with **-date_format** to specify, inclusively, the end date and time for reporting. The format of the date is "YYYY-MM-DD HH:MM:SS" and must be inside the double quotes due to the space between the year and the hour. If not given, the current date and time is used.

-object_type

Specifies an object type to be included in the report. All item and item subtypes are included.

-f

Specifies the function that **data_report** should perform. Valid values are **collect_data** (gather updated data from sites) and **generate_report** (generate reports based on gathered data). By default, **data_report** performs both actions.

-h

Displays help for this utility.

EXAMPLES

Required log-in information is omitted from the following examples.

- Gather data and generate reports for all sites defined by **MS_Dashboard_Supported_Sites**:

```
data_report
```

- Gather data for a specific site:

```
data_report -site=site01 -f=collect_data
```

- Generate reports for all sites and the Multi-Site federation using data previously collected:

```
data_report -f=generate_reports
```

- Generate reports for all sites and the Multi-Site federation for only specific categories of issues:

```
data_report -include=itemOwnershipReports,duplicateIDReports
```

- Gather data and generate reports for a specific site for a specific time period:

```
data_report -site=site01 -date_format=creation -from="2020-06-01 00:00:00"  
-to="2020-06-30 23:59:59"
```

- Gather data and generate reports for all sites and the Multi-Site federation using custom SQL data queries:

```
data_report -dir=d:\reports -sql_file=d\reports\msPredefinedSQLs.json
```

generate_admin_data_report

Generates a report showing the specified administration data for the site where you run the utility or for an export package. The export package can be from a remote site.

The report contains HTML pages for the administration data objects, showing their properties with hyperlinks to referenced objects. If an object is referenced by other objects, its HTML page contains a where-used table that indicates the categories and objects that have references to the current object.

The report has a summary showing all the administration data types included in the report and the instances of each element present within the category. The report also has a glossary page with descriptions of the administration data categories and the elements available in each of the categories.

SYNTAX

```
generate_admin_data_report -u=user-ID {-p=password | -pf=password-file}
-g=group
-adminDataTypes=Admin-data1,Admin-data2,...,Admin-dataX | all
[-inputPackage=input-package-path]
-outputDir=path-to-directory-for-report-files
[-listTypes]
[-h]
```

ARGUMENTS

-u

Specifies the user ID. The user must have administrative privileges.

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.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

For more information about managing password files, see *Teamcenter Utilities* in Teamcenter help.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

-adminDataTypes

Specifies the types of administrative data to include in the compare report. You provide the data types as a comma-separated list (no spaces). You may also specify the **all** value to include all data types defined in the local system or the specified input package.

Tip:

Use the **-listTypes** argument to get a list of available administration data types.

If the report contains multiple data types, it includes a where used table showing where each object is referenced.

-inputPackage

Specifies the full path, including the file name, of the export administration data package from the site for which the report is generated. If you do not specify this argument, the utility generates a report for the local site.

-outputDir=

Specifies the path to directory where you want the report saved. You must specify this argument.

-listTypes

Displays a list of the available administration data types that you can include in the report.

-h

Displays help for this utility.

ENVIRONMENT

As specified in *Manually configure the Teamcenter environment* in *Teamcenter Utilities* in Teamcenter help.

This is a Java utility that, by default, has the maximum Java heap size set to 1024M. For reports that contain a large number of objects, you may need to increase maximum Java heap size to avoid out-of-memory errors or poor performance. If possible, set the maximum heap to at least to 4096M for large reports. You can set this value using the **BMIDE_SCRIPT_ARGS** environment variable, for example:

```
set BMIDE_SCRIPT_ARGS=-Xmx4096M
```

Note:

Java standards require that no more than 25 percent of total RAM be allocated to virtual memory (VM). If the amount allocated to the Java VM exceeds this percentage, degradation of performance can occur.

FILES

As specified in *Log files produced by Teamcenter* in *Teamcenter Utilities* in the Teamcenter help.

EXAMPLES

- Generate a list of the administration data types that you can export:

```
generate_admin_data_report -u=admin-username -p=admin-password -g=dba
-listTypes
```

- Generate a report containing the preferences and their values at the local site:

```
generate_admin_data_report -u=admin-username -p=admin-password -g=dba
-adminDataTypes=Preferences
-outputDir=C:\temp\admin_data\siteA\preferences_report
```

- Generate a report containing the Access Manager and Organization administration data from an export package of a remote site:

```
generate_admin_data_report -u=admin-username -p=admin-password -g=dba
-adminDataTypes=AccessManager,Organization
-inputPackage=C:\temp\admin_data\siteB\siteB.zip
-outputDir=C:\temp\admin_data\siteB\am_and_organization_report
```

log-level

Changes the log level of Teamcenter microservice framework components and compliant microservices. Among these are the service dispatcher and the microservices **configurator**, **filerepo**, and **MPS**. Administrators can change logging levels without having to stop and restart the services.

Syntax

Host operating system	Syntax
Linux	<pre>./log-level operation [argument]</pre> On Linux hosts, it is necessary to set permissions on the <i>log-level.sh</i> file. To do so, enter the command <pre>chmod +x log-level.sh</pre>
Windows	<pre>log-level operation [argument]</pre>

Arguments

get

Gets the current log level of a microservice.

```
log-level get MICROSERVICE_NAME
```

```
>log-level get filerepo
Microservice Name: filerepo
Current Log Level: WARN
Possible Values: [INFO, ERROR, DEBUG, WARN, TRACE]
```

Available log levels depend on the microservice. See Server manager logging levels in the Teamcenter help for a description of log levels.

getall

Gets the current log level of all running microservices.

```
log-level getall
```

```
>log-level getall
-----
| Microservice Name | Log Level |
-----
| filerepo          | WARN     |
| mps               | ERROR    |
| servicedispatcher | INFO     |
-----
```

set

Changes the log level of a microservice. Possible values are INFO, ERROR, DEBUG, WARN, and TRACE.

```
log-level set MICROSERVICE_NAME LOG_LEVEL
```

```
>log-level set servicedispatcher DEBUG
INFO: Successfully updated log level for microservice servicedispatcher.
```

reset

Changes the current log level back to the default log level for a microservice.

```
log-level reset MICROSERVICE_NAME
```

```
>log-level reset servicedispatcher
INFO: Successfully reset log level for microservice servicedispatcher.
```

help

Prints the instructions for using the log level command line interface.


```
log-level help
```

Environment

The utility works on the master microservice framework node for a Teamcenter environment, irrespective of whether the host is running a Linux or Windows operating system.

The utility is located in the *TC_ROOT/microservices/bin* folder and must be run from this location.

req_word_html_converter

As an administrator, you can selectively convert requirement content in HTML format to Microsoft Word format (or the reverse, depending on the option selected). You can perform a selective conversion by providing a criteria file or a comma separated file containing fullText UIDs or directory containing multiple such Fulltext UID files. In non-permanent selective conversion from rich text to Microsoft Word, the converted requirements have the necessary named reference attachments as **MSWordXPart**; however, the content type remains rich text. Similarly, in non-permanent conversions from Microsoft Word to rich text, the converted requirements have the named references attachments as **Arm0HTML** and **Arm0HTMLIMG**, but the content type is still rich text.

Note:

Before you run this utility, ensure that the Active Workspace Requirements Management feature is installed. Several files are created in the transient volume folder when you run the utility:

- *failed_ID.txt* contains failed UIDs separated by commas
- *invalidIDs.txt* contains invalid UIDs
- *validIDs.txt* contains valid UIDs
- *dumpLogs.log* if you use the **dumpLogs** option
- The utility performs conversion for the following objects only: **SpecElementRevision**, **SpecElement**, or **SpecificationRevision**.

SYNTAX

```
req_word_html_converter [-u=user-id -p=password | password-file -g=group]
] [-path=full file path of HtmlConverter01.exe
] [-dumpLogs] [-forceUpdate] [-h][-htmlToWord] [-wordToHtml]
[-processObjectList=full path of the text file containing FullTextUIDs -in=full file path of criteria file
][-permanentConvert] [-dryRun]
```

ARGUMENTS

-u=user_id

Specifies the user ID to be used for the upgrade.

This is a user with Teamcenter administration privileges.

-p=password

Specifies the password for the specified user ID.

-g=group_name

Specifies the group associated with the user.

-path=*full file path of WordHtmlConverter.exe (including the .exe extension)*

Specifies the full path to the **WordHtmlConverter.exe** utility, including the filename **WordHtmlConverter.exe**.

-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.

-forceUpdate

(Optional) Forces the update and repair of all requirements in database even if they were previously converted.

-h

Displays help for this utility.

-htmlToWord

Converts requirements from HTML format to Microsoft Word.

-wordToHtml

Converts requirements from Microsoft Word format to HTML.

-processObjectList=*<full file path of the process object list text file>*

Specifies the full path of the text file containing FullText UIDs. The text files must contain FullText IDs separated only by commas; no spaces allowed.

-in=*<full file path of the criteria file>*

Performs a selective conversion of an entire requirement specification structure to html. The input is a criteria file that defines this structure.

The schema for the file is as follows:

KEYWORD_SEPARATOR=KWS

KEY_VALUE_SEPARATOR=KVS

RULE_SEPARATOR=RS

keys KWS key1=value1 KVS keyN=valueN RS **rev_rule** KWS revision-rule RS **topline_rev** KWS **topline** rev

keys KWS key1=value1 KVS keyN=valueN RS **rev_rule** KWS revision-rule RS

keys KWS key1=value1

Note the following conditions:

- The **KEYWORD_SEPARATOR**, **KEY_VALUE_SEPARATOR**, and **RULE_SEPARATOR** entries are required and you must enter in the order indicated. These entries define the separators that segregate the key-value pairs and keywords from corresponding values. You cannot use the equal sign (=) as the value for these entries. For example, the entry **RULE_SEPARATOR= =** is invalid.
- Separate all entries with the new-line character (**\n**).
- Starting in the fourth row, define the entries as follows:
 - *Segment 1 (Required)*

Consists of multi-field key-value (MFK) pairs. *Multifield keys* are IDs assigned to each object to ensure their uniqueness in the database. Based on the MFK, specifications/requirements are retrieved, and then further configuration is applied based on Segment 2 and Segment 3. This segment is required to uniquely identify the top object in the database.

Syntax

keys KWS key1=value1 KVS key2=value2 KVS keyN=valueN

Example

keys : item_id=REQ-000002 , object_type=Requirement

- *Segment 2 (Optional unless Segment 3 is defined)*

Denotes the revision rule to apply to the revision retrieved based on the MFK pairs in Segment 1.

If you do not define this segment, then the defined revision rule as read from preference **TC_config_rule_name** is applied.

Syntax

rev_rule KWS revision_rule_name

Example

rev_rule : Latest Working

- *Segment 3 (Optional)*

Defines the revision of the item retrieved on the basis of the MFK pairs defined in Segment 1. You can use this segment to determine the structure of interest. If you do not define this segment, then the latest revision is considered for configuring structure. If this segment is defined then you must define Segment 2 also.

For example, if the MFK pairs in Segment 1 return a particular item such as **item_id=REQ-000002**, then there can be different structures for different revisions of this item. Revision A as a top line might contain 6 children items; revision B might contain 12 children items.

Syntax

topline_rev *KWS topline_revision*

Example

topline_rev : A

- *Example: valid input criteria file*

KEYWORD_SEPARATOR=:

KEY_VALUE_SEPARATOR=,

RULE_SEPARATOR=|

keys: item_id=REQ-000001

keys: item_id=REQ-000001 , object_type=Requirement | rev_rule:Latest Working

keys: item_id=REQ-000001 | rev_rule:Precise Only

keys: item_id=REQ-000001 | rev_rule:Any Status; Working | topline_rev:A

- *Example: invalid input criteria file*

KEYWORD_SEPARATOR= = (invalid because the equals sign (=) is not a valid separator)

KEY_VALUE_SEPARTOR=, (invalid because **KEY_VALUE_SEPARATOR** is misspelled)

RULE_SEPARATOR| (invalid because the equals sign (=) is missing)

keys: item_id=REQ-000001 | rev_rule: | topline_rev:A (invalid because the revision rule name is not provided)

rev_rule:Latest Working | keys: item_id=REQ-000002 , object_type=Requirement (Invalid because Segment 1, Segment 2, and Segment 3 must be defined in that order: Segment 1 is the MFK, Segment 2 is the revision rule, and Segment 3 is the topline revision)

keys: item_id=REQ-000003 | topline_rev:A (Invalid because Segment2 is skipped and directly Segment3 is defined)

-permanentConvert

(Optional) Permanently converts requirements to one of the following formats:

- If you include the **-wordToHtml** switch, converts to HTML format, which is editable in the CK Editor in the **Documentation** tab.
- If you include the **-htmlToWord** switch, converts to Word format, which is not editable in the CK Editor in the **Documentation** tab.

-dryRun

These files are created: failed_ID.txt with failed UUIDs separated by comma, invalidUUIDs.txt with the invalid UUIDs, and validUUIDs.txt with the valid. The conversion does not take place.

The files are described below:

1. invalidUUIDs.txt: any UUID not present in the database or has a blank content_type property.
2. failed_ID.txt : any UUID having corrupted data, cannot be opened in Word, the user does not have write access on it, or is checked out by another user or a replica object.
3. validUUIDs.txt : All UUIDs are valid.
4. alreadyConvertedUUIDs.txt : UUIDs that are already HTML for wordToHtml conversion case or already Word for HtmlToWord conversion, is contained in this file.

EXAMPLES

req_word_html_converter -u=user_id -p=password -g=group -path=full file path of WordHtmlConverter.exe -forceUpdate -htmlToWord

req_word_html_converter -u=user_id -p=password -g=group -path=full file path of WordHtmlConverter.exe -dryrun -htmlToWord

req_word_html_converter -u=user_id -p=password -g=group -path=full file path of WordHtmlConverter.exe -dumpLogs -htmlToWord

req_word_html_converter -u=user_id -p=password -g=group -path=full file path of WordHtmlConverter.exe -in=full path of the criteria file for selective conversion -wordToHtml

req_word_html_converter -u=user_id -p=password -g=group -path=full file path of WordHtmlConverter.exe -processObjectList=full path of the process uidst text file -wordToHtml

req_word_html_converter -u=user_id -p=password -g=group -path=full file path of WordHtmlConverter.exe -dir=full path of the folder that contains text files with process uids -htmlToWord

req_word_html_converter -u=*user_id* -p=*password* -g=*group* -path=*full file path of WordHtmlConverter.exe* -permanentConvert -htmlToWord

Caution:

The following conditions are not supported:

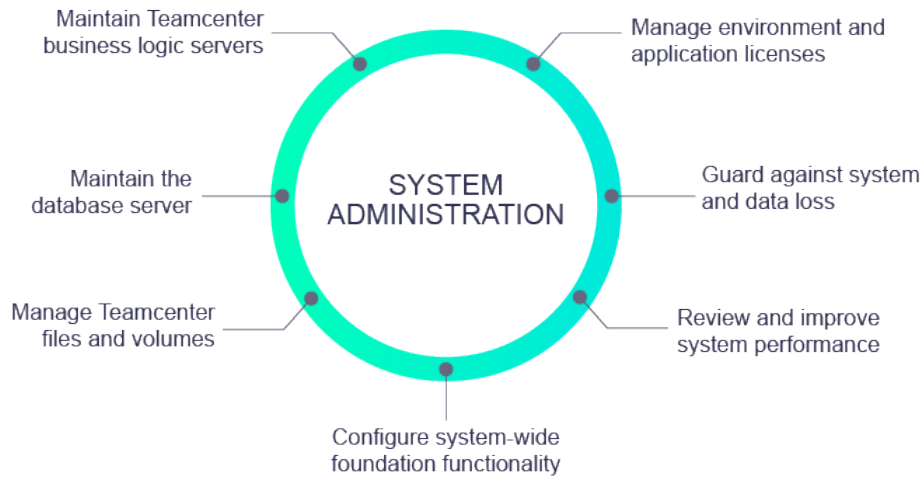
- Creating a requirement in Teamcenter Rich Client and modifying it in Active Workspace (and vice-versa).
- Creating a requirement in Teamcenter Rich Client and creating a sibling in Active Workspace and exporting to Word. The sibling cannot be edited.
- Creating a requirement in Teamcenter Rich Client and exporting it using microservices.

To create a specification that can be edited in Active Workspace, use **IMPORT SPECIFICATION** with the **Enable Editing** checkbox selected, or use the microservice-based import.

Part II: System Administration

17. Teamcenter system administration

Once Teamcenter is deployed, Teamcenter system administrators maintain the environment. Administrators can perform maintenance tasks using command line utilities, rich client perspectives, Active Workspace workspaces, or various dedicated administration consoles.



Where do I go from here?

Teamcenter Administrator	
Task	Documentation
Manage environment and application licenses	Licensing overview: seats, features, and bundles Using workspaces to control access to commands and pages in Active Workspace
Guard against system and data loss	Backing up a Teamcenter environment Cryptography and FIPS compliance in Teamcenter server Teamcenter Key Manager
Review and improve system performance	Techniques for improving four-tier performance What is Registry Editor? Teamcenter logging Active Workspace logging The Active Admin workspace
Configure system-wide foundation functionality	Introduction to Teamcenter client communication system (TCCS) Enabling external password management for Teamcenter
Manage Teamcenter files and volumes	Introduction to File Management System

 Teamcenter Administrator	
	Introduction to Volume Management
Maintain the database server	Oracle MS SQL Server
Maintain Teamcenter business logic servers	Scheduled maintenance for Teamcenter servers What is the server manager?

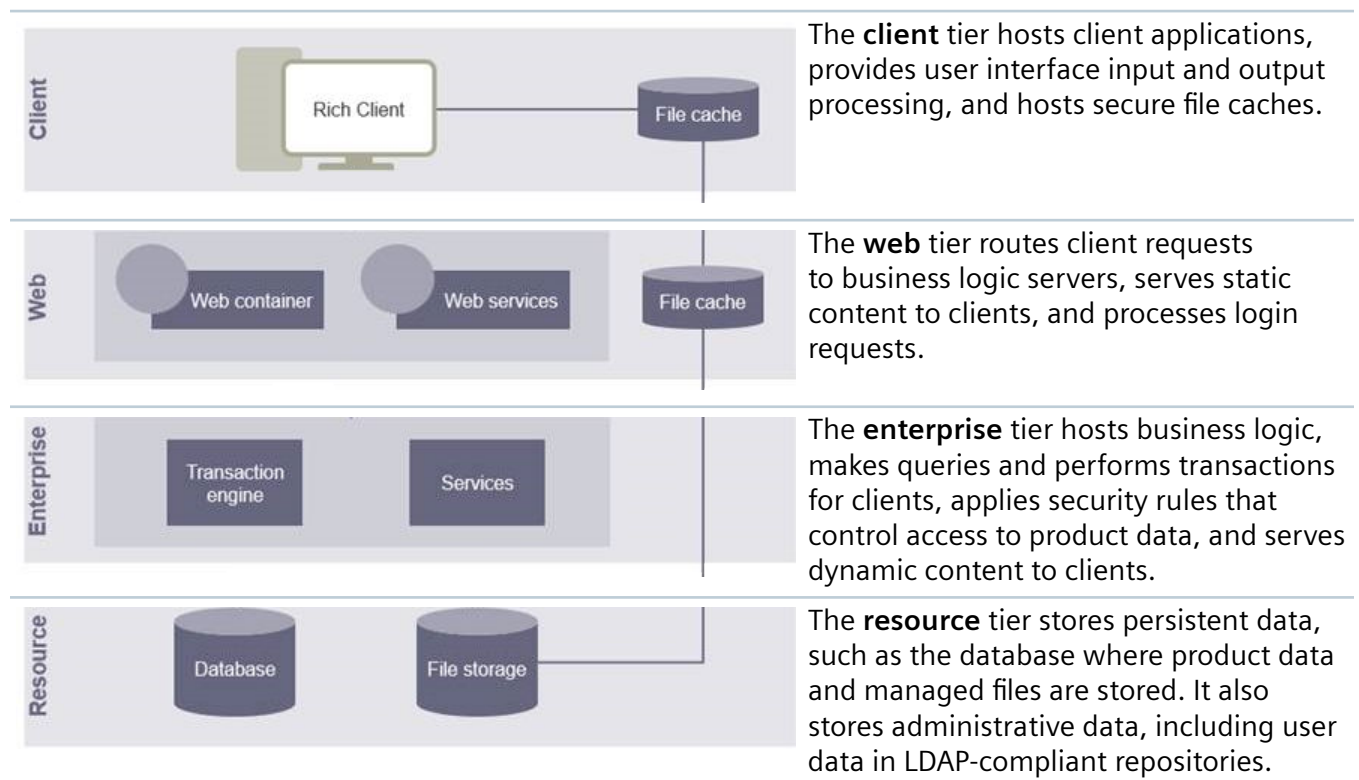
18. System logging and journaling

Teamcenter logging

Multiple Teamcenter components across the four-tier architecture can be involved in servicing an action request such as "expand the BOM". As a component processes the request, the component logs any related messages to its own log file. Administrators can review the logged messages to monitor and troubleshoot system performance.

An action could be (but is not limited to): an SOA service request; an FMS file upload/download; a microservice call; or a spawned child process.

The four-tier architecture comprises the following tiers:



Example transaction flow

A client sends a request that is received first by the web tier. A message recording the request is written to the **WebTier.log** file.

The same identifying information is recorded in the **ServerManager.log** file, where the server manager assigns the **tcserver1** process for this user.

The request path can be traced in the **tcserver1.syslog** file where messages have the same identifying information.

Log message format

To facilitate retrieval of messages from the logs of components involved in processing the request, most component messages follow a consistent format, and each message includes identifying information that is generated when the request is initiated: a **Session ID**, **User Action**, and **Correlation ID**.

Messages use the following format:

```
<Time Stamp>-<Log Level>-<Class Name>-<Session ID>-<User Action>-<Correlation ID>-<Message>
```

Format element	Description
Time Stamp	The coordinated universal time (UTC) when the message occurred.
Log Level	The message severity, one of FATAL, ERROR, WARN, INFO, DEBUG, or TRACE
Class Name	(Optional) A fully qualified class name identifying the message origin. For example <code>Teamcenter.CoreModelGeneral.Preferences</code> . The value may also contain the file path and line number.
Session ID	(Optional) A value identifying the client session or root application that initiated the request. This value applies to all initiated requests for the duration of the client session or root application.
User Action	(Optional) A meaningful label that identifies a group of requests initiated by a single user action or an application block of work.
Correlation ID	(Required) A unique value that identifies a particular request. This value appears in all related messages from all the components that work on processing that request.
Message	The actual message logged by the component.

Example:

A message populated with all format elements:

```
2022/09/23-22:06:54.574 UTC - INFO - Teamcenter.MetaModel - 5F0B0A62 - BOMExpand - 1388EFEER138 - Using existing MetadataCache
```

A message populated with only required format elements:

```
2022/09/23-22:06:54.574 UTC - INFO - - - - 945DE249R294 - Using existing MetadataCache
```

Default base log file location

Log files for business logic servers, the server manager, the web tier, and other components are written within a single base log file location on the host where the components are running. Expression of the default location depends on the host operating system:

Host OS	Default base log file location
Windows	%USERPROFILE%\Siemens\logs\
Linux	\$HOME/Siemens/logs/

Custom base log file location

To specify a custom base log file location, create the environment variable **SIEMENS_LOGGING_ROOT** and set it to your preferred location.

Managing FSC proxy log files

FSC proxy logs containing identical messages can quickly accumulate in the logs folder. You can limit or disable FSC proxy logging altogether by setting the **logging.logger.Teamcenter.FMS.FSCProxy** property to a value of **OFF** in the *TC_DATA\logger.properties* file:

```
logging.logger.Teamcenter.FMS.FSCProxy=OFF # Logger for the FSC Native
Proxy that communicates from the TcServer to the FSC
```

Logging for business logic servers

Business logic server (TcServer) logging levels

Administrators can use event logs to monitor system activity and troubleshoot session issues. Administrators can set startup logging levels for all business logic servers (TcServers) and can dynamically set session logging levels for a single TcServer.

Startup logging level - all TcServers

Administrators typically set startup logging to minimal levels for better performance. For all loggers used by TcServers, startup logging levels are specified in the **logger.properties** file within the *TC_DATA* directory. These levels apply to all business logic servers in server pools that use the *TC_DATA* directory.

Session logging level - selected server

Administrators can use the Teamcenter Management Console to **dynamically change logging levels** for a particular user session. Changing logging methods in this manner affects only the selected business logic server, and the changes persist only for the user session.

Administrators can set the following logging levels:

Logging level	Description
FATAL	Logs only severe error events that cause the application to fail. This is the least verbose logging level.
ERROR	Logs error events that may allow the application to continue running.

Logging level	Description
WARN	Logs potentially harmful situations, such as incomplete configuration, use of deprecated APIs, poor use of APIs, and other run-time situations that are undesirable or unexpected but do not prevent the application from functioning.
INFO	Logs informational messages highlighting the progress of the application at a coarse-grained level.
DEBUG	Logs fine-grained informational events that are useful for debugging an application.
TRACE	Logs detailed information, tracing any significant application step. This is the most verbose logging level.
DEFAULT	Inherit logging level from the parent logger.

Set TcServer startup logging levels

Use the following procedure to set startup logging levels for all business logic servers (TcServers) in all server pools that use the host server's `TC_DATA` directory.

1. Open the `TC_DATA/logger.properties` file.
2. Change the logging level of a logger:
 - a. Scroll to the logger whose logging level you want to change.
 - b. Type a valid logging level after the equal sign (=).
3. Save the file.

Administrators can click **Restart Warm Servers** in the **server manager administrative interface** to restart all warm servers and implement changes to the logging levels.

Swapping logger property files for debugging purposes

You can modify your standard `logger.properties` file to adjust logging levels for debugging, or you can specify an alternative settings file in either of two ways:

Use the provided `logger.debug.properties` file

The default settings of the provided `logger.debug.properties` debug file generate useful debugging messages.

Caution:

Using the `logger.debug.properties` file significantly increases the size of log files. Use it only for debugging purposes or when requested by Siemens Digital Industries Software support.

To use the `logger.debug.properties` file for setting logging levels, do one of the following:

- Set the **UGII_CHECKING_LEVEL** environment variable to **1**.
- Set the **TC_LOGGER_CONFIGURATION** environment variable to **TC_DATA/logger.debug.properties**.

Specify a custom logger properties file

To use a custom logger properties file for debugging, set the **TC_LOGGER_CONFIGURATION** environment variable to the whole file path of a custom logger properties file or to the path of the directory containing a custom **logger.properties** file.

Log files for Teamcenter tiers

Web tier logs

The location of web application server logs depends on the web application server:

Web application server	WebTierLogs location
JBoss, WildFly, and Tomcat	BaseLogLocation \Teamcenter\WebTiers\<Cluster>\<host name>\ For example, %USERPROFILE%\Siemens\logs\Teamcenter\WebTiers\<Cluster>\<host name>\
.NET	Analogous logs are written to Windows event logs in C:\Windows\System32\winevt\Logs\TcWebTier.evtx .

The log file *BaseLogLocation\Teamcenter\WebTiers\<Cluster>\<host name>\WebTier\process\WebTier.log* contains transaction messages sent to application process logs. Its configuration is defined in the file *BaseLogLocation\Teamcenter\WebTiers\<Cluster>\<host name>\WebTier\metadata\process\WebTier.log.xml*.

The following application process log files are located in *WebTierLogs\MLD\process*.

Log file	Log contents
plmconsole.txt	Console messages from the application implementing MLD.
RTEvents.txt	Sampled gauge values and threshold notifications. This local file is overridden by any JMX interface configuration.
RTConsole.txt	Default log file for response-time monitoring.
RTRemotes.txt	Notifications processed by remote Response Time GaugeListeners and TraceListeners .
RTTraces.txt	Local file for response-time-tracing messages and logging.

Enterprise tier logs

The following tables list types of enterprise tier logs and where they can be found.

Teamcenter business logic server logs
 Miscellaneous enterprise tier logs
 PLM XML logs
 Multi-Site logs

Teamcenter business logic server logs

Log file	Log contents	Log file location within BaseLogLocation \Teamcenter\ ServerManagers\<Cluster>\<PoolID>\
ServerManager.log	Records messages from the server manager application. ServerManager.log configuration: Increase the information logged to this file by increasing the severity level of the TC_ROOT\pool_manager\confs\configuration-name\log4j2.xml file.	ServerManager\process
tcserverpid.syslog	Diagnoses errors. Captures information, errors, and warnings. tcserverpid.syslog configuration: For information on logging levels, see Configuring business logic server logging.	TcServer\process
tcserverpid.jnl	Tracks objects accessed from the database and the activities performed on those objects. This session log also performs a trace through software modules. Each time you invoke or exit a module, the Log Manager posts an entry to this file.	

Miscellaneous enterprise tier logs

Log file	Log contents	Log file location
install.log	Tracks Teamcenter installation messages. The date-time stamp represents the date and time Teamcenter Environment Manager was run. For example, install1322241627.log indicates that Teamcenter Environment Manager was run at 4:27 on February 24, 2013. install.log configuration: Define the TC_TMP_DIR environment variable in the TC_DATA\tc_profilevars.bat file.	TC_TMP_DIR

Log file	Log contents	Log file location
pomutilities.log	Standard output from the POM utilities called by Teamcenter Environment Manager. pomutilities.log configuration: Define the TC_TMP_DIR environment variable in the TC_DATA\tc_profilevars.bat file.	
audit.log	Tracks selected properties for specified actions in the database. These audit logs are created in Audit Manager.	TC_LOG The default setting is TC_DATA\log_ORACLE-SERVER_ORACLE-SID where ORACLE-SERVER is the Oracle server network node and ORACLE-SID is the unique name of the Oracle database instance.
fscid_startup.log on Linux. fscidstdout.log and fscidstderr.log on Windows. Replace fscid with the FSC ID defined in FSC_HOME\fsc.xml .	Captures all FMS server cache (FSC) process output generated from stdout and stderr . This output is useful in diagnosing failure-to-start issues. The file also contains the entries generated to the runtime log.	Windows: FSC_HOME Linux: /tmp

PLM XML logs

Log file	Log contents	Log file location
xml-file-name.log or plmxml_log_timestamp.log	Provides complete information regarding the current PLM XML export or import.	TC_TMP_DIR For command line export, if TC_TMP_DIR is not set, the log file is generated at the same location as the XML file. For export from the rich client, the log file is generated at the same location as the XML file.
	xml-file-name.log or plmxml_log_timestamp.log configuration: Specify the logging level with the PLMXML_log_file_content preference.	

Multi-Site logs

Log file	Log contents	Log file location
idsm/D.log	Tracks actions performed on objects at the session level, such as imported or exported objects.	TC_TMP_DIR on the machine hosting the IDSM server.
idsm/D.syslog	Diagnoses errors. Captures information, errors, and warnings.	
idsm/D.jnl	Tracks objects accessed from the database and the activities performed on those objects. This session log also performs a trace through software modules. Each time you invoke or exit a module, the Log Manager posts an entry to this file.	
ods/D.log	Tracks actions performed on objects at the session level.	TC_TMP_DIR on the machine hosting the ODS server.
ods/D.syslog	Diagnoses errors. Captures information, errors, and warnings.	
ods/D.jnl	Tracks objects accessed from the database and the activities performed on those objects. This session log also performs a trace through software modules. Each time you invoke or exit a module, the Log Manager posts an entry to this file.	

Resource tier logs

The following tables list types of resource tier logs and where they can be found.

File Management System (FMS) Business Modeler IDE

File Management System (FMS)

Log file	Log contents	Log location
fscid_startup.log on Linux.	Log entries regarding server run-time operations.	BaseLogLocation\FSC\FSC\process\
fscidstdout.log and fscidstderr.log on Windows. Replace <i>fscid</i> with the FSC ID defined in <i>FSC_HOME\fsc.xml</i> .	fscidxxstxx.log configuration: Set the logging level in the <i>FSC_HOME\fsc.xml</i> file by supplying a value for the parameter \$FSC_HOME/FSC_\$fscid_\$USER.xml . Replace <i>fscid</i> with the FSC ID defined in the <i>fsc.xml</i> file.	

Log file	Log contents	Log location
	Valid values are FATAL , ERROR , WARN , INFO , and DEBUG . Siemens Digital Industries Software recommends that you never run an FSC in debug mode. Generally, WARN and INFO provide sufficient logging information.	
fscid.log Replace <i>fscid</i> with the FSC ID defined in <i>FSC_HOME\fsc.xml</i> .	Log entries regarding server run-time operations. <i>fscid.log</i> configuration: Change the location of this file by setting the LogVolumeLocation property in the <i>FSC_HOME\log.properties</i> file.	
FSC_host-name_user-ID_plmconsole.txt	Default console log file for response-time logging.	BaseLogLocation\FSC\MLD\process\
FSC_host-name_user-ID_RTEvents.txt	Sampled gauge values and threshold notifications. This file is overridden by any JMX interface configuration.	
FSC_host-name_user-ID_RTTraces.txt	Response-time-tracing messages and logging.	
FSC_host-name_user-ID_RTRemotes.txt	Notifications processed by remote Response Time GaugeListeners and TraceListeners .	

Business Modeler IDE

Log file	Log contents	Log location
deploy.log	Deployment messages.	output/deploy/serverProfileName/date-timestamp
Migration logs	Migration logging messages.	output/migration

Dispatcher logs

The Dispatcher sends request asynchronously to machines with the resource capacity to run the requests. A grid technology manages the job distribution, communication, translation, security, and error handling. Dispatcher requests are triggered by workflow, data checkin, batch mode, or on-demand operations.

The Dispatcher component has the following logs, none of which require configuration. The *LogVolumeDirectory* is specified in the *DISPATCHER_ROOT\AdminClient\conf\transclient.properties* file.

Log file and location within <i>LogVolumeDirectory/Dispatcher Server/</i>	Log contents
<i>task-ID/task-ID_m.log</i>	Dispatcher Module messages while processing the task.
<i>task-ID/task-ID_s.log</i>	Dispatcher Scheduler messages while processing the task.
<i>task-ID/task-ID_dc.log</i>	Dispatcher Client messages while processing the task. The Dispatcher Client receives Dispatcher Requests from the client and sends them to the Dispatcher Scheduler.
<i>process/Scheduler.log</i>	Scheduler messages.
<i>process/Module_ID.log</i>	Module messages.
<i>process/Adminclient.log</i>	Adminclient messages.
<i>process/History_DispatcherClient_IP-Address_port.log</i>	A history of events and state transitions performed on the requests.
<i>process/DispatcherClient_IP-Address_port.log</i>	Dispatcher Client process messages such as startup and cleanups .

Performance journaling

To analyze the performance of Teamcenter processes and programs, you can generate performance journal files (.pjl). The Pjl file format can also be generated by processes or programs other than Teamcenter. You can open Pjl files generated by different processes in a system simultaneously. For example, you can open Pjl files from a client and from its assigned TcServer, allowing you to view the performance of the entire session as a whole.

You can use Siemens Digital Industries Software's licensed application, *JournalWorkbench*, to read the Pjl files and analyze the data.

Finding error codes

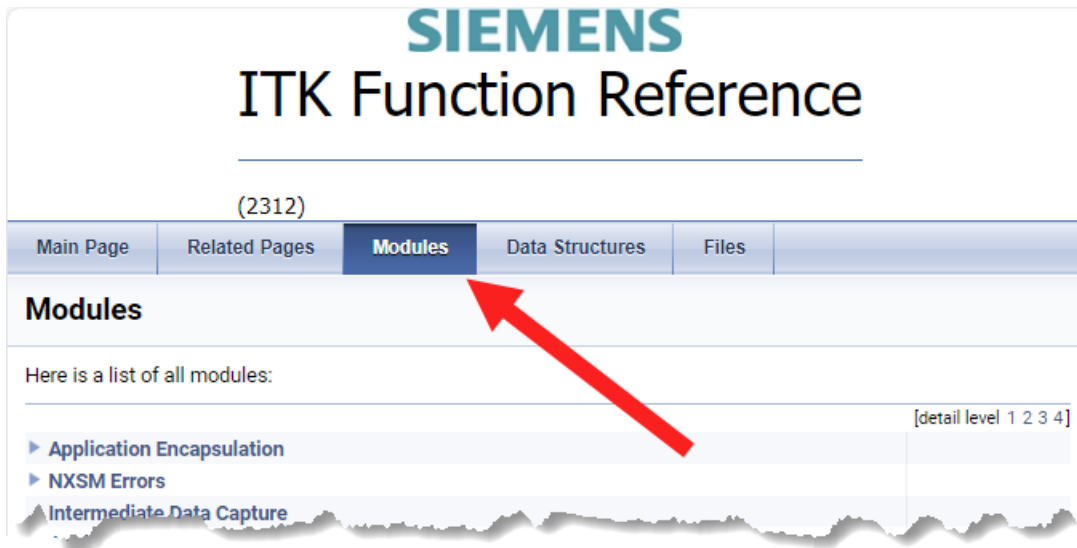
All error codes are documented in the *Integration Toolkit Function Reference*. Error codes are grouped by module. For example, Application Encapsulation (AE) errors are listed within the AE module, Appearances errors are listed within the Appearances module, most workflow errors are displayed in the Enterprise Process Modeling (EPM) module, and so forth.

To display a list of error messages:

1. Go to the Help Library and open the *Integration Toolkit Function Reference*.

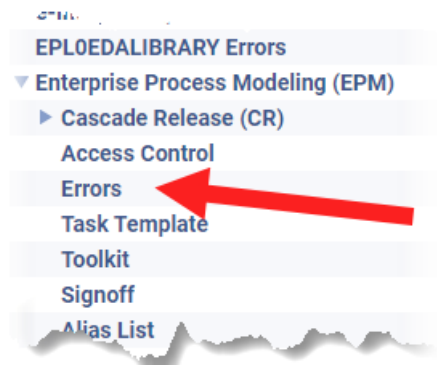
Version-specific editions of the *Integration Toolkit Function Reference* are available on Support Center.

- At the top of the page, select the **Modules** header.



- In the **Modules** page, scroll down to the appropriate module.

For example, to see all Enterprise Process Modeling (EPM) errors, which contain the majority of workflow errors, scroll to **Enterprise Process Modeling (EPM)**, expand the item, and click its **Errors** link.



The error page displays all errors for that module. Error numbers are defined as *module base value + error code*.

For example, **EPM_internal_error** has an error code of **EMH_EPM_error_base + 1**.

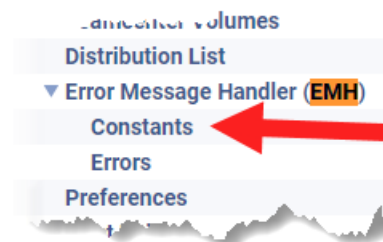
```

#define EPM_insufficient_data_to_generate_email (EMH_EPM_error_base + 370)
#define EPM_interactive_env_required (EMH_EPM_error_base + 22)
#define EPM_internal_error (EMH_EPM_error_base + 1)
#define EPM_interprocess_task_dependency_not_met (EMH_EPM_error_base + 126)
#define EPM_invalid_action_for_task_state (EMH_EPM_error_base + 324)
#define EPM_invalid_approver (EMH_EPM_error_base + 350)

```

4. To determine the value for a referenced error base:
 - a. Return to the **Modules** page.
 - b. At the upper right-hand site of the list, next to **detail level**, click **4** to fully expand the modules list.

With the list expanded, you can use the browser page search function to find an error code constant link.
 - c. Press **Ctrl+F** and search the page for the error code module. For example, **EMH**.
 - d. Under **Error Message Handler (EMH)**, click the **Constants** link.



The Error Message Handler (EMH) Constants page displays the error base of each module.

For example, the error base value of **EMH_EMP_error_base** is **33000**.

```

#define EMH_EPL_error_base 207000
#define EMH_EPM2_error_base 22000
#define EMH_EPM_error_base 33000
#define EMH_EPP0PRODUCTCONFIGURATOR_error_base 391000
#define EMH_EPW0WORKINSTRUCTIONS_error_base 414000

```

Thus, the error number for **EPM_internal_error** is the concatenation of the EPM modules error base (**33000**) and the error code (**1**), creating an error number of **33001**.

19. Configuring Teamcenter for performance

Configuring the four-tier architecture for performance

Techniques for improving four-tier performance

It can be useful to configure the Teamcenter four-tier architecture for optimum performance. The default settings for application servers are rarely appropriate for production scalability or good transactional performance.

Techniques for improving four-tier performance include:

- Disable the display of the **checked-out** symbol that displays to the right of each Teamcenter object. It is a quick indicator of whether the object is checked out. When an object is checked out, another user has exclusive access to it, preventing you from making changes to the object until it is checked in.

Set the **TC_show_checkedout_icon** preference to **True** to display the symbol and enhance usability or to **False** to hide the symbol and improve rich client startup performance.

- Determine whether your virus scanner monitors all HTTP communications from the host. Because Teamcenter four-tier clients (such as the rich client, and NX) use HTTP to communicate with the web tier, such scanning negatively impacts performance.

Deactivate this functionality, or configure your virus scanner to ignore communications from the web tier.

- Size the server pool appropriately. The pool-specific parameters are set in the **serverPool***database-name.properties* file in the *TC_ROOT\pool_manager\conf\configuration-name* directory.

Managing server memory

Introduction to managing server memory

Objects loaded on the Teamcenter server remain in the server memory throughout the session. If unused objects are not removed, long-running sessions generate significant memory usage which can cause memory errors.

To avoid these errors, Teamcenter provides automatic cleanup of unused objects. The default configuration of the memory cleanup operation unloads unused objects from the server in the order they were last accessed. The least recently accessed objects are unloaded first. The operation begins when the memory server size reaches 12,000 KB, and stops when the memory server size reaches 5,000 KB.

This default configuration of memory cleanup behavior is sufficient for most sites. The default settings are not optimal when:

- Users work with very large assemblies.
- All sessions are short-running sessions with minimal server memory needs; therefore, the site has no need of automatic memory cleanup.

Default automatic memory cleanup settings

Automatic memory cleanup unloads unused objects from the server in the order they were last accessed. The least recently accessed objects are unloaded first. The default settings initiates memory cleanup when the memory server size reaches 12,000 KB, and stops when the memory server size reaches 5,000 KB.

You can fine-tune automatic cleanup functionality by modifying the default values of the following preferences.

Preference	Default value	Description
UNLOAD_TRIGGER_CEILING	12000	Initiates the memory cleanup operation. Specifies (in kilobytes) what memory size must be reached to initiate the unloading of unused objects from the server memory. This value must be a positive integer higher than the value set for the UNLOAD_TRIGGER_FLOOR preference. If an invalid value is set, the default value of 12000 is used. Automatic cleanup is performed during the operation occurring <i>after</i> the specified ceiling is reached, possibly causing the specified memory size to be exceeded if the final loading action uploads a large amount of data. Setting this value very low reduces the memory server footprint, but decreases performance.
UNLOAD_TRIGGER_FLOOR	5000	Stops the memory cleanup operation. Specifies (in kilobytes) what memory size must be reached to stop the unloading of unused objects from the server memory after memory

Preference	Default value	Description
		cleanup begins. This value must be a positive integer lower than the value set for the UNLOAD_TRIGGER_CEILING preference. If an invalid value is set, the default value of 5000 is used. Automatic cleanup continues until the memory size specified by this preference is achieved. However, if the total size of unmovable object exceeds the floor amount, the specified memory size is not reached. For example, if objects cannot be removed because their life span does not meet the requirements specified by the UNLOAD_MINIMUM_LIFE preference, and the memory space consumed by these objects exceed the specified floor size, the specified memory size is not reached. Determining the optimum value for this preference requires investigation, and possibly experimentation in a working environment.
UNLOAD_MINIMUM_LIFE	1800	Determines which unused objects are qualified for memory cleanup, based on when objects were last accessed. Specifies (in seconds) when an object was last accessed. Objects accessed more recently than the specified value are not removed from server memory.
UNLOAD_LOGGING_LEVEL	0	Determines the level of detail logged in the in-session and end-session reports sent to the .unloadlog log file. 0 disables logging. Values 1 through 4 enable logging in progressively more detail. Setting this preference to 1 through 4 does not produce a log file unless the FCC is restarted on the client, for example:

Preference	Default value	Description
		<i>FMS_HOME\bin\fccstat -restart</i>

Determining optimum performance settings for a two-tier environment

Individual users running a two-tier environment can set the **UNLOAD_TRIGGER_CEILING** and **UNLOAD_TRIGGER_FLOOR** user preferences to best fit their individual needs.

- Users can use the default settings if they do not work with large amounts of data.
- Users can increase the value of the **UNLOAD_TRIGGER_CEILING** user preference if they work with large amounts of data, such as very large assemblies. Increasing the value delays the unloading of objects, improving performance.

Users can determine the necessary settings by estimating an average object size of .5 KB. For example, with the **UNLOAD_TRIGGER_CEILING** default setting of 12000 KB, automatic unload occurs when approximately 24,000 objects are loaded.

The 5,000 KB default setting for the **UNLOAD_TRIGGER_FLOOR** preference means unloading continues until approximately 10,000 objects remain in the unloadable memory area.

These settings are adequate for most two-tier environments, including the default Teamcenter installation running only the **Foundation** template.

Determining optimum performance settings for a four-tier environment

In a four-tier environment, site administrators must determine the needs of a typical user at the site and configure the **UNLOAD_TRIGGER_CEILING** and **UNLOAD_TRIGGER_FLOOR** preferences accordingly.

To determine the optimum setting of the **UNLOAD_TRIGGER_CEILING** preference at your site:

- Determine the size of the free memory available before any Teamcenter server is started.
- Determine the number of users at the site accessing the server concurrently.
- Determine the size of the Teamcenter server instance (including all site customizations and configurations) immediately after user login.
- Divide the amount of free memory available by the number of concurrent users to determine the average size of memory required for each user per session.

For example, if the average size of the free memory required per user session is 100 MB, and if users typically work with assemblies containing approximately 100,000 objects, the site administrator must increase the value of the **UNLOAD_TRIGGER_CEILING** preference to a minimum value of 50,000 KB, based on an estimated average size for an object being 0.5 KB. Because the total size of free memory in this case is 100 MB, a value between 50,000 KB and 100,000 KB is acceptable, but it should not exceed 100,000 KB to give the optimal result for all users.

Disabling automatic memory cleanup

Automatic memory cleanup cannot be disabled with a preference or with a button in the interface. However, you can stop automatic memory cleanup from functioning at your site by either:

- Setting the **UNLOAD_TRIGGER_CEILING** preference very high. Setting the ceiling value near the size of total memory essentially prevents the cleanup operations from initiating.

Siemens Digital Industries Software recommends using this method to disable automatic memory cleanup.

- Setting the **UNLOAD_MINIMUM_LIFE** preference very high. Setting the length of time in which an object must not have been accessed very high (years) prevents any objects from qualifying for memory cleanup.
- Adding the **ObjectUnloadable** business object constant to the **POM_object** and setting the constant to **false**. This setting makes all **POM_object** children invalid for unloading; therefore, no objects are ever selected for automatic memory cleanup.

Server memory cleanup log files

While the Teamcenter server is running, object unloading activity is logged in a log file. At the end of the session, a summary of the unloading operation is also logged. The data is stored in the **.unloadlog** file, typically stored in the **TC_TMP_DIR** directory. The file contains information such as the peak/least memory used by unloadable objects, the total number of unloaded objects, the total number of unload operations, the average time for an unload operation, and so on.

Determine the level of detail reported to the log file by setting the **UNLOAD_LOGGING_LEVEL** preference to a value between **0** and **4**. Setting this preference to **0** disables logging.

Following is the data logged in the files for each logging level.

Level 0	
In-session report	Creates no log file. No log information is reported.
End-session report	N/A

Level 1

In-session report	Specifies the number of objects in the cache. Specifies the number of objects removed from the cache. Specifies the time (in seconds) to unload the cache. Specifies the number of objects remaining in the cache.
End-session report	Specifies the peak memory used by unloadable objects for the entire session. Specifies the least amount of memory used by unloadable objects for the entire session. Specifies the total number of objects unloaded. Specifies the total number of unloading operation requests. Specifies the total number of actual unloading operation. Specifies the average time of an unloading operation.

Level 2

In-session report	Specifies the number of objects in the cache. Specifies the number of objects removed from the cache. Specifies the time (in seconds) to unload the cache. Specifies total memory usage of all areas (in bytes) before unloading. Specifies total memory usage of all areas (in bytes) after unloading. Specifies the number of objects remaining in the cache.
End-session report	Specifies the peak memory used by unloadable objects for the entire session. Specifies the least amount of memory used by unloadable objects for the entire session. Specifies the total number of objects unloaded. Specifies the total number of unloading operation requests. Specifies the total number of actual unloading operations. Specifies the average time of an unloading operation.

Level 3

In-session report

Specifies the number of objects in the cache.
 Specifies the number of objects removed from cache.
 Specifies the time (in seconds) to unload the cache.
 Specifies total memory usage of all areas (in bytes) before unloading and before callbacks. Usage shown by area.
 Specifies total memory usage of all areas (in bytes) after unloading and before callbacks. Usage shown by area.
 Specifies the number of objects remaining in the cache.

Setting the **UGII_CHECKING_LEVEL** environment variable to **1** provides the following callback and memory usage information:

Callbacks. (Lists callbacks triggered. Specifies the time (in seconds) callbacks took.)
 Specifies memory usage of all areas (in bytes) before callbacks after unloading. Usage shown by area.
 Specifies memory usage of all areas (in bytes) after callbacks after unloading. Usage shown by area.
 Specifies the number of objects remaining in the cache.
 Specifies the time (in seconds) to complete the call.

End-session report

Specifies the peak memory used by unloadable objects for the entire session.
 Specifies the least amount of memory used by unloadable objects for the entire session.
 Specifies the total number of objects unloaded.
 Specifies the total number of unloading operation requests.
 Specifies the total number of actual unloading operations.
 Specifies the average time of an unloading operation.
 Provides a summary of unloading per type, including type name and unload count.

Level 4

In-session report

Specifies, if adding to the cache:
 Object String =object string , type: type of object, time: when it was accessed
 Specifies, if removing from the cache:
 Unloading Object: Object String =object string ,

Level 4

type: type of object, time: when it was accessed
 Specifies the number of objects in the cache.
 Specifies the number of objects removed from cache.
 Specifies the time (in seconds) to unload the cache.
 Specifies total memory usage of all areas (in bytes) before unloading and before callbacks. Usage shown by area.
 Specifies total memory usage of all areas (in bytes) after unloading and before callbacks. Usage shown by area.
 Specifies the number of objects remaining in the cache.

Setting the **UGII_CHECKING_LEVEL** environment variable to **1** provides the following callback and memory usage information:

Callbacks. (Lists callbacks triggered. Specifies the time (in seconds) callbacks took.)
 Specifies memory usage of all areas (in bytes) before callbacks after unloading. Usage shown by area.
 Specifies memory usage of all areas (in bytes) after callbacks after unloading. Usage shown by area.
 Specifies the number of objects remaining in the cache.
 Specifies the time (in seconds) to complete the call.

End-session report

Specifies the peak memory used by unloadable objects for the entire session.
 Specifies the least amount of memory used by unloadable objects for the entire session.
 Specifies the total number of objects unloaded.
 Specifies the total number of unloading operation requests.
 Specifies the total number of actual unloading operations.
 Specifies the average time of an unloading operation.
 Provides a summary of unloading per type, including type name and unload count.

Shared memory

Shared memory overview

Shared memory is a mechanism for sharing commonly used data across multiple Teamcenter processes. The shared memory implementation is applied to all the collocated Teamcenter processes that are run on the same machine, and the shared memory appears in the address space of all Teamcenter servers, utility processes, and service (daemon) processes on the machine.

When large amounts of data need to be shared, shared memory is the fastest interprocess communication mechanism between co-located processes. After the memory is mapped into the address space of the processes that are sharing the memory region, the processes can access the shared memory area like regular working memory with no overhead incurred. Because only one copy of the data in the shared memory is resident for all the processes that are sharing the memory region, using shared memory for common data reduces the overall system memory footprint.

Shared memory provides the following benefits:

- Reduces the memory footprint for each server process and utility process.
- Large amounts of data are loaded into shared memory. This data includes text server data, site preference data, localized list of values data, and metadata. As a result, the private memory footprint for a server process is reduced. For example, in a Teamcenter 9.1.2 environment, the savings in the private memory footprint for a server process are as follows:

Text server: 6MB
 Preferences: 2–4 MB
 Lists of values: 5 MB
 Metadata: 19MB

- Improves Teamcenter scalability

Because the private memory footprint is reduced per server, more Teamcenter servers can be configured on the same machine than previously.

- Improves Teamcenter performance

All the shared data is populated once in shared memory and saved in local backing store files. After backing store files are generated, all the subsequent servers do not need to populate data into shared memory. Therefore, the logon time for all the subsequent servers is improved.

In the case of the shared metadata, the metadata cache is pregenerated (for example, during install and upgrade). This helps to reduce further the metadata population time. As a result, there is approximately 18% improvement in logon time.

Shared metadata memory helps to improve overall performance for other use cases as well. Each server is able to access the already-populated shared metadata cache that would otherwise have to be on-demand initialized by each server if shared metadata cache were not used.

Configure shared memory

Set the **TC_SHARED_MEMORY_DIR** environment variable to specify the root directory where the backing store files are stored. Set this to a valid directory and do not change it. If you do not set this environment variable on Windows systems, the directory specified by the **TEMP** environment variable is used, and on Linux systems, the **/tmp** directory is used. On Windows systems, if the **TEMP** environment variable is not specified, the **C:\temp** directory is used.

The shared memory root directory name is also used for making a unique **semaphore** name among all the collocated Teamcenter servers. When the **TC_SHARED_MEMORY_DIR** environment variable is set to a different root directory, a new semaphore with a new name is created per shared memory module.

Define a fixed directory for shared memory map files

Each time a process stores a shared memory map file in a new location, a new semaphore is created. Defining a fixed directory for shared memory map files minimizes the semaphore count. This is particularly useful on Windows if the value of your **TEMP** environment variable is frequently modified, and on Linux systems because the operating system does not relinquish semaphores when processes are complete.

1. Set the **TC_SHARED_MEMORY_DIR** environment variable to a valid, fixed directory.

The shared memory map files are created in a subdirectory of the directory defined by this environment variable.

2. Stop the text server process (**TcServer**).
3. Log on as root.
4. Display the list of semaphores owned by Teamcenter by typing:

```
ipcs -sb |grep user-id-used-to-start-pool-manager
```

5. Free all the semaphores in the list by typing:

```
ipcrm -s ID_1... -s ID_n
```

ID_1 is the semaphore identifying number from the list.

The first process started after this change creates the shared memory map files and required semaphores. All following processes use this existing information, with no need to re-create the map files or additional semaphores.

Clean up semaphores

Shared memory functionality consumes named semaphores. On Linux platforms, the count of the used semaphores may reach the operating system limit. In such a scenario, the following message is logged in the Teamcenter server **syslog** files and the Teamcenter server reverts to its in-process memory mode.

```
ACE_Malloc_T<ACE_MEM_POOL_2, ACE_LOCK, ACE_CB>::ACE_Malloc_T: Not enough space
```

If this problem occurs, you must clean up semaphores in the following steps:

1. Ensure the system is idle.

No Teamcenter process (for example, Teamcenter server, utility process, and service process) can be running.

2. Log on as root.
3. Display the list of semaphores owned by Teamcenter by typing the following command:

```
ipcs -sb | grep user-id-used-to-start-pool-manager
```

4. Free all the semaphores in the list by typing the following command:

```
ipcrm -s ID_1... -s ID_n
```

ID_1 is the semaphore identifying number from the list.

Shared memory for metadata

Introduction to shared memory for metadata

You can configure Teamcenter to use a shared memory cache, which reduces the memory footprint by eliminating metadata duplication among Teamcenter servers. The following types of metadata are stored in the shared memory cache:

- Types
- Property descriptors
- Constants

In a four-tier environment, multiple Teamcenter servers access the shared memory cache (reducing the overall footprint of the deployment). A metadata cache file is exported and mapped to the shared memory cache during each **TcServer** startup. By default, all Teamcenter servers access the shared memory cache.

To enable shared memory for metadata, set the **TC_USE_METADATA_SHARED_MEMORY** environment variable to **TRUE**. The metadata cache is stored within the **Metadata** subdirectory created within the directory specified by the **TC_SHARED_MEMORY_DIR** environment variable. The metadata cache files are stored as sequentially numbered files.

Note:

If a Teamcenter server cannot access the **Metadata** directory, it accesses its local metadata cache. For example, if the system runs out of semaphores, each Teamcenter server accesses its own local metadata cache.

Configuring the shared memory cache

You can enable shared memory cache for metadata from either Business Modeler IDE while performing live updates or from Teamcenter Environment Manager (TEM) during installation or upgrade.

- From Business Modeler IDE, when deploying a template, select the **Generate Server Cache** check box.
- From TEM, select the **Generate Server Cache** check box in the **Foundation Settings** panel.

Synchronizing the shared memory cache

When you create new types, property descriptors or constants in the Business Modeler IDE, the changes made to the metadata in running Teamcenter servers must be synchronized with the database.

The synchronization is accomplished by the **shared_server_metadata_cache_mgr** executable. This executable runs as a service on Windows and as a daemon on Linux. By default, the service/daemon starts automatically. To install the daemon, in the **Features** panel in Teamcenter Environment Manager (TEM), choose **Server Enhancements**→**Database Daemons**→**Teamcenter Shared Metadata Cache Service**.

Two preferences manage the behavior of this executable:

- You can use the **TC_shared_server_metadata_cache_mgr_cloning_interval** preference to stop and restart the service to avoid any possible memory leaks. Typically, this is not necessary. Accept the default setting (100 minutes).
- Use the **TC_shared_server_metadata_cache_mgr_sleep_minutes** preference to specify how frequently the service/daemon compares the latest metadata cache file against the database, exporting the latest metadata cache file in the **Metadata** directory if changes are found, and then mapping the file to the shared memory cache.

To force the regeneration of the cache, even though no metadata changes exist in the database, run the **generate_metadata_cache** utility using the **-force** argument.

Shared memory for Text Server data

Introduction to shared memory for Text Server data

Shared memory functionality improves memory performance; memory consumption is reduced and performance is increased. However, this implementation requires some administration that in-process memory functionality does not:

- Users must manage memory mapped files when new text key-value pairs are added to the Text Server XML files. When this occurs, all Text Server related memory backing store files must be removed.
- Administrators running on a Linux platform must manage semaphore usage.

The memory backing store files represent the persistent cache for the shared memory contents. As long as the physical backing store file exists, the file contents are used for the Text Server shared memory data on that machine. The initial XML text files are only read and parsed by the very first Teamcenter server process, to populate the shared memory cache. Subsequent processes use the shared memory technique. Updating the XML files through installation or customization requires a refresh of the shared memory cache, which is done by **removing the backing store files** (its persistent representation).

Shared memory functionality consumes semaphores. Each time a process stores a shared memory map file in a new location, a new semaphore is created. When a process reuses a location for a shared memory map file, the same semaphores are reused.

Remove memory backing store files

If you are using shared memory and have updated your text XML files, you must refresh the shared memory cache by removing the memory backing store files.

1. Ensure the system is idle. No Teamcenter server processes can be running.
2. Remove the memory backing store files. The **TC_SHARED_MEMORY_DIR** environment variable value specifies the directory where the backing store files are stored. If you do not set this environment variable, the **TEMP** environment variable (Windows) or **/tmp** (Linux) directory is used.

Note:

On Windows, if the **TEMP** environment variable is not available, the **C:\temp** directory is used.

The shared memory files are created under the *version/last-build-date/database-site-ID/TextSrv/language* directory located in the temporary repository. The many values of the *language* directory depend on the language used by the Teamcenter server processes, which are the ones used by the connection with Teamcenter. Server error messages and strings are saved respectively in the **emh_text.xml.mem** and **tc_text.xml.mem** files.

At the next process startup, the system finds the shared memory state is no longer initialized, reads the text XML files, and populates the shared memory cache, thus creating and populating the shared memory backing store file.

Note:

You can disable shared memory functionality and revert system behavior to in-process text storage by setting the **TC_NO_TEXTSRV_SHARED_MEMORY** environment variable to **TRUE**.

Shared memory for localized LOV data

The **TC_USE_LOV_SHARED_MEMORY** environment variable determines whether the Teamcenter server loads localized LOV display names into shared memory. Set this environment variable to either **TRUE** or **FALSE**. The default setting is **TRUE**.

If shared memory cannot be used, the system uses process memory. In rare instances when the system reports a problem with shared memory and cannot fall back to using process memory, set this environment variable to **FALSE** and restart the system.

Note:

Because localized LOV display can slow system performance, you can use the **FndOLOVDisplayAsEnabled** global constant to disable loading localized LOV display names.

Sharing a TcServer instance with multiple applications

A **TcServer** instance can sometimes be shared by multiple applications on the same machine. If clients log on with the same user from applications that use sharing, the server is shared, and the clients stay synchronized.

While the **TcServer** architecture does not support threading service requests from different users, it can support multiple sessions (client applications) for a single user. In the rich client, this sharing is enabled by default. In a four-tier environment, whether the rich client connects to the same **TcServer** as other clients on the same machine can be controlled by the **shareSession** property in the **site_specific.properties** file. This file is stored in the following location:

Teamcenter-install-location/portal/plugins/configuration_version

- Setting the **shareSession** property to **true** enables sharing.

All rich client sessions on the client desktop to which the user logs on with the same user name share the same instance of the Teamcenter server.

Other client applications that are also configured for session sharing, and logged on with the same user name, also share the same instance of the Teamcenter server. Any changes in session state (for example, a change of group or role) made for a client are reflected in the other clients.

- Setting the **shareSession** property to **false** disables sharing.

Each rich client session on the client desktop has its own dedicated instance of the Teamcenter server.

Tuning the web tier

Tuning application servers

The Teamcenter web tier manages communication between the Teamcenter clients and the enterprise tier. Tune the web tier to ensure performance. The successful deployment of the Teamcenter web application or Enterprise application is not complete without tuning the application server. The default settings for application servers (for example, WebSphere) are rarely appropriate for production scalability or good transactional performance.

Ensure that the applicable application server vendor tuning documentation is followed. Of critical importance are items such as the following:

- Set any JSP or servlet check intervals to very long values.

This prevents unwanted and expensive extraction of these components from the deployed WAR files. In some cases, application servers may be configured by default to check these upon every invocation. That is good for a developer environment but very bad for performance.

A good value is on the order of once a day or longer.

- Tune the application Java Virtual Machine (JVM) run-time parameters appropriate for high transactional throughput (server) applications:
 - Set JVM heap sizes to match the RAM available on the machine, while keeping the size to a limit manageable by the garbage collection (GC) algorithm and the power of the CPU(s) available to implement that algorithm. In this case, both too small and too large are potential performance problems.
 - Select the appropriate garbage collection algorithm to match the number of CPUs available to implement it.
 - Set the JVM heap generational sizes to be appropriate for the GC algorithms and available heap.

In all cases, the relevant performance tuning documentation provided by the application vendor and Java Runtime Environment (JRE) should be consulted for optimum tuning values. Each distinct application server has distinct configuration mechanisms.

Each major version of the JRE has unique Garbage Collection tuning options available.

You can [use third-party applications to view web tier administration interface data](#) in a more comprehensive manner.

Web tier monitoring

Web tier monitoring provides notification when the following metrics reach the specified level, indicating a critical event. Use the [Teamcenter Management Console](#) to manage the Server Manager and the web tier.

The following metrics are defined in the **webtierMonitorConfig.xml** file:

Metric	Description
Abandoned Servers	Servers are assigned, but attempts to connect fail.
No Server Available	The web tier is unable to assign a server to a user because none are available.
Server Comm Failure	A Teamcenter server closed the connection while the web tier was waiting for a response.
Failed Login Attempts	A user provided invalid credentials.
Number Pending Requests	The number of client requests for which the web tier is awaiting a response.
Long-running Requests Current	The web tier waited longer than the specified time for a server response
Long-running Request Completed	The number of requests taking longer than the specified amount of time to complete.
Grave Events	Fatal or unexpected errors occurring in the web tier.

Configure web tier monitoring using the `webtierMonitorConfig.xml` file, located in the deployed `.war` file in the `lib\mldcfg.jar` file.

You can revise the **`webtierMonitorConfig.xml`** file and save it back into the `.war` file before deployment by the web application. Or if you prefer, you can place the revised file directly into the root directory of the web application server's run-time environment, and thus override the version of the file in the `.war` file. For example, on WebSphere place it into the **profile** directory, or on JBoss place it into the **bin** directory.

Tip:

- You should review all monitoring settings and ensure that the thresholds are set correctly for your site.

If you do not know the optimum monitoring setting for any given critical event, then set the value to **COLLECT**. Collect the data and review it to determine normal activity levels. Then set notification values slightly higher than normal activity levels.

- The contents of the email notifications are generated from the **`webtierMonitorConfigInfo.xml`** file. (This is a companion file to the **`webtierMonitorConfig.xml`** file.) See this file for a complete list of possible causes and recommended actions for web tier monitoring.

The **`webtierMonitorConfigInfo.xml`** file is stored in the WAR file by default in the **`lib\mldcfg.jar`** file. A copy can be placed on the application server's current directory to override the default version.

You can configure web tier metrics and logging behavior to better administer web tier processes. Use the **`pref_export.xml`** file to enable web tier metrics. This file is stored in the WAR file by default in the **`lib\mldcfg.jar`** file. A copy can be placed on the application server's current directory to override the default version.

- Use the **`ExtendedEnabled`** element to enable extended nested metrics upon reset.
- Use the **`ResponseTimeSampler`** element to enable sampling and logging of response times.

Configure monitoring with the `webtierMonitorConfig.xml` file

1. Open the **`webtierMonitorConfig.xml`** file stored in the WAR file, application sever startup directory, or on the application server **`classpath`** (allowing settings to be tailored to a machine if required). By default, the **`webtierMonitorConfig.xml`** file is stored in the `.war` file in the **`lib\mldcfg.jar`** file.

2. On the **`ApplicationConfig`** tag, set the **`mode`** to one of the following:

- **Normal**

Enables monitoring of all the metrics listed in the file.

- **Disable_Alerts**

Enables monitoring of all the metrics listed in the file, but disables all notifications of critical events, regardless of individual notification settings on any metric.

- **Off**

Disables monitoring of all the metrics listed in the file.

3. (Optional) To be notified when criteria reaches the specified threshold, specify from whom, to whom, and how frequently email notification of critical events are sent by setting the following **`EmailResponder`** values.

You can specify more than one **`EmailResponder id`**.

All **`EmailResponder id`** values in all subsequent monitoring metrics in this file must match one of the **`EmailResponder id`** values set here.

- **EmailResponder id**

Specify an identification for this email responder. Multiple email responders can be configured, in which case the identifiers must be unique.

- **protocol**

Specify the email protocol by which notifications are sent. SMTP is the only supported protocol.

- **hostAddress**

Specify the server host from which the email notifications are sent. In a large deployment (with multiple server managers, or the web tier running on different hosts) the host address identifies the location of the critical events.

- **fromAddress**

Specify the address from which the notification emails are sent.

- **toAddress**

Specify the address to which the notification emails are sent. You can specify multiple email addresses, separated by semi-colons.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress email notification of critical events.

- **emailFormat**

Specify the format in which the email is delivered. Valid values are **html** and **text**.

4. (Optional) To be notified when criteria reaches the specified threshold, specify to which file critical events are logged by setting the following **LoggerResponder** values.

All **LoggerResponder** values in all subsequent monitoring metrics in this file must match the **LoggerResponder id** value set here.

- **LoggerResponder id**

Specify an identification for this logger responder. Multiple logger responders can be configured, in which case the identifiers must be unique.

- **logFileName**

Specify the name of the file to which critical events are logged.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress logging of critical events to the log file.

5. Configure the criteria for a critical event for any of the metrics in the file by:

- a. Specifying a particular **EmailResponder**.
- b. Specifying a particular **LoggerResponder**.
- c. Setting the metric monitoring mode to one of the following:

- **Collect**

Collect metric data and display results in the **server manager administrative interfaces** for this metric.

This is the default setting.

- **Alert**

When critical events exceed the specified threshold, collect metric data, send email notifications (set up with **EmailResponder** values), write messages to log files (set up with **LoggerResponder** values), and display results in the server manager administrative interfaces for this metric. Log file messages and email notifications are only issued when monitoring mode is set to **Alert**.

- **Off**

No metric data is collected.

- d. Setting the remaining values to specify criteria that must be met to trigger a critical event for the metric.

6. Save the file.

web tier monitoring is enabled for the metrics you configured.

Note:

You can revise the **webtierMonitorConfig.xml** file and save it in the `.war` file before deployment by the web application. Or if you prefer, you can place the revised file directly into the root directory of the web application server's run-time environment, and override the version of the file in the `.war` file. For example, on WebSphere, place it in the **profile** directory; or on JBoss, place it into the **bin** directory.

Sample webtierMonitorConfig.xml code

In the following example, two of the four monitoring metrics are configured for email notification of critical events. And two **EmailResponder** elements are configured. Email notification is sent to **EmailResponder1** when the **No Server Available** metric achieves critical event status. Email notification is sent to **EmailResponder2** when the **Grave Events** metric achieves critical event status. Data is collected for the other metrics, but no email notifications are sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
@<COPYRIGHT>@
=====
Copyright 2011.
Siemens Product Lifecycle Management Software Inc.
All Rights Reserved.
=====
@<COPYRIGHT>@
-->
<!-- Webtier Health Monitoring Configuration -->
<ApplicationConfig id="Webtier" mode="Off" version="1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation=
    "healthMonitorV1.0.xsd">
  <RespondersConfig>
    <EmailResponder id="EmailResponder1">
      <protocol value="smtp"/>
      <hostAddress value="svc1smtp.company.com"/>
      <fromAddress value="tcsys@company.com" />
      <toAddress value="admin1@company.com" />
      <suppressionPeriod value="4200"/>
      <emailFormat value="html"/>
    </EmailResponder>
    <EmailResponder id="EmailResponder2">
      <protocol value="smtp"/>
      <hostAddress value="svc1smtp.company.com"/>
      <fromAddress value="tcsys@company.com" />
      <toAddress value="admin2@company.com" />
      <suppressionPeriod value="4200"/>
      <emailFormat value="html"/>
    </EmailResponder>
    <LoggerResponder id="LoggerResponder1">
      <logFileName value="WebtierMonitoring.log" />
      <suppressionPeriod value="0"/>
    </LoggerResponder>
  </RespondersConfig>
  <MetricsConfig>
    <Metric name="Abandoned Servers" id="AbandonedServers" maxEntries="100" mode="Collect"
      metricType="integer"
      description="The web tier was unable to connect to the tcserver that the tree cache
        indicates is assigned to the session.">
      <ThresholdWithPeriod>
        <ThresholdValue name="NumberAbandonedServers" value="10"
          description="Alert if number of abandoned servers exceeds this limit for
            time period" />
        <ThresholdPeriod name="TimePeriodInSec" value="600"
          description="Period over which this metric is monitored for exceeding
threshold" />
      </ThresholdWithPeriod>
    </Responders>
```

```

        <ResponderRef id="EmailResponder1"/>
        <ResponderRef id="LoggerResponder1"/>
    </Responders>
</Metric>
<Metric name="No Server Available" id="NoServerAvailable" maxEntries="100"
mode="Alert"
    metricType="integer"
    description="The server pool was unable to assign a tcserver.">
    <ThresholdWithPeriod>
        <ThresholdValue name="NumberNoServerAvailable" value="10"
            description="Alert if number of times no server available exceeds this limit
            during time period" />
        <ThresholdPeriod name="TimePeriodInSec" value="600"
            description="Period over which this metric is monitored for exceeding
threshold" />
    </ThresholdWithPeriod>
    <Responders>
        <ResponderRef id="EmailResponder1"/>
        <ResponderRef id="LoggerResponder1"/>
    </Responders>
</Metric>
    <Metric name="Failed Login Attempts" id="FailedLoginAttempts" maxEntries="100"
        mode="Collect" metricType="integer"
        description="Excessive failed login attempts have been detected.">
        <ThresholdWithPeriod>
            <ThresholdValue name="NumberFailedLoginAttempts" value="2"
                description="Alert if number of failed login attempts exceeds this limit
                during time period" />
            <ThresholdPeriod name="TimePeriodInSec" value="600"
                description="Period over which this metric is monitored for exceeding
threshold" />
        </ThresholdWithPeriod>
        <Responders>
            <ResponderRef id="EmailResponder2"/>
            <ResponderRef id="LoggerResponder1"/>
        </Responders>
    </Metric>
    <Metric name="Grave Events" id="GraveEvents" maxEntries="100" mode="Alert">
        <Responders>
            <ResponderRef id="EmailResponder2"/>
            <ResponderRef id="LoggerResponder1"/>
        </Responders>
    </Metric>
</MetricsConfig>
</ApplicationConfig>

```

Configuring the rich client for startup performance

Overview of configuring the rich client for startup performance

For the Teamcenter rich client to start up and logon to the Teamcenter server, hundreds of megabytes of resources are loaded from the local hard disk into memory. In the *warm* case where the files were recently read into memory and remain in the RAM file cache of the operating system, this initial load can take a few seconds. However, in a *cold* case such as after reboot of the client computer, the limiting factor on performance is how quickly the bytes are read from the hard disk into RAM.

The following situations negatively impact cold file read performance:

- Virus scanning software

Exclude the entire **TC_ROOT/portal** folder and all of its subfolders from virus scanning, as well as the **Teamcenter/RAC** folder under the user folder where the rich client workspace folder is maintained.

- Large **PATH** statement

Minimize the size of your system **PATH** environment variable and remove non-local folders from the **PATH** statement.

- Low hard disk space

Ensure the hard drive where the rich client is installed remains defragmented and never exceeds 75 percent capacity.

- Running additional applications

Minimize use of other resource intensive applications that are competing for pages in memory while the rich client starts.

- Starting the FCC at rich client logon

Start the FMS client cache (FCC) at operating system logon and keep it running in the background so the rich client does not have to start the FCC while the rich client is logging on.

One way to achieve near warm startup times in a cold situation is to warm the rich client files found under the **TC_ROOT/portal** folder using the file warmer capability of the FCC application. The file warmer loads the specified files from the hard disk to the disk cache, effectively changing them to a warm state.

It is beneficial to configure file warmer functionality when all the following conditions exist at your site:

- Rich client startup is very slow.
- The FCC can be started when the user logs on to the operating system or can be manually started a few minutes before the rich client.
- The FCC can be kept active in memory until the user logs off.

And when none of the following conditions exist at your site:

- The client workstation employs very fast media, such as solid-state disk (SSD) media. In this situation, startup is already as fast as possible.
- The client workstation supports multiple simultaneous users on Linux or Citrix server machines.

- There is not enough hard disk cache on the machine to cache the necessary rich client startup files. The hard disk cache requirement is related to the amount of memory (RAM) on the system more than the capacity of the hard disk media. Siemens Digital Industries Software recommends a minimum of 512 MB of available hard disk cache, which provides approximately 2 GB of RAM on Windows.
- There is significant competition for the available disk cache. For example, additional third-party applications are using a similar technique to warm their files.

If all the requiring conditions are met, and none of the preventative conditions exist, configuring the FCC to warm rich client startup files can improve startup performance.

Configuring the FCC file warmer

Configure the filewarmer.properties file

File warmer behavior is controlled by property settings in the **filewarmer.properties** file that are read by the FCC at startup. A sample of this file is provided at *FMS_HOME/filewarmer.properties.template*.

1. Open the **filewarmer.properties** file in the *FMS_HOME* directory.

If this file does not exist, copy the **filewarmer.properties.template** file to the *FMS_HOME* directory and rename it as **filewarmer.properties**.

2. Set the **filewarmer.filelist** option to the name and location of the file containing the list of files to be warmed.

If a list file is not specified, file warming is disabled. If the file is modified, you must restart the FCC for the changes to take effect.

3. Set the **filewarmer.interval** option to the amount of time (in seconds) between warming updates.

The default setting is **1800** (30 minutes.)

4. Set the **filewarmer.mapfiles** option.

If set to **true**, the system memory maps each file and reads a selected part of the data. This method is more efficient for files larger than a few tens of kilobytes.

If set to **false**, the reading option is used.

5. Set the **filewarmer.readfiles** option.

If set to **true**, the system opens and reads each file into a large buffer.

If both the mapping and reading options are set to **true**, the mapping option is performed first.

For sample settings, see the *FMS_HOME/filewarmer.properties.template* file.

Configure the `filelist.txt` file

The **filelist.txt** file contains the list of files to be warmed. This file is used by the **filewarmer.filelist** option in the **filewarmer.properties** file.

1. Open the **filelist.txt** file in the *FMS_HOME* directory.

If this file does not exist, copy the **filelist.txt.template** file to the *FMS_HOME* directory and rename it as **filelist.txt**.

2. List the files and directories to be included (or excluded) from file warming, using the following formatting rules:

- Commented lines (lines beginning with a hash mark (#)) and blank lines are ignored.
- Specify **include** mode by typing **@include** alone on a line. Specify **exclude** mode by typing **@exclude** alone on a line.

By default, the file begins in **include** mode. All files and directories listed in this mode are included in the file warming process. All files and directories listed in **exclude** mode are excluded from the file warming process.

- Enter one file or directory per line.
- Do not use quotation marks.
- Do not specify environment variables.
- Do not use wildcards.
- Do not use relative paths.
- Use any platform-specific directory separators consistently. Do not use double backslashes to represent Windows directory separators.
- Use any path aliases consistently.

The specified directories are scanned at the start of each cycle, allowing the file warmer to adapt to dynamic content changes. The directories are scanned recursively, unless otherwise specified.

If the same file or directory is listed as both included and excluded, the exclusion is ignored.

Example:

Siemens Digital Industries Software recommends setting the following paths:

```
@include
rich client -install-path\portal\plugins
rich client -install-path\portal\features
rich client -install-path\portal\registry
rich client -install-path\portal\configuration
rich client -install-path\portal\Teamcenter.exe
rich client -install-path\portal\Teamcenter.ini
rich client -install-path\portal\.eclipseproduct
RAC-install-path\portal\jre\lib\rt.jar
@exclude
RAC-install-path\portal\plugins\FoundationViewer
```

For additional sample settings, see the *FMS_HOME/filelist.txt.template* file.

Configure file warmer logging behavior

You can use file warmer log files as a diagnostic tool. Logging should not be configured for a production environment.

By default, file warmer logs **CONFIG** output to the FCC log on startup and **EVENT** output to the FCC log at each cycle.

To configure more detailed logging:

1. Set the **FCC_LogLevel** element in the **fcc.xml** file to **TRACE**.
2. Set the **FCC_TraceLevel** element in the **fcc.xml** file to **ADMIN**.

Configure the FCC to locate the filewarmer.properties file

The FCC looks for the **filewarmer.properties** system property at startup. If this value is undefined, file warmer functionality is not enabled. You can define this system property by either editing the **fcc.properties** file or by editing the **startfcc** script.

To edit the **fcc.properties** file:

1. Open the **fcc.properties** file in the *FMS_HOME* directory.

If this file does not exist, copy the **fcc.properties.template** file to the *FMS_HOME* directory and rename it as **fcc.properties**.

2. Add one of the following properties:

- Windows systems:

```
filewarmer.properties=C:\\Program Files\\Teamcenter\\fcc\\filewarmer.properties
```

Use the full path to the properties file. Use double backslashes as directory separators.

- Linux systems:

```
filewarmer.properties=/usr/bin/teamcenter/fcc/filewarmer.properties
```

Use the full path to the properties file. Use single forward slashes as directory separators.

To edit the **startfcc** script:

1. Open the **startfcc.bat** (Windows) or **startfcc.sh** (Linux) file in the *FMS_HOME* directory.
2. Add one of the following properties:

- Windows systems:

```
-Dfilewarmer.properties=C:\\Program Files\\Teamcenter\\fcc\\filewarmer.properties
```

Use the full path to the properties file. Use double backslashes as directory separators.

- Linux systems:

```
-Dfilewarmer.properties=/usr/bin/Teamcenter/fcc/filewarmer.properties
```

Use the full path to the properties file. Use single forward slashes as directory separators.

Configure TCCS to start when users log on to a Windows operating system

If file warming for the FMS client cache (FCC) is configured, you can also configure a Windows system to launch TCCS each time the system starts and cache rich client files to main memory. Using this functionality in conjunction with FCC file warming improves system startup performance

Note:

If implemented when Kerberos authentication is not configured for zero sign-on, users are prompted to authenticate any proxy servers when TCCS is started. The consequence is that each time users log on to Windows, they are prompted to authenticate any proxy servers.

Warning:

JRE shutdown hooks are not honored, preventing the FCC from closing cleanly. If the TCCS/FCC instance remains running when users log off (or shut down), these operating systems, the FCC segment cache may be corrupted.

Siemens Digital Industries Software recommends you add the **fccstat -kill** command to all user logoff scripts and to any relevant Windows shutdown scripts for Teamcenter clients.

1. Create a script to automatically start TCCS.

Create a batch (**.bat**) script containing the following instructions:

- a. Set the **FMS_HOME** environment variable.

The **FMS_HOME** environment variable points to the folder where FMS is installed. By default, this location is the **tccs** directory in the Teamcenter installation folder.

For example:

```
set FMS_HOME=C:\Progra~1\Siemens\Teamcenter\version-number\tccs
```

C:\Progra~1\Siemens\Teamcenter\version-number is the Teamcenter installation folder. The FCC runs as a part of TCCS and is installed in the same folder.

- b. Set the **JRE_HOME** environment variable.

The **JRE_HOME** folder points to the directory where the Java JRE is installed on the system. The Java version must be 1.7 or later.

For example:

```
set JRE_HOME=C:\Progra~2\Java\jre7
```

- c. (Optional) Enable startup logging for the FCC.

Include this instruction only if the script is being used for debugging purposes. If included, the FCC creates a log for its startup events at the given file path.

For example:

```
set FMS_FCCSTARTUPLLOG=C:\fccstartup.log
```

- d. Start TCCS.

```
call FMS_HOME\bin\fccstat -start
```

FMS_HOME is the value set in the first step.

- e. Set the **_EL** variable to the correct FCC error level.

If the FCC does not start correctly, exiting with an error code, the FCC sets **ERRORLEVEL** to the correct FCC error code. You can use this value for debugging.

For example:

```
set _EL=%ERRORLEVEL%
```

ERRORLEVEL is the level set by the FCC.

- f. Exit if startup is successful.

If TCCS starts correctly, the script is instructed to close.

```
if "%_EL%" == "0" goto worked
```

- g. Retry TCCS startup.

If TCCS did not start correctly in the previous step, instruct the script to retry FCC startup step a few seconds later. The number after **-n** is the approximate number of seconds to wait. Siemens Digital Industries Software recommends setting this between 10 and 30 seconds. (Accuracy of this timing is not critical to the operation of this script.)

For example:

```
@ping 127.0.0.1 -n 30 -w 1000 > nul
goto retry
```

- h. Mark the completion of script execution.

Instruct the script to print **FCC successfully started** on the console upon successful completion.

```
:worked
echo FCC successfully started.
```

An example of the completed script is:

```
set FMS_HOME=C:\Progra~1\Siemens\Teamcenter9\tccs
set JRE_HOME=C:\Progra~2\Java\jre7
set FMS_FCCSTARTUPLUG=C:\fccstartup.log
:retry
call %FMS_HOME%\bin\fccstat -start
set _EL=%ERRORLEVEL%
if "%_EL%" == "0" goto worked
@ping 127.0.0.1 -n 30 -w 1000 > nul
goto retry:
:worked
echo FCC successfully started.
```

2. Configure TCCS to use the script.

- a. Store the script in the appropriate Windows startup directory (Windows 7 or Server2008):

- Single-user system:

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup

- Multiuser system:

C:\Users\user-name\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

Note:

These directories may be hidden by default.

- b. (Optional) Set the **TCCS_CONFIG_HOME** environment variable to the TCCS home directory.

This step is required only when the default home location is not used and a custom TCCS home location is created.

- c. (Optional) Set the **TCCS_CONFIG** environment variable to the TCCS configuration directory containing information about the various TCCS environments.

This step is required only when the default TCCS configuration name is not used.

Setting the PATH and AUX_PATH environment variables for enhanced performance

Cold start performance is improved when the operating system's **PATH** environment variable is shortened to its minimum. When this operating system environment variable is used to track a large number of locations, performance declines.

The rich client startup script sets the operating system **PATH** environment variable before opening the client. To reduce the overall size of the environment variable's value, Teamcenter excludes the existing system **PATH** value from the final **PATH** value used for the rich client startup.

If your Teamcenter deployment integrates applications with the rich client, and the integrations require that path locations are added to the operating system's **PATH** environment variable, add the paths to the **AUX_PATH** Teamcenter environment variable. For example:

- Windows systems:

```
set AUX_PATH=C:\new\path;%AUX_PATH%
```

- Linux systems (using **ksh**):

```
export AUX_PATH=/new/path:$AUX_PATH
```

Note:

Adding too many paths to the **AUX_PATH** environment variable defeats the purpose of shortening **PATH**.

Might deleting my rich client cache improve performance?

In some cases, deleting the rich client cache may improve Teamcenter performance.

To improve performance and tailor the user experience, the Teamcenter rich client caches information that is local and specific to each user. The cache includes information about session windows, current plug-ins, system configuration, dialogs, display preferences, and so on. However, in the following situations, contents of the cache can degrade performance:

- When a plug-in is updated with new or modified code, the cached information does not match the new or modified plug-in, and the client may not work properly.
- When a user opens thousands of items, the current application can become slow to respond. The next time that user logs on to the rich client, the system attempts to open the previously accessed items and can take a long time to complete the startup.

In these situations, removing the cache is recommended. The cached information can be safely removed because the client regenerates any needed information when the user next logs in.

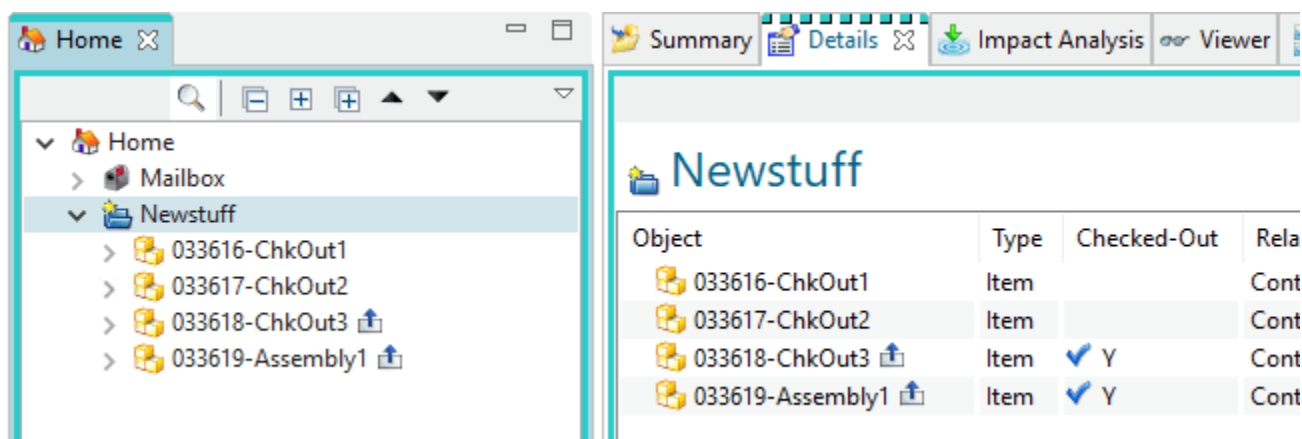
Caution:

If TCCS advanced configurations are enabled, a TCCS configuration (*cfg*) directory exists within the ...*Siemens* cache folder. While user cache files can be deleted, the ...*Siemens**cfg* configuration directory must not be deleted or TCCS will stop working correctly.

When TCCS is configured for	The configuration directory is here	
	Windows operating system	Linux operating system
Public (shared)	<i>C:\ProgramData\Siemens\cfg</i>	<i>/etc/Siemens/cfg</i>
Private	<i>C:\users\{USERID}\Siemens\cfg</i>	<i>\$HOME/Siemens/cfg</i>

What can I do to improve my startup performance?

By default, in rich client tree views a **checked-out** symbol displays to the right of checked out Teamcenter objects. It is a quick indicator of whether the object is checked out. When an object is checked out, a user has exclusive access to it, preventing others from making changes to the object until it is checked back in.

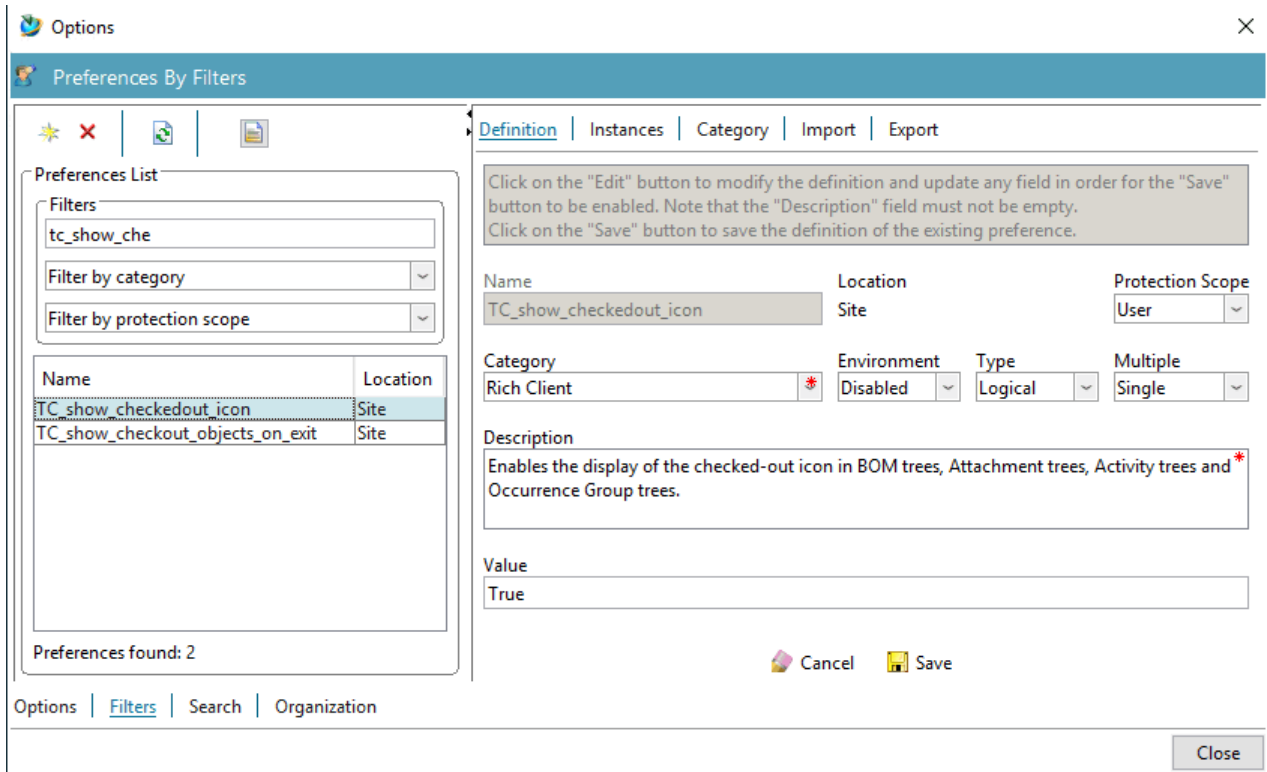


However, displaying this symbol for every object in a very large assembly (5,000 or more objects) can degrade performance. Turning off the display of this symbol in tree views (assembly trees, attachment trees, and so forth) improves performance. Turn off the display of the symbol by setting the **TC_show_checkedout_icon** preference to **FALSE**.

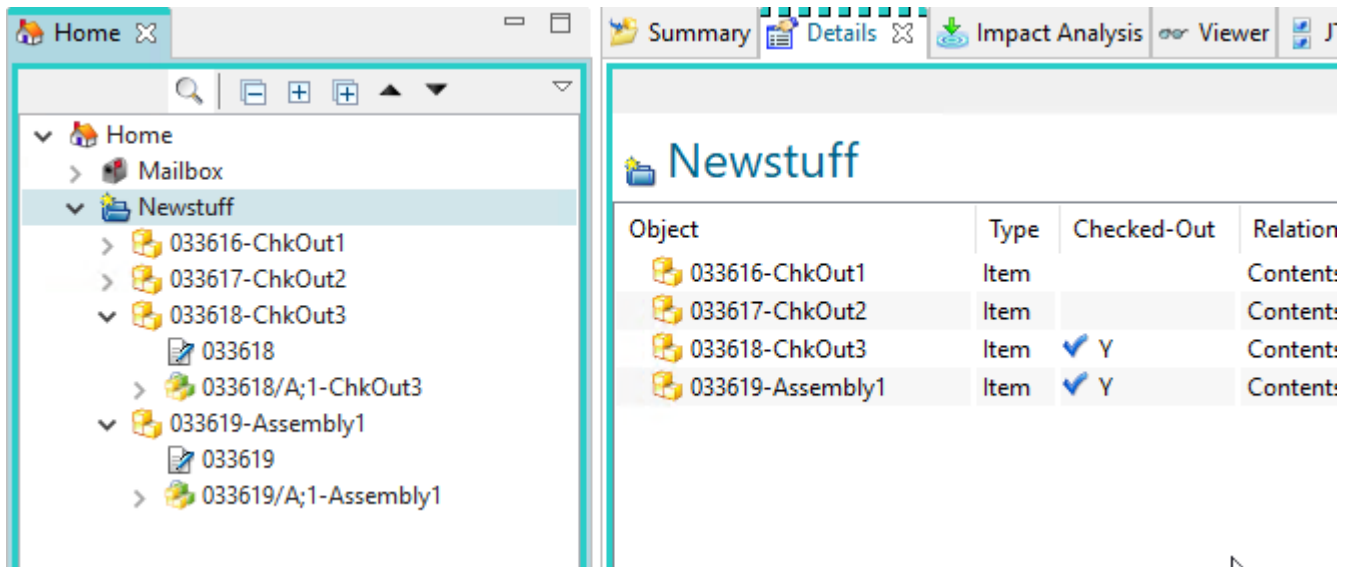
Checkout status always appears in the **Details** view, regardless of how you set the display of the checked-out symbol. If you turn off the display of the checked-out symbol, refer to the **Checked Out** column in this view to determine the check out status of objects.

To turn off the display of the checked-out symbol:

1. Choose **Edit→Options** to display the **Options** dialog box.
2. At the bottom left of the dialog box, click the **Filters** tab to display the **Preferences by Filters** pane.



3. In the first **Filters** box, type as much of **TC_show_checkedout_icon** as needed to reduce the list of preferences so that the preference displays in the **Preferences List**.
4. Select the preference from the list. In the **Value** box, change the value to **FALSE**.
5. Click **Save**.
6. Click **Close** to close the dialog box.
7. Log off your session. Log on again to see the change take effect. The symbol no longer appears to the right of objects in tree views.



POM_timestamp maintenance

The **POM_timestamp** table records the time of an object's most recent modification. The **POM_timestamp** table holds time stamps until manually cleared. To avoid an overly large **POM_timestamp** table and any related performance issues, Siemens Digital Industries Software recommends that you provide periodic maintenance of the **POM_timestamp** table.

Do so by running the following command on a periodic basis:

```
install -tidy_timestamps -u=user -p=password -g=group
```

Consider setting up a timed job that executes the previous command from within the Teamcenter environment.

Use the following commands to manage the maintenance of the **POM_timestamp** table.

- To enable manual mode:

```
install -set_pom_param -u=user -p=password -g=group TC_TIMESTAMP_TIDY_MODE manual
```

- To perform manual maintenance on the **POM_timestamp** table (nightly or as appropriate):

```
install -tidy_timestamps -u=user -p=password -g=group
```

- To disable manual mode, remove the manual parameter configuration:

```
install -set_pom_param -u=user -p=password -g=group TC_TIMESTAMP_TIDY_MODE
```

- To view the current configuration:

```
install -ask_pom_param -u=user -p=password -g=group TC_TIMESTAMP_TIDY_MODE
```

Note:

If a configuration is not displayed then the default configuration (automatic mode) is in effect.

- To adjust the time stamp threshold value:

```
install -set_pom_param -u=user -p=password -g=group TC_TIMESTAMP_THRESHOLD hours
```

hours must be an integer.

Cleaning the backpointer table after upgrade

In Teamcenter 9.1 and later versions, **relation_type** object references and **ImanRelation** primary and secondary object references are no longer stored in the backpointer table, but are stored only in the **ImanRelation** table. In addition, where-referenced queries now search the **ImanRelation** table for **ImanRelation** references, rather than searching the backpointer table. These changes improve performance by significantly reducing the size of the backpointer table.

To enable this performance improvement, you must run the **clean_backpointer** utility on your Teamcenter database after you upgrade from a version earlier than Teamcenter 9.1. The **clean_backpointer** utility is not run automatically during upgrade because this cleanup operation can be time-consuming.

If your initial Teamcenter deployment was Teamcenter 9.1 or later, you do *not* need to perform this procedure. If you performed this procedure during a previous upgrade to Teamcenter 9.1 or later, you do *not* need to perform it again.

The **clean_backpointer** utility scans the backpointer table for **relation_type** object references and **ImanRelation** primary and secondary object references, confirms their existence in the **ImanRelation** table, and deletes the instances from the backpointer table. The utility's performance may vary, depending on infrastructure elements such as database load, network performance, server configuration, and so on.

Because performance varies, Siemens Digital Industries Software recommends following these best practices:

1. Run the utility with the **-m** argument set to **INFO** to determine whether the **POM_BACKPOINTER** table needs to be cleaned. (This option returns the number **BACKPOINTER** records to be removed.)
2. To determine the amount of time it will take to clean the backpointer table using the **CLEAN** or **DO_IT** options, perform the operation on a backup or test system.
3. Clean the backpointer table:

- Use the **CLEAN** option if you wish to perform the analysis just prior to cleaning the **POM_BACKPOINTER** table.
- Use the **DO_IT** option to clean the **POM_BACKPOINTER** table in the most efficient manner. To use this option, you must run the utility with the **INFO** option first.

Caution:

Due to renaming of the **POM_BACKPOINTER** table, when running the **clean_backpointer** utility in **CLEAN** mode or **DO_IT** mode, make sure the system administrator has exclusive use of the system.

Improving Eclipse expression evaluation

The rich client uses Eclipse menus, commands, and the shortcut menu. Eclipse may evaluate several hundreds or thousands of expressions to control the visibility of any specific command in a menu. In some cases, the Eclipse expression evaluation mechanism may cause problems.

Possible cause	Problem
In the Manufacturing Process Planner and Structure Manager applications, a user performs Select All , Copy , Paste or Remove actions on a large number of BOM lines.	This may cause Teamcenter to hang.
In the My Teamcenter application with a dataset attached to an item revision, a user: • Opens an item revision into the Details pane, right-clicks the dataset, and chooses Check In/Out→Check Out. • Selects an item revision in the Home view and chooses Tools→Check In/Out→Check Out. • Right-clicks an item revision in the Home view and chooses Check In/Out→Check Out.	The Check-in and Check-Out menus are disabled.

To avoid these issues, Siemens Digital Industries Software recommends adding **-Dexpressioncache.enable=Y** to the **TC_ROOT/portal/portal.bat** file. For example:

```
start Teamcenter.exe %* -vm "%JRE_HOME%\bin\javaw.exe" -vmargs -Xmx%VM_XMX%
-XX:MaxPermSize=128m
-Dexpressioncache.enable=Y -Xbootclasspath/a:"%JRE_HOME%\lib\plugin.jar";
"%JRE_HOME%\lib\deploy.jar";"%JRE_HOME%\lib\javaws.jar" %DJIPJL_VMARG%
```

Optimizing read expression generation

Read expressions map to a given object and record its read access characteristics for any given user, group, role, and session parameter (for example, session projects).

Teamcenter automatically maintains read expressions by regenerating and updating them whenever objects are modified. Use the **am_read_expression_manager** utility to generate these expressions. By default, this utility is active as a service. Depending on which mode you select, this utility:

- Updates stale read expressions.
- Creates new read expressions.
- Manages changes based on rule tree updates.
- Deletes superfluous data.
- Objects without read expressions, which the system loads as part of any BOM expand or search, are marked to receive read expressions generated by the **am_read_expression_manager** service when it performs an update.

Read expression generation can be optimized using the following techniques:

- Read expression daemon execution parameters
- Pre-instrumentation of read expressions for specific sets of objects
- Rule tree updates

Read expression daemon execution parameters

Use the **daemon** mode of the **am_read_expression_manager** utility to update stale read expressions for business objects.

The following parameters determine how and when the read expression daemon process runs:

- **Sleep time**

Specifies the time, in seconds, during which the process remains dormant. The default is **10**.

- **Spawn intervals**

Specifies the number of repeated executions of the daemon before a new process is spawned. The default is **500**; however, this depends on the total number of objects processed.

You can specify these parameters either:

- In the command line if the daemon process is involved within a shell.
- Or, as a preference if the process runs as a service.

The service checks these values either when first executing or when re-spawning.

Preference Name	Type	Function
RE_MGR_sleep_time	Integer	Sleep time, in seconds, between executions. The default is 10 .
RE_MGR_exec_count	Integer	Execution count until a new daemon is spawned. The default is 500 .
RE_MGR_concurrency_objs_threshold	Integer	Threshold value of stale read expressions. The default is 1,000,000 .

To limit the read expression daemon process in the event of performance problems pertaining to system response times, Siemens Digital Industries Software suggests you set the daemon sleep time equal to the average daemon processing time per interval. You can view this time in the daemon execution console output. This suppresses the daemon speed to about 50 percent of maximum speed, thereby improving your system performance.

Pre-instrumentation of read expressions for specific sets of objects

The **am_read_expression_manager** utility allows identification of business objects for which read expressions can be force generated. To do this, query execution and/or UID list input is possible. This inserts new stale read expression place holders into the associated tables. However, these then have to be computed according to the rule tree and the objects involved, which can be time consuming.

You should perform this pre-instrumentation immediately after an install or upgrade, as not many objects have received read expressions. Otherwise, many business-relevant objects will have automatically received read expressions and the benefit of this pre-instrumentation is diminished.

Siemens Digital Industries Software recommends the following best practice if you require this upfront computation of read expressions:

1. For those sets of objects that have been identified to be relevant for business needs and should have read expressions as early as possible, perform a dry run of the associated query to obtain the total number of affected objects.

2. Determine how many objects per hour can be processed by the daemon in your system. You can calculate this using the daemon command shell output for individual processing intervals and computing an average.
3. Using these two numbers, establish a timeline for computation of such read expressions. You should perform this during a quiet time in your system, for example, at night or during a weekend.

Note that multiple pre-instrumentation daemon processes must run consecutively; multiple daemon processes cannot run in parallel.

Rule tree updates

Rule tree updates change the security profile of objects and potentially invalidate large numbers of read expressions of business objects. When such rule tree updates occur, the associated read expressions are examined and marked as stale if impact is perceived. Every attempt is made to limit the number of such invalidations; however, at times it may be necessary to invalidate large numbers of read expressions and potentially the entire read expression table.

If your rule tree requires updating a substantial number of objects, Siemens Digital Industries Software recommends you perform the rule tree update during the start of a quiet time to allow the read expression daemon adequate time to process updates before users are online.

Rich client timeouts

The rich client uses Eclipse menus, commands, and the shortcut menu. Eclipse may evaluate several hundreds or thousands of expressions to control the visibility of any specific command in a menu. In some cases, the Eclipse expression evaluation mechanism may cause problems.

To avoid these issues, Siemens Digital Industries Software recommends adding **-Dexpressioncache.enable=Y** to the **TC_ROOT/portal/portal.bat** file. For example:

```
start Teamcenter.exe %* -vm "%JRE_HOME%\bin\javaw.exe" -vmargs -Xmx%VM_XMX%
-XX:MaxPermSize=128m
-Dexpressioncache.enable=Y -Xbootclasspath/a:"%JRE_HOME%\lib\plugin.jar";
"%JRE_HOME%\lib\deploy.jar";"%JRE_HOME%\lib\javaws.jar" %DJIPJL_VMARG%
```

Active Workspace timeouts

The following diagram shows a typical Teamcenter layout.



Types of timeouts

Activity timeout

Starts a timer on the last activity and will timeout the resource if no new activity is detected within the timeframe.

Lack of response timeout

Encountered when a resource makes a request and does not receive a response within the timeout period

Teamcenter server manager (pool manager) timeout

This is an activity timeout.

If the tcserver doesn't receive any requests within the timeout period it is closed out.

Server manager timeout settings

Teamcenter web tier session timeout

This is an activity timeout.

The Teamcenter web tier delegates the web session. This is manifested via a session cookie:

- **JSESSIONID** for J2EE
- **ASP.NET_SessionId** for IIS

Session timeout represents the event occurring when a user does not perform any action on a web site during an interval (defined by a web server). The event on the server side, changes the status of the user session to 'invalid' (no longer used) and it instructs the web server to destroy it, deleting all data contained in it.

Session Timeout on owasp.org

J2EE web.xml

This is in terms of minutes

```
<session-config>
  <session-timeout>15</session-timeout>
</session-config>
```

IIS web.config

This is in terms of minutes

```
<configuration>
  <system.web>
```

```
<sessionState timeout="20"></sessionState>
</system.web>
</configuration>
```

or

```
<sessionState mode="InProc" cookieless="false" timeout="1440"></sessionState>
```

Active Workspace gateway timeout

This is a lack of response timeout.

This is a timer for how long the Active Workspace Gateway will wait for a response from a backend server (Teamcenter web tier, FMS, etc). Session timeout on requests originating with the server to backends/routes. The **node.js** default is 2 minutes. The parameter is in milliseconds.

- https://nodejs.org/api/http.html#http_request_settimeout_timeout_callback
- https://nodejs.org/api/https.html#https_server_settimeout_msecs_callback

You can configure per the CLI or via the configuration file.

autoLogout

Default value is 120000 ms (2 minutes).

This is a session logout timeout. Specifies the amount of time the gateway should wait before logging out the user session. The gateway starts waiting when either: All Active Workspace browser tabs to that session are closed or when the browser is completely closed.

Requires WebSocket communication between the browser and the gateway. When an Active Workspace client logs in using a browser, it establishes a WebSocket connection between the browser and the Active Workspace Gateway. If this WebSocket connection is lost for more than a timeout period, then the Active Workspace Gateway will use the session cookie to terminate the session via logout.

httpRequestResponse

Default value is 600000 ms (10 minutes).

This is a timeout on request responses originating with the server to backends/routes.

```
{
  "timeout": {
    "autoLogout": 120000,
    "httpRequestResponse": 600000
  }
}
```



```
}
}
```

Proxy timeout

This is a lack of response timeout.

Proxy servers like Apache and F5 have activity timeout as well as lack of response timeouts. Activity timeouts usually default to 300 seconds. They have been a problem for NX for years. NX manager user sessions sit for hours with no activity on the connection. So it can get caught by session timeout as well. The latest proxies have plenty of RAM so there isn't a problem keeping session for long period. In the past, administrators would want idle sessions clear to save resources.

A load balancer using session cookies to maintain persistence (stickiness) solves the problem because with the cookies the load balancer re-establishes the connections even when they have been timeout.

Apache Server (httpd)

<https://httpd.apache.org/docs/2.4/mod/core.html#timeout>

F5 Load Balancer

<https://www.f5.com>

Amazon Elastic Load Balancer

<https://aws.amazon.com/blogs/aws/elb-idle-timeout-control/>

Active Workspace client polling timeout

This is an activity timeout.

There are background SOA calls if the user leaves the client open in a browser. This timeout controls how long these polling calls happen after the last user activity in the browser for the Active Workspace client.

The Teamcenter preference name is **AWC_Polling_Timeout**. This is a site level preference and the value is in minutes. If the value is not greater than zero, there is no choking of the polling. The value must be greater than zero to enable this feature.

Following are related Teamcenter preferences that do not control timeouts, but rather the wait times between polling calls from the client.

AWS_Notifications_Polling_Interval	This value is in minutes. If set to 0 (zero), this will disable the polling.
AWS_Inbox_Polling_Interval	This value is in minutes. If set to 0 (zero), this will disable the polling.

User session timeout

When it comes to timeouts on the tcserver, you need to consider the following.

- Assuming you have a user logged into an AW client & they stop all interactive activity.
- The **AW Client polling timeout** needs to happen.
- The **TC web tier session timeout** and **TC Server Manager timeout** would begin as soon as the Active Workspace client stops making requests.

Once an Active Workspace client user goes idle, the **AW Client polling timeout** countdown begins. When this timeout is reached, the Active Workspace client blocks background polling SOA calls, and the Teamcenter web tier timeout countdown starts from the last SOA call. Once this timeout is reached, subsequent call to Teamcenter require re-authentication with the underlying identity provider.

Troubleshooting

the following information may be needed by support personnel.

- What are the timeout settings?
 - Preferences: **AWC_Polling_Timeout**, **AWS_Notifications_Polling_Interval**, **AWS_Inbox_Polling_Interval**
 - Active Workspace Gateway: http response timeout
 - Teamcenter web tier: session timeout
 - Teamcenter pool manager: inactivity timeout
- What is the tested scenario that is not working as desired?

For example:

- User logs into Active Workspace client.
- User performs specific actions.
- User stops interacting with the AW client browser tab.

Was the browser tab in the foreground or background?

- After a specified time, the user does a specific action which indicates that a timeout has occurred.

Was the user presented with an error, prompted for re-authentication, or resumed the session?

- What is the expected behavior?
 - What exactly was expected other than the observed behavior?
- Additional data that may be required.
 - Browser HAR of the failing scenario from login or page refresh to failure.

Verify that the preferences returned from **getPreferences** align with the configured timeout settings.

- Active Workspace log file.
- Active Workspace gateway **config.json** file.

20. Encryption key management

Teamcenter Key Manager

Teamcenter Key Manager is an optionally-installed feature that enables you to manage, edit, and share encryption keys and policies in a centralized secure manner for a distributed Teamcenter site.

Teamcenter Key Manager consists of a central key manager server which delegates key storage and cryptography to the public key cryptography standard PKCS #11 hardware or software plugin of your choice. Default Network Security Services (NSS) is delivered with Teamcenter Key Manager. Customized Network Security Services (NSS) and hardware security modules (HSM) are also supported.

Trusted Teamcenter satellites communicate with the key manager server using two-way SSL to share keys and perform cryptography for features and applications such as Teamcenter two-tier rich clients and Dispatcher. With Teamcenter Key Manager, you can define component-specific policies governing the algorithms used for key generation and the key encryption and decryption life times.

The key manager server, where all policies and keys are stored, resides on single host, accessed over secure HTTPS on a dedicated configurable port with two-way transport layer security. A key manager satellite resides on each enterprise tier host, and accesses the key manager server over HTTPS.

An optional secondary key manager server maintains synchronization with the primary key manager server, and automatically acts as a failover server for satellites if the primary key manager server becomes unavailable. The secondary server resides on a separate host than the primary server.

For sites using Teamcenter Key Manager, if key manager servers or the satellites are not running, users will not be able to log into Teamcenter.

Install Teamcenter Key Manager

If your site is using Teamcenter Key Manager for cryptographic services, use the following installation approaches:

When installing Teamcenter using TEM

Teamcenter Key Manager and the local key manager satellite must be installed and running before Teamcenter can be installed. Teamcenter Key Manager and key manager satellites are installed using Deployment Center as described in the following steps.

Once Teamcenter Key Manager is installed, install Teamcenter using Teamcenter Environment Manager (TEM). On the TEM **Key Manager Configuration** panel, check **Enable Key Manager** and specify the location of the key manager satellite.

When installing Teamcenter using Deployment Center

Teamcenter Key Manager and the local key manager satellite can be installed before or after Teamcenter foundation is installed using Deployment Center.

A key manager satellite must be installed on every machine that will be using Teamcenter Key Manager. SSL certificates must also be established on each machine involved in the Teamcenter Key Manager system.

Note:

Once a site is using Teamcenter Key Manager for cryptographic services, it cannot be reconfigured to use an alternate cryptographic system without reinstalling Teamcenter.

Install the key manager server and satellites

Teamcenter Key Manager is installed and deployed using Deployment Center. See the **Teamcenter Compatibility Matrix** to verify the version of Deployment Center required to install and deploy Teamcenter Key Manager.

The following steps are specific to installing and deploying Key Manager using Deployment Center. For detailed information on using Deployment Center, see the most recent Deployment Center help collection available on Support Center.

1. Download the Teamcenter Foundation software kit from Support Center and place it in the Deployment Center software repository.
2. Log onto Deployment Center and click the **Software Repositories** tile to view the Teamcenter Foundation software kit, verifying its availability in the software repository.
3. Click the **Environments** tile to display the environments scanned by Deployment Center. Select or add an environment in which to install Teamcenter Key Manager.
4. Click the **Deploy Software** tab. On the **Software** tab, add the Teamcenter Foundation software kit to the **Selected Software** list.
5. On the **Options** tab, select the options for your environment. Selecting **Single Box** allows you to install the key manager server and satellite on one machine. Selecting **Distributed** allows you to install the key manager server on one machine and satellites on one or more other machines.
6. On the **Applications** tab, click  to edit the selected applications. The list of available applications is displayed.
7. Under **Available Applications**, check **Key Manager**.
8. Remove the check from **Teamcenter Foundation** if any of these conditions apply:
 - You will be installing Teamcenter using TEM.
 - Teamcenter has already been installed using Deployment Center.

- You will install Teamcenter later using Deployment Center.

Click **Update Selected Applications**.

- On the **Components** tab, one key manager server and one key manager satellite component are listed.
 - If you have additional machines requiring satellites and selected **Distributed** on the **Options** tab, click ⊕ to add a satellite for each additional machine. Satellites are required for each machine requiring cryptographic services, such as machines with two-tier rich clients and Dispatcher.
 - If you plan to deploy a secondary key manager server, add a secondary server to the list.
 - If you are installing Teamcenter and have the **Server Manager** component checked, ensure that a Key Manager satellite is installed on the server manager machine.
- Bring each component to a 100% complete status by supplying any remaining required settings for each. With a component selected, move your cursor over fields in the right pane for tips on entering setting values for that component and the machine on which it will be deployed. Values will all be verified during deployment of the component. Be aware of the following items:
 - When setting a **PKCS#11 Configuration Type** value for the Key Manager Server, **Default Network Security Services (NSS)** is delivered with Teamcenter Key Manager. Selecting **Customized Network Security Services (NSS)** or **Hardware Security Module (HSM)** requires that those security systems are already installed at your site. Primary and secondary key manager servers must use the same PKCS#11 configuration type.
 - The Key Manager Server **Default Network Security Services (NSS)** PIN can contain numbers, letters, and special characters.
 - Record PINs, passwords, and other values for later reference.
- On the **Deploy** tab, generate the installation scripts. Review the **Deploy Instructions** information related to the generated installation scripts.
- Follow the steps in *Run the deployment scripts* in the Deployment Center help collection to run the deployment scripts for the primary server and satellites.
 - Do not run the deployment script for a secondary key manager server until the primary key manager server and satellites are deployed and running. Deploying a secondary server requires that the primary server has already been deployed. See **Installing a secondary Teamcenter Key Manager server** for more details.
 - If you selected **Distributed** on the **Options** tab and are deploying multiple satellites, deploy the Teamcenter Key Manager server before deploying satellites. Attempting to deploy satellites before deploying the server will fail.

If you selected **Single Box** on the **Options** tab, the Teamcenter Key Manager server will automatically deploy before the satellite.

- Record the key manager and satellite installation directories for use during Teamcenter installations on each machine requiring Teamcenter Key Manager.
13. If you are deploying Teamcenter Key Manager on Linux as a user without root privileges, log on as a user with root privileges and run the following scripts from the Teamcenter Key Manager installation directory (`$INSTALL_DIR`) for each server and satellite deployment:

Teamcenter Key Manager server:

```
$INSTALL_DIR/KeyManagerServer/root_post_tasks_key_manager_server<id>.ksh
```

```
$INSTALL_DIR/KeyManagerServer/root_post_tasks_key_manager_ca_server<id>.ksh
```

```
$INSTALL_DIR/KeyManagerServer/root_post_tasks_key_manager_admin<id>.ksh
```

Teamcenter Key Manager satellites:

```
$INSTALL_DIR/KeyManagerSatellite/root_post_tasks_key_manager_satellite<id>.ksh
```

14. Teamcenter Key Manager servers and satellites run as services on Windows, and are started automatically. Ensure the services were successfully started as part of the deployments. If they are not running, start them using the steps in **Start and stop Teamcenter Key Manager services**.

On Linux, Teamcenter Key Manager servers and satellites run as daemons that may need to be started manually if Java is not installed on the local machine. Ensure the following daemons are started. If necessary, start them as described in **Start and stop Teamcenter Key Manager daemons**.

- `rc.key.manager.ca.server.<your_env_name>`
- `rc.key.manager.admin<your_env_name>`
- `rc.key.manager.satellite.<your_env_name>`
- `rc.key.manager.server.<your_env_name>`

Establish SSL certificates

To provide keys and perform cryptography for programs running in the distributed Teamcenter environment, Teamcenter Key Manager is itself a separate distributed client-server system in which the client and the server trust each other using a PKI infrastructure.

As part of the installation of the key manager server and key manager satellites, certificates were generated that can be trusted by Teamcenter Key Manager.

Installing Teamcenter Key Manager for use with Multi-Site Collaboration

To support Multi-Site Collaboration file transfers, keys must be available on both sites.

Install a secondary Teamcenter Key Manager server

When using Teamcenter Key Manager for cryptographic services, you can optionally install a secondary key manager server for redundancy and failover purposes. The primary key manager server and satellites must be installed and running before the secondary server can be installed. The secondary key manager server must be installed on a different machine than that on which the primary key manager server is installed.

Generate the secondary key manager server deployment script

If you have already generated a deployment script for your secondary key manager server as described in [Install Teamcenter Key Manager](#), proceed to *Install the secondary key manager server*. Otherwise, generate your secondary server script using the following steps.

The secondary key manager server is installed and deployed using Deployment Center. See the *Hardware and Software Certifications* knowledge base article on Support Center to verify the version of Deployment Center required to install and deploy the secondary key manager server.

1. Download the Teamcenter Foundation software kit from Support Center and place it in the Deployment Center software repository.
2. Log onto Deployment Center and click the **Software Repositories** tile to view the Teamcenter Foundation software kit, verifying its availability in the software repository.
3. Click the **Environments** tile to display the environments scanned by Deployment Center. Choose the environment in which the primary key manager server has been installed.
4. On the **Components** tab, the key manager server and key manager satellite components are listed and shown to be 100% complete. Click ⊕, check **Key Manager Secondary Server**, and click **Update Selected Components**.
5. Bring the secondary server component to a 100% complete status by supplying any remaining required settings. Values will all be verified during deployment of the component.
 - Set the **PKCS#11 Configuration Type** value to be the same as that set for the primary key manager server.
 - If your site is using hardware security modules (HSM), the secondary server must use a different HSM server than that used by the primary server. Alternately, one server can be configured to use an HSM server and the other server can be configured to use the default Network Security Services or customized Network Security Services.

6. On the **Deploy** tab, generate the installation scripts. Review the **Deploy Instructions** information related to the generated installation scripts.

Install the secondary key manager server

Example:

These installation steps require the primary key manager server be disabled during the deployment of the secondary server installation script. Schedule the timing of this process to minimize impact at your site.

1. On the machine running the primary key manager server:
 - Stop the key manager server.
 - Start the Teamcenter Key Manager root certificate authority server if it is not running.
 - Start the Teamcenter Key Manager Administration application if it is not running.

See **Start and stop Teamcenter Key Manager services** for instructions on starting and stopping servers and applications on Windows. See **Start and stop Teamcenter Key Manager daemons** for instructions on starting and stopping servers and applications on Linux.
2. Follow the steps in *Run the deployment scripts* in the Deployment Center help collection to run the secondary server deployment script.
3. The secondary key manager server runs as a service on Windows that is started automatically. Ensure the Teamcenter Key Manager Server service was successfully started as part of the deployment. If it is not running, start it using the steps in **Start and stop Teamcenter Key Manager services**.

On Linux, the secondary key manager server runs as a daemon that may need to be started manually if Java is not installed on the local machine. Ensure the **rc.key.manager.server** daemon is started. If necessary, start it as described in **Start and stop Teamcenter Key Manager daemons**.

Ensure the **Teamcenter Key Manager Administration** application remains running on the secondary server at all times. (As it is read-only, leaving it running does not create a security risk.)

4. On the machine running the primary key manager server:
 - Restart the key manager server. The secondary key manager server will automatically begin synchronizing with the primary server.
 - Stop the Teamcenter Key Manager root certificate authority server.

- Stop the Teamcenter Key Manager Administration application.

See **Manage a secondary key manager server** for more information on working with your secondary key manager server.

Administering Teamcenter Key Manager

Work with Key Manager

Manage keys, certificates, and policies using the web-based Teamcenter Key Manager Administration application. Key Manager Administration is a web service or daemon that, for security purposes, is not running by default.

Use care when managing keys, certificates, and policies. Changes have broad impact, affecting users across the Teamcenter site.

Starting the Key Manager administration application web service or daemon

The process for starting the administration application is the same for the primary key manager server as it is for the secondary key manager server. On the appropriate machine, start the service or daemon as follows:

Windows service:

Start the Windows service using the Windows Services application. Its name has the following form:

"Teamcenter Key Manager Admin Web <your_chosen_environment_name>"

Linux daemon:

From the server machine console, navigate to the Teamcenter Key Manager server installation directory: `$INSTALL_DIR/KeyManagerServer/`

Run:

```
key_manager_admin -start
```

Though you can leave the web service or daemon running on the primary server, for security purposes, you may wish to just run it when you have an actual need to use Key Manager Administration. The administration web service or daemon must remain running on the secondary server. (As keys and policies cannot be changed on the secondary server, doing so is not a security risk.)

Running the Key Manager Administration application

Once the service or daemon is running, open the Teamcenter Key Manager Administration application by navigating to the following URL and entering the key manager administrator password:

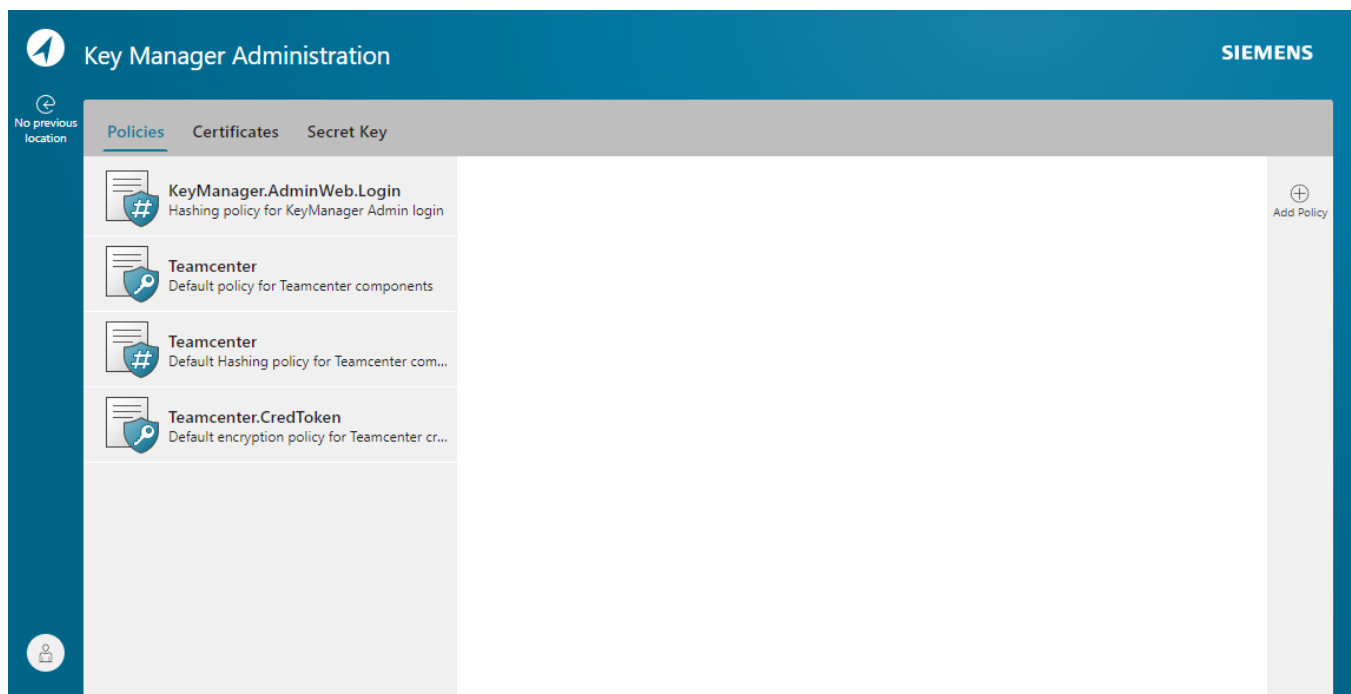
`https://<hostname>:<admin_web_server_port>`

- `<hostname>` is the name of the machine on which the Teamcenter Key Manager Administration web service is running.
- `<admin_web_server_port>` is **8204** unless changed when providing the Key Manager Server Component settings.
- The password is the one specified in Deployment Center for **Key Manager Administrator Password** when providing the **Key Manager Server Component** settings.

Note:

The first time you connect to the Key Manager Administration application, your browser may not authorize the self-signed CA Root certificate generated during the Key Manager installation. If so, follow your browser's specific steps for adding the security certificate to its store of trusted Root Certificate Authorities. These steps typically involve viewing the self-signed certificate, copying or exporting the certificate to a file, and then importing that certificate to your trusted Root Certificate Authorities store.

Once you have signed in, the Key Manager Administration application opens to the **Policies** tab as shown in the following example:



Manage policies

Key management policies control the manner in which keys are accessed and managed. Policies are defined for different uses of Teamcenter Key Manager by different Teamcenter components. Policies are uniquely identified by the component name in a hierarchical manner as follows:

Secret key policies

- Teamcenter
- Teamcenter.CredToken

Message digest (hashing) policies

- Teamcenter
- KeyManager.AdminWeb.Login

The Teamcenter Key Manager server performs and restricts requests based on these policies.

Separate types of policies are defined depending on the type of cryptography process. Each component may or may not use each of these types of processes, and may or may not define a component-specific policy for the process. If a component has no explicit policy defined for the process, the policy of the component's hierarchical parent is used for the request if possible. For example,

- If a component attempts to use a policy named "Teamcenter.ABC", and that policy is not defined in Key Manager, the parent policy "Teamcenter" is used.
- If a component attempts to use a policy named "NX", that policy is not defined in Key Manager, and there is no parent policy, an error is generated.

Key Manager supports symmetric encryption and hashing processes.

Create and manage your site's policies on the **Policies** tab of the Teamcenter Key Manager Administration application. Click the **Policies** tab to view a list of all of the defined policy instances sorted by component name.

Viewing a policy's details

On the **Policies** tab, each of the site's policies are listed. Click on a policy to view details about the policy. The list is updated after any policy change.

Creating a policy

1. With no policy selected, click **Add Policy** ⊕.
2. Several settings are displayed for you to specify the details of the policy. Select either **Secret Key** or **Message Digest** depending on the type of policy you are creating.

Specify the settings for the policy. Hover over the settings for descriptions and formatting requirements for values. You can cancel your policy creation at any time by clicking **Close**.

- When you have finished specifying the settings, click **Add Policy** to save the new policy. (**Add Policy** is not available until all required setting values are specified.) The new policy is added to the list of policies.

Editing and deleting policies

On the **Policies** tab, click on the policy in the you want to edit or delete.

To edit a policy, click **Edit Policy**  and change the settings.

You can delete policies that have no active keys generated against them. You cannot delete the policies delivered with Key Manager. Click **Delete Policy**  to delete the policy.

Manage secret keys

Manage your site's keys on the **Secret Key** tab of the Teamcenter Key Manager Administration application. Click the **Secret Key** tab to view a list of current and expired keys in the key store organized by component. Components can be expanded and collapsed. Click **Filter**  limit the types of keys listed:

- **All Keys** lists all keys in the secure store.
- **Latest Keys** lists only keys currently used for encryption and decryption for each component.
- **Active Keys** lists all keys that have not expired, including the most current keys and keys that may be used for decryption.
- **Expired Keys** lists keys that could be archived or deleted as they are no longer used for encryption or decryption.

Select a key to view its details.

Creating a key

You cannot manually create a key with the Key Manager Administration application. New keys are generated on demand according to policy settings. Keys can also be imported.

Exporting a key

Exchanging keys between Teamcenter sites is necessary when the sites exchange data. When exporting a key for use on a target key manager system, the key value is encrypted with a wrapping certificate created on that target system. This requires that a public certificate exists in the exporting key manager system's secure store. Before exporting a key from your system, export a wrapping certificate from the target system and import that certificate into your key store using the procedures in **Manage certificates**.

Select the key to export and click **Export Secret Key** 

Select a wrapping certificate created by the target system and save the exported key to an accessible directory.

Importing a key

Importing a key reads the definition of a key from a file exported by another key manager system and stores the key and attributes in the local secure store. To import a key exported from another system, you must first have provided the exporting system with a wrapping certificate in which the key is wrapped when exported. The imported key is decrypted using the private key corresponding to the certificate with which the key was wrapped and exported.

With no key selected, click **Import Secret Key** . Use **Choose File** to locate and open the key on your file system and click **Choose File** to import the key into the key store.

Deactivating (rolling over) a key

A key rolls over with a new key being generated when the old key's key lifetime is reached. You can also manually roll over keys that have been compromised or for which you have another reason for wanting a new key.

Select the key to be rolled over. Click **Roll Over**  and confirm the key deactivation.

The key is deactivated, a new key is generated, and the new key is used for future encrypting. The rolled over key is retained for future decrypting.

Archiving keys

Archiving a key removes it from the key store. Only expired keys can be archived. Archived keys can be reimported if needed at a later time.

Select the key to be archived. Click **Archive Secret Key** . The key is removed from the key store and archived in the key manager server data directory:

`INSTALL_DIR\KeyManagerServer\data\AdminWeb\Archive\policy_name`

Manage certificates

Securely export and import keys in encrypted form by wrapping the key using the destination site's public certificate. Doing so encrypts key information with the wrapping certificate of the destination site's secure store. Use the following procedures to work with wrapping certificates.

Click the **Certificates** tab to view a list of the certificates in the key store. Click **Filter**  limit the types of certificates listed:

- **All** lists wrapping certificates of all types.

- **Local Certificates** lists only wrapping certificates with public-private key pairs in the secure store that were generated on this key manager server.
- **Imported Certificates** lists only wrapping certificates without public-private key pairs in the key store that were imported from other systems.

In the **Certificates** list, click a certificate to view its details.

Creating a wrapping certificate

With no certificate selected, click **Create Certificate** .

Enter the fully-qualified name of the host on which you are running the Teamcenter Key Manager Administration application and click **Create Certificate** to generate and save the new certificate. The certificate is created in the store and appears in the list of certificates.

Exporting a wrapping certificate

Exporting a wrapping certificate to another key manager system enables that site to export keys you can then import to your key store.

Select a certificate to export, click **Export Certificate** , and save the exported certificate to an accessible directory. Provide this certificate to a site that will be exporting keys to your site for import.

Importing a wrapping certificate

Importing a wrapping certificate from another key manager system enables you to use the certificate to export keys to that key manager system.

1. With no key selected, click **Import Certificate** .
2. Enter the fully-qualified name of the host of the certificate in the **Subject CN** field.
3. Use **Choose File** to locate and open the certificate on your file system. Click **Upload** to import the certificate into the store.

Deleting a wrapping certificate

Click on a certificate. Click **Delete Certificate**  to delete the certificate from the store.

Manage a secondary key manager server

When both your primary and secondary key managers are running, changes made on the primary server are automatically synchronized with the secondary server. To account for possible network anomalies

or other communication interruptions, the secondary server periodically performs a full synchronization with the primary server.

Teamcenter satellites typically communicate with the primary key manager server to share keys and perform cryptography for features and applications. In the event communication with the primary server is interrupted, satellites automatically begin communicating with the secondary server. Communication with the secondary server continues until the primary server becomes available.

Monitor the current state of data on the secondary server by running the **Teamcenter Key Manager Administration application**. On the secondary server, the Teamcenter Key Manager Administration application gives a read-only view of the encryption keys and policies being managed by Teamcenter Key Manager. Run the **Teamcenter Key Manager Administration application** on the primary server to modify any keys or policies.

Manage your secondary key manager server using the following processes:

Manually synchronizing servers

The secondary server periodically performs a full synchronization with the primary server to ensure all changes on the primary server are reflected on the secondary server. Force a manual synchronization with the secondary key manager server as follows.

1. Run the Teamcenter Key Manager Administration application on the primary server.
2. Click **Synchronize** . The primary server is synchronized with the secondary server.

Changing the interval for fully synchronizing

By default, the secondary key manager is fully synchronized with the primary server every 1,200 seconds. To change this interval, adjust the value assigned to **KEY_MANAGER_SERVER_PERIODIC_SYNC_INTERVAL** in the following file on the primary key manager server:

```
%INSTALL_DIR%\KeyManagerServer\config\key_manager_server.properties
```

After changing the value, restart the primary key manager server process or daemon.

Viewing satellite logs

Teamcenter satellites log connection data to the following log file on the satellite machine:

```
%INSTALL_DIR%\KeyManagerSatellite\logs\KeyManager\process\KeyManager.log
```

Viewing server connection logs

Synchronization and server connection issues are logged to the following file on the primary key manager server machine:

```
%INSTALL_DIR%\KeyManagerServer\data\Synchronize\<machine_name>_sync.txt
```

Back up Teamcenter Key Manager data files

For sites configured to use the default network security services (NSS), data related to Teamcenter Key Manager operation is stored in the following Teamcenter Key Manager server machine installation directory. Ensure that you regularly back up the data in this directory as part of your site's broader backup and recovery plans.

```
$INSTALL_DIR\KeyManagerServer\data
```

Having a recent backup of this directory will aid in reestablishing your Teamcenter Key Manager installation in the case of a broader system data loss.

Sites using hardware security modules or custom NSS should back up their site's data as appropriate.

Start and stop Teamcenter Key Manager services

On Windows, Teamcenter Key Manager servers and satellites run automatically as Windows services. You can start, pause, and stop the following services as necessary using the Windows Services application. The services are named as follows:

Teamcenter Key Manager server (primary and secondary)

Teamcenter Key Manager Server <your_chosen_environment_name>

Teamcenter Key Manager satellite

Teamcenter Key Manager Satellite <your_chosen_environment_name>

Teamcenter Key Manager Administration application

Teamcenter Key Manager Admin Web <your_chosen_environment_name>

Teamcenter Key Manager root certificate authority server

Teamcenter Key Manager CA Server <your_chosen_environment_name>

Start and stop Teamcenter Key Manager daemons

On Linux, Teamcenter Key Manager servers and satellites run as daemons. Running these scripts requires root access. The server and satellite scripts are located in the following Teamcenter Key Manager installation directories respectively:

- `$INSTALL_DIR/KeyManagerServer/`

- `$INSTALL_DIR/KeyManagerSatellite/`

Each script has the following form:

```
script_name {-status | -start | -stop | -h}
```

Where

script_name

The script related to a particular daemon:

- **key_manager_server**

Controls **rc.key.manager.server.<your_env_name>**, the Teamcenter Key Manager server daemon (primary and secondary servers).

- **key_manager_ca_server**

Controls **rc.key.manager.ca.server.<your_env_name>**, the Teamcenter Key Manager Certificate Authority server daemon.

- **key_manager_admin**

rc.key.manager.admin.<your_env_name>, the Teamcenter Key Manager Administration application daemon.

- **key_manager_satellite**

rc.key.manager.satellite.<your_env_name>, the Teamcenter Key Manager satellite daemon.

-status

Displays the related daemon's current status.

-start

Starts the related daemon.

-stop

Stops the related daemon.

-h

Displays help for the script.

Unresponsive Teamcenter Key Manager satellite daemons

If a Teamcenter Key Manager satellite is not shut down properly on a Linux machine prior to a reboot, related Teamcenter applications connecting to the satellite may experience a hang (become unresponsive or unable to connect to the satellite). This may occur because a named pipe that is deleted

during a proper shutdown has not been deleted. When the Key Manager satellite restarts, it cannot create a new named pipe due to the existence of the old named pipe.

Manually delete the named pipe using the following steps:

1. Stop the Key Manager satellite.
2. Locate the old named pipe. The named pipe is defined with the **KEY_MANAGER_PIPE** property in *KeyManagerSatellite.properties*. The default location for the named pipe is */tmp/tctp*.
3. Delete the old named pipe.
4. Restart the Key Manager satellite.
5. Run the following command from the Key Manager satellite installation directory to verify that Key Manager is functioning properly:

```
%INSTALL_DIR%\KeyManagerSatellite\key_manager_client_test_connection
```

Install and uninstall Teamcenter Key Manager services

Teamcenter Key Manager server and satellites run as services on Windows. If you need to manually uninstall or reinstall the services, run the following batch files located in the local Teamcenter Key Manager installation directory (*INSTALL_DIR*).

Uninstall the key manager server service

```
%INSTALL_DIR%\KeyManagerServer\uninstall_key_manager_server_as_windows_service.bat
```

Uninstall the key manager satellite service

```
%INSTALL_DIR%\KeyManagerSatellite\uninstall_key_manager_satellite_as_windows_service.bat
```

Install the key manager server service

```
%INSTALL_DIR%\KeyManagerServer\install_key_manager_server_as_windows_service.bat
```

Install the key manager satellite service

```
%INSTALL_DIR%\KeyManagerSatellite\install_key_manager_satellite_as_windows_service.bat
```

Grant group access to a Teamcenter Key Manager satellite

By default, the Teamcenter Key Manager satellite can only be accessed by the operating system user that starts Key Manager satellite on the workstation. This user is typically the same operating system user that installs Key Manager satellite.

When Teamcenter Key Manager is enabled, for example, Key Manager satellite access is necessary to execute Teamcenter utilities on the satellite workstation. To allow additional users to execute

Teamcenter utilities on the workstation, these users must be added to an operating system group that is granted access to Key Manager satellite using Deployment Center.

Use the following procedure to grant permissions to other users to access the satellite. These steps must be performed by the user that installed Key Manager on the satellite workstation.

1. Create or use an existing operating system group containing the users that are to be granted group access to the Key Manager satellite.
2. Log onto Deployment Center.
3. Click the **Environments** tile to display the environments scanned by Deployment Center. Select the environment with the Key Manager satellite installed.
4. Click the **Deploy Software** tab. On the **Components** tab, choose the **Key Manager Satellite** component.
5. On the **Options** tab, click **Show all parameters** and navigate to **Group Permission Setting**. Check **Enable Group Permission?** and enter the name of the operating system group containing the users requiring access.
6. On the **Deploy** tab, generate the installation scripts. Review the **Deploy Instructions** information related to the generated installation scripts.
7. Follow the steps in *Run the deployment scripts* in the Deployment Center help collection to run the deployment scripts for the satellite.
8. If you are deploying Teamcenter Key Manager on Linux as a user without root privileges, log on as a user with root privileges and run the following script from the Teamcenter Key Manager installation directory (`$INSTALL_DIR`):

```
$INSTALL_DIR/KeyManagerSatellite/  
root_post_tasks_key_manager_satellite<id>.ksh
```

9. The Teamcenter Key Manager satellite is run as a services on Windows, and is started automatically. Ensure the service was successfully started as part of the deployment. If it is not running, start it using the steps in [Start and stop Teamcenter Key Manager services](#).

The Teamcenter Key Manager satellite is run as a daemon that may need to be started manually if Java is not installed on the local machine. Ensure the following daemon is started. If necessary, start it as described in [Start and stop Teamcenter Key Manager daemons](#).

```
rc.key.manager.satellite.<your_env_name>
```


21. Teamcenter server cryptography

Cryptography and FIPS compliance in Teamcenter server

Teamcenter can be configured to use FIPS (Federal Information Processing Standard) standards by using the FIPS-certified Teamcenter Cryptographic Module (TCM). When using the TCM, Teamcenter uses the FIPS PUB 140-2 standard to perform all cryptographic functions. The TCM provides FIPS-140-2 validated encryption, hashing, digital signatures, and random number generation.

All Teamcenter cryptographic functions are handled with the OpenSSL-based **TcCrypto** library.

Teamcenter Cryptographic Module (TCM)

The TCM is a software library providing an API for use by applications that require cryptographic security. The TCM is classified by FIPS 140-2 as a software module, multichip standalone module embodiment. The Siemens Digital Industries Software Teamcenter Cryptographic Module FIPS 140-2 security Policy (Certificate #2624) is available at <https://csrc.nist.gov>.

The TCM is a cryptographic engine library that is used only in conjunction with additional software. Aside from the use of the NIST-defined elliptic curves as trusted third party domain parameters, all other FIPS 186 - 3 assurances are outside the scope of the TCM, and are the responsibility of the calling process.

The TCM software version for this validation is provided in the Siemens Digital Industries Software Teamcenter Cryptographic Module FIPS 140-2 security Policy (Certificate #2624) available at <https://csrc.nist.gov>.

The TCM has the following characteristics:

- The physical cryptographic boundary is the general purpose computer on which the TCM is installed.
- The logical cryptographic boundary of the TCM is the **TcCryptoFips** object module. It is designed to be a shared library.
- The **TcCryptoFips** library communicates only with the calling **TcCrypto** library, which is responsible for invoking the TCM services in FIPS mode.
- The TCM requires the following initialization sequence.
 - Upon load, the **TcCryptoFips** library runs the integrity test followed by the self tests.
 - When a calling application requests the module to be in FIPS mode, the TCM invokes **FIPS_mode_set()** implemented in the **TcCryptoFips** library. Doing so verifies the user password and reruns the algorithms test, returning a 1 for success or 0 for failure.

If **FIPS_mode_set()** fails, all subsequent cryptographic services in the FIPS module fail. The module can later be initialized in domestic mode and an application can test whether FIPS mode has been successfully enabled.

See section 9.5 in Implementation Guidance for FIPS 140-2 and the Cryptographic Module Validation Program available at <https://csrc.nist.gov> for further information on initialization sequences.

Configure Teamcenter to use FIPS standards

Configure Teamcenter to use FIPS standards by setting the **TCCRYPTO_FORCE_FIPS_MODE** environment variable to a value of `true`.

You can also define and set **TCCRYPTO_FORCE_FIPS_MODE** in the file *tc_profilevars.bat* (Windows) or *tc_profilevars.sh* (Linux).

22. Managing secrets using Teamcenter vault

Teamcenter vault overview

The Teamcenter vault manages cross-application and infrastructure secrets (passwords, keys, certificates, and so on) for your Teamcenter environment. Teamcenter vault manages secrets using third-party secrets-management solutions. By default Teamcenter supports HashiCorp Vault's free and Enterprise editions. Refer to vaultproject.io for additional information about HashiCorp Vault.

See [Install and enable the Teamcenter vault](#) to customize your Teamcenter vault installation. You can optionally enable high availability of the Teamcenter vault to help ensure vault operation with minimal downtime during failures and outages. See [Enable Teamcenter vault high availability](#).

Install and enable the Teamcenter vault

Install Teamcenter as described in *Teamcenter Installation Using Deployment Center*, ensuring the following items are configured appropriately in Deployment Center.

- On the **Components** tab, ensure **HTTPS Config** is configured. Distributed environments have multiple **HTTPS Config** components.
- On the **Components** tab, select the **Teamcenter Vault** component and confirm the following settings.
 - Verify **Machine Name** is set to the name of a machine specified by an **HTTPS Config** component.
 - By default, **Vault Configuration** is set to **Open Source**. This setting installs the Teamcenter implementation of HashiCorp's Vault free edition. Verify that **Port** and **Cluster Port** are set to available open ports on the specified machine.

If your site uses HashiCorp's Vault Enterprise, change the **Vault Configuration** value to **Enterprise Vault**. Using HashiCorp's Enterprise Vault, create a Teamcenter namespace and enter the related values in the Deployment Center **Vault Configuration** section.

For further information on configuring the Teamcenter Vault component in Deployment Center, see the Teamcenter Vault section of *Create a Teamcenter environment using Deployment Center*.

Start and stop Teamcenter vault services and daemons

Use the following processes to start and stop the Teamcenter vault service or daemon if necessary.

Manually starting and stopping the Windows Teamcenter vault service

Start and stop the vault service using the Windows Services application. The service name is a combination of the vault service name ("Teamcenter Vault" by default) and the Deployment Center environment name.

Manually starting and stopping the Linux Teamcenter vault daemon

Start and stop the vault daemon service using the `TC_ROOT/tc_vault/rc.tc.vault_env` script, where *env* is the Deployment Center environment name. For example, to start the vault daemon for the Site1 environment, use the following command from the `tc_vault` directory:

```
rc.tc.vault_Site1 start
```

To stop the vault daemon, use the following command:

```
rc.tc.vault_Site1 stop
```

Adjusting vault connection tolerances

If your site is experiencing issues with Teamcenter applications accessing Teamcenter vault, adjust the following vault-related tolerance parameters.

Open the `TC_ROOT\tc_vault\tls\vault_connector.properties` file in a text editor and add or adjust the values of the following parameters as necessary. After doing so, **restart the Teamcenter vault service or daemon**.

VAULT_CONNECTION_TIMEOUT_IN_SEC

Sets the timeout (in seconds) to establish the connection with the vault.
Default value: **30**

VAULT_REQUEST_TIMEOUT_IN_SEC

Sets the timeout (in seconds) for the entire request-response operation.
Default value: **10**

VAULT_REQUEST_RETRIES

Number of times to retry a vault request.
Default value: **6**

VAULT_REQUEST_RETRIES_INTERVAL_IN_SEC

Interval (in seconds) before retrying a vault request.
Default value: **5**

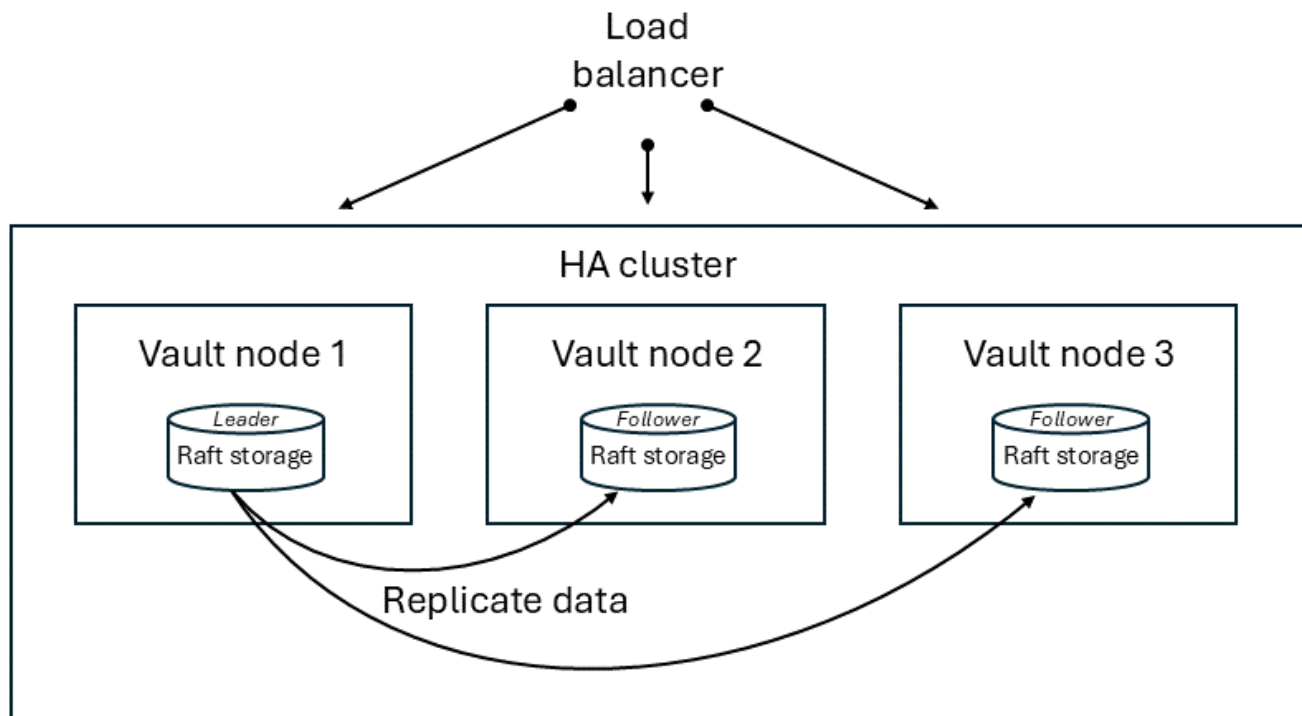
VAULT_HEALTH_CHECK_TIMEOUT_IN_MINS

Set the timeout (in minutes) to wait for TcVault to be available.
Default value: **3**

Enable Teamcenter vault high availability

Teamcenter vault high availability (HA) helps ensure the vault operates at a high level without interruption during failures and outages through use of redundant vault nodes.

In HA mode, vault instances (nodes) in a cluster run in active and standby states. Within the cluster, one vault instance runs in the active state and handles all requests. Each other vault instance in the cluster redirects requests to the active instance.



Prerequisites

- Teamcenter is installed with Teamcenter vault configured. See [Install and enable the Teamcenter vault](#)
- An understanding of Teamcenter vault, an implementation of HashiCorp Vault. Refer to vaultproject.io for additional information.
- A basic understanding of your load balancer is necessary. The following procedure uses nginx as the sample load balancer. See nginx.org for additional information.

Procedure

Use the following process to configure Teamcenter vault for high availability. nginx is used as an example load balancer. Apply similar approached with other load balancers.

1. Update *vault_config.hcl* on vault node1 using the following steps.
 - a. Stop the Teamcenter vault service or daemon on node 1. See [Start and stop Teamcenter vault services and daemons](#).
 - b. Copy the file *TC_ROOT\tc_vault\tls\rootCACertificate.pem* to the *TC_ROOT\tc_vault* directory.
 - c. Open *TC_ROOT\tc_vault\vault_config.hcl* in a text editor. Create *retry_join* sections using the following form:

```
storage "raft" {
  path="storage"
  retry_join {
    leader_api_addr = "[machine_2 https_addr]"
    leader_ca_cert_file = "[machine_1 root_ca_cert]"
  }
  retry_join {
    leader_api_addr = "[machine_3 https_addr]"
    leader_ca_cert_file = "[machine_1 root_ca_cert]"
  }
  ...
}
```

Siemens Digital Industries Software recommends configuring an odd number of vault nodes for best performance.

Example:

This example illustrates an updated *vault_config.hcl* file on the vault node 1 machine. The configuration uses vault nodes with the following HTTPS addresses:

Node	HTTPS address
Vault node 1 machine	VC6S009
Vault node 2 machine	VC6S008
Vault node 3 machine	VC6S007

```
storage "raft" {
  path="storage"
  retry_join {
    leader_api_addr = "https://VC6S007:8200"
    leader_ca_cert_file = "C:/apps/tc/TR/tc_vault/rootCACertificate.pem"
  }
  retry_join {
    leader_api_addr = "https://VC6S008:8200"
```

```

    leader_ca_cert_file = "C:/apps/tc/TR/tc_vault/rootCACertificate.pem"
  }
}

listener "tcp" {
  address="[:,]:8200"
  tls_disable="false"
  tls_cert_file="C:/apps/tc/TR/tc_vault/
VC6S009.NET.YOUR.COMPANY.COM_host_certificate.pem"
  tls_key_file="C:/apps/tc/TR/tc_vault/
VC6S009.NET.YOUR.COMPANY.COM_host_key.pem"
  cluster_address="[:,]:8201"
}

api_addr="https://VC6S009:8200"
cluster_addr="https://VC6S009:8201"

disable_mlock="true"

```

2. Use the following steps to copy the configured node 1 Teamcenter vault directory to the additional vault node machines, update each additional node's *vault.hcl* file, and start the vault services (or daemons) on each additional node.
 - a. Copy the entire *TC_ROOT\tc_vault\tls* directory from the vault node 1 machine to the *TC_ROOT* directory on each of the additional vault node machines.
 - b. On all of the additional vault node machines, delete the files and folders in each machine's *TC_ROOT\tc_vault\tls\storage* directory.
 - c. In the *vault_config.hcl* file on each node, update the values for each tls certificate, key file, *retry_join*, *api_addr*, and *cluster_addr*. Find your root CA certificate, host certificate, and private key .pem files in each node's *TC_ROOT\https_config* directory.

Example:

This configuration example illustrates the vault node 2 *vault_config.hcl* file changes. Note that this node's file contains *retry_join* sections for vault node 1 and vault node 3. This example uses the same HTTPS addresses for nodes as in the earlier example.

```

storage "raft" {
  path="storage"
  retry_join {
    leader_api_addr = "https://VC6S007:8200"
    leader_ca_cert_file = "C:/apps/tc/TR/tc_vault/rootCACertificate.pem"
  }
}

```

```

retry_join {
  leader_api_addr = "https://VC6S009:8200"
  leader_ca_cert_file = "C:/apps/tc/TR/tc_vault/rootCACertificate.pem"
}

listener "tcp" {
  address="[:,]:8200"
  tls_disable="false"
  tls_cert_file="C:/apps/tc/TR-Vault/tc_vault/
VC6S008.NET.YOUR.COMPANY.COM_host_certificate.pem"
  tls_key_file="C:/apps/tc/TR-Vault/tc_vault/
VC6S008.NET.YOUR.COMPANY.COM_host_key.pem"
  cluster_address="[:,]:8201"
}

api_addr="https://VC6S008:8200"
cluster_addr="https://VC6S008:8201"

disable_mlock="true"

```

Example:

This configuration example illustrates the vault node 3 *vault_config.hcl* file changes. Note that this node's file contains `retry_join` sections for vault node 1 and vault node 2. This example uses the same HTTPS addresses for nodes as in the earlier examples.

```

storage "raft" {
  path="storage"
  retry_join {
    leader_api_addr = "https://VC6S009:8200"
    leader_ca_cert_file = "C:/apps/tc/TR/tc_vault/rootCACertificate.pem"
  }
  retry_join {
    leader_api_addr = "https://VC6S008:8200"
    leader_ca_cert_file = "C:/apps/tc/TR/tc_vault/rootCACertificate.pem"
  }
}

listener "tcp" {
  address="[:,]:8200"
  tls_disable="false"
  tls_cert_file="C:/apps/tc/TR-Vault/tc_vault/
VC6S007.NET.YOUR.COMPANY.COM_host_certificate.pem"
  tls_key_file="C:/apps/tc/TR-Vault/tc_vault/
VC6S007.NET.YOUR.COMPANY.COM_host_key.pem"
  cluster_address="[:,]:8201"
}

```

```

}

api_addr="https://VC6S007:8200"
cluster_addr="https://VC6S007:8201"

disable_mlock="true"

```

- d. Update each target machine's (nodes 2 and 3 in this example) `TC_ROOT\vc_vault\vault_installer.properties` as follows.

Note:

These changes require **VAULT_KEYSTORE_PASSWORD** and **VAULT_TRUSTSTORE_PASSWORD** .p12 keystore files. To properly generate these files for each machine (nodes 2 and 3), use Deployment Center to create a temporary environment with which you then install Teamcenter vault on simulated nodes 2 and 3. Use a temporary installation folder for the Teamcenter vault component, as this vault will not be used in production. For example, `c:\temp_vault`.

Run the deployment script to install the temporary vaults. Doing so generates the `vault_installer.properties` files used to in this step to install the Teamcenter vault service.

No changes are required in node 1's `vault_installer.properties` file. It is provided here as an example:

```

VAULT_ROTATE_SECRET=false
VAULT_APPROLE_AUTH=TcV:V0:VGU2YGY2YjVgew1gMGZ5YDFsYnk2MW1leTFjZmV
nbGQxZmZnbA==
VAULT_AUTH_STRATEGY=APP_ROLE_AUTH
VAULT_KEYSTORE=C:/apps/tc/TR/https_config/VC6S009_keystore.p12
VAULT_CLUSTER_URL=https://VC6S009:8201
VAULT_CAPEM=C:/apps/tc/TR/tc_vault_tls/root-ca.pem
VAULT_TRUSTSTORE=C:/apps/tc/TR/https_config/
VC6S009_truststore.p12
VAULT_TRUSTSTORE_PASSWORD=TcV:V0:TDwodTw2AnsM
VAULT_AUTH_PATH=tc
VAULT_URL=https://VC6S009:8200
VAULT_TLS_ROOT=C:/apps/tc/TR/tc_vault_tls
VAULT_APPROLE_ID=TcV:V0:R3ImdSR2cXAhanBydHRqcyQjcWomI34ianV/f3R/
JHNydXVwcQ==
VAULT_KEYSTORE_PASSWORD=TcV:V0:VmkRciIyEGN3

```

Following is node 2's `vault_installer.properties` file with differences from the node 1 `vault_installer.properties` file highlighted:

```

VAULT_ROTATE_SECRET=false
VAULT_APPROLE_AUTH=TcV:V0:VGU2YGY2YjVgew1gMGZ5YDFsYnk2MW1leTFjZmV
nbGQxZmZnbA==

```

```

VAULT_AUTH_STRATEGY=APP_ROLE_AUTH
VAULT_KEYSTORE=c:/temp_vault/https_config/VC6S008_keystore.p12
VAULT_CLUSTER_URL=https://VC6S008:8201
VAULT_CAPEM=C:/apps/tc/TR/tc_vault_tls/root-ca.pem
VAULT_TRUSTSTORE=c:/temp_vault/https_config/VC6S008_truststore.p12
VAULT_TRUSTSTORE_PASSWORD=TcV:V0:QzcPFnMoEXZi
VAULT_AUTH_PATH=tc
VAULT_URL=https://VC6S008:8200
VAULT_TLS_ROOT=C:/apps/tc/TR/tc_vault_tls
VAULT_APPROLE_ID=TcV:V0:R3ImdSR2cxAhanBydHRqcyQjcWomI34ianV/f3R/
JHNydXVwcQ==
VAULT_KEYSTORE_PASSWORD=TcV:V0:UCoOKTlgHmlx

```

Following is node 3's *vault_installer.properties* file with differences from the node 1 *vault_installer.properties* file highlighted:

```

VAULT_ROTATE_SECRET=false
VAULT_APPROLE_AUTH=TcV:V0:VGU2YGY2YjVgEW1gMGZ5YDFsYnk2MW1leTFjZmV
nbGQxZmZnbA==
VAULT_AUTH_STRATEGY=APP_ROLE_AUTH
VAULT_KEYSTORE=c:/temp_vault/https_config/vc6s007_keystore.p12
VAULT_CLUSTER_URL=https://VC6S007:8201
VAULT_CAPEM=C:/apps/tc/TR/tc_vault_tls/root-ca.pem
VAULT_TRUSTSTORE=c:/temp_vault/https_config/vc6s007_truststore.p12
VAULT_TRUSTSTORE_PASSWORD=TcV:V0:TngCGG87BXxx
VAULT_AUTH_PATH=tc
VAULT_URL=https://VC6S007:8200
VAULT_TLS_ROOT=C:/apps/tc/TR/tc_vault_tls
VAULT_APPROLE_ID=TcV:V0:R3ImdSR2cxAhanBydHRqcyQjcWomI34ianV/f3R/
JHNydXVwcQ==
VAULT_KEYSTORE_PASSWORD=TcV:V0:Qy87NSQ6Gnp8

```

- e. (Windows only.) For each additional node, update the **VAULT_BASE_LOCATION** and **VAULT_SERVICE_NAME** values in *TC_ROOT\tc_vault\installTcVaultService.bat* with your actual installation directories and service names. Then, open an administrator command prompt and run the file to register the vault Windows service.

Example:

Vault Node 2 *installTcVaultService.bat*:

```

...
set VAULT_BASE_LOCATION=D:\Teamcenter\tr\tc_vault
set VAULT_BASE_LOCATION=%VAULT_BASE_LOCATION:/=\%
rem echo %VAULT_BASE_LOCATION%

set VAULT_SERVICE_NAME=Teamcenter_Vault_Service_HA_Node_2

```



```
if "%VAULT_SERVICE_NAME%" == "" (
    set VAULT_SERVICE_NAME= Teamcenter_Vault_Service_HA_Node_2
    ...
```

Example:

Vault Node 3 *installTcVaultService.bat*

```
...
set VAULT_BASE_LOCATION=D:\Teamcenter\tr\trc_vault
set VAULT_BASE_LOCATION=%VAULT_BASE_LOCATION: /=\%
rem echo %VAULT_BASE_LOCATION%

set VAULT_SERVICE_NAME=Teamcenter_Vault_Service_HA_Node_3
if "%VAULT_SERVICE_NAME%" == "" (
    set VAULT_SERVICE_NAME= Teamcenter_Vault_Service_HA_Node_3
    ...
```

- f. (Linux only.) For each additional node, update the TC_VAULT_DIR and TC_VAULT_SERVICE_NAME values in *TC_ROOT\trc_vault\rc.tc.vault*. Then, start the vault daemon.
3. Use the following steps to configure the nginx load balancer.
 - a. Download the most recent version of the nginx load balancer from nginx.org. Unzip the download to a location on node 1.
 - b. Create an *ssl* directory in the nginx installation directory (*nginx_install*). Copy the machine host certificate, private key, and root CA *.pem* files to the *ssl* directory to enable the TLS server.
 - c. Open *nginx_install\conf\nginx.conf* in a text editor and add the following entries to the file. Use the following example to update the file with values specific to your site.
 - `upstream vault_ha_cluster` section with machine names.
 - `server_name`, `ssl_certificate`, `ssl_certificate_key`, `ssl_trusted_certificate`, `proxy_pass`, `proxy_ssl_certificate`, `proxy_ssl_certificate_key`, and `proxy_ssl_trusted_certificate` parameters and values.

Example:

In the following example, the nginx application is installed in the Vault node 1 Machine (VC6S009) with the following entries added to the *nginx.conf* file. The configuration uses vault nodes with the following HTTPS addresses:

Node	HTTPS address
Vault node 1 machine	VC6S009
Vault node 2 machine	VC6S008
Vault node 3 machine	VC6S007

```

upstream vault_ha_cluster {
    server vc6s009:8200;
    server vc6s008:8200;
    server vc6s007:8200;
}

# HTTPS server
#
server {
    listen                8443 ssl;
    server_name vc6s009.net.your.company.com;

    ssl_certificate      D:/apps/nginx-1.26.1/ssl/
VC6S009.NET.YOUR.COMPANY.COM_host_certificate.pem;
    ssl_certificate_key  D:/apps/nginx-1.26.1/ssl/
VC6S009.NET.YOUR.COMPANY.COM_host_key.pem;
    ssl_trusted_certificate D:/apps/nginx-1.26.1/ssl/rootCACertificate.pem;

    ssl_session_cache    shared:SSL:1m;
    ssl_session_timeout  5m;

    #    ssl_ciphers  HIGH:!aNULL:!MD5;
    #    ssl_prefer_server_ciphers  on;

    location / {
        proxy_pass https://vault_ha_cluster;
        proxy_ssl_certificate  D:/apps/nginx-1.26.1/ssl/
VC6S009.NET.YOUR.COMPANY.COM_host_certificate.pem;
        proxy_ssl_certificate_key  D:/apps/nginx-1.26.1/ssl/
VC6S009.NET.YOUR.COMPANY.COM_host_key.pem;
        proxy_ssl_trusted_certificate D:/apps/nginx-1.26.1/ssl/
rootCACertificate.pem;
    }
}

```

- d. Start nginx. For example,

```
d:\apps\nginx-1.26.1>start nginx.exe
```

```
d:\apps\nginx-1.26.1>nginx.exe -s reload
```

4. In a text editor, open the *TC_ROOT\tc_vault\tls\vault_connector.properties* file and set **VAULT_URL** to the load balancer URL. The URL has the form **https://load-balancer-machine-host:port**, for example,

```
VAULT_URL=https://vc6s015.net.your.company.com:8443
```

5. Check the vault status on each node using the following steps.

- From a Teamcenter command prompt navigate to the node's *TC_ROOT\tc_vault* folder.
- Set the **VAULT_ADDR** and **VAULT_CACERT** environment variables as appropriate for your environments. Continuing the examples in this procedure, the node values are set as follows:

Node 1:

```
VAULT_ADDR=https://VC6S009:8200
VAULT_CACERT=rootCACertificate.pen
```

Node 2:

```
VAULT_ADDR=https://VC6S008:8200
VAULT_CACERT=rootCACertificate.pen
```

Node 3:

```
VAULT_ADDR=https://VC6S007:8200
VAULT_CACERT=rootCACertificate.pen
```

- Run **vault status** on each node. Verify the following status values:

Key	Value
HA Enabled	true
Cluster Name	Same value on each node.
Cluster ID	Same value on each node.
HA Mode	One node has this key set to active . Each other node has this key set to standby .

6. Restart the Teamcenter vault service or daemon on each node. See [Start and stop Teamcenter vault services and daemons](#).

Updating vault passwords

Use the following process to update passwords stored in the Teamcenter vault.

1. Run Deployment Center, open your environment, and click the **Components** tab.
2. In the **Selected Components** list, select the component for which you want to change the password. For example, if you want to change server manager database password, select the Server Manager Cluster Configuration component.
3. In the right panel, update and confirm the new password.
4. Click **Save Component Settings** when complete.
5. Generate and deploy the installation script.

Generate new vault keys

If you encounter a security issue, you can generate new Teamcenter Vault authentication secrets, keys, and certificates.

Run Deployment Center and ensure the following items are configured appropriately. Then, deploy the scripts created using Deployment Center.

1. In the **Selected Components** list on the **Components** tab, select the **Teamcenter Vault** component.
2. Click **Show All Parameters** .
3. Ensure **Rotate Authentication Parameters?** is checked.
4. Click **Save Component Settings** when complete.
5. Generate and deploy the installation script.

Back up and restore Teamcenter vault data

Teamcenter vault data is backed up when you implement **the backup strategy recommended by Siemens Digital Industries Software** both before the cited Teamcenter environment-related actions and the periodically.

By default, Teamcenter vault manages secrets using HashiCorp Vault secrets management solutions. With this secrets implementation, Siemens Digital Industries Software recommends you review and implement the backup approaches described in **the documentation available from Hashicorp** if you wish to implement an additional vault-focused backup strategy.

23. Backing up and recovering a Teamcenter environment

Backing up a Teamcenter environment

Caution:

Failure to back up all environment components runs the risk of data loss in the event of the need to restore the environment.

Backup software requirements

Third-party software is used to back up and restore files in volumes. Ensure the user running the third-party backup software is the same user as the volume owner.

When to back up

In addition to the normal backup of information for purposes of disaster recovery, back up the Teamcenter environment before performing any of the following actions:

- Installing or updating Teamcenter software
- Changing the machine environment, such as updating operating system software or 3rd party software
- Changing the data model, especially by deploying a new or modified template
- Other tasks that modify a particular directory in the installation

Example:

The addition of another FSC modifies the `TC_ROOT\fsc` directory. In such cases, that directory and any other related dependencies (database, volumes, and so on) should also be backed up.

Components of a Teamcenter environment backup

- For each configured database:
 - Oracle, Microsoft SQL Server, or Aurora database
 - `TC_DATA` directory and its contents
 - Teamcenter volume directories

- On each workstation where Teamcenter is installed:
 - `TC_ROOT\install` directory
 - `TC_ROOT\bmide` directory and any custom Business Modeler IDE project folders stored in a location other than `TC_ROOT\bmide`
 - `TC_ROOT\fossrepo` directory
 - `TC_ROOT\tc_vault` directory
 - `TC_ROOT\vault_tls` directory
 - Deployment Center installation directory
- Depending on the environment, additional components should be backed up. For example:
 - Dispatcher root directory (if updating the dispatcher configuration)
 - `TC_ROOT\fsc` (if updating the FSC configuration)

Tip:

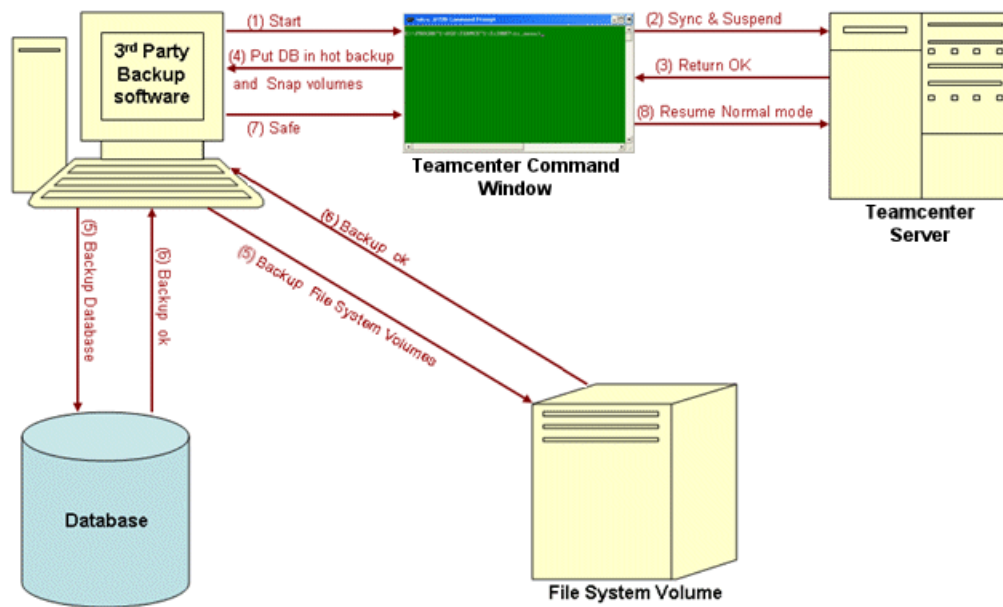
In the case of Teamcenter environment components installed in virtual machines, taking a snapshot of the affected virtual machines is also a viable means of backup.

Overview of the integrated backup and recovery process

Using the **backup_modes** utility to place Teamcenter in an appropriate operation mode, an administrator can allow Teamcenter to operate continually during online backup of metadata and files by third-party backup systems.

Teamcenter can operate in three volume backup modes: read-only, blobby volume, and normal.

Mode	Description
Read-only mode	Teamcenter pauses writing files to the volume during backup.
Blobby volume mode	Teamcenter writes to a temporary volume while third-party backup software takes a snapshot of the normal volume data, thus allowing continuous availability.
Normal mode	Teamcenter writes to the normal volume.



The following steps describe the process flow of a third-party integrated backup. Note that before running the integrated backup, the administrator should terminate Teamcenter sessions as described in *Back up new installations (Linux)* or *Back up new installations (Windows)*.

1. The third-party backup software requests that Teamcenter freeze all operations on its file system volumes.

Note:

The method of the request depends on how the third-party backup software is integrated with Teamcenter. In a loosely coupled integration, the request can be a reminder or email to the system administrator to begin the backup process. A tightly integrated system can trigger the read-only mode using the **backup_modes** command line utility.

To set the backup mode, use the **-m** (mode) argument in the **backup_modes** utility with the following options: **rdonly**, **normal**, and **blobby**. Use the **current** option to show the current mode in effect.

2. Teamcenter starts database and file system volume synchronization by ensuring there are no open writes to volumes. (The system pauses until all open writes are completed, and suspends future writes by putting file system volumes in read-only mode.)
3. Teamcenter returns an **OK** message after a successful math and metadata synchronization.
4. The third-party backup software puts the database in hot backup mode and creates a snapshot of the file system.
5. Third-party storage management systems start the backup of database and file system volumes.

Optionally, during the backup, the third-party software can request that Teamcenter operate in blobby volume mode. The blobby volume (a temporary file system area) serves as an alternate volume location for file writes during the hot backup, allowing for continuous availability. The blobby mode can be triggered using the **backup_modes** command line utility.

6. The third-party backup software completes the backup operation of database and file system volumes.
7. The third-party backup software requests that Teamcenter resume normal mode. The contents under blobby volumes are moved back to the original volume location.

Normal mode can be triggered using the **backup_modes** command line utility.

8. Teamcenter resumes normal mode.

Oracle Recovery Manager (RMAN)

Introduction to the Oracle Recovery Manager (RMAN)

Siemens Digital Industries Software recommends the Oracle Recovery Manager (RMAN) product be used with the Teamcenter integrated backup application. RMAN is an Oracle utility that backs up, restores, and recovers database files. This is a feature of the Oracle database server and does not require separate installation. Recovery Manager uses database server sessions to perform backup and recovery operations. It stores metadata about its operations in the control file of the target database, and optionally, in a recovery catalog schema in an Oracle database. You can invoke RMAN as a command line executable from the operating system prompt or use some RMAN features through the Enterprise Manager interface.

The features of RMAN are available using the Oracle Backup Manager interface. This is a command line interface, similar to **SQL*DBA**. It provides a powerful operating-system-independent scripting language and works in interactive or batch mode. The RMAN environment consists of the utilities and databases that play roles in a backup and recovery strategy. A typical RMAN setup utilizes the following components:

- RMAN executable
- Target database
- Recovery catalog database
- Media management software

Of these components, only the RMAN executable and target database are required. RMAN automatically stores its metadata in the target database control file, so the recovery catalog database is optional. Siemens Digital Industries Software recommends maintaining a recovery catalog. If you create a catalog on a separate machine and the production machine fails completely, you have all the restore and recovery information you need in the catalog.

When configuring backup and recovery, you must specify all of the following items to ensure a complete recovery:

- The database (such as Oracle and MS SQL).
- All database volumes.
- The `TC_DATA` directory.
- The `TC_ROOT\install` directory, which stores configuration data.
- The `TC_ROOT\bmide` directory (if the Business Modeler IDE is installed). This directory can contain database templates and custom templates under project folders.
- All local Business Modeler IDE project folders, including project folders within source control management (SCM) systems.

You can hot backup the database and volumes. You must cold backup the remaining items.

Benefits of RMAN

The following table lists a comparison between the Oracle Recovery Manager (RMAN) and user-managed methods.

Recovery Manager	User-managed method
Uses a media management API so that RMAN works seamlessly with third-party media management software. More than 20 vendors support the API.	Does not have support of a published API.
When backing up online files, RMAN rereads fractured data blocks to get a consistent read. You do not need to place online tablespaces in backup mode when performing backups.	Requires placing online tablespaces in backup mode before backing them up and then taking the tablespaces out of this mode after the backup is complete. Serious database performance and manageability problems can occur if you neglect to take tablespaces out of backup mode after an online backup is complete.
Performs incremental backups, which back up only those data blocks that changed after a previous backup. You can recover the database using incremental backups, which means that you can recover a NOARCHIVELOG database. However, you can only take incremental	Backs up all blocks, not just the changed blocks. Does not allow you to recover a NOARCHIVELOG database.

Recovery Manager	User-managed method
backups of a NOARCHIVELOG database after a consistent shutdown.	
Computes checksums for each block during a backup and checks for corrupt blocks when backing up or restoring. Many of the integrity checks that are normally performed when executing SQL are also performed when backing up or restoring.	Does not provide error checking.
Omits never-used blocks from datafile backups so that only data blocks that have been written to are included in a backup.	Includes all data blocks, regardless of whether they contain data.
Stores RMAN scripts in the recovery catalog.	Requires storage and maintenance of operating system-based scripts.
Allows you to easily create a duplicate of the production database for testing purposes or easily create or back up a standby database.	Requires you to follow a complicated procedure when creating a test or standby database.
Performs checks to determine whether backups on disk or in the media catalog are still available.	Requires you to locate and test backups manually.
Performs automatic parallelization of backup and restore operations.	Requires you to parallelize manually by determining which files you need to back up and then issuing operating system commands in parallel.
Tests whether files can be backed up or restored without actually performing the backup or restore.	Requires you to actually restore backup files before you can perform a trial recovery of the backups.
Performs archived log failover automatically. If RMAN discovers a corrupt or missing log during a backup, it considers all logs and log copies listed in the repository as alternative candidates for the backup.	Cannot failover to an alternative archived log if the backup encounters a problem.
Uses the repository to report on crucial information, including: Database schema at a specified time.	Does not include any reporting functionality.

Recovery Manager**User-managed method**

- Files requiring backup.
- Files that have not been backed up in a specified number of days.
- Backups that can be deleted because they are redundant or cannot be used for recovery.
- Current RMAN persistent settings

Features of RMAN

Feature	Description
Incremental backups	Up to four levels; level 0 and levels 1 and 4 .
Corrupt block detection	During backup: • v\$backup_corruption, v\$copy_corruption • Also reported in the databases alert log and trace files. Restore
Easy management	Distributing database backups, resources, and recoveries across clustered nodes in an Oracle parallel server.
Performance	• Automatic parallelization of backup, restore, and recovery. • Multiplexing prevents flooding any one file with reads and writes while keeping a tape drive streaming. • Backups can be restricted to limit reads per file, per second to avoid interfering with OLTP work. • No generation of extra redo during open database backups. • Easy backup of archived redo logs.
Limit file size	• Limits number of open files.

Feature	Description
	• Size of backup piece.
Recovery catalog	Automates restore and recovery operations.
Selective backups	Backs up an entire database, selected tablespaces, or selected data files.

ARCHIVELOG mode considerations

Running an Oracle database in **ARCHIVELOG** mode is necessary in 24x7 environments. If the archive log destination runs out of space, the database enters into freeze mode until free space is available in the destination directory. The immediate reaction is to delete some of the archive log files. Deleting archive redo logs creates holes in the archived log sequence. This can cause database recovery to fail. Use the following procedures to avoid inadvertently deleting archived logs.

Note:

Oracle documentation should be consulted for exact details of this operation. These procedures are offered as solutions to be considered, and may not be best for all environments.

- Redirect the archive log destination

Maintain two archive log destinations: primary and secondary. Once the primary log is filled to 85 or 90%, a switchover to secondary destination can be performed and vice versa. After switch over, the archived logs in the primary destination can be backed up and subsequently purged from the disk.

- Move archive logs to a temporary directory

Once the archive logs are moved to a temporary directory, Oracle will begin functioning again. Backup the archives logs in the archive log destination directory, and temporary directory, and subsequently purge them to release space.

- Selectively delete oldest archived logs

This is the last resort. List the logs based on time stamp, and selectively delete the oldest archived logs that have already been backed up. (Ensure you back it up before manual delete.) The best practice is to perform Oracle database backups at regular intervals, which can be used to ensure complete recovery while using minimal space.

Restoring purged files

Single file recovery (SFR)

Single file recovery (SFR) allows users to easily search for and restore purged versions of files from the backup medium. This feature also helps restore the files from the backup if accidentally deleted by a user. The scope of SFR is limited to restoring files from your backup medium.

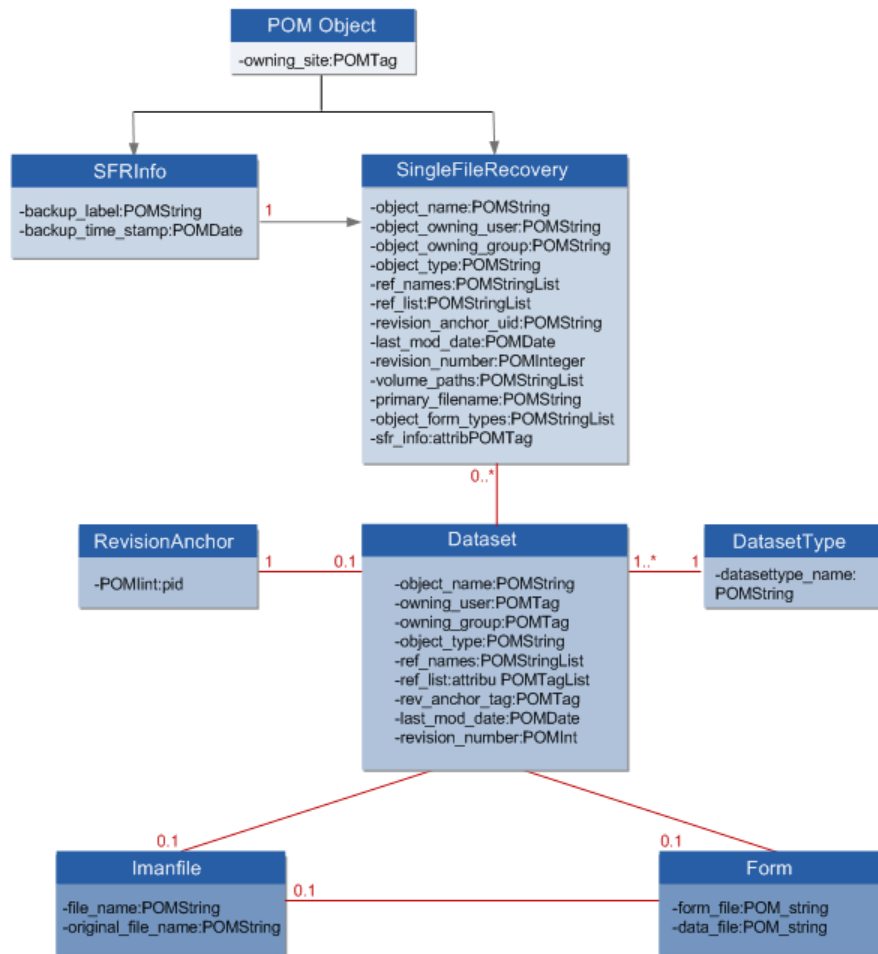
In Teamcenter, files revised beyond the set revision limit (the default is 3) are eclipsed. These eclipsed files are stored in the volume and are no longer referenced by the dataset once the revision limit is reached. A typical day-to-day backup preserves the revision limit versions of the file in the Teamcenter backup volumes. These files cannot be referenced in Teamcenter but are stored on backup media. Because the files are no longer associated in Teamcenter, it is tedious to manually bring the file back into Teamcenter. Rather than performing this task manually, Siemens Digital Industries Software recommends using SFR to recover a single file.

SFR uses File Management System (FMS) to search for and restore files within the time limits specified by the **TC_sfr_recovery_interval** and **TC_sfr_process_life_time** preferences. The third-party backup software recovers the purged versions of files from the Teamcenter volume. If the file is found, it is imported into Teamcenter as a new dataset and placed in the user's **Newstuff** folder. If the file is not found under the Teamcenter volume location, even after the maximum time duration specified by the **TC_sfr_process_life_time** preference, a message appears user stating that the file cannot be recovered.

Single file recovery object model

SFR contains several classes and tables. These tables are installed as a part of the Teamcenter integrated backup application. SFR also derives attributes from various Teamcenter classes. These attribute values are copied and therefore do not impact in the base classes from where they were derived.

The following graphic shows the single file recovery object model.



Single file recovery in Teamcenter rich client

SingleFileRecovery instances are created using the **sfr_instances** utility. The instances are generally created just before the routine backup by third-party backup systems. The backup label used in generation of these instances is subsequently used by third party backup systems to identify their backup sets with this label. On recovery command issued by Teamcenter, the user exit API contacts the 3rd party backup systems based on this backup label and restores the file to a common area.

The user exit API must be integrated with a third-party backup system to make this operational. The background process **sfr_bg** eventually retrieves the dataset containing the Teamcenter files to the users' **Newstuff** folder. The number of **sfr_instances** quickly grows in the database. Depending on the retention policy backup data of your site, the old instances should be deleted using the **sfr_instances** utility. The user exit API is:

```

extern USER_EXITS_API int SFR_recover_files_to_location(
const char *      dstClient,      /**<(I)  Volume host(node) name */
const char *      destination,    /**<(I)  The common area where files need to be
recovered */
int              no_of_files,     /**<(I)  Number of files to be recovered */
char *           bkup_lab,        /**<(I)  Backup Label associated with recovered files */

```

```
char **          volPath          /**<(I) Absolute Volume paths of recovered files */
);
```

Single file recovery query

The **SingleFileRecovery** query, located in the My Teamcenter search pane, is used to locate single files for recovery.

Once you run the query, the system displays the following message:

```
A Reference of File in the Database OR a Copy of file from Backup will be
recovered in the Newstuff folder.
```

The recovery process starts in the background, to compensate for the lead time in recovering the file from the backup medium. Specify the time intervals of this functionality using the following preferences:

- **TC_sfr_recovery_interval**

Specifies, in seconds, how often the system checks for recovered files after a user-initiated single file recovery action is activated.

- **TC_sfr_process_life_time**

Specifies, in minutes, how long the system continues to check for recovered files after a user-initiated single file recovery action is activated.

Alternative hot backup and recovery procedures

Back up and restore SQL Server database files

If you do not use third-party storage management systems, Siemens Digital Industries Software recommends that you use the SQL Servers Enterprise Manager or T-SQL backup/restore scripts to perform the hot backup and recovery operations on the database.

MS SQL Server supports online (hot) backups from Enterprise Manager that can back up, restore, and recover database files. This is a feature of the MS SQL Database Server and does not require separate installation.

1. Open the SQL Enterprise Manager by clicking **Start→All Programs→Microsoft SQL Server→Enterprise Manager**.
2. Expand the **SQL Server Group** to view all existing servers.
3. Expand the required SQL Server.
4. Expand **Databases** to list the available databases.

5. Right-click the required database and select **All Tasks→Backup Database.../Restore Database...**.
6. Click **Backup Database** to start the database backup process.

The **SQL Server Backup** dialog box is displayed.

7. On the **General** tabbed page, choose a type of backup from the **Backup** options.

After you create a complete database backup (using the **Database-complete** option), you must also create a **Transaction log** backup. To do this, open the **SQL Server Backup** dialog box and select the **Transaction log** backup option.

8. Select the destination (tape or disk) for the backup.

A backup device is a location that stores backup files. A backup file can hold multiple backups.

If this is the first backup of your database, create a backup device by performing the following steps:

- a. Click **Add**.

The **Backup Destination** dialog box is displayed.

- b. Type a name for the backup in the **Name** box.

The database name with a **.bak** extension is used as the default name for the backup.

- c. Click **File Name** (backup device).
- d. Define the appropriate path for the file.
- e. Click **OK**.

The **SQL Server Backup** dialog box is displayed again.

- f. Select the device or file you created from the **Destination** list.
9. Select the appropriate overwrite option for your backup. You can use this option to add multiple backups to file or device or to overwrite previous backups with the most current backup.

Warning:

Selecting the **Schedule** box automates the database backup process. Siemens Digital Industries Software does not recommend using this option, because it only backs up the database. It does not automate the backup of volumes.

10. Click **OK**.

The **SQL Server Backup** dialog box is displayed.

11. Click the **Options** tab.
12. Select the options that are appropriate for your organization.
13. Click **OK**.

A message is displayed indicating successful completion of the backup operation.

Back up and restore Teamcenter volumes

1. Use **backup_modes** utility to Set Teamcenter to read-only or blobby volume mode.
2. Copy the volumes from the source location to the destination device/location.
3. After completing the backup, return Teamcenter to normal mode.

Back up with SQL Server Virtual Device Interface (VDI)

Microsoft's Virtual Device Interface (VDI) is integral to all third-party SQL Server backup solutions, whether hardware or software. The VDI provides third-party vendors access to a collection of high-performance backup-and-restore mechanisms. By writing to the VDI, developers can substitute a virtual device for a server's local disk file or tape.

VDIs enable third-party software to control SQL Server backup and restore, which allows vendors to create a seamless interface between their own high-performance technologies and the SQL Server. Third-party applications can use the VDI interface to perform snapshot as well as standard SQL Server backup and restore.

However, VDI under the SQL Server 7.0 does not permit true third-party integration. It supports only standard SQL Server backup and restore. SQL Server, through its support for snapshot backups, treats the third-party solutions as true backups.

In this case, the backup of volumes and databases must be performed in the traditional manner.

The ability to back up and restore data in a reliable and timely manner requires the SQL Server VDI (Virtual Backup Device API). Most third-party independent software vendors use the VDI application programming interface to integrate SQL Server into their products. The advantages and disadvantages of VDI are as follows:

Advantages

- This API is engineered to provide maximum reliability and performance to support the full range of SQL Server backup and restore functionality.
- This API is used by most third-party vendors to perform backup/recovery operations. The VDI provides parallel high-speed data transfer between the SQL Server and the snapshot provider with minimal overhead.

Disadvantages

- Only utilities that interact with the SQL Server through the VDI can issue the backup and restore command snapshot option. Users cannot issue the snapshot option directly.
- SQL Server supports only the backups and restorations saved in the snapshot format. It does not support differential and transaction log backups saved as snapshots.
- Backup/restore of the SQL Server database is possible using the SQL Server Enterprise Manager or VDI. However, an integrated solution to back up both the SQL database and volumes is not possible using these approaches. Backup/recovery of Teamcenter volumes must be performed using the traditional copy and paste method.
- VDI combined with the third-party integrations can effectively back up and restore Teamcenter metadata and files that require minimal downtime and require SQL Server 2000/2005 to be in hot backup mode. This combination enables you to achieve greater flexibility by ensuring 24 X 7 availability.

24. Teamcenter licenses

Licensing overview: seats, features, and bundles

Managing how users access your company data is an important factor in information security. Users may be employees within your company, or they may be outside suppliers or contractors. Users gain authorized access to Teamcenter functionality through named user licensing.

With named user licensing, each user has appropriate access to the system from any Teamcenter client, and purchase of licenses and keys can be minimized.

When Teamcenter is installed, a **Default Local License Server (DLLS)** is defined. This DLLS is used as the default license server for the site, which in turn becomes the default license server for a user. If multiple license servers exist, an administrator can assign a different license server.

When a user logs on through any Teamcenter interface, such as rich client and Active Workspace, NX and Solid Edge integrations, SOAs, and mobile applications, Teamcenter requests checkout of a seat license from the license server assigned to the user. As the user performs actions beyond foundation functionality, such as actions that are part of **Change Manager** and **Requirements Manager** features, Teamcenter requests checkout of corresponding feature keys from the license server.

A named user on a site can be assigned only one license server at a time. A license server can serve licenses to more than one site.

Seat level

A seat license enables access to the system and foundation (base) Teamcenter functionality. By default, users are assigned an **Author** license seat level. An administrator can assign the user a different seat level, such as **Consumer**. Typically, license files include three license seat level options plus an internal seat type for administration:

Seat level	Primary focus
Author	Create and edit data.
Consumer	Query, view, print, and use basic viewer capabilities.

	 Author	 Consumer
Create/Edit item	Yes	No
Create/Edit BOM	Yes	No
Create/Edit change	Yes	No ¹

1

Consumer can create a Problem Report (PR) or Issue Report.

Seat level	Primary focus	
	 	
	Author	Consumer
	Create/Edit dataset	Yes Yes ²
	Create/Edit folder	Yes Yes
	Initiate workflow	Yes No ³
	Review/Approve workflow	Yes Yes ⁴
Occasional author	Create and edit data as at the Author level, but only for a limited time per month. This level is intended to support two use cases: • Access is required several days per month, but for short periods. • Access is infrequently required, with long periods. Limited time is defined as the combination of the following: • Maximum of twenty hours per month. • Maximum of five unique calendar days when logged on for over an hour each day. Teamcenter counts a minimum of 1 hour use per logon, and a maximum of 7.5 hours per day.	
Admin	A special system administration level assigned to the infodba and dcproxy user definitions provided for every site. The infodba and dcproxy users cannot be switched to another seat level, nor can other users be assigned the Admin seat license level. The infodba user must only be used for installation and upgrade, and dcproxy must be used only for the Dispatcher. The infodba and dcproxy users must not be used to perform any actions as regular Teamcenter users. Day-to-day administration activities (including scheduled cron jobs such as running the data_sync utility) must be performed by a named user who is a member of an administrator group. Members of an administrator group may have similar privileges and authorizations as infodba, but have greater accountability because their activities are logged with their identity.	

Feature keys

Feature keys are licenses for optional Teamcenter modules, and are required to use certain Teamcenter solutions, such as **Change Manager** and **Requirements Manager**. Checkout of a key gives a user access to the module from all interfaces to Teamcenter.

² Consumer can annotate and mark up PDFs, images, and snapshots, but cannot edit the sources directly.

³ Consumer can initiate a workflow on PRs and Issues.

⁴ Consumer cannot approve a workflow if approval would make updates to an object other than PR, ECR, or ECN.

License bundles

Companies can purchase license bundles as well as licenses for individual seats and features. Teamcenter license bundles typically include seat-level licenses for base (foundation) Teamcenter functionality and keys for a group of features. Bundles are created such that upon logon a user consumes a seat license and the entire group of features in the bundle, and users not assigned the bundle cannot independently leverage the features in the bundle. If the license server assigned to the user includes a license bundle in the license file, an administrator can assign the user a license bundle.

Support for license bundles includes:

- Assigning license bundles to users using the rich client.
- Assigning license bundles to users using the **make_user** utility.
- Enforcing a user's seat license level based on the base seat license available in the assigned bundle.
- Ensuring that only as many active users are assigned a bundle as there are licenses for that bundle.
- Allowing check out of feature keys that are not part of the assigned bundle, but are available separately in the license file.
- Recording use of feature keys and license bundle in the FlexLM license logs.

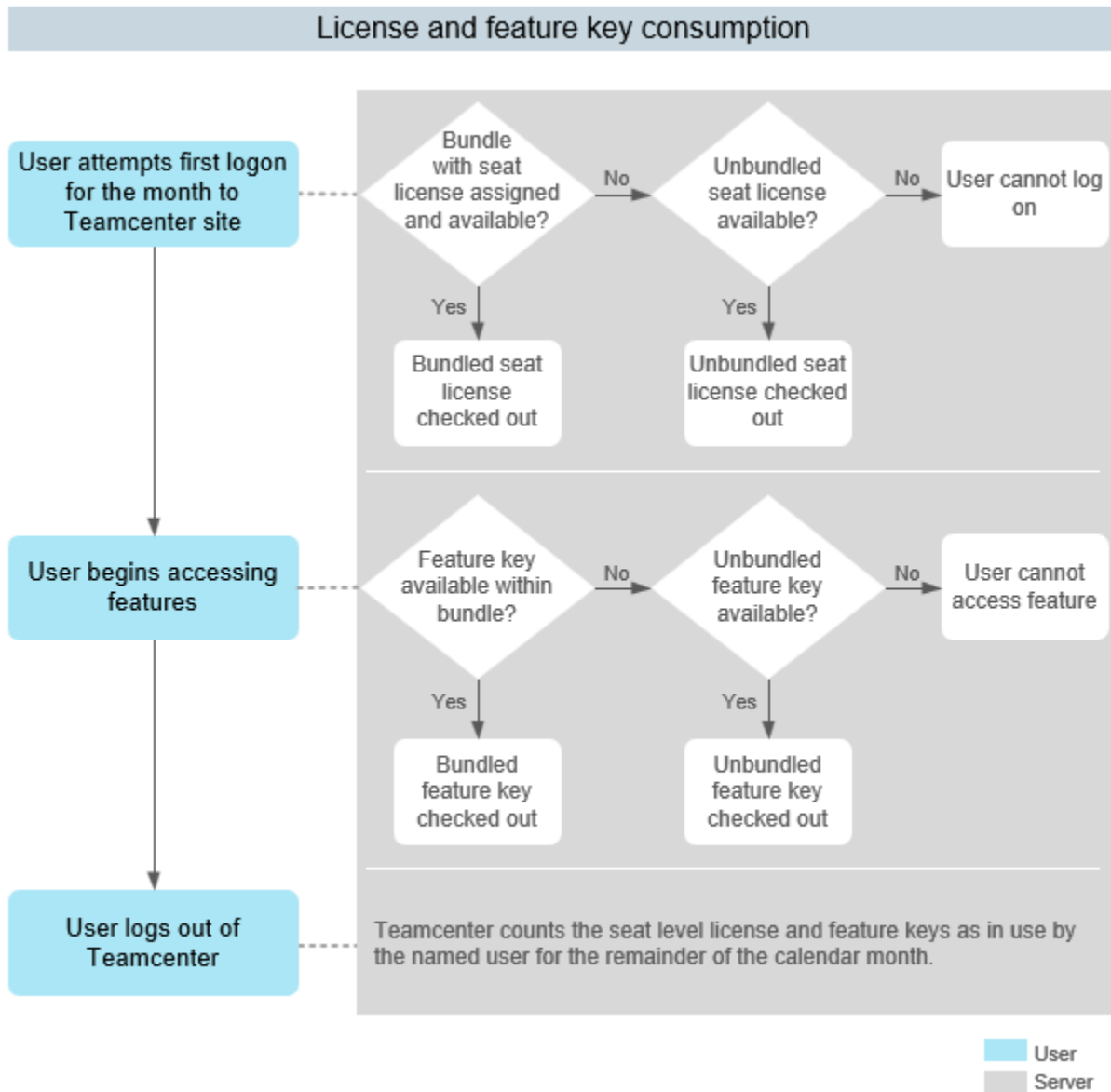
How license use is calculated

When a user logs on, Teamcenter requests checkout of a seat license from the assigned license server. As the user performs actions beyond foundation functionality, such as actions that are part of **Change Manager** and **Requirements Manager** features, Teamcenter requests checkout of corresponding feature keys from the license server.

Tip:

The Siemens Digital Industries Software Common Licensing Server is described in the **SPLMLicensing_user_guide.pdf** (*Siemens PLM Licensing User Guide*), available for download from Support Center.

With this system, each user has appropriate access to the system from any Teamcenter client, and purchase of licenses and keys can be minimized.



The license-used count for each seat license level comprises the sum of the following:

- All users defined as active at the level.
- Plus those users who logged on during the current calendar month with that level, but are now either set to inactive or are assigned a different seat level.

An *inactive* user who never logs on during a calendar month is not added to the used count.

Example:

A customer purchases 10 Author and 100 Consumer licenses. The customer adds 10 active Author users and 100 active Consumer users to Teamcenter. When a user logs on to Teamcenter, the system checks out that license from the license server. If all of the active Teamcenter users log on, then the customer has exactly the correct amount of licenses and all users have access.

When on January 1 user X, an Author level user, logs on, one Teamcenter Author license is checked out to user X for the remainder of the calendar month. On January 5, the administrator sets one of the 100 active Consumer users who has never logged on to inactive, and then changes user X to a Consumer level license. On January 10, user X logs on, and one Consumer license is checked out to user X for the remainder of the calendar month.

In this case, user X consumes both an Author license and a Consumer license until February 1. Because all 10 Author licenses and all 100 Consumer licenses are assigned to active users or checked out, the administrator cannot add another active Author user until February 1.

Example:

Chris, an **Occasional author**, logs on for 45 minutes a day each of seven days early in a month.

7 logons x 1 hour minimum = 7 hours

On two days later in the month, Chris logs on multiple times throughout the entire day, accumulating the daily maximum for each day.

2 days x 7.5 hours = 15 hours

At this point, Chris has accumulated 22 hours use, more than the maximum 20 hours, but still less than the maximum of five unique calendar days at over an hour per day. Chris can continue to log on until both the maximum hours and maximum days are reached.

If a user who is assigned to the **Occasional author** seat level regularly exceeds the time limit, then the user should be reassigned to the **Author** license seat level.

Federated license servers

A single Teamcenter site can potentially be accessed by multiple users from inside and outside a company. Conversely, a user may need to access multiple sites. Rather than requiring that a user with the same exact name and user ID consume site-specific seat and feature licenses when logging in to different sites, or that all users of a site draw from the same license server, multiple license servers with varying license files can be defined for a site. This capability is referred to as federated license servers.

Administrators can define license servers for a site and assign them to users using either the **site_util** utility or the Organization perspective in the rich client.

The following scenarios are some examples in which multiple license servers are used to appropriately provide access.

Different business units want control over their own licenses

Having separate license files and servers can provide business units better control over licenses, and can help avoid cross charging within a company.

Multisite environment

A user may generally work in SiteA, but might need to approve a Global workflow in SiteB. The user can be added to SiteB and point to the original license server to get authorized access.

Test, Development, and Production environments

Similarly to multisite environments, a user can be added to multiple environments and point to and use the single license from the original license server. Note that licenses of type "Test/QA" may not be used in a production environment or for any other purpose.

Adding new licenses

Say that company ABC has seat and feature licenses for a current set of users. The company wants to add seats and different feature licenses for process planning purposes to a new set of 50 users. In this case, the company can add the new 50 users and assign a new license server that has the required features without having to combine all the licenses into one file.

Global and regional licenses

License files can be purchased and generated for either global or regional use. Typically, an uplift charge is incurred for a global license.

Global licensing can facilitate Teamcenter administration and access for users who travel outside the region where the license server exists.

Purchasing both types of license files can make sense when a company has a sizable group of users who do not travel outside a region as well as users who do travel outside a region.

General principles of global vs. regional licenses

- A soldto ID cannot have both global and regional licenses. If both license types are required, then they must have separate soldto IDs.
- A license server can serve only one type of licenses, either global licenses or regional licenses.
- A global license can provide access to users physically located outside the region of the global license server as well as users located within the region of the license server.
- A Teamcenter user may only point to 1 license server. All Teamcenter licenses and add-on modules the user needs must exist on that 1 license server.

Example:

A user who requires the use of an Author seat, and Change and Classification features, must point to a license server that contains all three licenses.

- If regional license users travel outside their home license server region, then they cannot access their home region license. They must use a global license or a regional license served by a license server in the region where they are physically located.

Global vs. regional licenses Q & A

When licensing Teamcenter, does the location of the Database Server matter?

No, only the location of the license server and the physical location of the users.

When licensing Teamcenter, does the location of the Application (Enterprise) Server matter?

No, only the location of the license server and the physical location of the users.

In a cloud deployment, does every license need to be global?

No.

Can a company have both a regional and a global license server if they only have one Teamcenter database server and Application server?

Yes, as long as they are separate servers.

Can a company run a Multi-Site deployment in 2 regions with 2 regional licenses?

Yes.

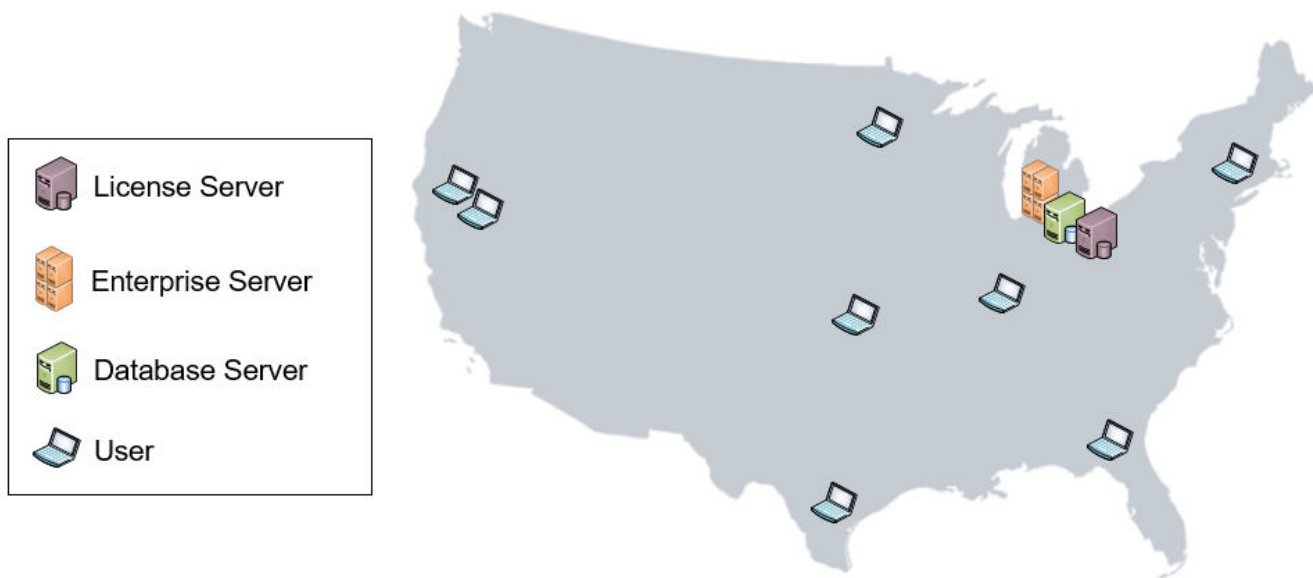
If a small company located in 1 region was to deploy to the cloud, where would their license server need to be? Could they use regional licenses or do they require global?

Their license server could be either on-premise or in the cloud. They could use a regional license since they have no users outside of the 1 region.

Example Scenario 1 - Single region Teamcenter deployment

Enterprise server, Database server and License server(s) exist in the same region as the User. Multiple organizations (such as Manufacturing and Engineering) in a company can have separate license servers/soldtos.

A single regional license is all that is required as long as the users remain in the region.

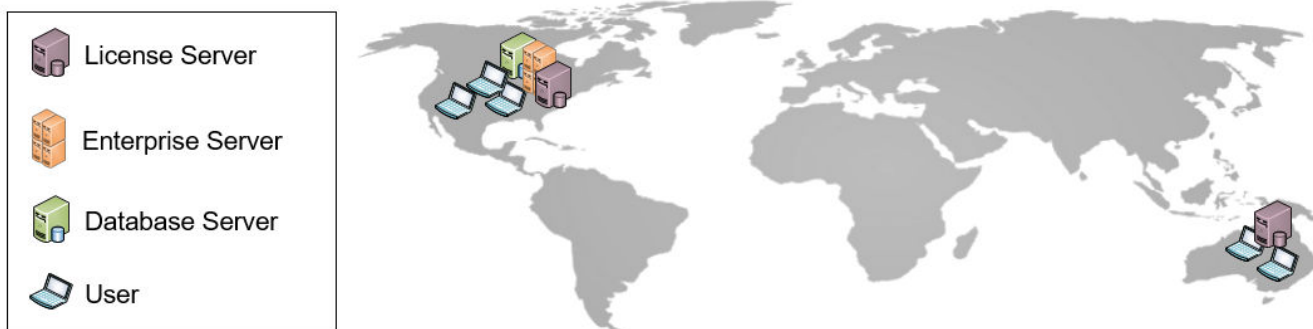


Example Scenario 2 - Two-region Teamcenter deployment

Teamcenter Application Server and Database server are in the US with clients in both US and Asia.

Three licensing options are possible:

- 1 global license – All licenses require the global uplift
- 2 regional licenses (1 in the US and 1 in Australia) – No global uplift
- 1 regional license server to serve the US and 1 global license server to serve outside of the US – Global uplift paid on licenses on the global license only.



Example Scenario 3 - Three-Region Deployment

3 region Teamcenter deployment using Multi-Site in 2 regions – Teamcenter application servers in the US and Europe. Database in the US shared with Europe via Multi-Site. All users accessing the US Site are in the US. Europe site used by clients in Europe and Asia.

Three licensing options are possible:

- 1 global license – All licenses require the global uplift
- 3 regional license servers (1 in each of the US, Europe, and Asia) – No global uplift
- Combination of 1 or 2 regional license servers to serve the local region(s) and 1 global license server to serve outside of the region(s) – global uplift paid on licenses on the global license only.



Example Scenario 4 - Cloud deployment

Teamcenter deployed to the Cloud – Teamcenter Enterprise Server and Database server are in the Cloud with clients in US, Europe and Australia

Licensing options Q & A

Does it matter in what country the cloud service provider is located?

No.

Can a Cloud Deployment use global and/or regional licenses?

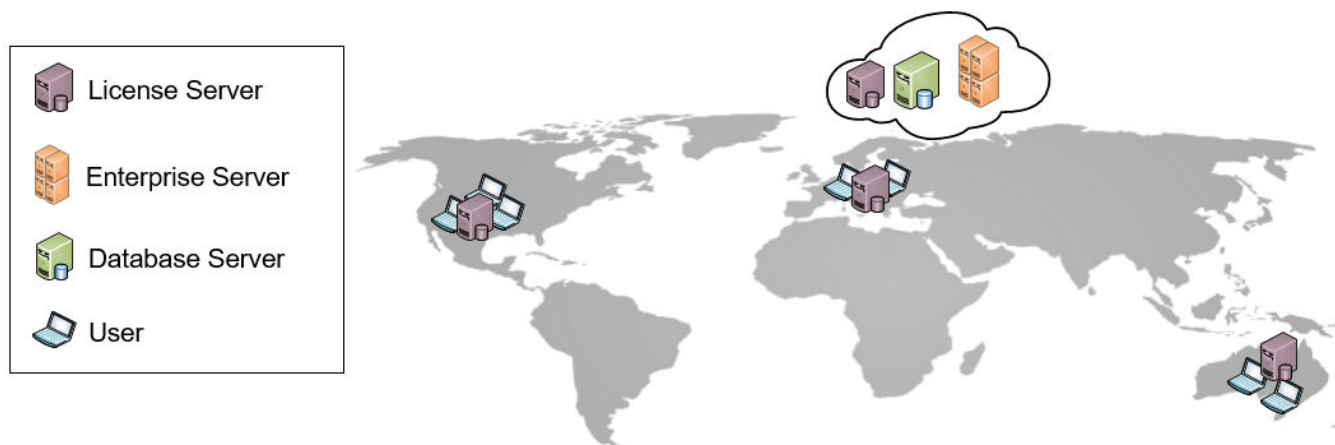
Can use either or a combination of the two.

Can each site have its own regional license server serving regional licenses?

Yes.

In a hybrid Multi-Site environment, on-premise in 1 region and in the cloud in another region, how would the environment be licensed?

Can be a combination of regional and global as needed.



Teamcenter licensing preferences

To configure Teamcenter license management, you can set the following preferences.

To do this	Set this preference
Adjust the threshold at which a warning message appears if, when you create or modify a user, the number of available base Teamcenter licenses equals or falls below the threshold.	license_warning_level The default preference value is 5.
Specify identifiers of administrative users who receive Teamcenter notifications when an occasional author exceeds the allotted usage and/or grace period limits specified for the occasional author license level.	LicenseUsage_admin_notifier_list
Specify remaining duration (in days) of allotted usage when occasional logged-on users start receiving notification of approaching their allotted usage days.	LicenseUsage_days_warning_level
Specify remaining duration (in hours) of allotted usage when occasional logged-on users start receiving notification of approaching their allotted usage hours.	LicenseUsage_hours_warning_level
Specify logins remaining in the month when administrators start receiving warning messages that feature key usage is approaching the allowed limit.	LicenseUsage_module_usage_warning_level
Control inclusion of the actual user ID in generated license usage reports.	LicenseUsage_show_userId_in_report

To do this	Set this preference
Specify the email ID of the Siemens Digital Industries Software licensing account team that receives notifications when any module usage limit exceeds its allowed usage.	Siemens_PL_email_id
Allow use of any account with DBA role to cause the SPLM_LICENSE_SERVER environment variable/Default Local License Server (DLLS) comparison and update to occur.	License_allow_dba_to_change

Configure failover license servers

To define a reference to a trio of license servers that are set up in Redundant Server Configuration, use the following procedure in the Teamcenter rich client Organization perspective.

1. Define references to the failover license servers (the second and third servers listed in the Redundant Server Configuration enabled license file).

Each reference consists of **Host**, **Port**, and **Protocol** information.

2. Define a reference to the primary license server (the first server listed in the license file).

In addition to the **Host**, **Port**, and **Protocol** information, add the references for the failover license servers to the **Failover License Server(s)** list.

3. When the Redundant Server Configuration is intended to be the default license server for the site, the best practice is to do the following:
 - In the **License Servers** category, set the **Default Local License Server** definition to the first server listed in the license file.
 - In the **Site** object definition, set the license server to the **Default Local License Server** reference.

Managing Teamcenter licenses

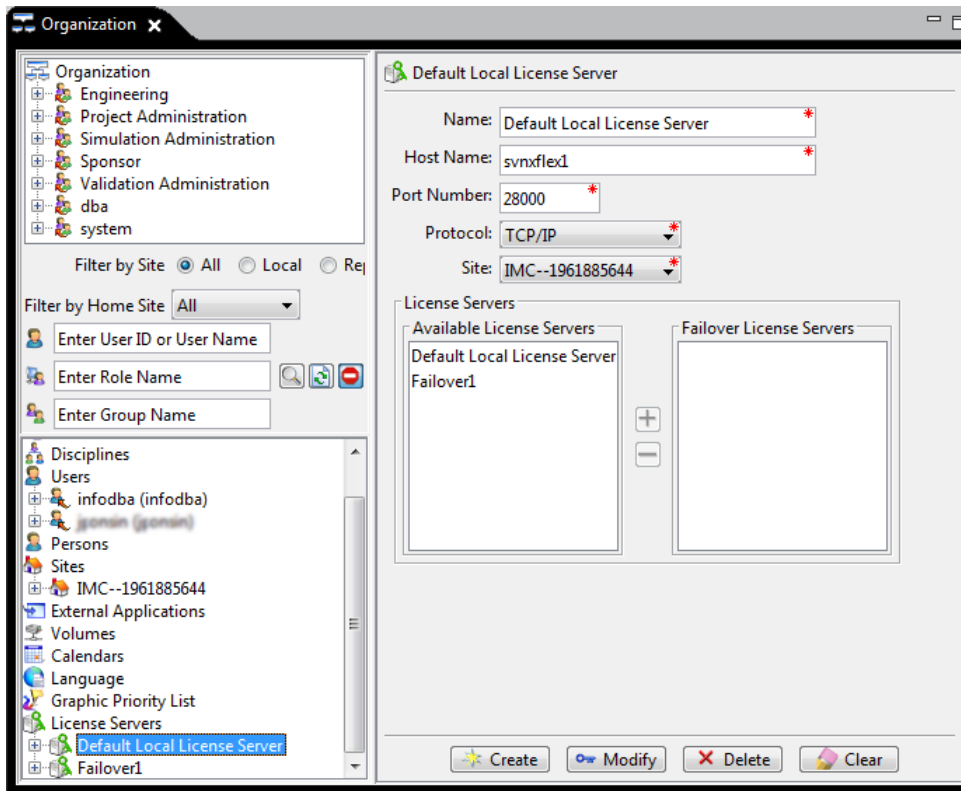
Create a license server reference

A license server is a process dedicated to serving software licenses and tracking license usage. Multiple license servers can be used to serve licenses. A reference to a license server is needed when specifying a default license server for a site, configuring failover license servers, or assigning a specific license server to a user.

1. Log on to the rich client as an administrator and open the Organization perspective.

2. Select **License Servers**, enter the required information, and click **Create**.

The **Failover License Servers** list is needed only when defining the primary license server for a Redundant Server Configuration (federated servers). For details, see [Configure failover license servers](#).



Change the Teamcenter default license server

The Default Local License Server (DLLS) is a special license server reference that is created during installation of Teamcenter. It is used as the default for the site, and thereby the default for all users. An administrator can change the value of the DLLS in either of two ways:

- In the rich client Organization perspective, select **Default Local License Server** and change the DLLS value. This approach is the most direct and obviously intentional.
- On the Teamcenter corporate server (enterprise server), do the following steps. This approach may be more convenient when multiple enterprise servers are deployed, or when changed or corrupted license server values do not allow logging in to the rich client.
 1. Set the **SPLM_LICENSE_SERVER** environment variable so that the first port and server name value in the file is the desired value for the DLLS.

Note:

In the case that the **SPLM_LICENSE_SERVER** environment variable lists multiple license servers (where each server is separated by a semicolon), Teamcenter only reads the first license server value.

2. Log on as a user with Teamcenter administration privileges.

When an administrator logs on, Teamcenter compares the value of the value of the **SPLM_LICENSE_SERVER** environment variable to the value of the DLLS. If the two values differ, then Teamcenter updates the DLLS with the value of the **SPLM_LICENSE_SERVER**.

Note:

If the **License_allow_dba_to_change** preference is set to **True**, then logging on using any account with DBA role causes the **SPLM_LICENSE_SERVER** environment variable/DLLS comparison and update to occur.

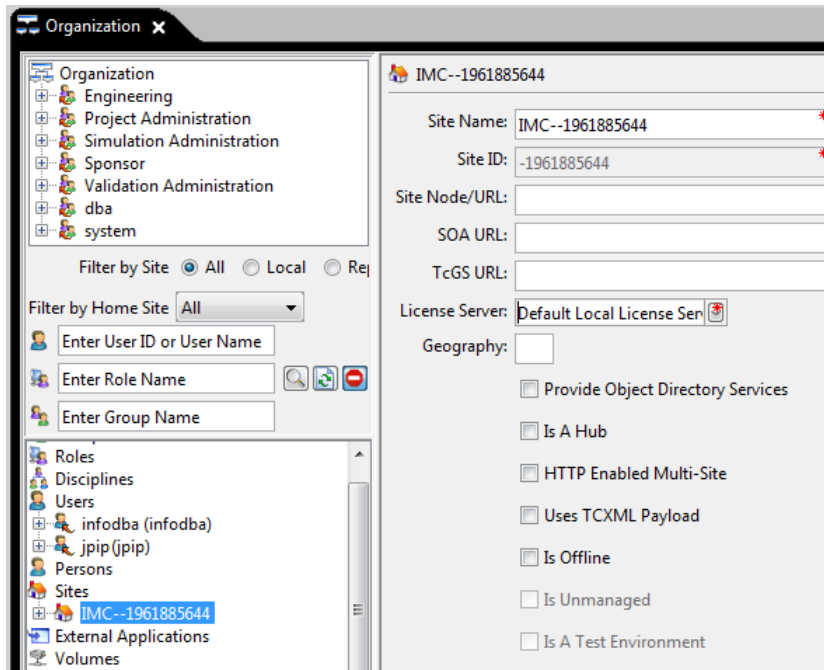
Set a default license server for a site

During Teamcenter installation, a Default Local License Server (DLLS) is defined. That DLLS is the default license server for users logging on to the site. An administrator can specify a different license server as the default for the site either indirectly or directly:

- *indirectly* by **changing the Teamcenter default license server**, which affects all sites in the Teamcenter environment that point to the DLLS.
- *directly* for a single site by ensuring that a **license server reference** for the intended license server exists in the site.

You can use either of two methods to assign a license server reference to the site:

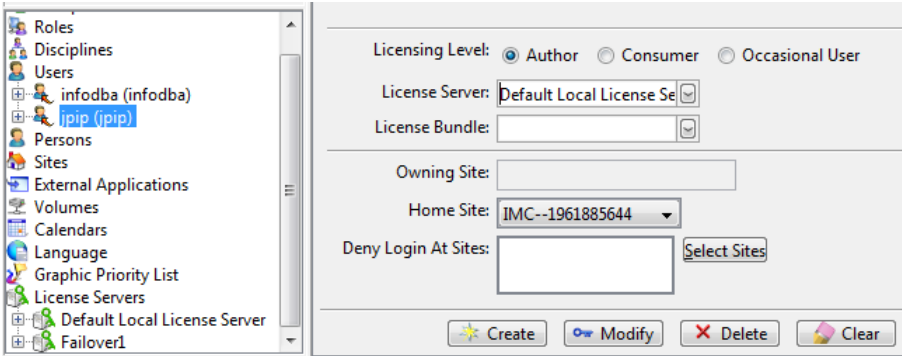
- Use the **site_util** utility with the **-license_server** argument.
- Use the Organization application.
 1. As an administrator, log on to the rich client and open the Organization application.
 2. Select the site and in the **License Server** box, assign a license server to a site.



Assign Teamcenter licenses to users

A Teamcenter administrator can assign a Teamcenter seat license, license server, and license bundle to users using either the rich client or a command line utility.

For this approach	Do this
Teamcenter rich client	1. Log on to the rich client as an administrator and open the Organization perspective. 2. Open the Users pane. 3. The default license level is Author. To assign a different seat license level, do one of the following: • Select a Licensing Level. • (For license bundles that include a Teamcenter seat license) From the License Bundle list, select a Teamcenter license bundle. Note: When you create or modify a user, Teamcenter compares the number of purchased licenses with the license-used count. If all licenses at the requested seat level are currently used, then you cannot create or modify a user at that seat licensing level.

For this approach	Do this
	4. (As needed) To override the site's default license server assignment for this user, select a value from the License Server list. Each user can be assigned to only one license server. If multiple license server references have been defined in Teamcenter, then administrators can assign a user to a specific license server. Assignment to a single license server does not preclude assignment to the first server in a Redundant Server Configuration, which allows for failover in case a license server fails. Note: This single-server license attachment differs from versions prior to Teamcenter 11.2, where a user could query multiple license servers with distinct license files (Multiple Server Configuration). Overriding the site default license server assignment might be necessary when, for example, both regional and global license servers exist, when different groups use different licenses (federated servers), or when load balancing among license servers is necessary. 
make_user utility	Use the make_user utility with the -licenselevel, -licenseserver, or -licensebundle arguments to set the respective values for a user. To update more than one user's licensing information, use the utility's batch mode processing capability.

Releasing checked out licenses

When attempting to release checked-out licenses, consider the following use cases:

- **Use case 1:**

Acme Corporation has X number of licenses. It purchases an additional number (Y) of licenses, which requires a new license file.

1. Shut down the license server, let the license server read the new license file, and restart the license server with the new license file.
2. Stop the pool manager.

By doing this, the cache carrying the licensing information gets reset.

3. Restart the pool manager, which allows Teamcenter to access the newly purchased licenses.

Note:

The licenses-used count within Teamcenter is not cleared. However, the additional newly purchased licenses are immediately available for use.

- **Use case 2:**

Acme Corporation has X number of licenses and all licenses have already been checked out this month.

Acme Corporation wants a different set of users to consume the same licenses and wants to release the checked-out licenses.

This cannot happen because Teamcenter counts the licenses as used until the end of the calendar month. The licenses used count resides in the database and cannot be reset. The stability of the licenses used count is critical for auditing purposes and for tracking license usage.

Monitor seat license level violations

When a user logs on to Teamcenter, the seat license consumed is at one of two seat level types: author or consumer.

The pairing of a given operation (action) performed on a given business object (type) is categorized as either author level or consumer level.

When a user logged on with a consumer level license performs an author level operation on a given operation/business object pair, Teamcenter counts a license violation. When the user exits the session, the violations are recorded in a log file on the server.

The log file is saved in the **consumer_violations** directory within the Teamcenter log location and named with the pattern *user-id_license_violation.log*.

By default, the Teamcenter log location is the value present in the TEMP environment variable. If a TC_LOG_DIR environment variable has been set on the server, then the **consumer_violations** directory is created within that location.

Use the **consumer_license_violation_report** utility to summarize license violation data from individual user log files. Users assigned a consumer level license who regularly perform author level operations should be reassigned an author level seat license.

After generating a license violation report, an administrator may choose to delete user-specific license violation log files. However, reports generated after these files have been deleted do not contain the historical license violation data of the users.

Example:

At the end of a two-week period, an administrator generates a summary report of license violations, and then deletes all user-specific license violation log files. In the following two weeks, new user-specific license violation log files are generated. At the end of the week four, the administrator generates a new summary report. The original summary report contains the violations for weeks one and two, and the new summary report contains only the violations for weeks three and four.

Reporting license usage

License usage information logged in Teamcenter

The following license usage details are logged in the Teamcenter database.

- Users logged in, seats used, and features used per calendar month.
- For **Occasional author** level users:
 - Hours of use per month.
 - Logs a minimum of 1 hour per logon. If a named user logs on for only 1 minute, usage is logged for 1 hour.
 - Logs a maximum of 7.5 hours per day. If a named user is logged on continuously, usage is logged for only 7.5 hours.
 - Simultaneous logons by the same user, such as when a single user is logged on to both the rich client and the Teamcenter Client for Microsoft Office.

The system counts the usage from the first logon time for that named user until the last logoff time. Time spent logged on to multiple clients simultaneously is counted once, not multiple times.

- Days of use per month.
- Number of logons per month.
- Any use violations.

You can **Generate a license usage report** of license use information logged within Teamcenter.

License bundle logging

The following are examples of license logs when license bundles are used.

- When a user with assigned bundle (**Bundle2**) logs in, the base license feature key is checked out:

```
09:59:36 (ugslmd) OUT: "Bundle2" gopinats@TcServer 09:59:36 (ugslmd) OUT:
"Bundle2_teamcenter_author" gopinats@TcServer
```

- When user creates a Requirement item, the **requirements_user** feature key is checked out:

```
10:42:11 (ugslmd) OUT: "Bundle2_requirements_user" gopinats@TcServer
```

- When user queries for a schedule object, the **prog_exec_access** feature key is checked out:

```
11:21:34 (ugslmd) OUT: "Bundle2_prog_exec_access" gopinats@TcServer
```

- When the user logs out, all the checked out feature keys along with the bundle are released:

```
11:30:34 (ugslmd) IN: "Bundle2_prog_exec_access" gopinats@TcServer
11:30:34 (ugslmd) IN: "Bundle2_requirements_user" gopinats@TcServer
11:30:34 (ugslmd) IN: "Bundle2_teamcenter_author" gopinats@TcServer
11:30:34 (ugslmd) IN: "Bundle2" gopinats@TcServer
```

Generate Teamcenter license reports

As an administrator, you must ensure that sufficient software seat and module (feature) licenses are available and that you comply with the terms of your license agreement. To assist in that task, you can generate reports of license and module usage information logged in the Teamcenter database.

License Login Report	If geography access has been configured for Teamcenter, reports the locations from which a user logs on.
License Seat Level Report	Summarizes seat level licenses assigned to active and inactive users by license type, allowing you to monitor license consumption and optimize license usage.
License Usage Report	Summarizes each named user's use per month of seat and feature (module) licenses. To assist you in ensuring that seat license levels are appropriately assigned to users in compliance with your license, Teamcenter provides the utility consumer_license_violation_report . The utility forms a consolidated report of users who repeatedly perform author-level operations (actions) while logged on with a consumer-level seat.
Module Usage Report	Summarizes module (feature) license use per month.

Note:

Specifics of your purchased license levels, their detailed descriptions, and their quantities are contained in your master license agreement. Your site's license file determines which license level options are available in the rich client **Organization** perspective.

Generate license reports from the rich client

1. In the rich client, open the My Teamcenter perspective and choose **Tools>Reports>Report Builder Reports**.

The **Report Generation Wizard** appears.

2. Select one of the license reports. For example, select the **License Usage Report**.
3. Click **Next**.
4. At the **Fill in Criteria** page, specify criteria for limiting the report.

For a **License Usage Report** or **License Login Report**, you must specify the users to include.

- For example, in **User ID** type an asterisk (*) to include all users.
- To get a report only of usage violations, in **Usage Violation** type **True**.

For a **License Seat Level Report**, ensure the parameters are set as follows:

- Set **License Level** to a value of **All**.
- Set **License Server** to the name of your license server. If your organization is using federated license servers, specify the name of the license server for which you want to generate a seat level report.
- Set **Summary Report** to **true** to produce a status summary report of seat licenses in use.

Set **Summary Report** to **false** to produce a detailed report on seats in use.

- Leave **message_id** set to **SeatLicensingUsageReport**. Do not change this value.

Tip:

For criteria that require keyed entry, hover over the criterion box to see a description of valid input parameters.

5. Set **Report Stylesheets** to the type of report you wish to generate.

- Choose **`your_report_type_status_html.xml`** to produce an HTML report displayed in your browser.
- Choose **`your_report_type_status_text.xml`** to generate and display a comma separated values text file. For best readability, save the output to a local file and then open it in a spreadsheet application such as Microsoft Excel.

6. Click **Finish**.

Generate license reports from Active Workspace

Follow the general procedure for a summary report as described in web client *Fundamentals* documentation "Generate an item, summary, or a custom report".

The following table provides report filter descriptions for the license reports.

Report	Filter	Description
License Usage Report	User ID	Required. Enter an asterisk (*) to include all users, or enter system user names. Use of wildcards for patterns is supported.
	Seat Level	Optional. Select seat levels from the list.
	Usage Recorded Year	Optional. Enter a four digit year.
	Usage Recorded Month	Optional. Select months from the list.
	Usage Violation	Optional. Select to include only usage violations.
Module Usage Report	From Month	Optional. Enter a numeral for the report start month, where January is 1.
	From Year	Required. Enter the four digit year for the report start year.
	End Month	Optional. Enter a numeral for the report inclusive end month, where January is 1.
	End Year	Optional. Enter the four digit year for the report inclusive end year.
	message_id	Accept the default value, which is ModuleUsageReport .
License Login Report	Geography	Optional. Select countries from the list to include in the report.
	Last Login Time	Optional. Select a time from which to start reporting logins.
	Login Count	Optional. Enter the number of logins to report per user.
	Login IP	Optional. Enter an IP address to report. Use of wildcards for patterns is supported.
	User	Required. Enter an asterisk (*) to include all users, or enter system user names. Use of wildcards for patterns is supported.
	Month First Created	Optional. Select a month to start reporting.

Report	Filter	Description
License Seat Level Report	Year First Created	Optional. Enter a four digit year to start reporting.
	License Status Message	Optional. Select a status message to include in the report for applicable logins.
	License Level	Set to All .
	License Server	Specify the name of your license server. The default value is Default Local License Server . If your organization is using federated license servers, specify the name of the license server for which you want to generate a seat level report.
	Summary Report	Set to true to produce a status summary report of seat licenses in use. Set to false to produce a detailed report on seats in use.
	message_id	Accept the default value, which is SeatLicensingUsageReport .

When generating your report, set **Report Stylesheets** to the type of report you wish to generate.

- Choose *your_report_type_status_html.xsl* to produce an HTML report displayed in your browser.
- Choose *your_report_type_status_text.xsl* to generate and display a comma separated values text file. For best readability, save the output to a local file and then open it in a spreadsheet application such as Microsoft Excel.

License usage report

Administrators can generate and review Teamcenter license usage reports to monitor compliance with license terms. Features assigned to a user within each reported month are consolidated on a single line for that user for the month.

If multiple months are reported, a user may appear on multiple lines.

userid	pseudo id	seat level	feature keys	month	used hours	usage days	number of logins
User1	b89a94465bc9a42c14a7	teamcenter_author	teamcenter_author; change_access; visview_base	11-Apr	1	1	1
User2	69cb49	teamcenter_author	teamcenter_author; visview_base	11-Apr	1	1	1
User3	0098	fb11501beb	teamcenter_author; change_access; visview_base	11-Apr	21	6	6
User4	56d851	0ffd78d37197	teamcenter_author	11-Apr	15	3	3
User5	362784168dea615e4c4	teamcenter_author	teamcenter_author; visview_base	11-Apr	20	7	7
User6	11113b01d046d3290d	teamcenter_author	teamcenter_author	11-Apr	26	8	8
User7	51543a114d441004486	teamcenter_consumer	teamcenter_consumer; change_access	11-Apr	4	4	3
User8	6d		teamcenter_author; access	11-Apr	22	7	7
User9	f5		teamcenter_author; base	11-Apr	27	9	9

Pseudo IDs are encrypted

Remaining fields match log analysis results

Features are grouped to minimize database table space and because with named user licensing, any feature used is allotted for the month.

Example license usage report displayed in a table

The columns of a license usage report output from Teamcenter match the columns from a FlexNet analysis log output, except that the pseudo IDs are encrypted more securely, and the feature keys used are grouped together on one line. This grouping minimizes the database table space required, and is appropriate because any seat or feature used is allotted to the named user for the month.

Column header	Description
user id	The Teamcenter user ID.
Note: The LicenseUsage_show_userid_in_report preference controls whether or not to output the actual user ID; the preference default value is true. When the preference LicenseUsage_show_userid_in_report is set to false, generating license usage reports from the Teamcenter database provides a secure method of hiding user identity in the output reports.	
pseudo id	A unique ID generated to enable anonymous reporting of user data. The pseudo user ID is always output.
seat level	The highest seat level used in the month by the user.
license bundle	The license bundle from which a seat license was obtained.
license server	The license server that served the license.
feature keys	Feature keys that were allotted to the user.
month	The month being reported.
used hours	The number of hours used in a given month.

Column header	Description
usage days	The number of days used in a given month.
number of logins	The number of times the user logged on in a given month.
usage violation*	Displays the value Y when there are logon denials. Tip: Specify this value in query criteria to fetch the data for all the violation records.
usage violation reason*	Lists the reason for the usage violation of the license.
max day usage*	Displays the value Y if the maximum hours of usage (7.5) is being recorded in any day in that month for the user.
max value reason*	Lists the reasons for the Maximum Day Usage value being recorded as Y , such as no logoff in a day.
not released lic count	The number of times a license is not released at the end of the day in a month. This is incremented when there is no logoff for a particular session on the same day the user logged on.
grace days*	The number of grace days used by the user.
grace hours*	The number of grace hours used by the user.
grace logins*	The number of times a user who exceeded license limits (hours and unique days) was allowed to log on.
number denials*	The number of times an user who exceeded license limits attempted to log on but was denied access.
home site	The home site of the user.

*Tracked for Teamcenter occasional authors only.

License seat level report

Use the license seat level report to understand the seat level license status of users. The report provides an output in summary and detailed formats. The summary format provides an overall output of seat level license usage information to quickly assess the number of purchased (and/or licensed), allocated and remaining seats. The detailed format reports the seat level license for each user to assist in managing seat level licenses at the user level.

The summary report includes the following columns of information:

Column	Description
License Server	The license server that served the license.
License Level	They type of license assigned to the user.

Column	Description
Purchase Count	The number of total licenses purchased and/or licensed for this seat type.
Active Users	The number of active users for this seat type.
Inactive Users With Current Month Usage	The number of inactive users for this seat type for the current month. The number of users with a current status of inactive who have also logged in to Teamcenter at this seat level during the current month. Inactive users cannot log in. However, they may have previously been active during the month and logged in as an author or consumer and consumed a license for the current month.
Available Licenses	The number of unused licenses for this seat type.

In addition to using the License Seat Level Report, you can use the **license_usage** command line utility.

The detailed report includes the following columns of information:

Column	Description
User ID	The Teamcenter user ID. The LicenseUsage_show_userId_in_report preference controls whether or not to output the actual user ID; the preference default value is true . When the preference LicenseUsage_show_userId_in_report is set to false, generating seat license reports from the Teamcenter database provides a secure method of hiding user identity in the output reports.
Pseudo User ID	A unique ID generated to enable anonymous reporting of user data. The pseudo user ID is always output.
License Server	The license server that served the license.
License Level	The type of license assigned to the user.
Purchase Count	The number of total licenses purchased and/or licensed for this seat type.
User Status	The status of the user, either active or inactive.
Current Month Usage	The user's seat level license usage status for the current month.

Module usage report

The module usage report provides a summary of usage per month of optional module licenses (for example, change author, requirements access, and so on).

Module use reporting may occasionally show that the number of licenses or keys used exceeds the number of licenses purchased. This is typical where the number of purchased licenses is intentionally minimized. Siemens Digital Industries Software allows for a small over-use. Administrators are notified about the over-use so that they can work with their Siemens PLM account representatives to identify the total number of licenses and keys required and take appropriate steps to comply with the licensing agreement.

When entering parameter values to **generate a module usage report**, you can:

- Enter the month as either a number or text (3-character abbreviations or alphabetic month values).

For example, to indicate March, you can enter **3**, **Mar**, or **March**.

- Enter the year values.

For example, to run the module usage report for all months from January 2017 through December 2018, enter **2017** in **From Year** and **2018** in **End Year**.

- Leave the month fields blank and just enter the year value.

For example, to run the module usage report for 2018, enter **2018** in **From Year** and **2018** in **End Year**.

The Module Usage report saved to a .csv file and viewed in Excel resembles the following:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	FeatureKey	Type	Jan-12	Feb-12	Mar-12	Apr-12	May-12	Jun-12	Jul-12	Aug-12	Sep-12	Oct-12	Nov-12	Dec-12	Jan-13
2	collab_product_dev	Usages	35	28	78	117	135	17	11	85	55	32	32	53	
3	collab_product_dev	Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
4	cont	Usages	541	85	504	534	564	122	96	88	111	120	638	996	
		Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
		Usages	1377	59	1949	350	1989	1368	50	167	1804	376	413	225	
		Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
		Usages	30	127	167	211	16	182	5000	5000	5000	5000	5000	5000	5000
9	esw_cddm	Usages	194	50	70	256	289	208	105	301	194	50	70	256	
10	nx_integration	Usages	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
11	nx_integration	Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
12	plant_struct_edit	Usages	128	44	30	289	269	133	207	272	128	44	30	289	
13	plant_struct_edit	Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
14	platform_des_user	Usages	457	176	2722	1969	1019	454	300	522	292	1986	1899	528	2678
15	platform_des_user	Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
16	prog_exec_user	Usages	1026	254	504	2008	382	731	252	208	1364	1168	338	1063	2863
17	prog_exec_user	Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
18	requirements_user	Usages	157	41	349	132	280	242	42	94	283	292	129	171	258
19	requirements_user	Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
20	simulation_user	Usages	1865	725	4100	3560	3640	2985	615	935	2945	2920	3155	2625	4860
21	simulation_user	Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
22	teamcenter_author	Usages	1382	369	2895	3510	2912	1466	423	568	981	2143	3004	1641	3076
23	teamcenter_author	Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
24	visview_base	Usages	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
25	visview_base	Purchased	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000

Example module usage report

Note:

The module **Usages** numbers are accurate for all reported months. However, the **Purchased** numbers are not stored in the Teamcenter database; all months for **Purchased** licenses show the numbers pulled from the current license file at the time the report is generated. Therefore in the following situations the purchased numbers in the report are not historically accurate:

- License modules have been added to or deleted from the server's license file between usage and running of the report.

- The number of module licenses in the server's license file has increased or decreased.
- A module has expired in the server's license file between usage and running of the report.

License login (location) report

When geography access has been configured for Teamcenter, you can run a license login report to analyze the locations from which a user logs on. The report displays one line per user per login region and includes the following columns of information:

Column header	Description
user id	The Teamcenter user ID.
month	The month reported.
user declared geography	The user-reported region or country from which the logon occurred.
ip address	The IP address from which the logon occurred.
login count	The total number of times that the user logged on from a region or country.
last login time	The most recent time that the user logged on.
license status message	Messages regarding user and site geography.

	A	B	C	D	E	F	G
1	user id	month	user declared geography	ip address	login count	last login time	license status message
2	user1	Sep-19 BR		xxx.xxx.xxx.x>	1	2019-09-05T16:12:40Z	User declared geography and Site geography attributes are different.
3	tc_author_1	Sep-19 VI		xxx.xxx.xxx.x>	1	2019-09-05T16:10:13Z	User declared geography and Site geography attributes are different.
4	user1	Sep-19 US		xxx.xxx.xxx.x>	2	2019-09-05T16:05:08Z	
5	consumer	Sep-19 UY		xxx.xxx.xxx.x>	2	2019-09-05T15:42:11Z	User Declared Geography and/or Site Geography attribute is blank
6							

License login report saved to local file system and opened in a spreadsheet.

25. Configuring external management of passwords

Enabling external password management for Teamcenter

Administrators can install Security Services to enable user password management by an external identity service provider, rather than by Teamcenter. All other user administration tasks remain within Teamcenter.

Security Services is optional. To enable external password management, the Security Services services must be installed and an external identity service provider must be configured.

Configuring Teamcenter clients for Security Services

After Security Services is installed and the external identity service provider configured, configure Security Services for Teamcenter by manually entering the Security Services environment variables in the **tc_profilevars** properties file and the rich client properties in the **site_specific.properties** file.

To enable Security Services for Teamcenter clients other than the rich client, such as command line utilities and ODBC Driver, set the **TC_SSO_SERVICE** environment variable. When this environment variable is set, user authentication is performed by the external identity service provider (the user must enter the password defined in the external identity service provider); otherwise, the logon is unchanged.

To enable Security Services for the rich client, set the **TC_SSO_SERVICE** and **TC_SSO_LOGIN_URL** environment variables. For administrative purposes, the administrator can choose not to set the **TC_SSO_LOGIN_URL** environment variable.

In the rich client and ITK, the ability to set or change a password in the Teamcenter database is disabled when the **TC_SSO_SERVICE** environment variable is set. The **Change Password** command remains available from the **Actions** menu. In the Organization application, the password text box and blank password check box are disabled.

The following table lists the logon scenarios for users logging on to the Teamcenter rich client before logging on to any other Teamcenter product.

Environment variables set	Logon scenario
No Security Services environment variables are set.	Security Services is not enabled. User authentication is performed by Teamcenter: The rich client displays its standard logon dialog box; users enter the user IDs and passwords defined in Teamcenter.

Environment variables set	Logon scenario
TC_SSO_SERVICE is set in the tc_profilevars properties file. TC_SSO_LOGIN_URL is set in the client_specific.properties file (two-tier installation). For a four-tier installation, it is in the site_specific.properties file.	• To access other Teamcenter products, the rich client user must log on again. Security Services is enabled for the rich client and server. User authentication is performed by external identity service provider: • Security Services displays a browser logon page and starts a Security Services user session. Users enter user IDs defined in the Teamcenter database, but passwords are defined in the external identity service provider. If applet-free mode is configured, the sign-on process is performed without a Web browser and session applet. This mode requires HTTP 401-based authentication. If zero sign-on is configured using Kerberos authentication, no Teamcenter logon challenges appear. Zero sign-on in applet free mode requires HTTP 401-negotiate authentication. For information about configuring Kerberos authentication, see <i>Security Services Configuration</i>. For information about the results of different logon configurations, see <i>Teamcenter Basics</i>. • The logon uses default values for Teamcenter group and database. Users can change the group default using the User Settings dialog box; the default database can be changed using the HTTP_SERVER_n.URI property in the rich client site_specific.properties file. • Users can access other Teamcenter products that implement Security Services without logging on again during a Security Services user session.
TC_SSO_SERVICE is set in the tc_profilevars properties file.	Security Services is enabled for the rich client server only. User authentication is performed by the external identity service provider: • Security Services does not start a user Security Services session; the rich client displays its standard logon dialog box, but users enter passwords defined in the external identity service provider. • To access other Teamcenter products, user must log on again.

Environment variables set	Logon scenario
	A Security Services session is essential for logging on to other products using authentication by the external identity service provider. This configuration is for the following special uses: - Administrators can start multiple sessions as different user IDs to act on behalf of several users. - Users can log on using a group or database other than the default values.

When one external identity provider serves multiple Teamcenter sites, and each site authorizes a different set of users, the administrator must set the **TC_SSO_APP_ID** environment variable to identify the site. This site ID must match a site ID registered with the Security Services service. This environment variable is optional; when used, it must be set in both the **tc_profilevars** file for the server and the **site_specific.properties** file for the rich client.

When Security Services is installed and enabled, the rich client user logon defaults to the Teamcenter database, behaving as if users had entered user IDs and passwords in its standard logon dialog box but left the database blank. The default database is the database defined by the **HTTP_SERVER_n.URI** property (for four-tier rich client configurations) in the **client_specific.properties** file. This property is set using Teamcenter Environment Manager: the HTTP-based protocol enables SOAP-based HTTP communication and is implemented for a four-tier configuration. To specify a different default database for Security Services logons, you must manually modify the appropriate property in the rich client **site_specific.properties** file.

Configuring load balancer timeouts

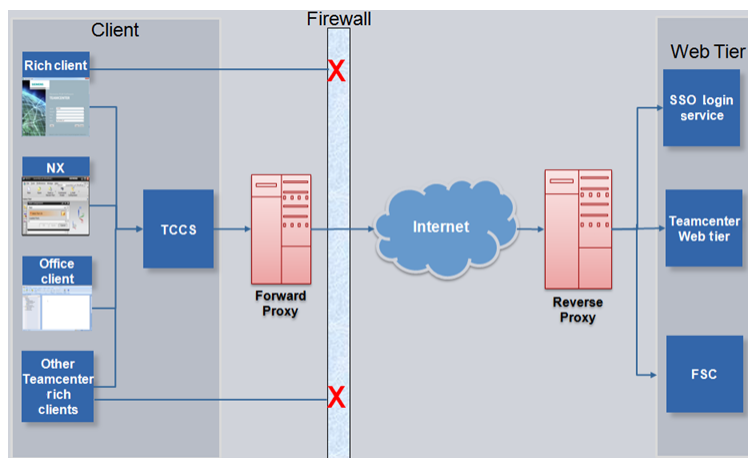
If you use a third-party load balancer, ensure its timeout setting values are equal to or greater than the timeout settings for Teamcenter as entered in the Web Application Manager **Session Timeout** box. This indicates how many minutes of inactivity are permitted before the web server terminates the session on the web server. From the **Modify Web Application** dialog box, click **Modify Web Application Information**. The Web Application Manager displays the **Modify Web Application Information** dialog box for the **Session Timeout** box.

The Teamcenter web tier and Teamcenter Security Services Login Service maintain client session information. When deployed behind a load balancer, it is important that all requests from a given client are routed to the same back-end web tier or Login Service instance. Ensure that you set the load balancer's session timeout interval to a value equal to or greater than the Teamcenter session timeout values. Also, load balancers typically have a stickiness or affinity setting, and this must be set as well in the load balancer configuration for these Teamcenter web applications. Otherwise, the load balancer timeout eclipses the Teamcenter timeout and can lead to apparently random and unexpected behavior as the load balancer switches between active web application instances.

26. Administering Teamcenter client communication system (TCCS)

Introduction to Teamcenter client communication system (TCCS)

As an enterprise class application, Teamcenter works within company infrastructure and security policies. Teamcenter accommodates company infrastructure such as forward proxies, reverse proxies, commercial single sign-on (SSO) integrations, and secure sockets layer (SSL), among others. Teamcenter client communication system (TCCS) is a centralized tool to configure these points of contact between a company's infrastructure and Teamcenter.



TCCS includes the following features for supported clients:

- **Forward proxy support**

A *forward proxy* server takes requests from an internal network and forwards them to the Internet.

TCCS provides centralized forward proxy support to supported Teamcenter clients configured to use TCCS for SOA/web tier requests. TCCS also provides forward proxy credentials sharing and reuse across supported Teamcenter client applications. The FCC also uses TCCS forward proxy support.

- **Reverse proxy support**

A *reverse proxy* server takes requests from the Internet and forwards them to servers in an internal network.

TCCS provides centralized reverse proxy support (for supported proxies) to supported Teamcenter clients configured to use TCCS for SOA/web tier requests. Reverse proxy challenges and cookie storage is managed by Security Services (SSO). Therefore, reverse proxy for WebSEAL is only supported in environments for which SSO is enabled. The FCC also uses TCCS reverse proxy support.

- **Centralized network configuration**

Provides centralized configuration for defining web tier, SSO URLs, and forward proxy configurations. Supported Teamcenter clients refer to a common configuration defined in TCCS. The ability to specify specific environments to be used with a given TCCS connection is provided.

Teamcenter client communication system (TCCS) provides transport-level support to manage communication between the following Teamcenter clients and the web tier or the FMS server cache (FSC):

- Business Modeler IDE
- Client for Office
- Lifecycle Visualization
- NX
- Rich client
- Solid Edge
- TcPMM

TCCS is a container application that contains the following Teamcenter applications:

- **Teamcenter server proxy (TSP)**

TSP manages HTTP communications for Teamcenter server (**TcServer**) requests. It accepts client requests over secured pipes using a proprietary protocol and submits the requests over HTTP to the web tier endpoint. TSP uses the **TcProxyClient** component and forward proxy configuration to support forward and reverse proxy servers.

Use the **tspstat** utility to administer and obtain run-time statistics from TSP.

- **Teamcenter model event manager (TcMEM)**

TcMEM manages event synchronization across service-oriented architecture (SOA) clients sharing the same Teamcenter server instance.

Use the **tcmemstat** utility to administer and obtain run-time statistics from TcMEM.

- **FMS client cache (FCC)**

The FCC provides a private user-level cache, just as web browsers provide a read file cache. It also provides a high-performance cache for both downloaded and uploaded files. The FCC provides proxy interfaces to client programs and connectivity to the server caches and volumes.

The FCC accepts client requests over secure pipe connections and submits them to the appropriate FMS server cache (FSC) process. It uses the **TcProxyClient** component and forward proxy configuration to support forward and reverse proxy servers.

Use the **fccstat** utility to administer and obtain run-time statistics from the FCC.

Note:

TCCS and all its container applications run simultaneously. Starting, stopping, or reconfiguring TCCS (or any of its applications) propagates the same action to all.

TCCS installation is included in a Teamcenter client installation.

Clients using TCCS can use client-certificate authentication through secure sockets layer (SSL) settings.

Tools to configure TCCS

You can configure Teamcenter client communication system (TCCS) using tools such as Teamcenter Environment Manager (TEM), the TCCS installer, and the Client for Office installer. The TCCS configuration options in these tools are similar. The remainder of the topics in *Administering Teamcenter client communication system (TCCS)* show how to use Teamcenter Environment Manager (TEM) to configure TCCS.

Teamcenter Environment Manager (TEM)

In TEM, configuration panels for TCCS appear during installation of 4-tier rich client and the 2 and 4-tier rich client, especially if you choose **Advanced** options. Once TCCS has been installed, in the **Feature Maintenance** panel of TEM, you can use the options in the **Client Communication System** section to configure TCCS.

TCCS stand-alone installer

To install TCCS for a client other than the rich client, such as NX or Lifecycle Visualization, that needs to communicate with a Teamcenter server, from the installation source files run the *tccinst.exe* file located in **additional_applications\tccs_install.zip**.

Client for Office installer

To configure TCCS as part of installing Teamcenter Client for Microsoft Office, from the installation source files run the *setup.exe* file located in **additional_applications\OfficeClient.zip** file.

Configure TCCS for rich client (two-tier and four-tier)

When the **Teamcenter Rich Client (2-tier and 4-tier)** feature is installed, you must set up server pointers in TCCS for both the two-tier client and the four-tier client.

1. In the **Feature Maintenance** panel of Teamcenter Environment Manager (TEM), under **Teamcenter Rich Client (2-tier and 4-tier)** select **Modify settings** and click **Next** until you reach the **Environment Settings for Client Communication System** panel.

The environments entered in this panel are displayed in the list of available environments when a client is launched. Use this panel to set up pointers to both the two-tier client and the four-tier client.

- In the **Environment Settings for Client Communication System** panel, type values in the following table cells:

Name	Specifies the name of the TCCS environment. Assign the TCCS name precisely to indicate a two-tier or four-tier environment.
URI/TC_DATA	Specifies the four-tier URI to the TCCS environment or two-tier <i>TC_DATA</i> location. TEM validates this field whether it is a valid URI or <i>TC_DATA</i> location.
Tag	Specifies a pattern to apply when filtering the list of available TCCS environments.
SSO App ID	Specifies the ID of the Security Services application you use with TCCS.
SSO Login URL	Specifies the URL to the Security Services Login Service application you use with TCCS.
TcServer Character Encoding Canonical Name (2-tier)	Specifies the TcServer character encoding set that the two-tier rich client uses to access the database by canonical name, for instance, Cp1252 . This entry is only needed for two-tier environments. This field is empty for four-tier environments.
Single Server	Specifies whether the connection is to a single server machine (true) or multiple servers in a distributed environment (false).

- To set up additional TCCS features such as forward and reverse proxy, SSL, and Kerberos authentication, click **Advanced**.

A tabbed panel opens in which you can set up the features.

When you have finished setting up the features, click **OK** to close the advanced features panel.

- Click **Next** until you reach the **Confirmation** panel and then click **Start**.

The environment information is saved. If the configuration is shared, the files are saved to the *AllUsersProfile\Siemens\cfg\tccs\Teamcenter\envs* directory. If the configuration is private, the files are saved to the *UserProfile\Siemens\cfg\tccs\Teamcenter\envs* directory.

When you log on to the client, the environments are displayed in a list when a client is launched.

Creating shared or private TCCS configurations

A TCCS configuration contains information about how TCCS connects to the server. A TCCS configuration can be either shared or private. If both private and shared configurations exist for a user, the private configuration takes precedence.

A *shared* configuration is used by all system accounts. Location of the files depends on the operating system:

For this operating system Shared TCCS files are located here

Windows 10	<i>AllUsersProfile\Siemens\cfg</i>
Linux	<i>etc/Siemens/cfg</i>

A *private* configuration is used by a single user. On Windows systems, the files are located in *UserProfile\Siemens\cfg*.

Perform the following steps to create shared or private TCCS configurations using Teamcenter Environment Manager (TEM).

Note:

This procedure describes how to create shared or private configurations using TEM. Depending on your environment, you can also **configure TCCS using other tools** such as the TCCS installer, the Client for Office installer, and web tier context parameters. TCCS configuration options in these tools are similar.

1. In TEM, navigate to the **Feature Maintenance** panel, and under **Client Communication System**, select **Use Configurations and Environments**.
2. In the **Client Communication System Switch** panel, to authorize the system to use the configuration, select **Use Configurations and Environments** and click **Next**.

To set up the configuration without enabling it in the system, leave the check box empty.

Tip:

If you have trouble launching Teamcenter after creating TCCS configurations, try clearing the **Use Configurations and Environments** check box. It could be that an incorrect setting in a configuration is the source of the problem.

3. In the **Configuration Selection for Client Communication System** panel, if you are a TCCS administrator and you want this configuration to be used by multiple users, select **Shared**. By default, the shared configuration is used by the system for all users connecting using TCCS.

If this configuration is for your use only, select **Private**.

A TCCS administrator can create both shared and private configurations. A non-TCCS administrator can only create a private configuration. If both private and shared configurations exist for a user (designated as **existing** in the configuration panel in TEM), the private configuration takes precedence.

4. Click **Next** and continue to set up the configuration (for example, for forward proxy, environments, and so on). Click **Next** until you reach the **Confirmation** panel and then click **Start**.

The configuration information is saved.

Configure TCCS for forward proxy

If Teamcenter client applications must use a forward proxy to access Teamcenter services such as the web tier and the FMS server cache (FSC), then install Teamcenter client communication system (TCCS) on the client machine and perform the following steps in Teamcenter Environment Manager (TEM) to configure TCCS to point to the forward proxy server:

Note:

This procedure describes how to configure TCCS for forward proxy using TEM only. You can also **configure TCCS using other tools** depending on your environment, such as the TCCS installer, the Client for Office installer, and web tier context parameters. The TCCS configuration options in most of these tools are similar.

Companies frequently use forward proxies to control access of clients to the Internet and to cache responses to optimize Internet access. Users in these companies typically set their web browsers to use the proxy server when accessing the Internet. When a Teamcenter client is deployed in such an environment, it must adapt to the requirements of the forward proxy server including submission of the request to a proxy address (in addition to the address for the Teamcenter service) and authentication to the proxy server.

Many companies configure proxy access centrally for all browsers using a mechanism called proxy autoconfiguration (PAC). Their web browsers are then set to download the autoconfiguration at startup. Teamcenter clients leverage this existing central configuration to fit in automatically to your deployment environment.

In TEM, navigate to the **Forward Proxy Settings** view, which is an advanced view of **Client Communication System** settings. You reach the view as part of initial installation of the rich client, or as modification of the **Client Communication System** feature configuration.

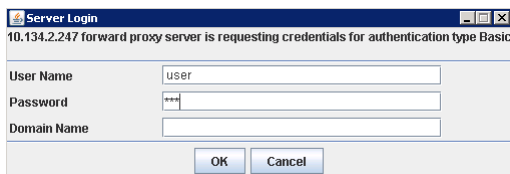
The settings in the **Forward Proxy Settings** view are similar to the settings found in web browsers for configuring proxies. Select the appropriate setting for your environment, adding values as needed.

For this situation	Do this
Your company does not use forward proxy.	Select Do not use forward proxy
Users in your company are required to use their web browser settings to point to the proxy server.	Select Use web browser settings . Selecting this option automatically retrieves the proxy settings from the client's web browser.
Your company uses a Web Proxy Auto-Discovery Protocol (WPAD) to point to the proxy server.	Select Detect settings from network . With WPAD, clients automatically locate a URL of a configuration file using the Dynamic Host Configuration Protocol (DHCP) or the Domain Name System (DNS).

For this situation	Do this						
Your company uses a proxy autoconfiguration (PAC) file to point to the proxy server.	Select Retrieve settings from URL. In the Proxy URL box, type the address where the PAC file is located.						
Your company does not use automatic methods to point to the proxy server.	Select Configure settings manually. <table> <tr> <th>In this situation</th><th>Do this</th></tr> <tr> <td>The same proxy server is used for all forward and reverse proxy requests.</td><td> Select All Protocols Host. Type the host name or IP address of the server in the Host box, and type the server port number in the Port box. </td></tr> <tr> <td>You have separate proxy servers dedicated to HTTP and HTTPS requests.</td><td> Select HTTP Protocols. Select HTTP Host and HTTPS Port and type the host and port information in the provided boxes. </td></tr> </table> In the Exceptions box, type a list of host names or IP addresses that are not proxied. Separate hosts by semicolons. For example, localhost; my_tc; 182 exempts http://localhost:8017/tc, https://my_tc:14327/tc, and http://182.0.0.27:8080/tc.	In this situation	Do this	The same proxy server is used for all forward and reverse proxy requests.	Select All Protocols Host. Type the host name or IP address of the server in the Host box, and type the server port number in the Port box.	You have separate proxy servers dedicated to HTTP and HTTPS requests.	Select HTTP Protocols. Select HTTP Host and HTTPS Port and type the host and port information in the provided boxes.
In this situation	Do this						
The same proxy server is used for all forward and reverse proxy requests.	Select All Protocols Host. Type the host name or IP address of the server in the Host box, and type the server port number in the Port box.						
You have separate proxy servers dedicated to HTTP and HTTPS requests.	Select HTTP Protocols. Select HTTP Host and HTTPS Port and type the host and port information in the provided boxes.						

The forward proxy information is saved in the **fwdproxy_cfg.properties** file.

To test forward proxy settings, you can launch the rich client and select a single sign-on (SSO) environment for which forward proxy is set. A forward proxy challenge dialog box is displayed.



Configure TCCS for reverse proxy form-based authentication

If Teamcenter client applications must connect to the Teamcenter web tier via a reverse proxy server that requires form-based authentication, and if the reverse proxy server is other than WebSEAL, then you must either configure TCCS with advanced reverse proxy criteria to recognize the form-based challenge, or disable reverse proxy criteria checking. Failure to do this may lead to errors about invalid SOA responses.

Note:

This procedure describes how to use TEM to configure TCCS reverse proxy settings. Depending on your environment, you can **configure TCCS using other tools** such as the TCCS installer, the Client for Office installer, and web tier context parameters. The TCCS configuration options in these tools are similar.

1. In TEM, navigate to the **Reverse Proxy Settings** view, which is an advanced view of **Client Communication System** settings.

You reach the view as part of initial installation of the rich client, or as modification of the **Client Communication System** feature configuration.

2. Depending on your situation, either configure TCCS with criteria to recognize the form-based challenge, or disable reverse proxy criteria checking.

In this situation	Do this
- Your reverse proxy performs form-based authentication using SiteMinder or another server type that does not send specific header information in the type 200 form-based challenge. - You are troubleshooting communication errors, your reverse proxy requires form-based authentication, and you suspect invalid criteria settings.	Select the Skip reverse proxy criteria check check box. If checking is disabled, then TCCS treats all reverse proxy text/html packages with HTTP status of 200 as form-based challenges that pass criteria checking.
Your reverse proxy is not configured for form-based authentication challenges, thus there is no occasion for TCCS to perform checking.	At your discretion, delete any existing criteria and form actions. Criteria are not needed. Ensure that the Skip reverse proxy criteria check check box is <i>not</i> selected.
Your reverse proxy server performs form-based authentication using WebSEAL, or using another server type that does send specific header information in the type 200 form-based challenge.	Ensure that the Skip reverse proxy criteria check check box is <i>not</i> selected. Add appropriate criteria for identifying the challenge. The Reverse Proxy Settings view includes two groups: Private Reverse Proxy Settings Lists the reverse proxy criteria currently defined. Each row is a criteria XML element defined in the specified format.

In this situation**Do this**

By default, the table is blank and no criteria are defined. A criterion string is of the following form:

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

Criteria Details

Lists the details of a criterion selected in the **Private Reverse Proxy Settings** group. Each criterion must contain at least one header name/header value pair or at least a single form action. To satisfy a criterion, a reverse proxy form-based challenge package must match every criteria element listed within a given criterion.

- a. Next to the **Private Reverse Proxy Settings** group, click **Add**.
- b. Next to the **Criteria Details** group, use the **Add** and **Remove** buttons to add or remove HTTP header names and values for the selected criterion.
- c. In the **Form Action** box, specify a form action.
- d. Click **Apply**.

The criterion is added to the criteria table.

Note:

If you have reached the **Reverse Proxy Settings** view in TEM, and you do deploy the advanced TCCS configuration, TEM saves the configuration to the **reverseproxy_cfg.xml** file, which overrides the default WebSEAL configuration. To ensure that TCCS correctly performs checking if you use WebSEAL, add the following criterion:

Header name	Header value	Form action
server	webseal	/pkmslogin.form

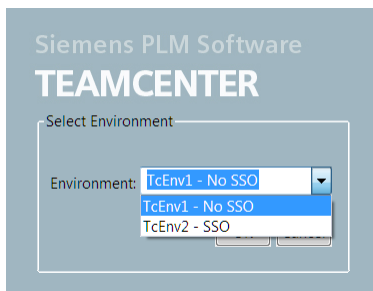
When you are through configuring criteria, click **Next** until you reach the **Confirmation** panel and click **Start**.

The reverse proxy information is saved in the **reverseproxy_config.xml** file.

Configure multiple TCCS environments

A single TCCS configuration can contain multiple environments, providing support for multiple versions or server databases. For example, you may want to install some TCCS environments with Security Services (SSO), some environments without SSO, some environments on one server, and other environments on another server.

If multiple environments are configured, all environments are displayed at rich client logon, allowing the user to select which TCCS environment to use.



Each environment contains one or both of the following service types:

- **sso**

The **sso** service endpoint corresponds to the SSO logon URL. **id** corresponds to **SSO AppID**.

- **tcserver**

The **tcserver** service endpoint is the URL of the web tier deployment.

Perform the following steps to create multiple environments using Teamcenter Environment Manager (TEM).

1. In the **Feature Maintenance** panel of TEM, under **Client Communication System** select **Modify Configurations** and click **Next** until you reach the **Environment Settings for Client Communication System** panel.

The environments entered in this panel are displayed in the list of available environments when a client is launched.

2. To add a TCCS environment, in the **Environment Settings for Client Communication System** panel, click **Add**.
 - a. For each environment, type a value in the **Name** and **URI** boxes.

Note:

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

- b. (Optional) Use the **Tag** box to tag the environment. When installing a rich client, you can optionally provide a **Client Tag Filter** value to filter the list of environments displayed in the rich client to those environments that match the filter.

Example:

You create 10 environments, three on **Server1** with SSO, three on **Server1** without SSO, and four on **Server2**.

Tag the environments **SSO**, **no SSO**, and **Server2**, respectively.

Tip:

You must have already created these tags. To create a tag, see the **Client Tag Filter** panel described later in this procedure.

- c. If you want to add a single sign-on environment, type the values for the single sign-on server in the **SSO App ID** and **SSO Login URL** boxes.

For example, in the **SSO App ID** box, type the value of the **SSO_APPLICATION_ID** context parameter from the web tier installation. In the **SSO Login URL** box, type the value of the **SSO_LOGIN_SERVICE_URL** context parameter.

Tip:

You must have already set up the single sign-on server.

3. Click **Next** until you reach the **Client Tag Filter** panel.

The **Client Tag Filter** value specifies a pattern to apply when clients filter TCCS environments. To specify multiple filter tags, separate the entries with a pipe (|).

Example:

To display only the environments containing the **SSO** tag, type **SSO** in the box.

To display the environments containing the **SSO** and **Server2** tags, type **SSO|Server2** in the box.

When installing a rich client, you can optionally provide a **Client Tag Filter** value to filter the list of environments displayed in the rich client to those environments that match the filter. (In a four-tier

environment, the **Client Tag Filter** context parameter sets the tag.) Environments that do not fit the pattern are not available to the rich client. For example, if the rich client **Client Tag Filter** value is **9.***, all TCCS environments with **Tag** values beginning with **9**. are available to the rich client. Environments with **Tag** values beginning with **10** are not available.

4. Click **Next** until you reach the **Confirmation** panel and then click **Start**.

The environment information is saved. If the configuration is shared, the files are saved to the *AllUsersProfile\Siemens\cfg\tccs\Teamcenter\envs* directory. If the configuration is private, the files are saved to the *UserProfile\Siemens\cfg\tccs\Teamcenter\envs* directory.

Configure TCCS for Kerberos

Kerberos, a network authentication protocol originally developed at MIT, provides authentication for client/server applications by using secret-key cryptography. With the Kerberos protocol, a client proves its identity to a server (and vice versa) across an insecure network connection. Kerberos allows a user to be authenticated without sending the user's password over the network. Instead, encrypted tickets derived from the password and secret key information are sent over the network.

After you install Kerberos, perform the following steps to set up an environment in TCCS so that users can log on to Teamcenter using Kerberos authentication:

In TEM, navigate to the **Kerberos Authentication Settings** view, which is an advanced view of **Client Communication System** settings. You reach the view as part of initial installation of the rich client, or as modification of the **Client Communication System** feature configuration.

1. Select the **Support Kerberos authentication** check box.

Note:

On Windows hosts, the Kerberos configuration file is **C:\Windows\krb5.ini**.

On Linux hosts, the Kerberos configuration file is **etc/krb5.conf**.

2. To always prompt for a Kerberos user name, select the **Always prompt for User ID** check box. If you want to enable zero sign-on functionality on Windows hosts, clear this check box.

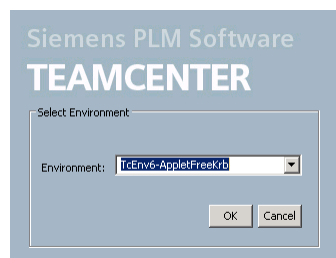
Zero sign-on allows Windows users to launch a Teamcenter client without being prompted to log on to Teamcenter. Zero sign-on functionality requires that you configure Security Services in applet-free mode.

3. Click the **Back** button until you reach the **Environment Settings for Client Communication System** panel.

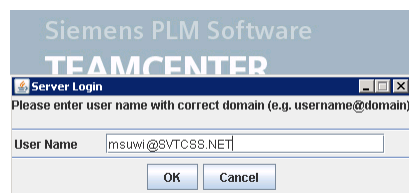
4. In the **Environment Settings for Client Communication System** panel, enter the Kerberos environment. (For applet-free mode, ensure that **/tccs** is appended at the end of the value in the **SSO Login URL** box.)
5. Click **Next** until you reach the **Confirmation** panel and then click **Start**.

The Kerberos environment information is saved.

When users log on, they select an environment from the list of configured environments.



When prompted for a user name, user must correctly enter the user name for the environment. For Kerberos authentication, the domain must be in all uppercase letters. (For example, *username@DOMAIN*).



Configure TCCS for SSL

Secure sockets layer (SSL) is a framework to provide an encrypted connection between the client and the server.

Following are some SSL terms and their meanings:

Cacerts	A file that holds certificates issued by trusted certificate authorities (CAs).
Certificate	A verification tool that consists of a public encryption key with identity information and an expiration date. Clients and servers trust certain certificate authorities (CAs) that issue SSL certificates.
Keys	An encryption key pair that consists of a private key and a corresponding public key. Data encrypted by one can only be decrypted by the other.
Keystore	A file used for storing private keys and certificates and their corresponding public keys.

Truststore A file used for storing certificates from other parties or from trusted certificate authorities. A cacerts file is a truststore.

If you have set up your enterprise to use SSL, perform the following steps to configure TCCS to use SSL:

1. Launch Teamcenter Environment Manager (TEM) in maintenance mode.

For this operating system	Do this
Windows	In the Windows start menu, choose Programs>Teamcenter 14 , and then right-click Environment Manager and choose Run as administrator . You can also run the tem.bat file in the install directory in the application root directory for the Teamcenter installation. Right-click the tem.bat program icon and choose Run as administrator .
Linux	a. Change to the install directory in the Teamcenter application root directory for your Teamcenter installation. b. Run the tem.sh script.

2. In the **Maintenance** panel, choose **Configuration Manager**.
3. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**.
4. In the **Old Configuration** panel, select the configuration that contains the four-tier rich client.
5. In the **Feature Maintenance** panel, under **Client Communication System**, select **Use Configurations and Environments**.
6. Proceed to the **Secure Socket Layer (SSL) Settings** panel and select one of the following:
 - **Use Internet Explorer Certificate Store (Recommended)**

Uses certificates stored in Microsoft Internet Explorer. This option is available only on Windows hosts.
 - **Disable SSL**

Turns off SSL.
 - **Configure Certificate Store Manually**

Allows you to certificate store you want to use.

- a. Select **Use trust store (supported type is JKS)** to use an established repository file of trusted certificates, such as Java KeyStore. Click the ellipse button in the **File** box to point to the file. This populates the **truststore** element in the **tccs.xml** file. (When this is unspecified, Java uses the standard Java Development Kit (JDK) CA keystore in the **jre/lib/security/cacerts** folder.)
- b. Select **Accept untrusted certificates** to allow self-signed certificates from untrusted parties. This sets the **allowuntrustedcertificates** element in the **tccs.xml** file to **true**.

Select this option if you do not want to provide warnings for untrusted certificates.

- c. Select **Use Key Store** to use a keystore file that holds private keys and certificates and their corresponding public keys. Click the ellipse button in the **File** box to point to the keystore file. This populates the **keystore** element in the **tccs.xml** file. Click the arrow in the **Type** box to select **JKS** or **PKCS12** as the keystore file format.

7. Click **Next** until you arrive at the **Confirmation** panel and then click **Start**.

The SSL settings are saved.

8. Verify the following SSL settings in the **tccs.xml** file:

This setting	Specifies this
keystore	The path to a key store file containing a user's private keys.
keystorepassword	The password for the key store file.
truststore	The path to a trust store file that hold trusted certificates. When this is unspecified, Java uses a standard JDK file cacerts.
truststorepassword	The password for the trust store.
allowuntrustedcertificates	Allows self-signed certificates from untrusted parties.

Modify four-tier server connection settings

You can change the list of web servers available for connection from the four-tier rich client.

1. In the **Feature Maintenance** panel of TEM, under **Client Communication System**, select **Modify 4-tier server settings** and click **Next**.
2. In the **4-tier server configurations** panel, in the **4-tier Servers** list, edit the list.
3. Click **Next** until you reach the **Confirmation** panel and then click **Start**.

TEM applies the four-tier server settings.

Modify FCC parent settings

You can specify the FMS server caches (FSCs) used by the FMS client cache (FCC). The FCC can have multiple parent FSCs to use in case of failover. FSCs are used in the priority you specify.

1. In the **Feature Maintenance** panel of TEM, under **Client Communication Setting**, select **Modify FCC Parent settings** and click **Next**.

The **FCC Parents** panel is displayed.

2. From the **FSC assignment mode** list, select the method to assign FSCs:

- **clientmap**

Routes FCC data requests to the **assignedfsc** elements specified within the **clientmap** section of the **fmsmaster_fscid.xml** configuration file.

- **parentfsc**

Routes FCC data requests to the list of **parentfsc** elements specified in the **fcc.xml** configuration file.

Use this setting when the default client mapping setting is not efficient.

3. In the **FCC Parents** table, add the FSC parents and set their priority. This table specifies from which FSCs to download the FMS configuration information. FSCs are used in the priority you specify. You may specify multiple parent FSCs to provide failover.

The values entered are placed into the **parentfsc** node of the **FMS_HOME/fcc.xml** file.

4. Click **Next** until you reach the **Confirmation** panel and then click **Start**.

The FCC parent settings are saved.

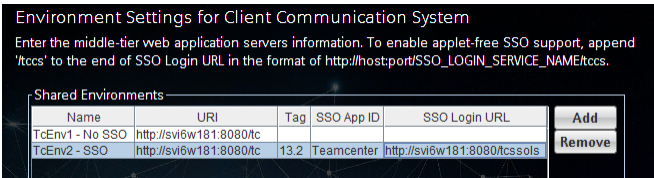
5. Verify the settings in the **TC_ROOT\tccs\fcc.xml** file:

```
<!-- default parentfsc - this is a marker that will be overwritten by the installer
-->
<parentfsc address="http://SVI6W181:4544/" priority="0" />
<parentfsc address="http://AHIU351:4544/" priority="1" />
<parentfsc address="http://LLB3W067:4544/" priority="2" />
```

Find Security Services settings for use in TCCS

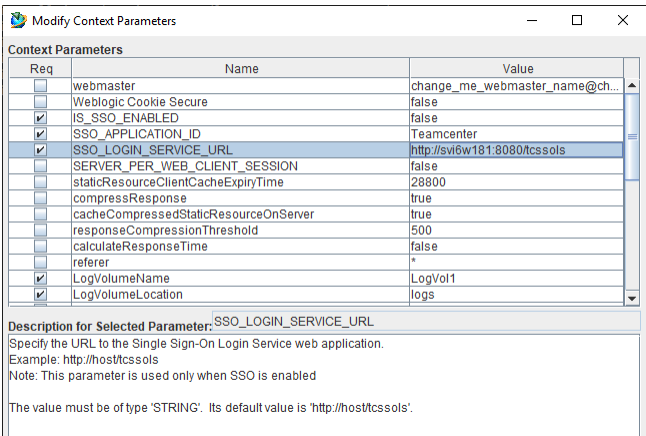
Teamcenter Security Services works in conjunction with Teamcenter client communication system (TCCS). When you configure TCCS, you must point to settings for Security Services.

For example, in the **Environment Settings for Client Communication System** panel, you must type Security Services settings in the **SSO App ID** and the **SSO Login URL** boxes.



The settings that are used with TCCS are configured when Security Services is first installed.

To review the settings, in the Web Application Manager, open the context parameters panel.



This context parameter value in the Web Application Manager	Corresponding value in the Environment Settings for Client Communication System panel
SSO_APPLICATION_ID	SSO App ID
SSO_LOGIN_SERVICE_URL	SSO Login URL

TCCS configuration files

Orientation to TCCS configuration files

After you **install TCCS**, you can configure TCCS using the following configuration files:

tccs.xml

Configures TCCS settings. This file includes a list of applications configured with TCCS and HTTP configuration values.

fdwproxy_cfg.properties

Configures forward proxy settings.

reverseproxy_config.xml

Configures reverse proxy settings.

TCCS configuration files location

When TCCS starts, it looks for TCCS configuration files in the location concatenated as a combination of parent and child folders. The folder locations can be set by the following environment variables.

TCCS_CONFIG_HOME

Sets the path to the TCCS configurations home directory (the parent folder).

OS	Default TCCS_CONFIG_HOME value
Windows	For a private installation: %USERPROFILE%\Siemens\cfg\tccs For a shared installation: %ALLUSERSPROFILE%\Siemens\cfg\tccs
Linux	For a private installation: \$HOME/Siemens/cfg/tccs For a shared installation: /etc/Siemens/cfg/tccs

TCCS_CONFIG

Sets the name of the TCCS configuration directory containing the TCCS environment configuration files (the child folder).

If TCCS_CONFIG is undefined, the default child folder value is **Teamcenter**.

When TCCS is upgraded or installed as part of a client, configuration files are always placed in the default location, even if TCCS_CONFIG_HOME or TCCS_CONFIG is set to other than the default value. In that case, adjust either the environment variable values or the file locations so that TCCS finds the correct files.

tccs.xml file

The **tccs.xml** file defines TCCS configuration settings.

The **tccs.xml** file contains the following sections:

- **List of applications configured with TCCS**

This section contains the list of hosted TCCS components and the startup information for each component.

- **Max idle time in minutes for TCCS before it shuts down**

The **maxidletime** attribute indicates the maximum time (in minutes) the application is idle before shutting itself down. The default idle time is 240 minutes.

- **Default HTTP Configuration**

The default HTTP configuration settings work for most Teamcenter environments. Components with the **initialize** attribute set to **true** are initialized during TCCS startup. These components apply to all the TCCS container applications using **TcProxyClient** transports, such as the FCC and the Teamcenter server proxy.

Note:

Alternative settings of the same components in the **tcserverproxy.xml** file take precedence for Teamcenter server proxy behavior.

- **allowchunking**

Allows the HTTP message body to be transmitted to the client as chunks that are stamped with the size of the chunks.

The default value for the attribute is **false**. This value must be **false** for WebSEAL proxy servers when you are using an IIS web server because the IIS web server does not support chunking. This value must be **false** when using a Squid proxy server because the Squid proxy server does not fully support HTTP version 1.1, which is the version that supports chunking.

- **allowuntrustedcertificates**

Allows TCCS to accept server certificates that are not signed by a trusted certificate authority.

- **connectiontimeout**

Sets the time period that a connection remains idle before it is closed.

- **httpversion**

Indicates the HTTP protocol version. The default is version **1.1**.

Note:

HTTP version 1.1 supports chunking; version 1.0 does not.

- **keystore**

Sets the path to the Java keystore containing the user's own certificate and private key used for two-way SSL. The value is set into the Java system's **javax.net.ssl.keyStore** property.

- **keystorepassword**

Specifies the password for the Java keystore. The value is set into the Java system's **javax.net.ssl.keyStorePassword** property.

- **maxconnectionsperhost**

Sets the maximum number of connections to the proxy server from a single host. The default value is **8**.

- **maxretriesreverseproxy**

Set the maximum number of times an HTTP request is attempted when receiving an authentication challenge from a reverse proxy server.

- **sslEnabled**

Sets whether secure sockets layer (SSL) is enabled. This attribute is optional and appears only when the installer selects one of the following on the **Secure Socket Layer (SSL) Settings** panel in Teamcenter Environment Manager (TEM):

- **Disable SSL**

- **Accept untrusted certificates**

- **Configure keystore**

Note:

The application assumes that SSL is enabled regardless of the **sslEnabled** tag in the XML file when the value of the **allowuntrustedcertificates** setting is false, the keystore and truststore paths are not specified, and SSL is not disabled.

- **sockettimeout**

Sets the maximum amount of time (in milliseconds) the **HttpClient** component (the TCCS HTTP transport library) waits for data when executing the method. A value of zero specifies an infinite time-out.

- **stalechecking**

Enables the **HttpClient** component (the TCCS HTTP transport library) to determine if the active connection is stale before executing a request.

Typically, stale checking is disabled to improve performance.

- **totalmaxconnections**

Sets the maximum number of connections to the proxy server from all hosts.

- **truststore**

Specifies the path to the Java keystore containing the certificate authority (CA) certificates trusted by the user. The value is set into the Java system's **javax.net.ssl.trustStore** property. If not specified, the Java default trust store is used.

- **truststorepassword**

Specifies the password for the Java trust store. The value is set into the Java system's **javax.net.ssl.trustStorePassword** property.

- **usesinglecookieheader**

When submitting multiple HTTP cookies to a server, allows TCCS to place all cookies in a single cookie header rather than using multiple cookie headers. Set to **true** to allow submitting a single cookie header. This is useful when working with HTTP servers that cannot process multiple cookie headers correctly.

- **Default Kerberos configuration**

If Kerberos authentication is configured, the following components are listed.

- **kerberosconfig**

Set **enable** to **true** to configure support for Kerberos authentication in TCCS. The default value is **false**.

- **alwayspromptforusername**

Determines whether TCCS prompts users for their user name with Kerberos authentication. Set to **true** for users to be prompted for their user name, preventing zero sign-on functionality. Set to **false** for the user name to be automatically obtained from the operating system logon, enabling zero signon functionality.

The default value is **false**. This attribute is only valid when **kerberosconfig enable** is set to **true**.

- **krb5path value**

Specifies the path to the **Krb5** file used with Kerberos authentication. This must be the absolute path to the **Krb5** file including the file name. If no value is provided, the **Krb5** file is located in the default location as specified in the Sun Java documentation.

fwdproxy_cfg.properties file

The **fwdproxy_cfg.properties** file contains forward proxy-related configuration properties used by TCCS for server requests. All these values are set in the Teamcenter Environment Manager (TEM) **Forward Proxy Settings** panel.

To view the panel in TEM, in the **Feature Maintenance Panel** under **Client Communication System**, select **Modify Configurations** and click **Next** until you reach the **Forward Proxy Settings** panel.

Note:

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

The file contains the following properties:

tcproxy.connection.type	Determines the type of connection for forward proxy servers. The default value is direct, indicating there is no forward proxy server. Other valid values for this property are: browser Uses the web browser proxy settings. If you have more than one browser installed, your designated default browser is used. network Detects the proxy settings from the network the client is on. url Retrieves the proxy autoconfiguration (PAC) file from the URL set in the tcproxy.connection.url property. manual Uses the proxy settings set in the manual configuration properties. These are the properties prefixed with tcproxy.connection.manual in this file. The manual configuration properties are ignored if the tcproxy.connection.type attribute is not set to manual.
tcproxy.connection.url	Sets the URL the TCCS application uses to retrieve the PAC file. This property must have a value if the tcproxy.connection.type attribute is set to url.
tcproxy.connection.manual.all.host	Sets the host name or IP address for the proxy server for all protocols. If you set a value for this property, the following protocol host and port properties are ignored:

tcproxy.connection.manual.all.port

Sets the port number for the proxy server for all protocols.

tcproxy.connection.manual.http.host

Sets the host name or IP address for the proxy server for the HTTP protocol. If using manual configuration properties and a value for this property is not provided, HTTP requests attempt to connect directly to the server host.

tcproxy.connection.manual.http.port

Sets the port number for the proxy server for the HTTP protocol.

tcproxy.connection.manual.https.host

Sets the host name or IP address for proxy server for the HTTPS protocol. If using manual configuration properties and a value for this property is not provided, HTTPS requests attempt to connect directly to the server host.

tcproxy.connection.manual.https.port

Sets the port number for the proxy server for the HTTPS protocol.

tcproxy.connection.manual.socks.host

Sets the host name or IP address for proxy server for the SOCKS protocol. If using manual configuration properties and a value for this property is not provided, the SOCKS protocol requests attempt to connect directly to the server host.

tcproxy.connection.manual.socks.port

Sets the port number for the proxy server for the SOCKS protocol.

tcproxy.connection.manual.exceptions

Contains a semicolon-delimited list of hosts where direct connections can be attempted.

tcproxy.advanced.address_caching

Indicates whether resolved proxy addresses are cached based on their host and port. Valid values are **on** and **off** (case-sensitive)

tcproxy.advanced.retry_delay

Specifies the number of minutes to wait before retrying a connection to a proxy server that failed a client connection. Valid values are any positive integer.

reverseproxy_config.xml file

The **reverseproxy_config.xml** file contains a list of criteria used for detecting a form challenge from a reverse proxy server.

A **criteria** element requires one or more **header** elements and can contain zero or one **form** element. Default **criteria** elements are provided for WebSEAL and SiteMinder. You can edit these or add additional criteria elements, for example:

```
<criteria active="true">
  <header name="server" value="Webseal"/>
  <header name="challengeType" value="Form"/>
  <form action="pkmsloginform"/>
</criteria>
```

The criteria list of values can be set in the Teamcenter Environment Manager (TEM) **Reverse Proxy Settings** panel. To view the panel in TEM, in the **Feature Maintenance Panel** under **Client Communication System**, select **Modify Configurations** and click **Next** until you reach the **Reverse Proxy Settings** panel.

TCCS and container applications

Administering the TCCS container

The TCCS container reads TCCS configuration files on startup and launches the container applications enabled in the configuration.

Enable each container application to run within TCCS by setting its **initialize** parameter to **true** in the **tccs.xml** file, for example:

```
<tccsconfig>
  <!--List of applications configured with TCCS -->
  <tccsapps>
    <tccsappconfig name="TcServerProxy" initialize="true" configfile="tcserverproxy.xml"/>
    <tccsappconfig name="FMSClntCache" initialize="true" configfile="fcc.xml"/>
    <tccsappconfig name="TcModelEventManager" initialize="true" configfile=""/>
  </tccsapps>
```

Consider the following behavior when working with the TCCS container:

- **TcProxyClient** component

The container initializes the **TcProxyClient** component that is the forward and reverse proxy support library for TCCS.

- Idle shutdown

TCCS checks each application to determine if it is idle, shutting down when all applications are idle for a configured amount of time. By default, this is set to 240 minutes. If set to zero, TCCS does not automatically shut down. It continues to run until manually shut down.

- Fatal application errors

Each application shuts down the TCCS container if it encounters an unrecoverable fatal error.

- Restart

You must restart TCCS after making changes in TCCS configurations, such as forward proxy, server end points, and HTTP parameters.

Note:

TCCS and all its container applications run simultaneously. Starting, stopping, or reconfiguring TCCS (or any of its applications) propagates the same action to all.

- TCCS lock file

The inappropriate shutdown of TCCS can cause the lock file to become stuck.

If the lock file has become stuck, remove the TCCS lock file. It is stored in the `%USERPROFILE%\user-name_lock_host-name` directory (Windows) or `HOME/.user-name_lock_host-name` directory (Linux).

The file name is `TCCS_CONFIG.lck`. By default, the `TCCS_CONFIG` environment variable is set to **Teamcenter**.

For example:

```
C:\Users\smith\.smith_lock_host123\Teamcenter.lck
```

Administering the Teamcenter server proxy

The Teamcenter server proxy manages HTTP communications for Teamcenter server (**TcServer**) requests. It accepts client requests over secured pipes using a proprietary protocol and submits the requests over HTTP to the web tier endpoint.

Use the **tcserverproxy.xml** file to set HTTP configuration elements related to Teamcenter server proxy behavior. (The file is located in the `USERPROFILE\Siemens\cfg\tccs\TCCS_CONFIG` directory.) These settings override any corresponding **tccs.xml** file settings applied to Teamcenter server proxy behavior. (Settings in the **tcserverproxy.xml** file do not affect corresponding **tccs.xml** file settings as applied to FCC behavior.)

Use the **tspsstat** utility to administer and obtain run-time statistics from the **TcServerProxy** component.

Note:

TCCS and all its container applications run simultaneously. Starting, stopping, or reconfiguring TCCS (or any of its applications) propagates the same action to all.

The **tcserversproxy.xml** file contains HTTP configuration elements related to the **TcServerProxy** component. These settings override any corresponding **tccs.xml** file settings.

It contains the following configuration elements:

- **maxconnectionsperhost**

Sets the maximum number of connections through each proxy server to a single host. The default value is **8**.

- **totalmaxconnections**

Sets the maximum number of connections through each proxy server to all hosts. The default value is **10**.

- **connectiontimeout**

Sets the time period that a connection remains idle before it is closed. The default value is **30000**.

- **sockettimeout**

Sets the maximum amount of time (in milliseconds) the **HttpClient** component (the TCCS HTTP transport library) waits for data when executing the method. The default value is **0**, which specifies an infinite time-out.

- **stalechecking**

Enables the **HttpClient** component (the TCCS HTTP transport library) to perform determine if the active connection is stale before executing a request. The default value is **false**.

Typically, stale checking is disabled to improve performance.

- **maxretriesreverseproxy**

Sets the maximum number of times an HTTP request is attempted when receiving an authentication challenge from a reverse proxy server. The default value is **5**.

- **idleconntimeoutms**

Sets the time period (in milliseconds) that a connection remains idle before it is closed. The default value is **30000**.

- **idleconnintervalms**

Sets the time period (in milliseconds) for the interval between closing each idle server connection. The default value is **10000**.

- **httpversion**

Indicates the HTTP protocol version. The default value is version **1.1**.

Note:

HTTP version 1.1 supports chunking, version 1.0 does not.

Administering the FCC for TCCS

The FMS client cache (FCC) provides a private user-level cache, just as web browsers provide a read file cache. The FCC also provides a high-performance cache for both downloaded and uploaded files. The FCC provides proxy interfaces to client programs and connectivity to the server caches and volumes.

- Use the **fcc.xml** file to manage FCC behavior, such as connection time-outs and the maximum number of retries.
- Use the **fccstat** utility to administer and obtain run-time statistics from the FCC.

Consider the following behaviors when working with the FCC as it runs with TCCS:

- Setting the **initialize** parameter of the **FMSClientCache** element to **true** in the **tccs.xml** file enables FCC functionality when TCCS starts. This is the default value.
- The only **tccs.xml** file the FCC accesses is the one stored in the TCCS configuration directory. Any other **tccs.xml** files are ignored.
- HTTP configuration parameters in the **tccs.xml** file are also applied to the FCC connections to an FMS server cache (FSC).

Note:

TCCS and all its container applications run simultaneously. Starting, stopping, or reconfiguring TCCS (or any of its applications) propagates the same action to all.

Note:

Native FCC does not run in the TCCS container.

Administering TcMEM

The Teamcenter model event manager (TcMEM) manages event synchronization across SOA clients sharing the same Teamcenter server instance.

Use the **tcmemstat** utility to administer and obtain run-time statistics from TcMEM.

Note:

TCCS and all its container applications run simultaneously. Starting, stopping, or reconfiguring TCCS (or any of its applications) propagates the same action to all.

Administering proxy support for clients not integrated with TCCS

Not all Siemens Digital Industries Software clients are integrated with TCCS, including:

- Teamcenter Systems Engineering

Consider the following when administering proxy support for clients not integrated with TCCS:

- Forward proxy support

A non-integrated client does not leverage the TCCS ability to send requests using a forward proxy server. Unless the client has its own support for forward proxy, the client must connect directly to any Teamcenter services it requires. A client with its own forward proxy support does not leverage the TCCS proxy configuration and must be separately configured.

Note:

Browser-based clients do not integrate with TCCS but do leverage the browser's capabilities and configuration.

- Reverse proxy support

A non-integrated client does not leverage the TCCS ability to recognize authentication challenges from a reverse proxy server such as WebSEAL.

If the Teamcenter web tier is protected by an authenticating reverse proxy server, the client must have its own support for recognizing and responding to authentication challenges. There is limited support for this recognition and response in some clients, including those based exclusively on the Teamcenter SOA client framework.

If only Security Services is protected by an authenticating reverse proxy server, a client does not directly receive authentication challenges from the reverse proxy. The challenges come only to the browser and applet used for **TcSS** authentication and session management. The browser and applet are capable of responding to these challenges.

TCCS logging

You can examine TCCS log files to troubleshoot problems. TCCS log files are stored in:

- Windows systems:

```
%USERPROFILE%\user-name\Siemens\logs\tccs\TCCS\process\
tccs_os-user-name_%TCCS_CONFIG_host-name.log
```

- Linux systems:

```
$HOME/user-name/Siemens/logs/tccs/TCCS/process/
tccs_os-user-name_$TCCS_CONFIG_host-name.log
```

Users can change the log file location by setting the **LOG_VOLUME_LOCATION** environment variable to a different location.

There are two logging configuration files, stored in the same location as the TCCS startup script.

- **log.properties**

When TCCS starts, it looks for this file's location in the **classpath**. The location is specified by the **LogVolumeLocation** parameter, which points to the **logs** value, which specifies the relative path to the TCCS startup script.

TCCS log files are output to the **TCCS/process** subdirectory within the logs directory. Users can change the location that TCCS log files are stored by setting the **LOG_VOLUME_LOCATION** environment variable. In which case, the TCCS log files are written to **LOG_VOLUME_LOCATION/TCCS/process**.

The **log.properties** file contains the **LogConfigLocation** parameter, which specifies the name of the logger definition file, stored in the same directory as the **log.properties** file. By default, this is the **log4j2.xml** file.

- **log4j2.xml**

This file contains an Appenders section and an MLD Loggers section. The appenders are referenced by the logger definitions. Users can add additional loggers to the file. The following must be specified for each logger entry:

- **logger name**

Specifies the name of the MLD logger.

- **level value**

Specifies the level of logging performed by the specified logger. See the following table for default logging levels.

By default, all logging levels are set to **warn**.

- **appender-ref**

Specifies the appender to reference.

For example:

```
<logger name="com.teamcenter.net.tcserverproxy" additivity="false">
  <level value="warn"/>
  <appender-ref ref="TCCSAppender"/>
  <appender-ref ref="ProcessConsoleAppender"/>
</logger>
```

Logging level name	Description
fatal	Logs only fatal errors. Fatal errors typically result in application shutdown.
error	Logs all errors.
warn	Logs all warnings and errors.
info	Includes debugging logs along with all warnings and errors.
debug	Includes debug log levels and debugging logs, along with all warnings and errors.

Users can define their own custom **log4j** configuration. This allows them to supply their own configuration without affecting other users sharing the same TCCS deployment environment. In this case, the **LOG_CONFIG_LOCATION** environment variable must be set to specify the location of the custom **log4j** file.

27. Volume Management application

Getting started with Volume Management

Introduction to Volume Management

Use the Volume Management application to monitor volume usage and improve Teamcenter performance by moving infrequently used data from primary storage space to either destination volumes or to third-party storage systems. All data continues to appear online to users, whether the data is stored online, near-line, or offline.

The **Volume Management** perspective consists of two views.

Volume Management view

Displays by default when you open the **Volume Management** perspective. If you close this view to better display the **Volume Monitor** view, you can open it again by choosing **Window→Show View→Volume Management**.

Use the **Volume Monitor** view to monitor volume usage and balance volume resources by reassigning default volumes for specific users or groups. After migration policies are created, you can use the **Volume Management** view to:

- Preview the impact of migration policies.
- Use the FMS server cache (FSC) process to migrate volume files from a source volume to a destination volume.
- Export pending migration file sets to third-party hierarchical storage management (HSM) systems to migrate/purge files from the online primary tier to near-line secondary tiers or offline tertiary tiers.
- Generate data migration reports.

Volume Monitor view

To open the **Volume Monitor** view, from the **Volume Management** perspective choose **Window→Show View→Volume Monitor**.

Use the **Volume Management** view to create migration policies by applying migration options and metadata query options to a specific policy.

Tip:

Use *volume management (VM) migration policies* to migrate and consolidate Teamcenter volume files and to migrate infrequently used Teamcenter files off of source volumes and on to destination volumes.

Use *hierarchical storage management (HSM) migration policies* to move lower priority files to third-party storage systems. Files stored on secondary and tertiary tiers can be read without moving them to the highest level.

Before you begin**Prerequisites**

- You need Teamcenter administrator privileges to use the Volume Management application.
- If you intend to use hierarchical storage management (HSM), you need a third-party HSM tool.

Enable Volume Management

Volume Management does not need to be enabled before you use it.

If you have trouble accessing Volume Management, see your system administrator. It may be a licensing issue.

Note:

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

Configure Volume Management

Volume Management does not need to be configured.

However, you can modify the following **Volume Monitor preferences** to change the levels at which volumes reach warning and critical states:

- **TC_VOLUME_MONITOR_WARNING_LEVEL**
- **TC_VOLUME_MONITOR_CRITICAL_LEVEL**

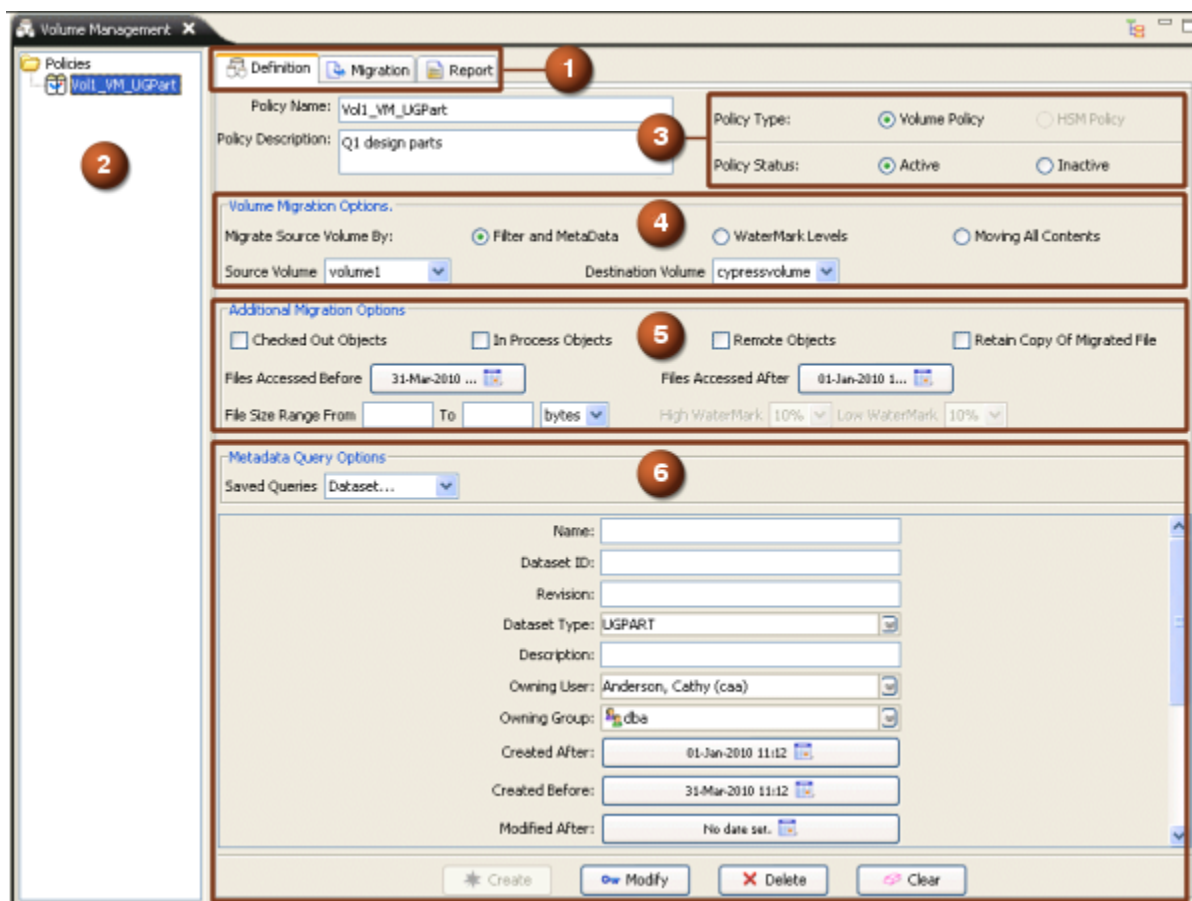
Start Volume Management

Click **Volume Management**  in the navigation pane.

Volume Management interface

Volume Management view

Volume Management view interface



- 1 Volume Management tabs

The **Definition** tab displays options for defining volume management (VM) migration policies and hierarchical storage management (HSM) migration policies. Use this tab to define various migration policies by applying migration options and metadata query options to a specific policy.

The **Migration** tab displays preview and migration options for specific migration policies. Use this tab to preview the impact of a migration policy, and to implement migration policies.

The **Report** tab displays migration report options. Use this tab to create pie charts and textual reports. Reports can be printed or saved to a file.

- 2 **Policies** tree

Displays migration policies that you have created. Migration policies define what data will be migrated. You can create VM migration policies and HSM migration policies.

3	Policy Type and Policy Status	Use the Definition tab to select the type of policy you want to manage: VM or HSM. You can also set the policy status.
4	Volume Migration Options	Selection of the policy type (VM or HSM) determines which migration options appear.
5	Additional Migration Options	Selection of the policy type (VM or HSM) also determine which additional migration options appear.
6	Metadata Query Options	Use the MetaData Query Options functionality to further filter which files are migrated. Select a workspace object type from the Saved Queries list to limit migration to files of the selected object type. You can further filter which files are migrated, refining the saved query using the workspace object fields (name , Item ID , and so on).

Pending migration requests buttons

Use the buttons in the **Migration** tab to work with pending migration file sets. These buttons are displayed to the right of the **Pending Migration Requests** pane after a migration is initiated.

Button	Name	Description
	Move up	Moves up the selected pending file set.
	Move down	Moves down the selected pending file set.
	Delete	Deletes the selected pending file set.
	Open	Opens the selected pending file set.
	Export	Exports the pending file set in either XML or text format.
	Migrate	Migrates the selected files and deletes the pending file set.
	Clear	Clears the Pending Migration Requests box.

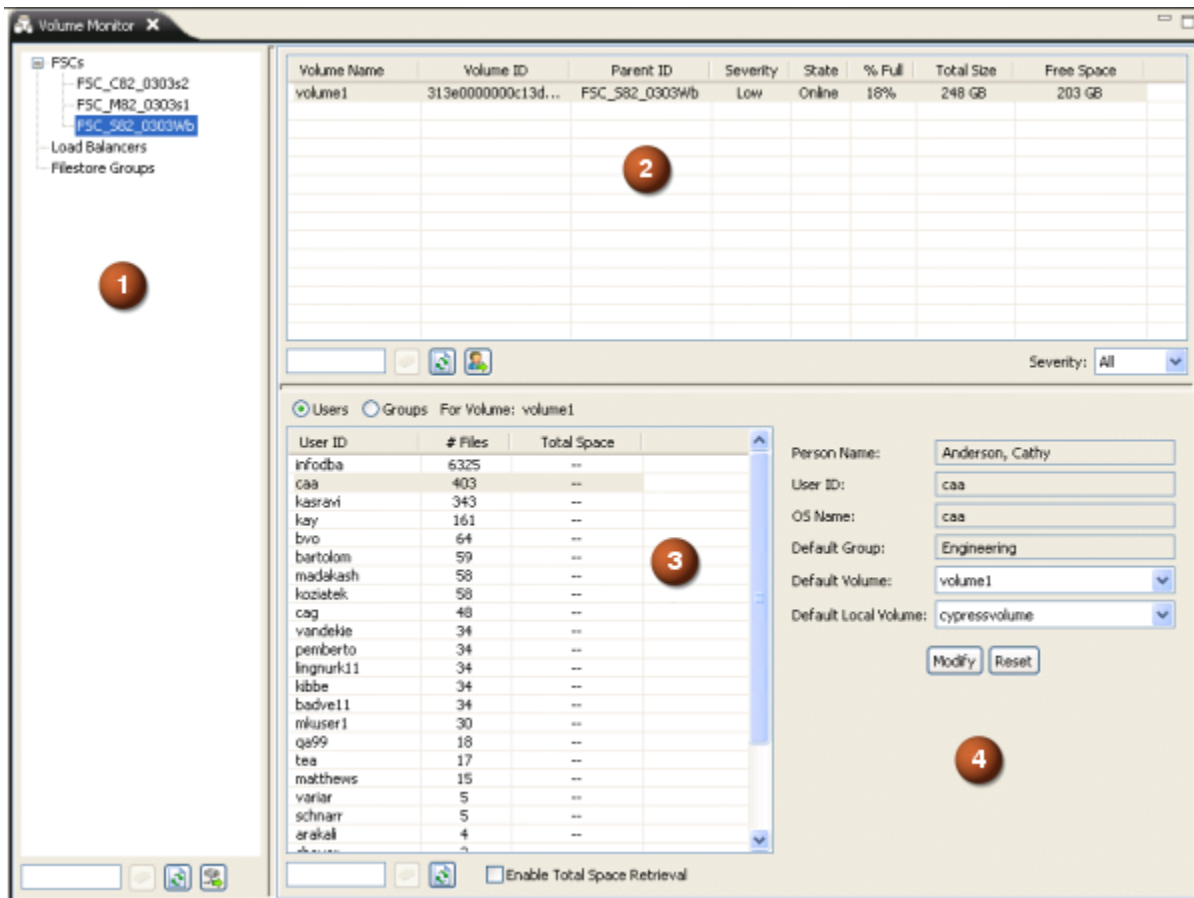
Report buttons in the Volume Management view

Use the buttons in the **Report** tab to select which report type to generate.

Button	Name	Description
	PieChart	Generates a report of how much data is migrated through a volume management (VM) migration policy. This graphical report illustrates the storage space saved by migrating the selected data, relative to the capacity of the source volume. You can save the report as a JPEG or PNG file.
	Report	Generates a report of how much data is migrated through a VM migration policy. This text-based report provides an itemized list of the migrated files, the total number of migrated files, and the total size (in bytes) of the migrated data.
	Save	Saves the report. Displays at the center-right of the Report tab.
	Print	Prints the report. Displays at the center-right of the Report tab.
	Clear	Clears the report. Displays at the center-right of the Report tab.

Volume Monitor view

Volume Monitor view interface



- 1 Filestore tree Displays all filestores defined in your FMS configuration that contain configured volumes.
- Add volumes to a filestore using the **filestore element** in the FMS master configuration file (*FSC_HOME/fmsmaster_fscid.xml*).

Note:

Only configured volumes display. If, for example, you have a load balancer defined in your FMS configuration with no volume definitions, the load balancer does not display in the filestore tree.

- 2 Volume table Displays the status, percentage full, and free space available for all volumes defined for the selected filestores.

- 3 Usage table Displays user and group usage information (number of files, and total space used) for the volume selected in the volume table.

Note:

Determining the total space used by users and groups requires calculating the size of every file on the volume. This can be a time-consuming operation if numerous files exist on the volume, taking hours to complete. By default, this calculation is disabled.

To display the total space used by users and groups, select the **Enable Total Space Retrieval** check box. You are prompted to confirm that you want to continue with this operation. Click **OK** to display the total space calculations for all users and groups on the selected volume.

- 4 User/group information pane Displays user information for the user or group selected in the usage table.

Filestore tree buttons

Use the buttons in the filestore tree pane to work with the contents of the filestore tree and to retrieve volume information.

Button	Name	Description
	Clear	Clears the filter box in the lower-left corner of the volume tree and restores tree contents back to an unfiltered state.
	Refresh with current items	Refreshes the filestore tree.
	Retrieve volume information	Retrieves volume information for the filestores selected in the filestore tree and displays the information in the volume table.

Volume table buttons

Use the buttons in the volume table pane to work with the volume table and to retrieve usage information.

Button	Name	Description
	Clear	Clears the filter box in the lower-left corner of the volume tree and restores tree contents back to an unfiltered state.
	Refresh with current items	Refreshes the volume table, based on the data most recently retrieved from the filestore tree.
	Retrieve user information	Retrieves user and group usage information for the volume selected in the volume table and displays the information in the usage table.

Configuring Volume Management

Setting Volume Management preferences

Use the following preferences to modify behavior of the Volume Management application.

- Use the **HSM_integration_enabled** preference to enable the Volume Management application to work with third-party hierarchical storage management (HSM) software.
- Use the **HSM_read_thru_supported** preference to declare whether read access to secondary and tertiary volumes is supported. If your HSM system or hardware platform does not support read access from secondary and tertiary volumes, the migrated files are moved to a higher tier to be read. Using **HSM_read_thru_supported** requires that the **HSM_integration_enabled** preference be set to true.

For example, the IBM Tivoli HSM product does not provide read-through capability, and the EMC DiskXtender HSM product only provides read-through capability on Windows. In such situations, use reverse migration to return files to the primary tier.

- Use the **HSM_primary_tier_hosts** preference to define the host names on which the primary tier volumes are managed by third-party HSM software.

HSM migration is applied to the volume files residing on hosts defined by this preference. The **hsm_capacity_alert** utility estimates the capacities of the volumes and uses the values defined in this preference to send email alerts when the primary tier capacity exceeds alert levels. Volumes residing on hosts not defined by this preference are ignored for HSM migration and the **hsm_capacity_alert** utility.

- Use the **HSM_secondary_tier_capacity** preference to send an email alert when the capacity of the second tier volume reaches capacity. Set this preference to the estimated capacity level of the second tier volume. Compute this estimate by totaling all storage capacities of all disks mounted on the secondary tier.

- Use the **TC_VOLUME_MONITOR_WARNING_LEVEL** preference to specify when a volume reaches warning level. By default, this preference is set to 80. When a volume becomes 80 percent full, it reaches warning level and displays in the volume table of the **Volume Monitor** tab with a yellow background.
- Use the **TC_VOLUME_MONITOR_CRITICAL_LEVEL** preference to specify when a volume reaches critical level. By default, this preference is set to 90. When a volume becomes 90 percent full, it reaches critical level and displays in the volume table of the **Volume Monitor** tab with a red background.

Running Volume Management command line utilities

Use the following Volume Management command line utilities in conjunction with the Volume Management application.

- The **hsm_capacity_alert** utility determines when the capacity of specified volume tiers exceed specified alert levels. When alert capacity is reached, the utility sends an email alert to the system administrator.

Note:

You can run this utility overnight by wrapping it as a **.bat** file and using Windows Scheduler (Windows) or wrapping it as a shell script and running a **cron** job (Linux).

- The **hsm_report** utility evaluates the specified hierarchical storage management (HSM) migration policies and generates the specified reports.

Note:

For better performance, generate migration reports using this utility rather than the **PieChart** and **Report** buttons in the Volume Management application. You can run this utility overnight using a Windows **at** command or running a Linux **cron** job.

- The **install_vminfo_acl** utility creates access control list (ACL) rules for migration of the existing Teamcenter volume files. The Volume Management application introduces changes to the Access Manager (AM) rule tree. This utility verifies whether the AM rule tree contains the required rules (**HSM_Info** and **VM_Info**). If not, the rules are added to inherit the access privileges of the named ACL **POM Open Access** in par with the **ImanFile** object and saves the changes.

This utility runs automatically at installation. Typically, there is no need for administrators to run the utility again. In cases where an administrator has overwritten the rule tree with custom rules, this utility can be run to ensure the required rules are added to the rule tree.

- The **mark_for_migrate** utility determines whether there are any volume files pending for migration, and if so, marks the files for migration.

Note:

For better performance, mark files for migration using this utility rather than the **Mark for Migration** button in the Volume Management application. You can run this utility overnight using a Windows **at** command or running a Linux **cron** job.

- The **unmigrate_from_hsm** utility is used for reverse migration of existing Teamcenter volume files. This utility removes the **hsm_info** object associated with file objects from a specified volume, or from all volumes. Use this utility to resolve error situations that occurred while migrating files to HSM, or to remove a volume from the purview of HSM.

Running this utility only removes the HSM functionality associated with the specified files. All physical volume files must still be returned to the primary tier if they were migrated to a secondary or tertiary tier.

Note:

Siemens Digital Industries Software recommends that you execute this utility on a specific volume when no other users are accessing Teamcenter, especially during maintenance or upgrades.

- The **vm_report** utility evaluates the specified volume management (VM) migration policies and generates the specified reports.

Note:

For better performance, generate migration reports using this utility rather than the **PieChart** and **Report** buttons in the Volume Management application. You can run this utility overnight using a Windows **at** command or running a Linux **cron** job.

Basic concepts about Volume Management

Why use Volume Management?

Primary disk space is expensive. Volume Management allows you to monitor volume usage and migrate infrequently used data to secondary and tertiary storage media. You can define various migration policies for specific types of data, and the system migrates the data based on the policies you define.

You can define both *volume management* (VM) migration policies and *hierarchical storage management* (HSM) migration policies.

VM migration policies specify data to be migrated from a source volume to a destination volume. These migration policies let you specify that data is migrated by saved queries, by watermark levels, or by migrating all data from the source volume.

HSM migration policies direct data to be migrated between storage tiers. Data can be migrated from a primary to a secondary tier, from a primary to a tertiary tier, or from a secondary to a tertiary tier. These migration policies let you choose whether to purge the original data after migration, or to purge when specified watermark levels are reached.

Monitoring volumes

It is important to know how much free space is available on each volume at your site, to avoid import errors and emergency fixes. Use the **Volume Monitor** tab to monitor free space and proactively manage volume usage for all volumes at your site.

The volume table lists the status, percentage full, and free space for any volume at your site. When free space is low, use the usage table to track space usage by group or by individual users. Balance volume resources by reassigning default volumes for specific users or groups.

As a general guideline to ensure optimal operation, Siemens Digital Industries Software recommends that you avoid storing more than 1 million files on any individual volume.

Volume Management migration overview

Migration moves infrequently used Teamcenter files from primary storage space to either third-party storage systems or to destination volumes. All data continues to appear online to users, whether the data is stored online, near-line, or offline.

Reverse migration is available only for hierarchical storage management (HSM) migration policies. This functionality returns files to the primary tier based on the number of times it has been accessed on the secondary or tertiary tier.

Migration policies in the Volume Management application

Migration policies are displayed in the **Policies** tree in the left pane of the application. They define the data to be migrated. You can create volume management (VM) migration policies and hierarchical storage management (HSM) migration policies. Create migration policies in the **Definition** pane by defining the migration options and metadata query options to be applied to a given migration policy.

Migration options differ depending on the type of migration policy being defined. Options for VM migration policies focus on source and destination volumes. Options for HSM migration policies center around purging methods and third-party storage locations.

Use the saved query options to further define the data to be migrated. Both HSM and VM migration policy options allow you to specify criteria using standard Teamcenter saved queries. You can define the queries using standard saved queries for items, item revisions, or datasets.

Following are the types of migration policies:

- VM migration policies

Use *VM migration policies* to migrate and consolidate Teamcenter volume files.

Migration options for this method focus on selecting a source and destination volume. You can then either select certain types of data to be migrated, choose to migrate data when specified volume watermark levels are reached, or choose to migrate all files.

These migration policies do not necessarily require a third-party HSM system. You can use VM migration policies to emulate HSM functionality *without* the use of third-party storage systems by migrating infrequently used Teamcenter files off of source volumes and on to destination volumes. The pending file sets defined by VM migration policies are migrated using File Management System (FMS).

Alternatively, VM migration policies can be used to loosely couple your third-party migration tool with Teamcenter. Pending migration file sets (**VM_Migration** files) are exported to a valid operating system directory, either in text or XML format. The third-party migration tool reads the exported file to perform the migration.

- HSM migration policies

Use *HSM migration policies* to move lower priority files to third-party storage systems. Files stored on secondary and tertiary tiers can be read-accessed without moving them to the highest level. HSM migration requires third-party HSM tool integration and configuration with Teamcenter.

Note:

Read-access from secondary and tertiary tiers is not supported by certain third-party storage systems or certain hardware platforms. For example, the IBM Tivoli HSM product does not provide read-through capability, and the EMC DiskXtender HSM product only provides read-through capability on Windows. In such situations, you must set the **HSM_read_thru_supported** preference to **false** and use reverse migration to return files to the primary tier for read access.

Migration options for this method include selecting purge options and third-party storage locations. Define the type of data to be migrated by selecting additional migration options (such as re-filed files and checked out objects). Further define data to be migrated by applying saved queries.

This method requires that your site implements third-party HSM systems. The HSM migration policies provide the following types of storage strategies:

- Online primary tier storage

Data is stored on third-party HSM software compatible disk media such as simple Windows or Linux SAN attached disks. This is the primary, direct, volume storage location.

- Near-line secondary tier storage

Data is stored on third-party HSM software compatible disk media that can be either simple Windows or Linux SAN attached disks or complex network access storage (NAS) systems from HP, EMC Celerra or NetApp NearStore.

- Offline tertiary tier

Data is stored on third-party HSM software compatible devices such as EMC Centera, WORM drives, Juke Box, and tape drives.

HSM pending file sets can be migrated by loosely coupling third-party HSM storage systems with Teamcenter. Pending migration file sets (**HSM_Migration** files) are exported to a valid operating system directory, either in text or XML format. The HSM storage systems read the exported file to perform the migration.

Alternatively, you can tightly couple third-party HSM storage systems with Teamcenter. Pending migration file sets are routed through API (user exit) calls to the HSM storage systems. This method allows you to replace the base user exit extension with a custom extension using Business Modeler IDE.

- Active and inactive migration policies

Active migration policies are displayed under the **Policies** tree in full color. Only active migration policies can be selected for evaluation and migration. Migration policies are made active at the time they are defined.

Inactive policies are grayed out in the **Policies** tree. **Make a migration policy inactive** to prevent future evaluations and migrations.

Basic tasks for Volume Management

Following are some of the basic tasks you perform with Volume Management:

- **Reallocating volume usage**

Volume usage is not always distributed evenly between users on a given volume. A few users (or groups) may use many times more volume space than others. When volume usage is significantly unequal, and volume free space is becoming limited, you can balance volume usage by reallocating heavy users or groups to a different volume with more free space.

- **Defining volume management (VM) migration policies**

Migration policies define the data to be migrated. VM migration policies specify data to be migrated from a source volume to a destination volume. These migration policies let you specify that data is migrated by saved queries, by watermark levels, or by migrating *all* data from the source volume.

- **Defining hierarchical storage management (HSM) migration policies**

Migration policies define the data to be migrated. Hierarchical storage management (HSM) migration policies direct data to be migrated between storage tiers. Data can be migrated from a primary to a secondary tier, from a primary to a tertiary tier, or from a secondary to a tertiary tier. These migration policies let you choose whether to purge the original data after migration or to purge when specified watermark levels are reached.

- **Data migration**

After defining the data to be migrated, you can migrate the data to the specified location. Migrating data involves previewing the migration results, marking the files for migrate, and then migrating the data.

- **Generating reports**

After migrating data, you can generate migration reports to determine how much data was migrated between volumes or tiers, and to see which migration policies migrated the data. You can refine reports by date, by policy, or you can generate a report covering all current migration policies. Reports can be printed or saved to a file.

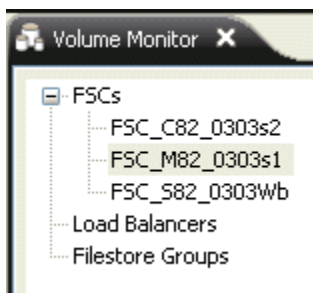
Monitoring volumes

Introduction to monitoring volumes

It is important to know how much free space is available on each volume at your site, to avoid import errors and emergency fixes. Use the **Volume Monitor** tab to monitor free space and proactively manage volume usage for all volumes at your site.

As a general guideline to ensure optimal operation, Siemens Digital Industries Software recommends that you avoid storing more than 1 million files on any individual volume.

Choose **Window→Show View→Volume Monitor** from the **Volume Management** perspective to open the **Volume Monitor** view. The filestore tree displays in the left side of the view.

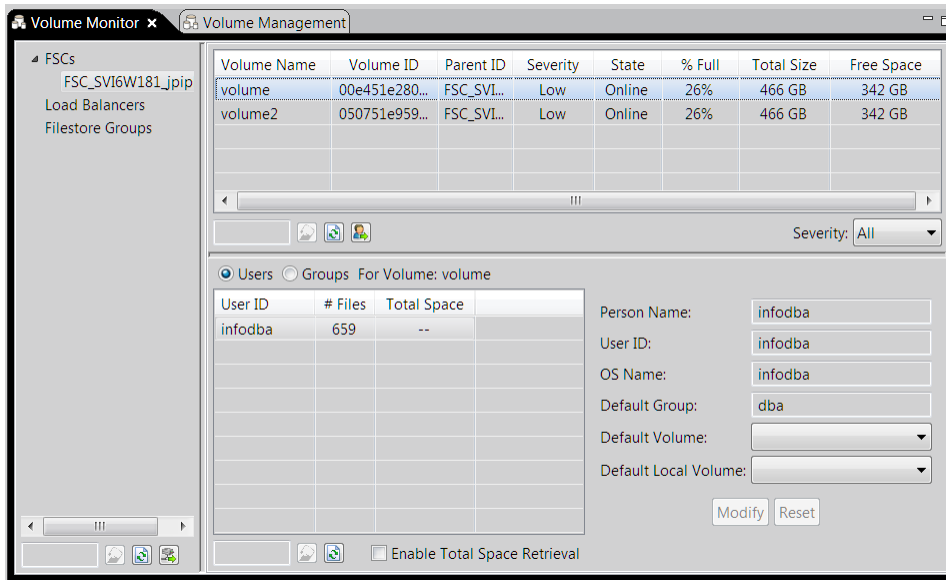


The filestore tree lists all filestores with volumes defined on them. The filestores in the tree are loaded from the bootstrap FSC, which is defined by the **Fms_BootStrap_Urls** preference. Filestores are grouped in the tree by FSC, load balancers, and filestore groups.

Selecting one or more filestores from the tree and clicking the **Retrieve volume information**  button at the bottom of the file store tree pane populates the volume table.

Selecting a volume from the volume table and clicking **Retrieve user information**  populates the usage table with user and group data.

To **retrieve volume usage information**, select a group or user from the usage table. This populates the boxes to the right of the volume table with usage information.



The screenshot shows the 'Volume Management' window. On the left is a tree view with 'FSCs' selected. The main area contains a 'Volume Table' with the following data:

Volume Name	Volume ID	Parent ID	Severity	State	% Full	Total Size	Free Space
volume	00e451e280...	FSC_SVI...	Low	Online	26%	466 GB	342 GB
volume2	050751e959...	FSC_SVI...	Low	Online	26%	466 GB	342 GB

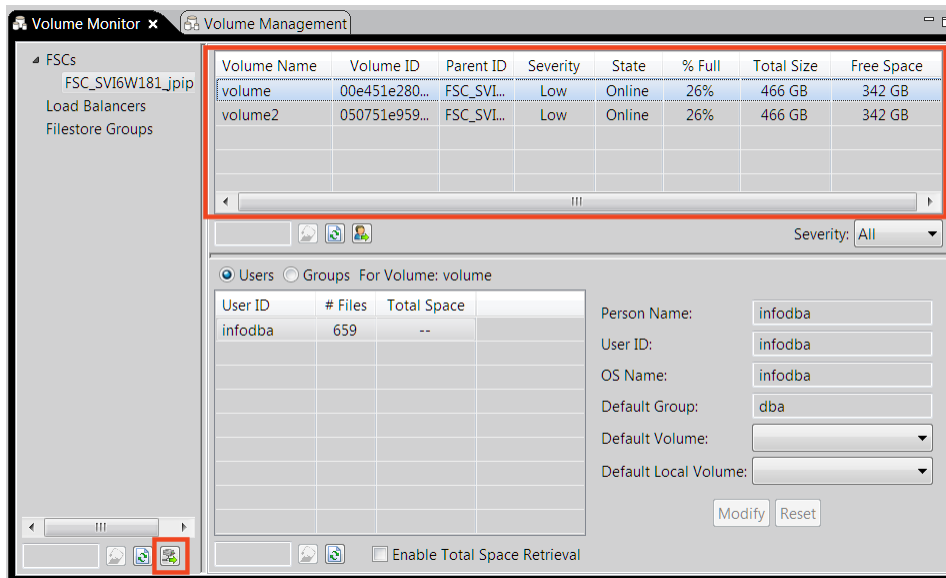
Below the table is a 'Severity' dropdown set to 'All'. Underneath is a section for 'Users' and 'Groups' with a radio button for 'Users' selected. The 'For Volume: volume' section shows a table with user information:

User ID	# Files	Total Space
infodba	659	--

To the right of this table are input fields for 'Person Name', 'User ID', 'OS Name', 'Default Group', 'Default Volume', and 'Default Local Volume', all containing the value 'infodba' or 'dba'. At the bottom right are 'Modify' and 'Reset' buttons. At the bottom left is a checkbox for 'Enable Total Space Retrieval'.

Retrieving volume information

Selecting one or more filestores from the tree and clicking **Retrieve volume information**  populates the volume table with the following information for each volume associated with the selected filestores.



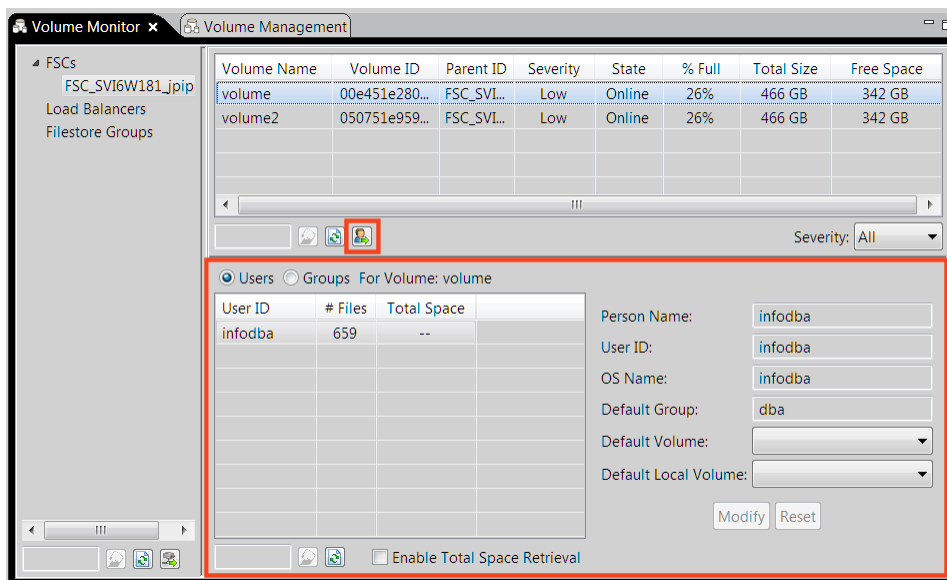
Data	Description
Volume Name	Specifies the name of the volume.
Volume ID	Specifies the volume ID.
Parent ID	Specifies the filestore ID of the parent filestore. If the volume is hosted on multiple filestores, all parents display in a comma-separated list.
Severity	Lists the possible severity levels: Low, Warning, Critical. The severity level determines the row color: • Low is clear. • Warning is yellow. • Critical is red. If an FSC volume is offline, it is listed as Critical. Set severity levels with the <code>TC_VOLUME_MONITOR_WARNING_LEVEL</code> and <code>TC_VOLUME_MONITOR_CRITICAL_LEVEL</code> preferences.
State	Indicates the state as Online or Offline .
% Full	Specifies the percentage of space consumed on the disk hosting the volume.

Data	Description
	File Management System (FMS) does not maintain quotas at a volume level, thus two volumes hosted on the same disk display the same value.
Total Size	Specifies the total amount of space, both used space and free space, available on the disk hosting the volume.
Free Space	Specifies the total amount of free space available on the disk hosting the volume.

Retrieving user information for a volume

Selecting a volume from the volume table and clicking **Retrieve user information**  populates the usage table.

Select the **Users** option to display the following user statistics for the selected volume.



The screenshot shows the 'Volume Management' window. A table lists volumes, with 'volume' selected. Below the table, the 'Users' tab is active, displaying a table of user statistics for the selected volume.

Volume Name	Volume ID	Parent ID	Severity	State	% Full	Total Size	Free Space
volume	00e451e280...	FSC_SVI...	Low	Online	26%	466 GB	342 GB
volume2	050751e959...	FSC_SVI...	Low	Online	26%	466 GB	342 GB

User ID	# Files	Total Space
infodba	659	--

Person Name: infodba
 User ID: infodba
 OS Name: infodba
 Default Group: dba
 Default Volume:
 Default Local Volume:
 Modify Reset

Data	Description
User ID	Lists the names of each user owning data on the selected volume.
# of files	Specifies the total number of files owned by the corresponding user.
Total space	Specifies the total space used by all the files owned by the corresponding user.

Select the **Groups** option to display the following group statistics for the selected volume.

Data	Description
Group Name	Lists the names of each group owning data on the selected volume.
# of files	Specifies the total number of files owned by the corresponding group. This number includes all files owned by all users in that group and subgroups.
Total space	Specifies the total space used by all the files owned by the corresponding group.

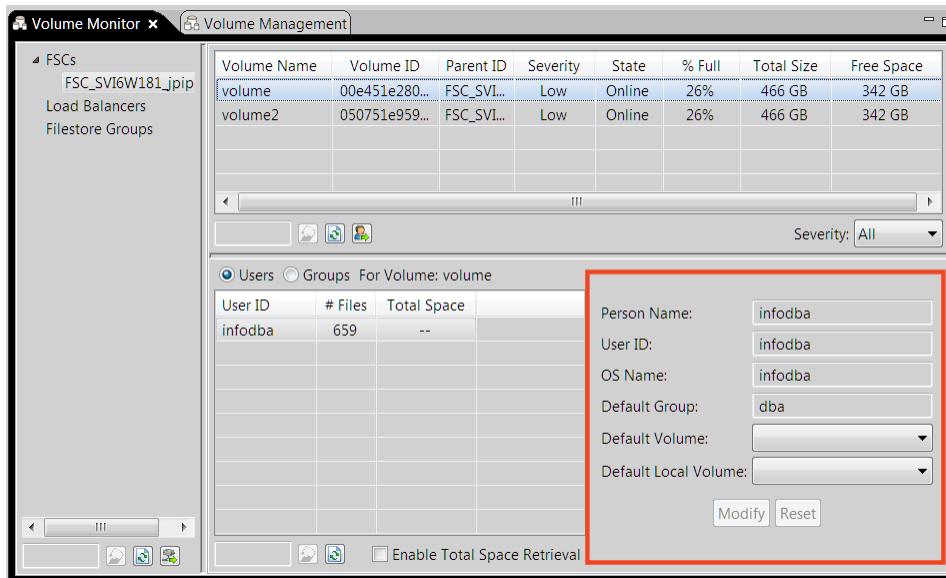
Note:

Determining the total space used by users and groups requires calculating the size of every file on the volume. This can be a time-consuming operation if numerous files exist on the volume, taking hours to complete. By default, this calculation is disabled.

To display the total space used by users and groups, select the **Enable Total Space Retrieval** check box. You are prompted to confirm that you want to continue with this operation. Click **OK** to display the total space calculations for all users and groups on the selected volume.

Retrieving volume usage information

Selecting a user from the usage table populates the boxes to the right of the table with the following usage information.



Data	Description
Person Name	Specifies the full name of the selected user.
User ID	Specifies the user ID used for Teamcenter logon.
OS Name	Specifies the selected user's operating system name.
Default Group	Specifies the default group assigned to the selected user. When a user belongs to more than one group, one of them is designated as the default group. The user's default group is used at logon unless the user specifies another group.
Default Volume	Specifies the default volume assigned to the selected user. Typically, default volumes are assigned per group, rather than per user. Siemens Digital Industries Software recommends that you do not define a default volume for each user, as it is time-consuming to change volumes for every user individually. (When a default volume is not specified when creating a user in the Organization application, the group's default volume information is used.) If volume usage is significantly unequal between users and volume free space is becoming limited, you can balance volume usage by reallocating users consuming large amounts of volume space to a different default volume.
Default Local Volume	Specifies the default local volume assigned to the selected user.

Data	Description
	This temporary local volume allows the file to be stored locally before it is automatically transferred to the final destination in the background. Once the file is stored in the default local volume, the user can continue working without having to wait for the upload to take place. If volume usage is significantly unequal between users and volume free space is becoming limited, you can balance volume usage by reallocating users consuming large amounts of volume space to a different default local volume.

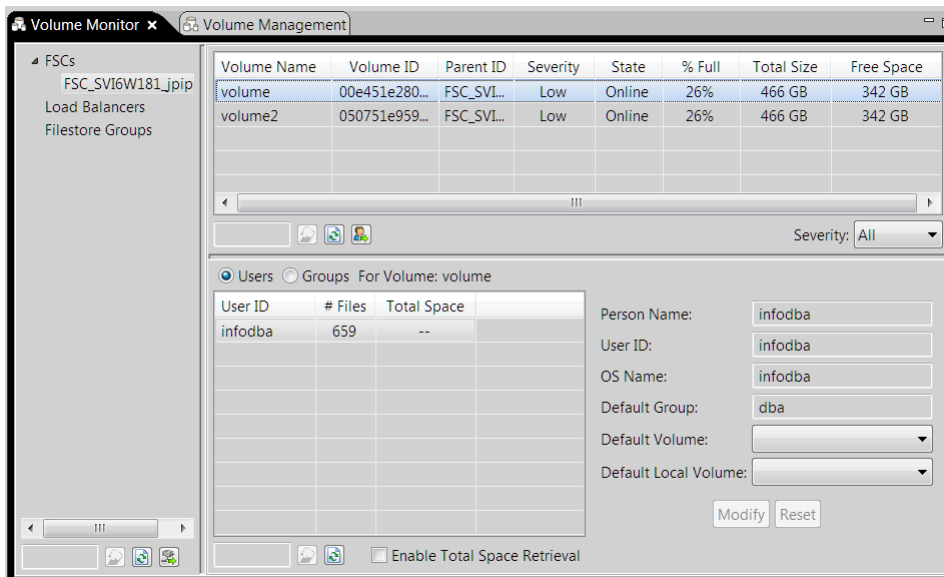
Selecting a group from the usage table populates the boxes to the right of the table with the following group information.

Data	Description
Group Name	Specifies the name of the selected group.
Description	Provides a description of the group, entered by the system administrator when creating the selected group.
To Parent	Specifies the parent group of the selected group.
Default Volume	Specifies the default volume assigned to the selected group. Typically, default volumes are assigned per group, rather than per user. (When a default volume is not specified when creating a user, the group's default volume information is used.) Warning: If you create a group without assigning a default volume, group members cannot save datasets. If volume usage is significantly unequal between groups and volume free space is becoming limited, you can balance volume usage by reallocating groups consuming large amounts of volume space to a different default volume.
Default Local Volume	Specifies the default local volume assigned to the selected group. This temporary local volume allows the file to be stored locally before it is automatically transferred to the final destination in the background. Once the file is stored in the default local volume, the user can continue working without having to wait for the upload to take place.

Data	Description
	If volume usage is significantly unequal between groups and volume free space is becoming limited, you can balance volume usage by reallocating groups consuming large amounts of volume space to a different default local volume.

How to monitor volumes

- Select one or more filestores from the filestore tree. If necessary, locate a filestore in the tree using one of the following methods:
 - Select **FSCs**, **Load Balancers** and/or **Filestore Groups** to select all filestores within the selected grouping. Selecting all three groupings effectively selects all volumes defined in your FMS configuration.
 - Filter the entries in the filestore tree by entering a filestore name in the search box. As you type in the box the tree refreshes, listing only filestores with names containing the characters typed in the search box.
 - If you recently added a filestore to your site, click **Refresh with current items**  to update the cached filestore information read from the bootstrap FSC.



- Click **Retrieve volume information** .

The volume table is populated with the following data for the selected filestores.

Volume Name	Volume ID	Parent ID	Severity	State	% Full	Total Size	Free Space

3. Monitor the selected filestore by reviewing the data in the volume table using one of the following methods:
 - Filter the entries in the table entering a volume name in the search box. As you type in the box the tree refreshes, listing only volumes with names containing the characters typed in the search box.
 - If you have recently modified filestores at your site, click **Refresh with current items**  to update the cached filestore information read from the bootstrap FSC.
4. Select the volume in the volume table for which you want to retrieve usage information.
5. Click **Retrieve user information**  in the volume table.

The usage table is populated with data for the selected volume. By default, user data is populated. Select the **Groups** option to display group data in the usage table.

6. Monitor usage information for the selected volume by reviewing either user or group data in the usage table.
7. Monitor user information for a specific user by selecting a user from the usage table.

If necessary, filter the entries in the usage table by entering all or part of a user name in the search box. As you type in the box the table refreshes, listing only users with names containing the characters typed in the search box.

The boxes to the right of the usage table are populated with volume information for the selected user.

8. Monitor group information for a specific group by selecting a group from the usage table.

If necessary, filter the entries in the usage table by entering all or part of a group name in the search box. As you type in the box the table refreshes, listing only groups with names containing the characters typed in the search box.

The boxes to the right of the usage table are populated with group information for the selected group.

9. If volume usage is significantly unequal between users or groups, and if volume free space is limited, balance volume usage by reallocating users or groups consuming large amounts of volume space to a different default volume, and/or a different default local volume.

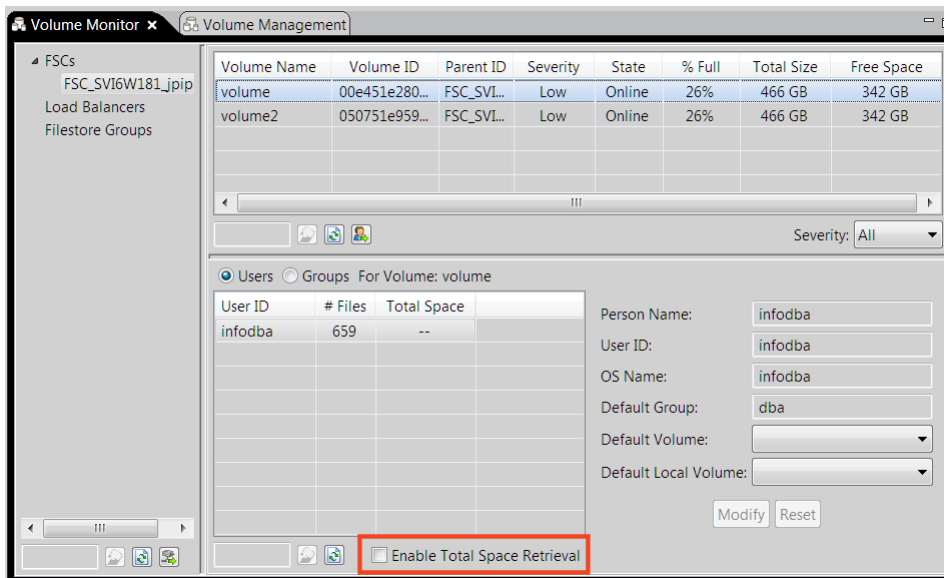
Reallocate volume resources

1. (Optional) Display the total space used by users or groups by selecting the **Enable Total Space Retrieval** check box.

When you click **Retrieve user information** , you are prompted to confirm that you want to continue with this operation. Click **OK** to display the total space calculations for all users and groups on the selected volume.

Note:

Determining the total space used by users and groups requires calculating the size of every file on the volume. This can be a time-consuming operation if numerous files exist on the volume, taking hours to complete. By default, this calculation is disabled.



2. Balance default volume usage by reallocating users or groups consuming large amounts of volume space to a different default volume by selecting the user or group whose default volume you want to change.

Note:

Default volumes specify the default final destination volume for Teamcenter files. Default volumes differ from default local volumes in that default volumes are the first and final destination for Teamcenter files. (Default local volumes are temporary storage locations.)

The boxes to the right of the usage table are populated with user/group information for the selected user or group.

3. Select a different default volume from the **Default Volume** list. Generally, this is a volume with more free space than the original default volume.

4. Click **Modify** to send the change to the database.

Alternatively, click **Reset** to reset the value to what is set in the database.

5. Balance default local volume usage by reallocating users or groups consuming large amounts of volume space to a different default local volume by selecting the user or group whose default local volume you want to change.

Note:

Default local volumes are temporary local volumes that allow files to be stored locally before they are automatically transferred to the final destination volume. This functionality improves end-user file upload times from clients by uploading files to a temporary volume. Users can continue to work on their files from the temporary location. The system moves the files to their final destination according to administrator-defined criteria. Files are accessible to FMS at all times. This behavior is also referred to as the store and forward of files.

The boxes to the right of the usage table are populated with volume information for the selected user or group.

6. Select a different default local volume from the **Default Local Volume** list. Generally, this is a volume with more free space than the original default local volume.
7. Click **Modify** to send the change to the database.

Alternatively, click **Reset** to reset the value to what is set in the database.

Defining migration policies

Migration policy options

Migration policies define the data to be migrated. You can create either volume management (VM) migration policies or hierarchical storage management (HSM) migration policies. Either type of migration policy allows you to select various migration options. Depending on the options chosen, you can further refine the data to be migrated by filtering using standard saved queries for commonly used workspace objects, such as items, item revisions, and datasets.

When defining volume management policies you must select a source and destination volume. The following table defines the different migration types and migration options available.

Volume policy migration options	Description	Metadata query options
Filter and MetaData	Refines a migration policy by checked-out objects, in-process objects and/or remote objects. It further refines a migration policy by file size and date access. You can also choose to retain a copy of migrated files. This is the most granular method of creating a VM migration policy.	Further refines a migration policy by defining attributes of the following saved queries: item, item revision, dataset.
Water Mark Levels	Specifies the high and low watermark levels of the source volume. When the high watermark level is reached, the data is migrated from the source volume to the destination volume. Migration stops when the low watermark is reached. You can further	No saved queries attributes can be used with this method.

Volume policy migration options	Description	Metadata query options
	refine the data to be migrated by file size.	
Moving All Contents	Migrates all valid files from the source volume to the destination volume.	No saved queries attributes can be used with this method.

When defining HSM management policies, you must select a purge option and a migration tier option. The following table lists all available options.

HSM policy migration options	Definition
Purge After Migrate	Purges files immediately after migration.
Purge Watermark	Purges files after the specified high watermark has been reached. Purge continues until the low watermark is reached.
Primary-Secondary migration tier	Migrates files from the primary source tier to the secondary destination tier.

HSM policy migration options	Definition
Secondary-Tertiary migration tier	Migrates files from the secondary destination tier to the tertiary destination tier.
Primary-Tertiary migration tier	Migrates files from the primary source tier to the tertiary destination tier.
Re-Filed Files	Includes refiled files in the migration evaluation.
Checked Out Objects	Includes the files of checked-out objects in the migration evaluation.
In Process Objects	Includes the files of in-process objects in the migration evaluation.
Remote Objects	Includes the files of remote objects in the migration evaluation.
Files Accessed Before	Considers files accessed before the specified date in the migration evaluation.
Files Accessed After	Considers files accessed after the specified date in the migration evaluation.
File Size Range	Includes files of the specified size range in the migration evaluation.
Reverse Migrate File Accessed	Used with reverse migration. Any file in the secondary or tertiary tier that is accessed more than the selected number of times (1 or 2) is selected for reverse migration.

Define a VM policy by filters and metadata

This method creates migration policies using logic filters and standard saved queries. It is the most granular method for defining volume management (VM) migration policies. It is the default method.

1. Open the **Volume Management** view.
2. Type a name for the migration policy in the **Policy Name** box.

If you create both VM and hierarchical storage management (HSM) migration policies at your site, implementing a naming convention that differentiates between the two types simplifies managing your policies from the **Policies** tree after the policies are created.

3. Type a description of the migration policy in the **Policy Description** box.

4. Click the **Definition** tab.
5. In the **Policy Type** section in the upper-right corner of the pane, click **Volume Policy**.
6. In the **Policy Status** section in the upper-right corner of the pane, click **Active**.
7. In the **Volume Migration Options** section, click **Filter and MetaData**.

The remaining migration options become specific to this migration method.

8. Select a source volume from the **Source Volume** list. Select the volume from which you want to migrate the data.

Note:

The list displays all defined volumes for the Teamcenter database. If you do not see a volume you created during this session, close and reopen the Volume Management application.

9. Select a destination volume from the **Destination Volume** list. Select the volume to which you want the data migrated.
10. (Optional) In the **Additional Migration Options** section, select **Checked Out Objects** to include the files of all checked-out objects in the migration.
11. (Optional) Select **In Process Objects** to include the files of all in-process objects in the migration.
12. (Optional) Select **Remote Objects** to include the files of all remote objects in the migration.
13. (Optional) Select **Retain Copy of Migrated File** to retain a copy of the migrated file in the source volume. Previously issued FMS read tickets on the volume files are honored. When these tickets have expired, you must remove the copy of the migrated file by running the **review_volumes** utility.
14. (Optional) Select a date from the **Files Accessed Before** calendar to limit migration to files accessed before the specified date.
15. (Optional) Select a date from the **Files Accessed After** calendar to limit migration to files accessed after the specified date.
16. (Optional) Type a minimum and maximum number in the **File Size Range From** and **To** boxes and select the size measurement type (bytes, KB or MB) from the list to limit migration to files within the specified size range.
17. (Optional) In the **MetaData Query Options** section, select a workspace object type from the **Saved Queries** list to limit migration to files of this object type.

18. If you select an object type in the previous step, type information into any of the boxes to further define the saved query.
19. After you finish defining the data to be migrated, click **Create** at the bottom of the pane to create the migration policy.

The migration policy appears under the **Policies** tree.

After one or more migration policies have been defined, you can preview the results of the migration, then migrate the data.

Define a VM policy by watermark levels

This method creates a volume management (VM) migration policy that directs the system to migrate or purge files when the specified watermark level (capacity) is reached within the source volume. The only migration options that you can specify using this method are the source and destination volumes, and the watermark levels of the source volume. You cannot define any additional migration options or query options.

1. Open the **Volume Management** view.
2. Type a name for the migration policy in the **Policy Name** box.

If you create both VM and hierarchical storage management (HSM) migration policies at your site, implementing a naming convention that differentiates between the two types simplifies managing your policies from the **Policies** tree after the policies are created.

3. Type a description of the migration policy in the **Policy Description** box.
4. Click the **Definition** tab.
5. In the **Policy Type** section in the upper-right corner of the pane, click **Volume Policy**.
6. In the **Policy Status** section in the upper-right corner of the pane, click **Active**.
7. In the **Volume Migration Options** section, click **Water Mark Levels**.

The remaining migration options become specific to this migration method.

8. Select a source volume from the **Source Volume** list. Select the volume from which you want to migrate the data.

Note:

The list displays all defined volumes for the Teamcenter database. If you do not see a volume you created during this session, close and reopen the Volume Management application.

9. Select a destination volume from the **Destination Volume** list. Select the volume to which you want the data migrated.
10. Select a watermark percentage from the **High Water Mark** list. When the capacity of the source volume reaches this percentage of capacity, migration begins.
11. Select a watermark percentage from the **Low Water Mark** list. When the capacity of the source volume drops to this percentage level, migration ends.
12. After you finish defining the data to be migrated, click **Create** at the bottom of the pane to create the migration policy.

The migration policy appears under the **Policies** tree.

After one or more migration policies are defined, you can preview the results of the migration, then migrate the data.

Define a VM policy by moving all files

This method creates a volume management (VM) policy that migrates *all* files from the source volume to the destination volume. The only migration options that can be specified using this method are the source and destination volumes. No additional migration options or query options can be defined. This is the least refined of the VM migration policies.

1. Open the **Volume Management** view.
2. Type a name for the migration policy in the **Policy Name** box.

If you create both VM and hierarchical storage management (HSM) migration policies at your site, implementing a naming convention that differentiates between the two types simplifies managing your policies from the **Policies** tree after the policies are created.

3. Type a description of the migration policy in the **Policy Description** box.
4. Click the **Definition** tab.
5. In the **Policy Type** section in the upper-right corner of the pane, click **Volume Policy**.
6. In the **Policy Status** section in the upper-right corner of the pane, click **Active**.
7. In the **Volume Migration Options** section, click **Moving All Contents**.

The remaining migration options become specific to this migration method.

8. Select a source volume from the **Source Volume** list. Select the volume from which you want to migrate the data.

Note:

The list displays all defined volumes for the Teamcenter database. If you do not see a volume you created during this session, close and reopen the Volume Management application.

9. Select a destination volume from the **Destination Volume** list. Select the volume to which you want the data migrated.
10. (Optional) Select **Retain Copy of Migrated File** to retain a copy of the migrated file in the source volume. Previously issued FMS read tickets on the volume files are honored. When these tickets have expired, you must remove the copy of the migrated file by running the **review_volumes** utility.
11. After you finish defining the data to be migrated, click **Create** at the bottom of the pane to create the migration policy.

The migration policy appears under the **Policies** tree.

After one or more migration policies are defined, you can preview the results of the migration, then migrate the data.

Define an HSM policy to purge after migration

In this method, third-party hierarchical storage management (HSM) systems purge migrated data from the source tier immediately after migration.

All HSM migration policies migrate specified lower priority files to third-party storage systems. Files stored on secondary and tertiary tiers can still be read-accessed without returning them to the source volume.

Note:

HSM migration policies require a third-party HSM system.

1. Open the **Volume Management** view.
2. Type a name for the migration policy in the **Policy Name** box.

If you create both VM and HSM migration policies at your site, implementing a naming convention that differentiates between the two types simplifies managing your policies from the **Policies** tree after the policies are created.

3. Type a description of the migration policy in the **Policy Description** box.
4. Click the **Definition** tab.
5. In the **Policy Type** section in the upper-right corner of the pane, click **HSM Policy**.
6. In the **Policy Status** section in the upper-right corner of the pane, click **Active**.
7. In the **Purge Options** section, click **Purge After Migrate**.
8. From the **Migration Tier** section, specify the migration path for the selected data. Click either **Primary-Secondary**, **Secondary-Tertiary**, or **Primary-Tertiary**.

For example, **Primary-Secondary** migrates data from the primary tier to the secondary tier.

9. (Optional) In the **Additional Migration Options** section, select **Re-Filed Files** to refile the latest version of data to the same tier from which it originated.
10. (Optional) Select **Checked Out Objects** to include the files of all checked-out objects in the migration.
11. (Optional) Select **In Process Objects** to include the files of all in-process objects in the migration.
12. (Optional) Select **Remote Objects** to include the files of all remote objects in the migration.
13. (Optional) Select a date from the **Files Accessed Before** calendar to limit migration to files accessed before the specified date.
14. (Optional) Select a date from the **Files Accessed After** calendar to limit migration to files accessed after the specified date.
15. (Optional) Type a minimum and maximum number in the **File Size Range From** and **To** boxes and select the size measurement type (bytes, KB or MB) from the list to limit migration to files within the specified size range.
16. (Optional) Select **1** or **2** from the **Reverse Migrate File Accessed More Than** list. Any file in the secondary or tertiary tier accessed more than the selected number of times is selected for reverse migration.
17. (Optional) In the **MetaData Query Options** section, select a workspace object type from the **Saved Queries** list to limit migration to files of this object type.
18. If you selected an object type in the previous step, type information into any of the boxes to further define the saved query.

19. After you finish defining the data to be migrated, click **Create** at the bottom of the pane to create the migration policy.

The migration policy appears under the **Policies** tree.

After one or more migration policies are defined, you can preview the results of the migration, then migrate the data.

Define an HSM policy to purge at watermark levels

In this method, third-party hierarchical storage management (HSM) systems purge migrated files from the source tier only after the source tier capacity reaches a specified watermark level.

All HSM migration policies migrate specified lower priority files to third-party storage systems. Files stored on secondary and tertiary tiers can still be read-accessed without returning them to the source volume.

Note:

HSM migration policies require a third-party HSM system.

1. Open the **Volume Management** view.
2. Type a name for the migration policy in the **Policy Name** box.

If you create both VM and HSM migration policies at your site, implementing a naming convention that differentiates between the two types simplifies managing your policies from the **Policies** tree after the policies are created.
3. Type a description of the migration policy in the **Policy Description** box.
4. Click the **Definition** tab.
5. In the **Policy Type** section in the upper-right corner of the pane, click **HSM Policy**.
6. In the **Policy Status** section in the upper-right corner of the pane, click **Active**.
7. In the **Purge Options** section, click **Purge Water Mark**.
8. Select a watermark percentage from the **High Water Mark** list. When the capacity of the source tier reaches this percentage of capacity, migration begins.
9. Select a watermark percentage from the **Low Water Mark** list. When the capacity of the source tier drops to this percentage level, migration ends.

10. From the **Migration Tier** section, specify the migration path for the selected data. Click either **Primary-Secondary**, **Secondary-Tertiary**, or **Primary-Tertiary**.

For example, **Primary-Secondary** migrates data from the primary volume to the secondary volume.

11. (Optional) In the **Additional Migration Options** section, select **Re-Filed Files** to refile the latest version of data to the same tier from which it originated.
12. (Optional) Select **Checked Out Objects** to include the files of all checked-out objects in the migration.
13. (Optional) Select **In Process Objects** to include the files of all in-process objects in the migration.
14. (Optional) Select **Remote Objects** to include the files of all remote objects in the migration.
15. (Optional) Select a date from the **Files Accessed Before** calendar to limit migration to files accessed before the specified date.
16. (Optional) Select a date from the **Files Accessed After** calendar to limit migration to files accessed after the specified date.
17. (Optional) Select **1** or **2** from the **Reverse Migrate File Accessed More Than** list. Any file in the secondary or tertiary tier accessed more than the selected number of times is selected for reverse migration.
18. (Optional) In the **MetaData Query Options** section, select a workspace object type from the **Saved Queries** list to limit migration to files of this object type.
19. If you selected an object type in the previous step, type information into any of the boxes to further define the saved query.
20. After you finish defining the data to be migrated, click **Create** at the bottom of the pane to create the migration policy.

The migration policy appears under the **Policies** tree.

After one or more migration policies have been defined, you can preview the results of the migration, then migrate the data.

Define a Volume Management policy based on an existing policy

You can define a new volume management (VM) or hierarchical storage management (HSM) migration policy based on an existing policy. Use this method when you want to create a migration policy similar to an existing one.

1. Open the **Volume Management** view.
2. Select an existing migration policy from the **Policies** tree.
3. Click the **Definition** tab.

The selected migration policy settings appear.

4. Type a new name for the migration policy in the **Policy Name** box.

If you create both VM and HSM migration policies at your site, implementing a naming convention that differentiates between the two types simplifies managing your policies from the **Policies** tree after the policies are created.

5. Type a new description of the migration policy in the **Policy Description** box.
6. Modify the remaining settings as desired.
7. After you finish defining the data to be migrated, click **Modify** at the bottom of the pane to create the migration policy.

The migration policy appears under the **Policies** tree.

After one or more migration policies are defined, you can preview the results of the migration, then migrate the data.

Deactivate a migration policy in Volume Management

Make a migration policy inactive to prevent future evaluations and migrations. Inactive policies are grayed out in the **Policies** tree.

1. Open the **Volume Management** view.
2. Select an existing migration policy from the **Policies** tree.
3. Click the **Definition** tab.

The selected migration policy's settings appear.

4. In the **Policy Status** section, click **Inactive**.

Migrating volume data

Introduction to migrating volume data

After defining the data to be migrated, you can migrate the data to the specified location. Migrating data involves previewing the migration results, marking the files for migration, and then migrating the data.

Previewing migration results allows you to sort your volume files based on specified migration policies and review the list of files to be migrated. Use the preview functionality to run *what if* scenarios to assess the impact of a specific migration policy before implementing the migration. For example, if you change the last access date of a migration policy from 90 days to 200 days, you can easily determine how much more data will be moved between tiers or volumes by running a preview. You can print the result or save the results to a file.

When you are satisfied with the migration results, you can mark the files for migration and migrate the files.

Note:

For better performance, mark files for migration using the **mark_for_migrate** utility rather than the **Mark for Migration** button in the Volume Management application. You can run this utility overnight using a Windows **at** command or running a Linux **cron** job.

Migration methods differ depending on whether you are migrating data from a volume management (VM) migration policy or a hierarchical storage management (HSM) migration policy.

VM migration policies can be migrated using one of the following methods:

- *FMS-based migration* uses the FSC process to migrate files from the source volume to the destination volume. This method does not require a third-party migration tool and works without additional configuration.
- *Basic migration* loosely couples your third-party migration tool with Teamcenter. Pending migration file sets (**VM_Migration** files) are exported to a valid operating system directory, either in text or XML format. The third-party migration tool reads the exported file to perform the migration.

Note:

Before using this method, your third-party migration tool must be configured to work with Teamcenter.

Tip:

Before migrating VM migration policies, use the **review_volumes** utility to delete unreferenced files.

HSM migration policies can be migrated using one of the following methods:

- *Basic migration* loosely couples third-party HSM storage systems with Teamcenter. Pending migration file sets (**HSM_Migration** files) are exported to a valid operating system directory, either in text or XML format. The HSM storage systems read the exported file to perform the migration.

Tip:

Before migrating VM migration policies, use the **review_volumes** utility to delete unreferenced files.

- *API-based migration* tightly couples third-party HSM storage systems with Teamcenter. Pending migration file sets are routed through API (user exit) calls to the HSM storage systems. This method allows you to replace the base user exit extension with a custom extension using the Business Modeler IDE. Use the **HSM_migrate_files_msg** operation (under **HSM_Policy** business objects) in the Business Modeler IDE.

Preview migration results

Use the **Preview** functionality to evaluate how much data is migrated between volumes (for VM policies) or migration tiers (HSM policies) for a selected migration policy. The migration information appears in a pie chart illustrating the storage space saved by migrating the selected data (relative to the capacity of the source volume) or in a text-based report detailing the number and size of migrated files.

Note:

Siemens Digital Industries Software recommends that you preview migration information before actually migrating the data.

1. Open the **Volume Management** view.
2. From the **Policies** tree, select a policy, either a volume management (VM) policy or a hierarchical storage management (HSM) policy.
3. Click the **Migration** tab.

The policy name and description appear at the top of the **Migration** pane.

If you select a VM policy, the **Migration Volumes** section displays the source and destination volume for the select migration policy. If you select an HSM policy, this section displays the tier from which the data is being migrated and the tier to which it is being migrated.

4. In the **Function Options** section, click **Preview**.
5. Run the preview.

- At the bottom of the pane, click **PieChart** .

A pie chart appears in the **Preview Report** section.

- At the bottom of the pane, click **Evaluate** .

A text-based report appears in the **Preview Report** section.

6. (Optional) To the right of the **Preview Report** section, click **Save**  to save the preview to a file, or **Print**  to print the preview.

Migrate VM policy data using FMS-based migration

Use the **Migration** functionality to migrate VM migration policy data from a source volume to a destination volume using FMS-based migration. Teamcenter's FMS APIs for migration perform the file migration and commit migration-related changes to the database.

1. Open the **Volume Management** view.
2. From the **Policies** tree, select a volume management (VM) policy.
3. Click the **Migration** tab.

The policy name and description display at the top of the **Migration** pane. Verify that the selected policy is a VM migration policy.

The **Migration Volumes** section displays the source and destination volume for the select migration policy.

4. In the **Function Options** section, click **Migration**.
5. At the bottom of the pane, click **Mark For Migration**.

Any files defined by the selected migration policy that meet the migration criteria are added to a *pending migration request* and displayed in the **Pending Migration Requests** section.

6. Manage the pending migration requests in the list using any of the buttons to the right of the **Pending Migration Requests** section.
7. Select the pending file request to be migrated and click the **Migrate** button  to the right of the **Pending Migration Requests** section.

The **Migration** dialog box appears.

8. Select **Use FMS APIs for Migration** and click **OK**. A progress indicator displays the migration progress as the FMS APIs perform the file migration and the changes are committed to the database.

Note:

If an error occurs, the current file is rolled back; files from the migration set that already migrated without error are stored to the database. If you rectify the error and reselect the same pending file set for migration, the second iteration of the migration begins from the current file.

9. When the migration completes, a **Migration successful** message appears. Click **OK**.

Migrate VM policy data using basic migration

Use the **Migration** functionality to migrate volume management (VM) migration policy data from a source volume to a destination volume using basic migration. This method requires third-party migration tools.

Note:

Before using this migration method, you must export the pending file sets to the operating system directory upon which your third-party migration tool is running. The migration tool reads the exported file and performs the actual migration. The following steps ensure the migration information is committed to the database. The pending file sets can be exported in either XML or text format.

1. Open the **Volume Management** view.
2. From the **Policies** tree, select a VM policy.
3. Click the **Migration** tab.

The policy name and description appear at the top of the **Migration** pane. Verify that the selected policy is a VM migration policy.

The **Migration Volumes** section displays the source and destination volume for the select migration policy.

4. In the **Function Options** section, click **Migration**.
5. At the bottom of the pane, click **Mark For Migration**.

Any files defined by the selected migration policy that meet the migration criteria are added to a *pending migration request* and displayed in the **Pending Migration Requests** section.

6. Manage the pending migration requests in the list using any of the buttons to the right of the **Pending Migration Requests** section.
7. Select the pending file request to be migrated and click the **Export** button  to the right of the **Pending Migration Requests** section.

The pending file sets are exported in either XML or text format to the operating system directory upon which your third-party migration tool is running. The migration tool reads the exported file and performs the actual migration.

Caution:

Ensure this step actually takes place before proceeding.

8. Select the pending file request to be migrated and click the **Migrate** button  to the right of the **Pending Migration Requests** section.

The **Migration** dialog box appears.

9. Select **Use 3rd Party Tool for Migration** and click **OK**.
10. Click **Yes** when the following message appears to confirm you have exported the pending file set to the operating system for third-party tool migration:

Has the selected pending file set been exported for
third party tool migration?

A progress indicator appears as Teamcenter commits the file migration information to the database.

11. After the migration completes, a **Migration successful** message appears. Click **OK**.

Migrate HSM policy data using basic migration

Use the **Migration** functionality to migrate hierarchical storage management (HSM) migration policy data from a primary tier to a secondary or tertiary tier using basic migration.

Note:

Before using this migration method, you must export the pending file sets to the operating system directory upon which your third-party migration tool is running. The migration tool reads the exported file and performs the actual migration. The following steps ensure the migration information is committed to the database. The pending file sets can be exported in either XML or text format.

1. Open the **Volume Management** view.

2. From the **Policies** tree, select an HSM policy.
3. Click the **Migration** tab.

The policy name and description appear at the top of the **Migration** pane. Verify that the selected policy is an HSM migration policy.

The **Migration Volumes** section displays the tier from which the data is being migrated, and the tier to which it is being migrated.

4. In the **Function Options** section, click **Migration**.
5. At the bottom of the pane, click **Mark For Migration**.

Any files defined by the selected migration policy that meet the migration criteria are added to a *pending migration request* and displayed in the **Pending Migration Requests** section.

6. Manage the pending migration requests in the list using any of the buttons to the right of the **Pending Migration Requests** section.
7. Select the pending file request to be migrated and click the **Export** button  to the right of the **Pending Migration Requests** section.

The pending file sets are exported in either XML or text format to the operating system directory upon which your third-party migration tool is running. The migration tool reads the exported file and performs the actual migration.

Caution:

Ensure this step actually takes place before proceeding.

8. Select the pending file request to be migrated and click the **Migrate** button  to the right of the **Pending Migration Requests** section.

The **Migration** dialog box appears.

9. Select **Use Loosely Coupled (Basic) HSM Migration** and click **OK**.
10. Click **Yes** when the following message appears to confirm you have exported the pending file set to the operating system for third-party HSM tool migration:

Has the selected pending file set been exported for
third party HSM tool migration?

A progress indicator appears as Teamcenter commits the file migration information to the database.

11. After the migration completes, a **Migration successful** message appears. Click **OK**.

Migrate HSM policy data using API-based migration

Use the **Migration** functionality to migrate hierarchical storage management (HSM) migration policy data from a primary tier to a secondary or tertiary tier using API-based migration. This method tightly couples third-party HSM storage systems with Teamcenter. Pending migration file sets are routed through API (user exit) calls to your HSM storage system. The API calls perform the actual migration and commit the changes to the database.

Note:

HSM API-based migration requires tight integration with your third-party HSM system.

1. Open the **Volume Management** view.
2. From the **Policies** tree, select an HSM policy.
3. Click the **Migration** tab.

The policy name and description display at the top of the **Migration** pane. Verify that the selected policy is an HSM migration policy.

The **Migration Volumes** section displays the tier from which the data is being migrated, and the tier to which it is being migrated.

4. In the **Function Options** section, click **Migration**.
5. At the bottom of the pane, click **Mark For Migration**.

Any files defined by the selected migration policy that meet the migration criteria are added to a *pending migration request* and displayed in the **Pending Migration Requests** section.

6. Manage the pending migration requests in the list using any of the buttons to the right of the **Pending Migration Requests** section.
7. Select the pending file request to be migrated and click the **Migrate** button  to the right of the **Pending Migration Requests** section.

The **Migration** dialog box appears.

8. Select **Use Tightly Coupled (API Based) HSM Migration** and click **OK**.
9. Click **Yes** when the following message appears to indicate you are using a third-party HSM storage system:

Has the 3rd party file migration tool been integrated and configured with Teamcenter?

The selected data is migrated. A progress indicator displays the migration progress as the APIs perform the file migration and the changes are committed to the database.

Note:

If an error occurs, the current file is rolled back; files from the migration set which have already migrated without error are stored to the database. If you rectify the error and re-select the same pending file set for migration, the second iteration of the migration begins from the current file.

10. After the migration completes, a **Migration successful** message appears. Click **OK**.

Volume Management migrated file set format

Pending migrated file sets can be exported to any valid operating system directory in either XML or text format. Your third-party HSM application reads these files to perform the migration. The following code is an example of an XML DTD format for exporting pending migration sets.

The DTD is common to both VM and HSM migration policies.

```
<!-- main structure policyInfo.dtd file -->

<!ELEMENT policyInfo (enterpriseId, pendingFileSet+ )>

<!ELEMENT enterpriseId (#PCDATA)>

<!ELEMENT pendingFileSet (purgeInfo,tierInfo? )>

<!ELEMENT purgeInfo (purgeImmediately,purgeWaterMark?)>

<!ELEMENT purgeImmediately (#PCDATA)>

<!ELEMENT purgeWaterMark (lowWaterMark, highWaterMark)>

<!ELEMENT lowWaterMark (#PCDATA)>

<!ELEMENT highWaterMark (#PCDATA)>

<!ELEMENT tierInfo (migrateFromTier?, migrateToTier?,
volumeInfo+ )>

<!ELEMENT migrateFromTier (#PCDATA)>

<!ELEMENT migrateToTier (#PCDATA)>
```

```

<!ELEMENT volumeInfo (volumeName, nodeName, filePath+)>

<!ELEMENT volumeName (#PCDATA)>

<!ELEMENT nodeName (#PCDATA)>

<!ELEMENT filePath (#PCDATA )>

```

The following code is an example of a VM migration instruction in XML format:

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<policyInfo>
  <enterpriseId>-2035849880</enterpriseId>
  <PendingFileSet>
    <pendingObjUid>yTZFeW53opnVGC</pendingObjUid>
    <purgeInfo>
      <purgeImmediately>TRUE</purgeImmediately>
    </purgeInfo>
    <migrateFromSource>
      <volumeInfo>
        <volumeName>volume</volumeName>
        <volumeNode>SVI6W181</volumeNode>

<filePath>d:\apps\Siemens\volume\dba_51e2c745\mds_def_tex_j34034w4zk6p6.txt</filePath>
<filePath>d:\apps\Siemens\volume\dba_51e2c745\070000_unk_fho00tw4zk6r3.xlsm</filePath>
<filePath>d:\apps\Siemens\volume\dba_51e2c745\070000_unk_lxx00tw4zk6r1.xlsm</filePath>
<filePath>d:\apps\Siemens\volume\dba_51e2c745\070000_unk_1io00tw4zk6qz.xlsm</filePath>
<filePath>d:\apps\Siemens\volume\dba_51e2c745\070000_unk_j1p00tw4zk6qx.xlsm</filePath>
      </volumeInfo>
    </migrateFromSource>
    <migrateToDestination>
      <volumeInfo>
        <volumeName>volume2</volumeName>
        <volumeNode>SVI6W181</volumeNode>

<filePath>d:\apps\Siemens\volume2\dba_51e9a92a\mds_def_tex_j34034w4zk6p6.txt</filePath>
<filePath>d:\apps\Siemens\volume2\dba_51e9a92a\070000_unk_fho00tw4zk6r3.xlsm</filePath>
<filePath>d:\apps\Siemens\volume2\dba_51e9a92a\070000_unk_lxx00tw4zk6r1.xlsm</filePath>
<filePath>d:\apps\Siemens\volume2\dba_51e9a92a\070000_unk_1io00tw4zk6qz.xlsm</filePath>
<filePath>d:\apps\Siemens\volume2\dba_51e9a92a\070000_unk_j1p00tw4zk6qx.xlsm</filePath>
      </volumeInfo>
    </migrateToDestination>
  </PendingFileSet>
</policyInfo>

```

The following code is an example of a VM migration instruction in text file format:

```

# Note: Lines starting with hash(#) provide HSM migration data. If this file
# is used with IBM TSM HSM migration utility, it logs error code 8 on lines
# starting with #. The user can ignore those errors or alternatively edit this
# file to remove lines starting with #, after comprehending the information.

# enterpriseId = -2035849880
# BEGIN:
# purgeImmediately = TRUE

# migrateFromSourceVolume = volume
# nodeName = SVI6W181
d:\apps\Siemens\volume\dba_51e2c745\070000_unk_j1p00tw4zk6qx.xlsm
d:\apps\Siemens\volume\dba_51e2c745\070000_unk_1io00tw4zk6qz.xlsm
d:\apps\Siemens\volume\dba_51e2c745\070000_unk_lxx00tw4zk6r1.xlsm
d:\apps\Siemens\volume\dba_51e2c745\070000_unk_fho00tw4zk6r3.xlsm
d:\apps\Siemens\volume\dba_51e2c745\mds_def_tex_j34034w4zk6p6.txt

# migrateToDestinationVolume = volume2
# nodeName = SVI6W181

d:\apps\Siemens\volume2\dba_51e9a92a\070000_unk_j1p00tw4zk6qx.xlsm
d:\apps\Siemens\volume2\dba_51e9a92a\070000_unk_1io00tw4zk6qz.xlsm
d:\apps\Siemens\volume2\dba_51e9a92a\070000_unk_lxx00tw4zk6r1.xlsm
d:\apps\Siemens\volume2\dba_51e9a92a\070000_unk_fho00tw4zk6r3.xlsm
d:\apps\Siemens\volume2\dba_51e9a92a\mds_def_tex_j34034w4zk6p6.txt
# END:

```

The following code is an example of an HSM migration instruction in XML format:

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<policyInfo>
  <enterpriseId>-2035849880</enterpriseId>
  <PendingFileSet>
    <pendingObjUid>AuaFFEDvopnVGC</pendingObjUid>
    <purgeInfo>
      <purgeImmediately>TRUE</purgeImmediately>
    </purgeInfo>
    <tierInfo>
      <migrateFromTier>Secondary</migrateFromTier>
      <migrateToTier>Tertiary</migrateToTier>
      <volumeInfo>
        <volumeName>volume</volumeName>
        <volumeNode>SVI6W181</volumeNode>

<filePath>d:\apps\Siemens\volume\dba_51e2c745\mds_def_tex_j34034w4zk6p6.txt</filePath>

<filePath>d:\apps\Siemens\volume\dba_51e2c745\070000_unk_fho00tw4zk6r3.xlsm</filePath>

<filePath>d:\apps\Siemens\volume\dba_51e2c745\070000_unk_lxx00tw4zk6r1.xlsm</filePath>

<filePath>d:\apps\Siemens\volume\dba_51e2c745\070000_unk_1io00tw4zk6qz.xlsm</filePath>

<filePath>d:\apps\Siemens\volume\dba_51e2c745\070000_unk_j1p00tw4zk6qx.xlsm</filePath>
      </volumeInfo>
    </tierInfo>
  </PendingFileSet>
</policyInfo>

```

The following code is an example of an HSM migration instruction in text file format:

```
# Note: Lines starting with hash(#) provide HSM migration data. If this file
# is used with IBM TSM HSM migration utility, it logs error code 8 on lines
# starting with #. The user can ignore those errors or alternatively edit this
# file to remove lines starting with #, after comprehending the information.

# enterpriseId = -2035849880
# BEGIN:
# purgeImmediately = TRUE

# migrateFromTier = Secondary
# migrateToTier = Tertiary

# volumeName = volume
# nodeName = SVI6W181

d:\apps\Siemens\volume\dba_51e2c745\070000_unk_j1p00tw4zk6qx.xlsm
d:\apps\Siemens\volume\dba_51e2c745\070000_unk_1io00tw4zk6qz.xlsm
d:\apps\Siemens\volume\dba_51e2c745\070000_unk_lxx00tw4zk6r1.xlsm
d:\apps\Siemens\volume\dba_51e2c745\070000_unk_fho00tw4zk6r3.xlsm
d:\apps\Siemens\volume\dba_51e2c745\mds_def_tex_j34034w4zk6p6.txt
# END:
```

Volume Management migration reports

Introduction to generating volume migration reports

After migrating data, you can generate migration reports to determine how much data migrated between volumes or tiers and to see which migration policies migrated the data. You can refine reports by date, by policy, or generate a report covering all current migration policies. Reports can be printed or saved to a file.

Reports can be generated using one of the following methods:

- Text-based reports provide an itemized list of the migrated files, the total number of migrated files, and the total size (in bytes) of the migrated data.
- Pie chart reports illustrate the percentage of files that were migrated for each migration policy. These reports can be saved as JPEGs or PNGs.

Note:

For better performance, generate migration reports using either the **vm_report** utility or the **hsm_report** utility rather than the **PieChart** and **Report** buttons in the Volume Management application. You can run them overnight using a Windows **at** command or running a Linux **cron** job.

Generate a VM policy report

Use this report to determine how much data is migrated through a volume management (VM) migration policy. The text report provides an itemized list of the migrated files, the total number of migrated files, and the total size (in bytes) of the migrated data. The pie chart report graphically illustrates the storage space saved by migrating the selected data, relative to the capacity of the source volume. You can save the pie chart report as a JPEG or PNG file.

1. Open the **Volume Management** view.
2. From the **Policies** tree, select a VM policy.
3. Click the **Report** tab.

The policy name and description appear at the top of the **Report** pane. Verify that the selected policy is a VM migration policy.

4. In the **Volume Migration Options** section, select the source volume and destination volume for which you want to generate the report.
5. (Optional) In the **Migration Dates** section, select the beginning and end dates for which you want to generate the report.
6. Alternative to the two previous steps, select **Load Previously Generated Report** to load the last report created.
7. Run the report.
 - Click **Report**  at the bottom of the pane to generate the text-based report.
 - Click **PieChart**  at the bottom of the pane to generate the graphic report.

The report appears in the **Report** section.

8. Manage the report using any of the buttons to the right of the **Report** section.

Generate an HSM policy report

Use this report to determine how much data is migrated through a hierarchical storage management (HSM) migration policy. The text report provides an itemized list of the migrated files, the total number of migrated files, and the total size (in bytes) of the migrated data. The pie chart report graphically illustrates the storage space saved by migrating the selected data, relative to the capacity of the source volume. You can save the pie chart report as a JPEG or PNG file.

1. Open the **Volume Management** view.

2. From the **Policies** tree, select an HSM policy.
3. Click the **Report** tab.

The policy name and description appear at the top of the **Report** pane. Verify that the selected policy is an HSM migration policy.

4. From the **Migration Tier** section, specify the migration path for the selected policy. Click either **Primary-Secondary**, **Secondary-Tertiary**, or **Primary-Tertiary**.

For example, **Primary-Secondary** reports the migrated data from the primary tier to the secondary tier by the selected policy.

5. (Optional) In the **Migration Dates** section, select the beginning and end dates for which you want to generate the report.
6. Alternative to the two previous steps, select **Load Previously Generated Report** to load the last report created.
7. Run the report.

- Click **Report**  at the bottom of the pane to generate the text-based report.

Click **PieChart** at  the bottom of the pane to generate the graphic report.

The report appears in the **Report** section.

8. Manage the report using any of the buttons to the right of the **Report** section.

Generate a report on all policies

Use this report to determine how much data is migrated using all currently defined policies, either volume management (VM) policies or hierarchical storage management (HSM) policies. The report format can be either text-based or in a pie chart.

1. Open the **Volume Management** view.
2. From the **Policies** tree, select a policy, either a VM policy or an HSM policy.

If you select a VM policy, a report is run on all defined VM policies. If you select an HSM policy, a report is run on all HSM policies.

3. Click the **Report** tab.
4. Select **All Policies** to generate a report on all defined policies.

5. Refine the report.

- If you select a VM policy, in the **Volume Migration Options** section, select the source volume and destination volume for which you want to generate the report.
- If you select an HSM policy, in the **Migration Tier** section, specify the migration path for the selected data. Click either **Primary-Secondary**, **Secondary-Tertiary**, or **Primary-Tertiary**.

For example, **Primary-Secondary** migrates data from the primary volume to the secondary volume.

6. (Optional) In the **Migration Dates** section, select the beginning and end dates for which you want to generate a report.
7. Alternative to the two previous steps, select **Load Previously Generated Report** to load the last report created.
8. Click **PieChart**  at the bottom of the pane to illustrate the percentage of files that were migrated for each migration policy in the **Report** section.
9. Click **Report**  at the bottom of the pane to generate the total number and size of all files that are migrated by each policy in the **Report** section.
10. Manage the report using any of the buttons to the right of the **Report** section.

28. File Management System

Introduction to File Management System

Teamcenter's File Management System (FMS) is a file storage, caching, distribution, and access system. FMS provides global, secure, high performance and scalable file management.

Use FMS to centralize data storage volumes on reliable backup file servers, while keeping data close to users in shared data caches. This enables centralized storage and wide distribution of file assets to the needed locations within a single standard file management system. FMS provides WAN acceleration to effectively move large files across WAN assets.

FMS pulls files on demand as users request them. FMS efficiently transfers files across a wide area network (WAN). Also, FMS can locate caches closer to end user machines, for example, FMS server caches (FSCs). FMS uses a *file GUID*, a business neutral identifier for file contents, to determine when to pull a file from its local cache, rather than retrieving the file across a network from the vault's underlying file system. Every file in a Teamcenter vault has a single file GUID associated with every replicated copy of the file. If you move, copy, reassign to a new owner, or rename the file, its file GUID remains the same. However, if you change the file content by one bit or change its language encoding, a new file GUID must be created to describe the file's new contents.

Note:

Siemens Digital Industries Software reserves the right to change FMS behavior in the future to enhance performance or improve reliability.

FMS consists of two main components:

- FMS server cache (FSC)

Provides a shared, secure, server-level cache. It uploads and downloads files to other FSCs and to client caches.

An FSC can provide one or more modes of behavior, where each mode manages a type of data including volume files, cache files, transient files, and configuration files. A particular FSC can perform any or all of these functions simultaneously depending on your FMS configuration. All FSCs provide at least one mode in a properly configured FSC topology.

You define configuration, volume, and transient file modes explicitly in the FMS configuration files using XML statements. Cache server functionality is installed on each FSC but is only used if the FSC does not have direct access to volume files.

- FMS client cache (FCC)

Provides a private user-level cache, just as Web browsers provide a read file cache. The FCC provides a high-performance cache for both downloaded and uploaded files. The FCC provides proxy interfaces to client programs and connectivity to the server caches and volumes.

Any files captured by the FCC do not change, for either download or upload, and for either whole files or partial files. All file copies and file segment copies are identical throughout the system and never updated. New file versions are checked into the system with a new GUID, but a file with an existing GUID in the FMS system never changes. Therefore, there are no issues with file change or cache consistency.

FCCs also provide access to the transient volume for the business server in a Teamcenter two-tier configuration. The business server writes or reads temporary files directly to a disk directory, and the rich clients access those files using the standard FCC interfaces. This provides client independence from the system configuration and ensures that client programs operate the same in both two-tier and four-tier mode for file access functions.

Benefits of using FMS

The benefits of using FMS include:

- Data distribution

Administrators can distribute copies of data closer to end users by using FMS server caches at remote locations. FMS cache servers can be distributed worldwide, while retaining FMS volume data in central storage.

- Multisite support

FMS enables file transfer directly between servers in two different PLM systems, eliminating the need for an intermediate transfer directory.

- All network configurations supported

FMS supports LAN, WAN, and firewall configurations.

- Common caching system

FMS caches all data for all clients and provides standard interfaces for file access across client and server programs. FMS eliminates the need for client specific caches.

- Pull-through caching

FMS automatically caches data at the locations needed, based on what data users read and write to the system.

- No single point of failure

FMS provides the capability to administer a fully redundant configuration of configuration servers, volume servers, and cache servers. FMS routing algorithms automatically search for an alternate path in the case of a connection failure.

- Primary configuration server

FMS provides the capability to administer the FMS deployment configuration with a single primary (master) configuration file. FMS automatically distributes the configuration file to all FMS client and server processes.

- Managed caches

The FMS client and server caches are self-purging. The least recently accessed data is purged first.

- Secure server caches

FMS servers do not permit any direct access to cached file data. FMS permits file data access when the requester presents a valid security ticket.

- Secure volume servers

FMS servers do not permit any direct access to volume file data. FMS permits file data access only when the requester presents a valid security ticket.

- Private user caches

FMS automatically caches data downloaded or uploaded by Visualization clients in a private user cache, providing fast access to recently accessed files. The user cache automatically purges data to fit within a maximum size.

- Streamed data delivery

FMS streams data from volumes down to clients through any number of cache servers. Data becomes available to the user as soon as the first bits stream in, through any number of cache servers as needed. FMS also streams data from the client all the way to the volume on upload.

- Segment file cache and delivery

FMS allows applications to transfer only specific parts of a file, improving the overall transfer time and conserving network bandwidth.

FMS components

FMS server cache

The FMS server cache (FSC) is the name of the FMS server cache server process. The FSC is a shared, secure, server level cache. It uploads and downloads files to other FSCs and to FCCs.

An FSC can provide one or more modes of behavior, where each mode manages a type of data including volume files, cache files, transient files, and configuration files. A particular FSC can perform any or all of these functions simultaneously depending on your FMS configuration. All FSCs provide at least one mode in a properly configured FSC topology.

You define configuration, volume and transient file modes explicitly in the FMS configuration files using XML statements. Cache server functionality is installed on each FSC, but is only used if the FSC does not have direct access to volume files. The various FSC modes are as follows:

- Configuration server (optional)

One or more FSCs may be designated as a configuration server. An FSC configuration server reads the **fmsmaster_fscid.xml** configuration file and distributes that information to other FSCs and/or clients. The FSC configuration topology can be a single FSC or a tree of FSCs. The FSC configuration topology is separate from the FSC routing topology.

- Volume server (optional)

An FSC may contain zero or more mounted volumes. An FSC serves volume data by reading/writing the file directly from local or mounted disk, and writing/reading that data onto a TCP port in HTTP protocol.

- Cache server (conditional)

An FSC caches any data not directly available in a volume mounted on that FSC. An FSC routes and caches data from other FSCs if it does not have direct access to the volume containing the file.

- Transient file server (for Teamcenter four-tier configurations only, on each business logic server. Use FCC in transient file server mode with Teamcenter two-tier configurations).

Each business logic server in four-tier mode writes and reads data from a temporary disk location. The FSC provides the capability to deliver this temporary data to or from the client. Each business logic server should have an FSC transient server to deliver the temporary data.

The transient volume directory must reside on the same machine as the FSC and the Pool Server Manager.

FSC basic functions are:

- Segment read file cache

The FSC stores all file downloads as 16K file fragments in the read segment cache. Whole *files* are not stored. The segment cache uses the FMS fast cache. No purge policies are needed for the fast cache; it automatically purges file segments as needed.

- Segment write file cache

The FSC stores all file uploads as 16K file fragments in the write segment cache. Whole files are not stored.

- WAN acceleration (optional)

The FCC provides the capability to accelerate file downloads over high latency or noisy WAN lines. Required software is shipped as part of the base FMS install.

- Server configuration

The FSC reads and distributes FMS configuration data across site FSCs and FCC processes.

- Client configuration

The FSC processes the client configuration, analyzes the configuration, computes the configuration download for each client, and provides this information as a bootstrap download when the FCC process initializes.

- Failover (configuration dependent)

The FSC provides the capability to failover to alternate FSCs upon failure of a specific FSC. Failover is provided for access to both FSC configuration servers and FSC file servers.

- Whole file cache (WFC)

The files are whole files on the disk and not in a virtual memory mapped disk space.

FMS client cache

The FMS client cache (FCC) is the name of the FMS client cache server process. The FCC runs within the Teamcenter client communication (TCCS) module. The FCC provides a private user-level cache, just as web browsers provide a read file cache. The FCC provides a high performance cache for both downloaded and uploaded files. The FCC provides proxy interfaces to client programs and connectivity to the server caches and volumes.

Any files captured by the FCC do not change, for either download or upload, and for either whole files or partial files. All file copies and file segment copies are identical through out the system, and never

updated. New file versions are checked into the system with a new GUID, but a file with an existing GUID in the FMS system never changes. Thus, there are no issues with file change or cache consistency.

The FCC can act as a transient volume for the business server in a Teamcenter two-tier configuration. The business server writes or reads temporary files directly to a disk directory, and the rich clients access those files via the standard FCC interfaces. This provides client independence from the system configuration, and ensures that client programs operate the same in both two-tier and four-tier mode for file access functions.

FCC basic functions are as follows:

- Whole file read cache

The FCC caches downloaded whole files in a whole file read cache. Most applications access a whole file at a time, including Teamcenter rich client, Microsoft Office products, 2D viewers, and others.

- Whole file write cache

The FCC caches uploaded whole files in a whole file write cache. All rich client applications upload completely new files through this cache. Files are never changed once they are uploaded to the system, there are only new files. The cache is always consistent.

- Segment read file cache

The FCC caches file fragments read by smart clients. These clients are FMS aware, and call a specific FCC programming interface in order to reduce the amount of data downloaded to the client for processing and display.

- WAN acceleration (optional)

The FCC provides the capability to accelerate file downloads over high latency or noisy WAN lines. Required software is shipped as part of the base FMS install.

- Managed cache

The FCC automatically purges old cache data based on configuration parameters. Separate sizing parameters are provided for each cache type including whole file read, whole file write, and segment file read.

- Client configuration

The FCC reads a local **fcc.xml** configuration file and uses this information to bootstrap a server based configuration download. This provides the capability to centrally manage FCC configuration parameters with a minimum amount of configuration data installed on the client.

- Failover (configuration dependent)

The FCC provides the capability to fail over to alternate FSCs on failure of a specific FSC. Failover is provided for access to both FSC configuration servers and FSC file servers. FSC configuration servers are initially used to download the FCC configuration. Once the FCC configuration is downloaded, the FCC uses the FSC file servers specified in the downloaded configuration.

FMS configuration files

Introduction to FMS configuration files

FMS uses the following files to configure server-client file handling:

- FMS primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

See the *FSC_HOME/fmsmaster.xml.template* file for all available elements.

- FMS server configuration file (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

- FMS client configuration file (*FMS_HOME/fcc.xml*)

See the *FMS_HOME/fcc.xml.template* file for all available elements.

- FMS server configuration properties file (*FMS_HOME/fsc.properties*)

See the *FSC_HOME/fsc.properties.template* file for all available elements.

The **fmsmaster_fscid.xml** and **fsc.xml** files are located in the *FSC_HOME* directory (for example, *TC_ROOT\fsc*). The **fcc.xml** file is located in the **FMS_HOME** directory (for example, *TC_ROOT\tccs*).

Grammatical elements provided within the FMS configuration files syntax are as follows:

- Substitution elements

Use substitution elements such as **\$HOME** within string values to simplify parameter specifications within the configuration file.

- Directory paths

Directory paths may use either local computer conventions with forward or reverse slashes, or UNC style conventions such as **//server1/share/volume1**.

- Windows/Linux parameter values

Specify separate Windows and Linux parameter values by using a vertical **or** character, for example, **\$HOME/FscCache|tmp/FscCache**.

Editing FMS configuration files

FMS configuration files are XML files containing structures that must adhere to rules defined in their document type definition (DTD) files. If you are an administrator who edits XML files for use in Teamcenter, Siemens Digital Industries Software strongly recommends that you have training in XML editing and DTD files.

If any edits in a configuration violate the rules in the file's DTD, when you attempt to reload the file, the following error is reported:

```
Error, server returned status code: 400, status message:
CONFIGURATION_PARSE_ERROR_UNKNOWN_DETAILS_2
```

This message means the configuration file does not parse properly as defined by the DTD. Every configuration file declares the name of its particular DTD file in the **DOCTYPE** element found in the second line of the file.

Example:

```
<!DOCTYPE fmsworld SYSTEM "fmsmasterconfig.dtd">
```

If you encounter such an error, examine the details in the error message and correct the configuration file so that it parses correctly. You may need to examine the DTD file to learn which parsing rules were violated.

Find the DTD files in the *FSC_HOME\fmsutil.jar* file in the *com\teamcenter\fms\config* directory. The available DTD files include:

- *fmsmasterconfig.dtd*

Specifies the DTD for FMS primary configuration files.

- *fscconfig.dtd*

Specifies the DTD for FSC configuration files.

- *fccconfig.dtd*

Specifies the DTD for FCC configuration files.

FMS primary configuration file

The FMS primary configuration file (**fmsmaster_fscid.xml**) is used to manage the FMS configuration. This file contains routing information and the FSC and FCC defaults. The **fmsmaster_fscid.xml** file is located in the *FSC_HOME* directory. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements.

A basic FMS primary configuration file contains the following elements:

- **fmsworld**

The containing XML element.

- **fmsenterprise**

Contains the primary configuration content, including the FMS topology and the FSC and FCC parameter defaults.

May be followed by a `<ticketdigest value="SHA-256" />` tag as part of **enabling SHA-256 digest algorithm** for tickets.

- **fccdefaults**

Contains the parameter defaults for all the site FSCs and indicates which of these parameters can be overridden by a particular FCC installation.

- **fscGroup**

Contains a list of all the FSCs installed as part of a LAN configuration. One or more groups can be defined. FSCs within a defined group cannot be deployed across a WAN. FMS assumes that file transfers between two FSCs within an FSC group are directly routed with no WAN acceleration.

- **volume** (optional)

Describes where the FSC mounts volumes. An FSC may mount one or more volumes.

- **transientvolume** (optional)

Specifies the temporary directory used by Teamcenter business servers for writing and reading temporary files in four-tier configurations. Users access these files using the Teamcenter rich client.

Note:

Do not use the following symbols as delimiter characters within an **fmsenterprise** ID, **fscGroup** ID, or **fsc** ID:

- Slash (/)
- Question mark (?)
- Equal sign (=)

- Number sign (#)

FMS server configuration file

Overview of the FMS server configuration file

The FMS server configuration file (**fsc.xml**) is installed for each server. This file identifies the server configuration within the system. Optionally, it can override FSC and FCC parameter defaults specified in the **fmsmaster_fscid.xml** file. The **fsc.xml** file is located in the *FSC_HOME* directory. See the *FSC_HOME/fsc.xml.template* file for all available elements.

A basic FSC configuration file contains the following elements:

- **fscconfig**

The outer containing XML element.

- **fscdefaults**

Defines or overrides FSC default values from the primary configuration file (**fmsmaster_fscid.xml**).

- **fmsmaster**

Specifies the location from which to download FMS configuration information. The location can be either the disk location of the **fmsmaster_fscid.xml** file or the address of another FSC.

Downloaded FMS configuration information results from the merge of the **fmsmaster_fscid.xml** file and the **fsc.xml** file of the FSC from which the configuration is downloaded.

Resulting FSC configuration information results from the merge of the **fmsmaster_fscid.xml** file and the local **fsc.xml** file of the FSC from which the configuration is downloaded.

FSC configuration paths can branch and can be any depth. You can specify more than one FSC for configuration download to provide configuration server failover.

- **fsc**

Specifies the identity of the installed FSC. This identity must match an FSC defined in the primary configuration file and be within an FSC group.

Note:

fccdefaults elements have been deprecated from *fsc.xml*.

Place **fccdefaults** elements in *fmsmaster.xml*, as they apply to the configuration of clients assigned to the FSC in question using the **clientmap**. A complete set of each FSC's **fccdefaults**

parameter defaults (not including those set in the client's *fcc.xml* file) is available from any FSC configuration server. This is not the case when **fccdefaults** elements are placed in an *fsc.xml* configuration file, so Siemens Digital Industries Software recommends not placing any **fccdefaults** elements in *fsc.xml*.

General FSC configuration parameters

The following table lists some of the parameters defined in the FMS server configuration file (*FSC_HOME/fsc.xml*). See the *FSC_HOME/fsc.xml.template* file for all available elements.

Name	Default value	Description
FSC_BULKTRANSFERMAXTHREADS	8	Defines the maximum number of threads the server uses when performing bulk transfer operations for files between two volumes.
FSC_DelayedVolumeValidation	false	Enables delayed file store validation and allows offline volumes. By default, an FSC validates all disk locations before starting and before a configuration reload. Validation fails if any disk locations are unavailable. Set this element to true if your site has a very large number of disk locations. This defers disk validation to a background thread, reducing start up time. And access to offline disk locations return an error indicating an <i>offline</i> condition, rather than <i>file not found</i>.
FSC_EnableMonitoring	true	Not used.
FSC_LogLevel	WARN	Defines the FSC logging level. Valid values, in order of increasing detail, are OFF, ERROR, WARN (the default), INFO, DEBUG, and TRACE. (SEVERE can be used as a synonym for ERROR.) Caution: Setting FSC_LogLevel to DEBUG or TRACE produces a large volume of logging output. Avoid using these logging levels in production environments.
FSC_MaximumConnectionIdleTimeMs	90000	Sets the outgoing connection time-out (in milliseconds). This is essentially a write time-out. If the connection sends less than one byte of data to the recipient in this amount of time, it is closed.
FSC_MaximumHttpsConnectionIdleTimeMs	90000	Sets the idle connection time-out (in milliseconds). This is essentially a garbage connection

Name	Default value	Description
		cleanup. This parameter replaces the FSC_MaximumHttpConnectionIdleTimeMs parameter but only for HTTPS connections. If a secure (HTTPS) connection is not used for this amount of time, it is closed.
FSC_MaximumThreadIdleTimeMs	10000	Defines the amount of time (in milliseconds) a thread in the service pool remains idle before it is removed and destroyed. This parameter is subject to the value of the FSC_MinimumThreads parameter.
FSC_MaximumThreads	255	Defines the maximum number of threads the server uses to service incoming requests.
FSC_MinimumThreads	5	Defines the minimum number of threads the server retains to service incoming requests.
FSC_TraceLevel	2	Ignored. Tracing is not enabled.
FSC_TransferRemoteMetadataFile	true	Enables the transfer of metadata files used for data transmission verification .
FSC_UploadTimeoutMs	30000	Defines the amount of time (in milliseconds) an upload attempt waits to connect before failing over to another route, or failing the operation if no other route exists.

FSC whole file cache parameters

The following table lists the **whole file cache** parameters defined in the FMS server configuration file (*FSC_HOME/fsc.xml*). See the *FSC_HOME/fsc.xml.template* file for all available elements.

Name	Default value	Description
FSC_WholeFileCacheLocation	\$HOME/FSCCache /scratch/FSCCache	Specifies the absolute paths to the root directory of each partition containing the FSC whole cache files. The operating system's file system manages this storage by partition. Only one path should appear on each media partition. Directory permissions should be limited to the authorized user ID/account. You can map multiple partitions in this parameter, for example: <pre><property name="FSC_WholeFileCacheLocation" value="E:\Wholefilecache,F:\Wholefilecache,G:\Wholefilecache,"</pre>

Name	Default value	Description
		<pre>overridable="true" /></pre> Each partition's root path is separated by either a comma or a platform-specific path separator character (; on Windows or : on Linux). All cache partitions must be within 10% of each others' size. Note: If you add or remove partitions, you must run the FSCWholeFileCacheUtil utility with the -redistrib argument to redistribute the cache contents among the new partition spaces before changing the fmsmaster.xml configuration file and relaunching a new FSC instance. Otherwise, the existing partitions continue to own all of the slots, and no files are eligible for storage in the added partitions.
FSC_FilesPerWholeFileCacheDir	0	Specifies the maximum number of files per whole file cache subdirectory. Valid values are 1024 through 10240. Set to 0 to disable the whole file cache. If the number of files stored in the subdirectory exceeds the configured amount, files are automatically purged to the maximum number of files specified.
FSC_DiskPercentFreeGoal	30	Specifies the percentage of free disk space on the disk containing the whole file cache. Valid values are 0 through 100. When free disk space falls below the specified value, files are automatically purged until the specified percentage of free disk space is reached.
FSC_MaxCacheAgeDays	3650	Specifies (in days) the maximum time a file is not accessed. Cache files not accessed in the specified time are purged from the whole file cache. Valid values are 0 through 3650. Any files not accessed within the specified time are purged during the next scheduled purge.

Name	Default value	Description
FSC_WholeFilePurgePeriodMinutes	240	Specifies the time (in minutes) between background cache purge cycles. Valid values are 0 through 10080. Set to 0 to run the background purge process continuously. If a background purge cycle exceeds the specified time, one more purge cycles are skipped as necessary to maintain the specified schedule.
FSC_WholeFilePurgeInitialWaitMinutes	0	Specifies the time (in minutes) to wait after starting the FSC before starting the first background cache purge cycle. Valid values are -1 through 2880. Set to -1 to disable the background purge process. To run the background cache purge cycle as an overnight cron job, set this parameter to -1 to disable the automatic background cache purge, and then use a cron job to run the purgeFSCWholeFileCache script. The script is generated by the FSC on startup and stored in the FSC cache directory. The script uses the FSCWholeFileCacheUtil command line utility to purge the cache.

FSC read cache parameters

The following table lists the read cache parameters defined in the FMS server configuration file (*FSC_HOME/fsc.xml*). See the *FSC_HOME/fsc.xml.template* file for all available elements.

Name	Default value	Description
FSC_ReadCacheLocation	\$HOME\FSCCache/tmp/FSCCache	Defines the read cache location. The value is entered through Teamcenter Environment Manager during FSC installation. Modify the value either by running Teamcenter Environment Manager in maintenance mode or by manually editing the fmsmaster_fscid.xml file. All cache directories must be on a local hard disk. Cache directories cannot be

Name	Default value	Description
		on a file share directory or a mapped drive.
FSC_MaximumReadCache FilePages	40960	Defines maximum number of file pages in the read cache.
FSC_MaximumReadCache Segments	9216	Defines the maximum number of read cache segments.
FSC_ReadCacheHash BlockPages	2048	Defines the maximum number of hash block pages. Each hash block contains 128 hash entries.
FSC_MaximumReadCache ExtentFiles	3	Defines the maximum number of read cache extent files.
FSC_MaximumReadCache ExtentFileSizeMegabytes	256	Defines the size (in megabytes) of read cache extent files.
FSC_ReadCache PurgePolicy	age	Ignored. Reserved for future use.

FSC write cache parameters

The following table lists the write cache parameters defined in the FMS server configuration file (*FMS_HOME/fsc.xml*). See the *FSC_HOME/fsc.xml.template* file for all available elements.

Name	Default value	Description
FSC_WriteCacheLocation	\$HOME\FSCCache/tmp/FSCCache	Defines the write cache location. The value is entered through Teamcenter Environment Manager during FSC installation. Modify the value either by running Teamcenter Environment Manager in maintenance mode or by manually editing the fmsmaster_fscid.xml file. All cache directories must be on a local hard disk. Cache directories cannot be on a file share directory or a mapped drive.
FSC_MaximumWriteCache FilePages	40960	Defines maximum number of file pages in the write cache.

Name	Default value	Description
FSC_MaximumWriteCache Segments	9216	Defines the maximum number of write cache segments.
FSC_WriteCacheHash BlockPages	2048	Defines the maximum number of hash block pages. Each hash block contains 128 hash entries.
FSC_MaximumWriteCache ExtentFiles	3	Defines the maximum number of write cache extent files.
FSC_MaximumWriteCache ExtentFileSizeMegabytes	256	Defines the size (in megabytes) of write cache extent files.
FSC_WriteCache PurgePolicy	age	Ignored. Reserved for future use.

FSC internal cache parameters for idle file handles

The following table lists the internal cache parameters defined in the FMS server configuration file (*FSC_HOME/fsc.xml*). See the *FSC_HOME/fsc.xml.template* file for all available elements.

Name	Default value	Description
FSC_MaximumIdleFile Handles	100	Defines the maximum number of idle file handles the server can cache.
FSC_MaximumIdleFile HandlesPerFile	5	Defines the maximum number of cached idle file handles allowed for a single file.
FSC_MaximumIdleFile HandleAgeMs	10000	Defines the amount of time (in milliseconds) an idle file handle remains open before being closed. Siemens Digital Industries Software recommends the value be set between 1000 to 10000 ms.

FSC internal cache parameters for ticket caches

The following table lists the internal cache parameters for ticket caches defined in the FMS server configuration file (*FSC_HOME/fsc.xml*). See the *FSC_HOME/fsc.xml.template* file for all available elements.

Name	Default value	Description
FSC_MaximumCached Tickets	500	Defines the maximum number of valid tickets cached. Use this parameter to reduce the need to revalidate a valid ticket on a repeated request.

FSC WAN acceleration tuning parameters

Use the following parameters in the specified configuration files to tune FSC WAN acceleration. Additionally, use **the FSC_MaximumWANDownloadSockets environment variable** to control the number of sockets used for WAN acceleration downloads.

Name	Default value	Description
FSC_WANThreshold	256K	Defines the minimum file size allowed for the FSC to use WAN acceleration for file downloads. The minimum value is 32K. Reducing the value may improve performance when processing relatively small files. Specify this value in the FSC configuration file or in the fscdefaults section of the FMS primary configuration file. For example: <pre><property name="FSC_WANThreshold" value="256K" overridable="true"/></pre>
FSC_WANChunkSize	256K	The maximum file chunk size (from 32K to 2M) that the FSC is allowed to request. Increasing this value reduces the number of parallel requests, which may result in performance improvements when bandwidth is limited. Specify this value in the FSC configuration file or in the fscdefaults section of the FMS primary configuration file. For example: <pre><property name="FSC_WANChunkSize" value="256K" overridable="true"/></pre>
FSC_DoNotCompressExtensions		File types to be excluded from being compressed during download when the FSC is using WAN acceleration. All files of the specified types are processed without using WAN acceleration. Increase performance by not attempting to compress files that are already compressed. Specify this value in the FSC configuration file or in the fscdefaults section of the FMS primary configuration file. For example: <pre><property name="FSC_DoNotCompressExtensions"</pre>

Name	Default value	Description
		<pre>value="xlsm,xlsx,z,zip" overridable="true"/></pre>
maxpipes	10	The number of sockets (from 2 to 10) the FSC opens for a file download when using WAN acceleration. Decreasing this value allows for more requests to be handled simultaneously. Increasing this value allows processes to be processed more quickly. Setting Maxpipes to 1 effectively disables WAN acceleration. Specify this value in the linkparameters section of the FMS primary configuration file. For example: <pre><linkparameters fromgroup="mygroup" togroup="cacheGroup" transport="wan" maxpipes="5" /></pre>

Specifying an invalid value resets the parameter value to the nearest valid value.

FMS client configuration file

Overview of the FMS client configuration file

The FMS client configuration file (*FMS_HOME/fcc.xml*) is installed for each client. This file identifies the file server used to download the FCC configuration. Typically, the downloaded configuration information contains FCC routing information and default parameter values. The **fcc.xml** file is installed at **FMS_HOME** (for example, *TC_ROOT\tccs*). See the *FMS_HOME/fcc.xml.template* file for all available elements.

A basic FCC configuration file contains the following elements:

- **fccconfig**

The outer containing XML element.

- **fccdefaults** (optional)

Defines or overrides FCC default parameters downloaded from the parent FSC, if these parameters may be overridden.

- **parentfsc**

Specifies which FSC to download the FMS configuration information from. You may specify multiple parent FSCs to provide failover.

General FCC configuration parameters

The following table lists general parameters defined in the FMS client configuration file (*FMS_HOME/fcc.xml*). See the *FMS_HOME/fcc.xml.template* file for all available elements.

Name	Default value	Description
FCC_LogFile	\$HOME/fcc.log tmp/\$USER/fcc.log	Defines the name of the FCC log file. The value to the left of is used for Windows hosts. The value to the right of is used for Linux hosts.
FCC_LogLevel	WARN	Defines the FCC logging level. Valid values, in order of increasing detail, are OFF , ERROR , WARN (the default), INFO , DEBUG , and TRACE . (SEVERE can be used as a synonym for ERROR .)
		Caution: Setting FCC_LogLevel to DEBUG or TRACE produces a large volume of logging output. Avoid using these logging levels in production environments.
FCC_TraceLevel		Defines the FCC trace filtering level, allowing you to filter the log output to specific diagnostic areas. Valid values are PERF, CACHE, CLIENT, FSC, and ADMIN. Enter a combination of any of these values, separated by , for example: CACHE CLIENT FSC ADMIN Note: PERF outputs requested information in only enough detail to replay a series of events for diagnostic purposes. Defining this value invalidates all other values for this parameter.
FCC_WANThreshold	256K	Defines the minimum file size allowed for the FCC to use WAN acceleration for FSC downloads. The minimum value is 32K. Reducing the value may improve performance when processing relatively small files.
FCC_MaxWANSources	10	Defines the maximum number of network sockets (from 2 to 10) that can be used for each file download. Decreasing this value allows more requests to be handled simultaneously. Increasing

Name	Default value	Description
		this value allows processes to be processed more quickly.
FCC_WANChunkSize	256K	Defines the maximum size of a file segment (or chunk) the FCC is allowed to request. The minimum size is 32K. The maximum size is 5M. Raising this value reduces the number of parallel requests, which may result in performance improvements when bandwidth is limited.
FCC_ProxyPipeName	\\.\pipe\FMSClientPipe /tmp/FMSClientPipe	Defines the base name of the set of FIFOs (pipes) used to communicate with the FCCClientProxy. The value must exactly match the value of the FCC_PROXYPIPENAME environment variable set in the client application environment. The value to the left of is used for Windows hosts. The value to the right of is used for Linux hosts. Note: Siemens Digital Industries Software recommends you do not set this parameter except under certain circumstances where it may be required. If you have questions, contact your Siemens Digital Industries Software representative.
FCC_FSCConnectionRetry Interval	5000	Defines how often (in milliseconds) the FCC attempts to reconnect to FSCs it has determined are offline or malfunctioning.
FCC_StatusFrequency	1000	Defines how often (in milliseconds) the FCC sends status callback messages to the client during a lengthy transport operation.
FCC_EnableDirectFSC Routing	true	Determines whether the FCC routes data requests on volumes mounted within the local FCC group to FSCs which mount these volumes. If set to false, the FCC routes all requests to an assigned FSC for forwarding to the appropriate volume server.
FCC_IdleTimeoutMinutes	Not applicable.	This parameter is obsolete. Use the maxidletime attribute in the tccs.xml file instead; setting the

Name	Default value	Description
		idle time out for TCCS and all its contained applications, including the FCC.
FCC_TransientFileFSCSource	ticketuri	Determines whether to use the uniform resource identifier (URI) contained in every transient file ticket. Note: If your network uses IPv6 (128-bit) addresses, use the hostname in URIs and do not use the literal addresses, so the domain name system (DNS) can determine which IP address should be used. Transient file ticket URIs are defined with the Default_Transient_Server preference, which specifies a default transient file server location. This parameter applies only to a four-tier transient file accessed using an FCC. It is ignored during two-tier transient file access and regular volume access. If set to ticketuri, the FCC routes four-tier transient file requests using the ticket URI list, failing over to the directfscroutes list and then the assignedfsc list. If set to parentfsc, the FCC ignores the ticket URI list, and routes four-tier transient file requests using only the information obtained from the parentfsc element (such as the directfscroutes list and the assignedfsc list).

FCC cache parameters

The following table lists the values in the *FMS_HOME/fcc.xml* file available for FCC common cache parameters. See the *FMS_HOME/fcc.xml.template* file for all available elements.

Name	Default value	Description
FCC_CacheLocation	\$HOME\FCCCache/tmp/ FCCCache	Defines the FCC root cache location. Each user's FCC cache is in a separate subdirectory below this location. The value to the left of is used for Windows hosts. The value to the right of is used for Linux hosts.

The following table lists the values available for FCC whole file cache parameters.

Name	Default value	Description
FCC_CacheTableHashSize	1000	Deprecated. Defined the initial size of the internal FCC GUID lookup table.
FCC_WholeFileCache Subdirectories	30	Defines the number of FCC whole file cache subdirectories.
FCC_MaxWriteCacheSize	1G	Defines the maximum size (in bytes) of the FCC whole file write cache. Use the suffixes K , M , G and T to represent kilobytes, megabytes, gigabytes, and terabytes, respectively.
FCC_MaxReadCacheSize	1G	Defines the maximum size (in bytes) of the FCC whole file read cache. Use the suffixes K , M , G and T to represent kilobytes, megabytes, gigabytes, and terabytes, respectively.
FCC_MaximumRead CacheAge	180	Defines the maximum idle age (in days) of a file in the FCC whole file read cache. Files that have not been accessed in longer than the time period defined are removed from the cache.
FCC_MinimumRead CacheAgeMinutes	240	Defines the minimum read cache age (in minutes). The FCC deletes files from its WholeFileReadCache only if they have <i>not</i> been accessed within the specified amount of time. The minimum value is one minute.
Note: Setting this parameter to a very high value makes the FCC whole file read cache purge ineffective.		
FCC_MaximumWrite CacheAge	180	Defines the maximum idle age (in days) of a file in the FCC whole file write cache. Files that have not been accessed in longer than the time period defined are removed from the cache.
FCC_MinimumWrite CacheAgeMinutes	10	Defines the minimum write cache age (in minutes). The FCC deletes files from its WholeFileWriteCache only if they have <i>not</i> been accessed within the specified amount of time. The minimum value is one minute.
Note: Setting this parameter to a very high value makes the FCC whole file write cache purge ineffective.		
FCC_ReadCachePurgeSize Percentage	25	Defines the minimum percentage of free space purged when the FCC whole file read cache becomes full.

Name	Default value	Description
		For example, if set to 25, when the cache reaches 100% of its maximum size, the FCC begins to purge the least recently accessed files until the cache size is reduced to 75% or less of its maximum size.
FCC_WriteCachePurgeSize Percentage	25	Defines the minimum percentage of free space purged when the FCC whole file write cache becomes full. For example, if set to 25, when the cache reaches 100% of its maximum size, the FCC begins to purge the least recently accessed files until the cache size is reduced to 75% or less of its maximum size.
FCC_MinimumBackground IdleTimeSeconds	5	Allows prioritizing of background processing of whole file cache population requests. The FCC begins background processing of cache population requests only after it is free of foreground client data requests for the specified amount of time (in seconds). Once cache population begins, received foreground data requests do not interrupt a file download already in progress, they can introduce delays before the background cache population process progresses to the next file. Disable this behavior by setting this parameter to 0. Note: Setting this parameter to a very high value makes the background cache file population functionality ineffective.
FCC_MaxBackground Retries	3	Defines the maximum number of times a background whole file cache population request is retried due to a recoverable error. (Unrecoverable errors are not retried.) If the file is not successfully downloaded after the maximum number of retries, the request is discarded. Disable this behavior by setting this parameter to 0. The request is immediately discarded. Note: Setting this parameter to a very high value can degrade FCC performance, requiring it to continually retry downloads that fail in a manner that is not obviously unrecoverable.

The following table lists the values in the **fcc.xml** file available for FCC segment cache parameters.

Name	Default value	Description
FCC_MaximumNumberOfFilePages	40960	Defines the maximum number of header files.
FCC_MaximumNumberOfSegments	512	Defines the number of data segments in the base segment file. This number does not include extents.
FCC_HashBlockPages	2048	Defines the maximum number of hash block pages. Each hash block contains 128 hash entries.
FCC_MaxExtentFiles	32	Defines the maximum number of extant files.
FCC_MaxExtentFileSize Megabytes	16	Defines the maximum size (in megabytes) of each extent file. This number is rounded to a multiple of 16 megabytes. For example, a value of 258 results in 256 megabytes of segment data per extent file. The value of 256M represents 256 <i>million</i> megabytes, which is out of range.

Sizing FMS fast cache

The File Management System (FMS) fast cache is used by every FMS client cache (FCC) and FMS server cache (FSC) to store segment data. This is by default the primary cache in the FSC. It is the portion of the FCC where excerpted partial file data is stored. This cache is frequently used in conjunction with a whole file cache, which stores complete files.

Use the FMS cache sizing tool to determine your cache size requirements. The tool can be found in a ZIP file in the `TC_ROOT\fsclbin` directory, and the latest version is always available from Customer Support. The tool consists of a Microsoft Word instruction document and a Microsoft Excel spreadsheet.

Following are the contents of the *FMSCacheSizingTool_Ver_version.zip* file:

FMSCacheSizingTool_Instructions_vversion.zip
FMSCacheSizingTool_Ver_version.xlsx
FMSCacheSizingTool_Ver_versionmacros.xlsx

The tool's output is applicable to all versions of FMS, subject to the documented limitations of each version. For example, using the sizing tool does not make the FMS fast cache support 64-bit configurations when used in conjunction with a 32-bit Java run time.

Administering FMS

Introduction to administering FMS

An initial installation of Teamcenter using Teamcenter Environment Manager (TEM) provides a single FSC that mounts to a single volume; a typical configuration for small deployment. During installation, TEM prompts for the appropriate parameters, creates the **.xml** configuration files, and installs and starts the FSC. Using this method, the FSC is installed under a local user account.

After FMS is installed, you can start/stop an FSC so that you can add volumes. You can also customize your FMS configuration after the initial installation.

Teamcenter File Management System (FMS) supports UTF-8 encoding. Client applications can use existing 8-bit encoding (native), UTF-8 encoding (8-bit Unicode), or Unicode (**wchar**) APIs. The UTF-8 and Unicode (**wchar**) FCC and FSC and UTF-8 APIs operate consistently with any client locale or native encoding. Once a client application migrates to the new Unicode APIs, the native encodings of the FCC and FSC no longer need to match that of the Unicode client application.

Administering FSCs

Introduction to administering FSCs

The **fscadmin** utility monitors and controls FSC servers. Use this utility to check the status of a server, perform a shutdown, modify logging levels, query performance counters, and to clear or inspect caches. You can also use this utility to route these administrative commands from one FSC to any other FSCs in the local network.

In some cases, you need to manage your FMS file server caches. For example, when you make a change to the **FSC_HOME\fmsmaster_fscid.xml** file on the primary host, you must restart all FSCs. To manually change your FMS configuration, you may need to install and/or uninstall components.

Manage your FSC on Windows

Following are tools to assist you in administering your FMS configuration running on Windows.

Before running any of these tools, ensure the **JAVA_HOME** and **FSC_HOME** environment variables are set to the correct paths. (**FSC_HOME** is typically set to the **fsc** directory in your Teamcenter installation, for example, **TC_ROOT\fsc**).

- Start the FSC.
- When FSC is installed, it is typically installed as a Windows service. To view the service, choose **Start→Control Panel→Administrative Tools→Services→Teamcenter FSC Service fscid**).

Replace *fscid* with the FSC ID defined in the **FSC_HOME\fsc.xml** file.

To start the service, right-click the service and choose **Start**.

- If FSC is not installed as a Windows service, install the FSC service using the **installfsc** batch script by completing the following steps:

1. From a Teamcenter command prompt, change to the *FSC_HOME* directory.

To open a Teamcenter command prompt, choose **Start**→**Programs**→**Teamcenter version**→**configuration-name Command Prompt**.

2. Run **installfsc %JAVA_HOME% %FSC_HOME% fscid**.

Replace *fscid* with the FSC ID defined in the *FSC_HOME\fsc.xml* file. If **JAVA_HOME** or **FSC_HOME** contains spaces, they must be quoted on the command line.

Example:

```
c:\apps\Siemens\Teamcenter10\fsc>installfsc "C:\Program Files\Java\jre7"
"D:\apps\Siemens\Teamcenter10\fsc" FSC_SVI6W181_user1
```

- To verify the FSC is running, at a Teamcenter command prompt, enter the following from the *FSC_HOME* directory:

fscadmin -s http://hostname:port fsc-ID/status

Example:

```
c:\apps\Siemens\Teamcenter10\fsc>fscadmin -s
http://SVI6W181:4544 FSC_SVI6W181_user1/status
```

- To check the status of the read cache, enter the following:

fscadmin -s http://hostname:port fsc-ID/cachesummary/read

Example:

```
c:\apps\Siemens\Teamcenter10\fsc>fscadmin -s
http://SVI6W181:4544 FSC_SVI6W181_user1/cachesummary/read
```

- To check the status of the write cache, enter the following:

fscadmin -s http://hostname:port fsc-ID/cachesummary/write

Example:

```
c:\apps\Siemens\Teamcenter10\fsc>fscadmin -s
http://SVI6W181:4544 FSC_SVI6W181_user1/cachesummary/write
```

- To check the **FSC performance**, enter the following:

fscadmin -s http://hostname:port fsc-ID/perfcounters

Example:

```
c:\apps\Siemens\Teamcenter10\fsc>fscadmin -s
http://SVI6W181:4544 FSC_SVI6W181_user1/perfcounters
```

- To stop the FSC, enter the following:

fscadmin -s http://hostname:port fsc-ID/stop

Example:

```
c:\apps\Siemens\Teamcenter10\fsc>fscadmin -s
http://SVI6W181:4544 FSC_SVI6W181_user1/stop
```

Manage your FSC on Linux

Following are tools to assist you in administering your FMS configuration running on Linux:

- To verify the FSC is running, enter the following:

%./fscadmin.sh -s http://hostname:port fsc-ID/status

Example:

```
%./fscadmin.sh -s http://EVALSUN8:4544 FSC_EVALSUN8_user1/status
```

- To check the status of the read cache, enter the following:

%./fscadmin.sh -s http://hostname:port fsc-ID/cachesummary/read

Example:

```
%./fscadmin.sh -s http://EVALSUN8:4544 FSC_EVALSUN8_user1/cachesummary/read
```

- To check the status of the write cache, enter the following:

%./fscadmin.sh -s http://hostname:port fsc-ID/cachesummary/write

Example:

```
%./fscadmin.sh -s http://EVALSUN8:4544 FSC_EVALSUN8_user1/cachesummary/write
```

- To stop the FSC, enter the following:

```
%./fscadmin.sh -s http://hostname:port fsc-ID/stop
```

Example:

```
%./fscadmin.sh -s http://EVALSUN8:4544 FSC_EVALSUN8_user1/stop
```

If the FSC fails to start

The FSC can fail to start after a reboot of Linux systems. This can occur when other services upon which the FSC depends have not yet started.

For example, the **ypbind** service can take additional time to start. Because the **ypbind** service was not operational when the FSC attempted to start, the **su** operation failed to switch to the user required to start the FSC.

In this case, correct the reason the **ypbind** service launches slowly. Because Siemens Digital Industries Software is not directly contributing to the slow launch of the **ypbind** service, the resolution must be investigated by your IT department on a case-by-case basis. One workaround is to add a pause in the startup script. A sleep value of 120 seconds is sufficient in test systems. The **sleep** command must be added to the startup script in the **rcname.d** directory. Do not modify the **rc** scripts under the **TC_ROOT** directory; these are not the scripts first called.

Note:

Renaming the **rc** script in the **rcname.d** directory to force the service to boot later in the boot order does not provide sufficient time to allow the **ypbind** service to start in test systems.

If a RedHat Linux system does not accept connections from Windows systems or other sources

Connections may not be accepted due to a known RedHat Linux TCP daemon issue caused by third-party software. Perform the following steps on each RedHat Linux system if you encounter this issue.

1. Log in as a system admin (root).
2. Edit the */etc/sysctl.conf* file, adding the following setting:

```
net.ipv4.tcp_timestamps=0
```

3. Reboot the Linux system and restart all servers.

Query the FSC performance counters

Use the **fscadmin** utility to query the FSC performance counters:

```
c:\apps\Siemens\Teamcenter10\fsc>fscadmin -s  
http://cyi6w387:4544 FSC_cyi6w387_user1/perfcounters
```

Performance counters provide insight into the working of the file system cache. There are a number of FSC performance counters available.

```

LocalReadPartialFile
  Min - 0
  Max - 0
  Avg - 0
  Successes - 0
  Failures - 0
  KBytes - 0
  AdHoc:
    2012/04/19-20:10:13,654 UTC - cyi6w387 -
    name=Throughput,units=Bytes/Sec,gauge=0 : 128 avg=0 {0/0 (0)} span=60000 rate
basis=1000
  name=Size,units=Bytes,count=0 : 0 avg=0 {0/0 (0)}
  name=GetRequestCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
  name=HeadRequestCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
  name=TransactionRate,units=Transactions/Sec,gauge=0 : 128 avg=0 {0/0 (0)} span=60000
rate basis=1000

LocalTransfer
  Min - 0
  Max - 0
  Avg - 0
  Successes - 0
  Failures - 0
  KBytes - 0
  AdHoc:
    2012/04/19-20:10:13,654 UTC - cyi6w387 -
    name=Throughput,units=Bytes/Sec,gauge=0 : 128 avg=0 {0/0 (0)} span=60000 rate
basis=1000
  name=Upload,units=Bytes,count=0 : 0 avg=0 {0/0 (0)}
  name=TransactionRate,units=Transactions/Sec,gauge=0 : 128 avg=0 {0/0 (0)} span=60000
rate basis=1000

LocalReadWholeFile
  Min - 18
  Max - 1345
  Avg - 297
  Successes - 8
  Failures - 0
  KBytes - 29164
  AdHoc:
    2012/04/19-20:10:13,655 UTC - cyi6w387 -
    name=Throughput,units=Bytes/Sec,gauge=0 : 128 avg=0 {0/0 (0)} span=60000 rate
basis=1000
  name=Size,units=Bytes,count=29864040 : 8 avg=3733005 {2335/17774751 (5840329)}
  name=GetRequestCount,units=Transactions,count=8 : 8 avg=1 {1/1 (0)}
  name=HeadRequestCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
  name=Simultaneous Open,units=Transactions,gauge=0 : 8 avg=0 {0/0 (0)}
  name=TransactionRate,units=Transactions/Sec,gauge=0 : 128 avg=0 {0/0 (0)} span=60000
rate basis=1000

LocalWrite
  Min - 0
  Max - 0
  Avg - 0
  Successes - 0
  Failures - 0

```

KBytes - 0

AdHoc:

2012/04/19-20:10:13,655 UTC - cyi6w387 -
 name=Append,units=Bytes,count=0 : 0 avg=0 {0/0 (0)}
 name=Upload,units=Bytes,count=0 : 0 avg=0 {0/0 (0)}
 name=RollbackCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}

Ticketing

Min - 0

Max - 1

Avg - 1

Successes - 16

Failures - 0

KBytes - 0

AdHoc:

2012/04/19-20:10:13,655 UTC - cyi6w387 -
 name=TicketTimeParseErrorCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=MissingCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=ExpiredTicketCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=InvalidTicketCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=OnURLCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=InHeaderCount,units=Transactions,count=16 : 16 avg=1 {1/1 (0)}
 name=TicketCacheMissCount,units=Transactions,count=16 : 16 avg=1 {1/1 (0)}
 name=TicketCacheHitCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=Simultaneous Open,units=Transactions,gauge=0 : 16 avg=1 {0/2 (0)}
 name=ValidTicketCount,units=Transactions,count=16 : 16 avg=1 {1/1 (0)}

LoadFSCCache

Min - 0

Max - 0

Avg - 0

Successes - 0

Failures - 0

KBytes - 0

AdHoc:

2012/04/19-20:10:13,655 UTC - cyi6w387 -
 name=InvalidTicketsReceived,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=WaitingDownloads,units=Transactions,gauge=0 : 0 avg=0 {0/0 (0)}
 name=SuccessDownloads,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=FailedDownloads,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=ValidTicketsReceived,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}

LocalAdminRead

Min - 0

Max - 126

Avg - 13

Successes - 10

Failures - 0

KBytes - 0

AdHoc:

2012/04/19-20:10:13,655 UTC - cyi6w387 -
 name=FCCDefaultsCount,units=Transactions,count=3 : 3 avg=1 {1/1 (0)}
 name=Throughput,units=Bytes/Sec,gauge=0 : 128 avg=0 {0/0 (0)} span=60000 rate

basis=1000

name=FMSMasterFileCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=PutRequestCount,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=GetRequestCount,units=Transactions,count=8 : 8 avg=1 {1/1 (0)}
 name=Simultaneous Open,units=Transactions,gauge=0 : 10 avg=0 {0/0 (0)}
 name=FMSMasterMD5Count,units=Transactions,count=0 : 0 avg=0 {0/0 (0)}
 name=TransactionRate,units=Transactions/Sec,gauge=0 : 128 avg=0 {0/0 (0)} span=60000

```

rate basis=1000

localReadTotal
  Rate - 0 bytes/sec

ReadSegmentCacheStats
  MapName - FSC_cyi6w387_kimc-FSCReadMap
  HashBlockPages - 32
  HashEntries - 4096
  HashTableSize - 16384
  MaxExtentFiles - 0
  MaxExtentSegments - 0
  MaxFilePages - 512
  MaxSegmentExtentMBs - 0
  MaxSegments - 623
  MaxTotalSegments - 623
  SegmentEntries - 18432
  SegmentExtentFileSize - 16
  SegmentExtentFileSizeBytes - 16777216
  SegmentExtentsPerFile - 1024
  SegmentFileSize - 10207232
  HeaderPercentFull - 0
  NumExtentFiles - 0
  NumExtentSegments - 0
  NumFiles - 0
  NumFreeFilePages - 512
  NumFreeSegmentEntries - 623
  NumHits - 0
  NumMisses - 0
  NumUsedFilePages - 0
  NumUsedSegmentEntries - 0
  SegmentPercentFull - 0

WriteSegmentCacheStats
  MapName - FSC_cyi6w387_kimc-FSCWriteMap
  HashBlockPages - 32
  HashEntries - 4096
  HashTableSize - 16384
  MaxExtentFiles - 0
  MaxExtentSegments - 0
  MaxFilePages - 512
  MaxSegmentExtentMBs - 0
  MaxSegments - 623
  MaxTotalSegments - 623
  SegmentEntries - 18432
  SegmentExtentFileSize - 16
  SegmentExtentFileSizeBytes - 16777216
  SegmentExtentsPerFile - 1024
  SegmentFileSize - 10207232
  HeaderPercentFull - 0
  NumExtentFiles - 0
  NumExtentSegments - 0
  NumFiles - 0
  NumFreeFilePages - 512
  NumFreeSegmentEntries - 623
  NumHits - 0
  NumMisses - 0
  NumUsedFilePages - 0
  NumUsedSegmentEntries - 0
  SegmentPercentFull - 0

```

Configure an FSC manually

1. Run the **backup_xmlinfo** utility, located in the *TC_BIN* directory. The output is the **backup.xml** file, stored in the directory from which you ran the **backup_xmlinfo** utility.
2. Review the **backup.xml** file to determine the *enterpriseID*, *volumeUid*, and the *wntPath* or *unixPath*.

Enter the values from the following parameters into the appropriate parameters within the **FSC_HOME\fmssmaster_fscid.xml** file.

Parameter in backup.xml file	Parameter in fmssmaster_fscid.xml file
enterpriseID	fmsenterprise id
volumeUid	volume id
wntPath	volume id root
unixPath	transient id root

3. Edit the **FSC_hostname_user** and **fcc.xml** files, if necessary. For example, if you are changing cache locations.
4. Stop and restart the FSC process.

Maintaining the FSC whole file cache

You can configure either traditional FSC storage or an NFS style storage for FSC whole files known as *whole file cache* (WFC). (FSC/JT segments are still stored in FSC segment cache.) Whole file cache stores a copy of the file on the disk.

To maintain the whole file cache:

- To configure whole file cache, set the **whole file cache parameters** in the **FSC_HOME\fschostname_user.xml** file and the **fcc.xml** files.
- To monitor whole file cache, use the **fscadmin** utility with the **whole** options (for example, **cachesummary/whole**, **cachedetail/whole**, **clearcache/whole**, and **purgecache/whole**).
- To purge files from the whole file cache, run the **FSCWholeFileCacheUtil** utility stored in the **FSC_HOME\bin** directory. Purge the cache based on file age, subdirectory size, and/or available disk space. This utility also purges misplaced files.

The benefits of whole file cache for IT data storage deployments are:

- You can use NFS or other network storage for the FSC whole file cache. The files are whole files on a disk and not in a virtual memory mapped disk space. You no longer need a local disk or a special card to make the network disk appear local to the operating system.

- You can use platforms such as HP for large FSC whole file caches. Platforms that previously had issues with the segment cache or virtual memory mapping can be used with large FSC whole file caches by using the NFS style whole file disk storage (although JT segments that are not whole file are still stored in the segment cache). Whole JT files can be prepopulated to avoid segment caching.
- You can use redundant disk storage (or RAID) rather than dual FSC caches. You can still run two FSCs for high availability off of the network storage.

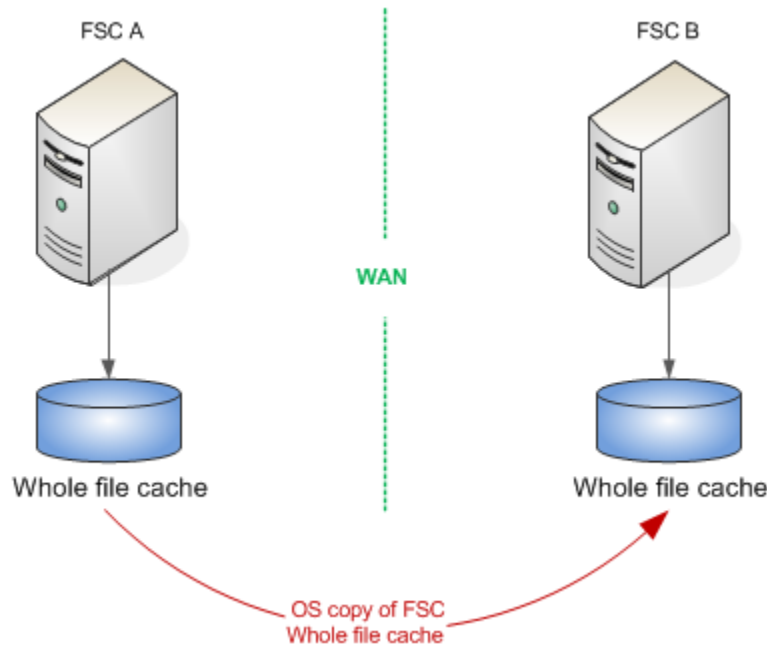
Though multiple file system caches can share the same whole file cache, no more than one FSC should be responsible for purging the shared cache. When managing a shared whole file cache, purge the shared cache using either of the following methods:

- Disable purging on all but one of the FSCs. (This is the typical deployment.)
- Disable purging on all FSCs and configure a purge job to run as an external cron job (Linux) or Scheduled Task (Windows).
- FSC whole file cache reliability is increased to OS level file reliability. Loss of a directory entry now results in single file loss, improving the durability and reliability of the whole file cache.
- You can do a disk copy to transfer an FSC cache, or copy the FSC whole file cache to a DVD and mail it. A background garbage collection process runs to prune the whole file cache. The cycle on this may be several hours for very large FSC whole file caches.

Copying the FSC whole file cache

You can copy all or part of the whole file cache using operating system commands, and then transfer the data electronically or physically to another site. This is useful when setting up a deployment site. Copying the whole file cache from a test environment to a deployment environment reduces the time it takes to populate the cache and improves performance.

This procedure works best when both FSCs belong to the same enterprise ID, providing superior performance over a WAN.



If the FSCs belong to different enterprises, follow these best practices:

- Set up a Multi-Site configuration between the two enterprises.
- Replicate the **ImanFile** objects belonging to the owning site as stubs at the remote site.

Note:

Massive WFC (Whole File Cache) sometimes can be spread across different physical disk partitions. The WFC copy works if both the source and destination have the same WFC configuration (the same number of disk partitions and the same number of hash directories).

Selective file caching

Selective file caching prevents designated datasets from being cached on specific, intermediate FSC servers. Doing so may be important for site security, political geography, or other reasons definable in terms that Teamcenter business logic can apply. Servers so designated are referred to as restricted servers.

By default, FMS caches most files on each intermediate FSC server. (There are exceptions for transient file data and administrative command results.) Selective file caching lets you designate files that will not be cached on these servers. Be aware of the following items when **enabling selective file caching**:

- FSC servers participating in selective file caching must run a version of FMS that supports selective file caching.
- When enabling selective file caching in Multi-Site Collaboration configurations, do not attempt to enable selective caching across Teamcenter sites. Each Teamcenter site should be configured for

selective file caching within its own File Management System. All participating sites configured for selective file caching should have a similar set of caching rules. That is, conditions and FSC lists should be the same.

- Enabling selective file caching does not impact how files are cached in the user's FMS Client Cache (FCC). Client caches cannot be associated with any restricted FSC ID or wildcard.

Enable selective file caching

Use the following steps to configure your site for selective file caching.

1. Make the following updates to the *fmsmaster.xml* and *fsc.xml* files:
 - Ensure FSC IDs contain only letters, numeric digits, underscores ("_"), and hyphens ("-"). Specifically, remove or change any of the following characters in any of the FSC IDs:
 - Commas (",")
 - Colons (":")
 - Semicolons (";")
 - Apostrophes ("'")
 - Question marks ("?")
 - Asterisks ("*")
 - Any characters that cannot be used as part of a file name, such as horizontal tabs and quotation marks.
 - Any characters or character sequences that may interfere with the parsing of regular expressions. For example, "[0-9]".
 - For any combination of FSC ID lists that will be deployed as part of the site's selective file caching implementation, shorten any FSC IDs that would collectively exceed 512 bytes of escaped, comma-separated, URL-encoded UTF-8 text. If shortening is problematic, consider using the asterisk ("*") character as a wildcard to represent all FSCs.
2. Using the Business Modeler IDE, set up the custom business rules by which selective file caching will work at your site.

Conditions return **true** if the dataset is to be cached. Conditions return **false** if the dataset is not to be cached.

- **Fnd0DatasetShouldFSCCacheThisDataset** condition

This condition defines the dataset instances that are not cached on restricted FSC servers, using the **Dataset_Selective_File_Caching_Rule_Condition_And_Blacklist** preference. By default, the condition is set to **isTrue()**, specifying that all datasets are to be cached. Override this condition with a custom template defining the business rules for your site's selective file caching.

The definition of this condition depends entirely on your site's security requirements. As an example, the following expression returns **true** for all datasets with an IP classification set to anything other than "**top-secret**".

```
o != NULL AND o.ip_classification != "top-secret"
```

Note:

Always add a check for `o != NULL` before accessing any field on the "o" object. Doing so ensures that files not associated with any dataset object are restricted by the FSC deny list configured in the rule preferences. Without this check, evaluating the condition on a file not associated with any dataset object could cause unexpected results.

- Additional **{DatasetType}ShouldFSCCacheThisDataset** conditions

The **{DatasetType}_Selective_File_Caching_Rule_Condition_And_Blacklist** preferences define dataset type-specific instances that are not cached on restricted FSC servers. Each applies to datasets of the type specified by **{DatasetType}**, for example, **TextShouldFSCCacheThisDataset** or **MSWordShouldFSCCacheThisDataset**. The dataset type-specific preferences override the default **Dataset_Selective_File_Caching_Rule_Condition_And_Blacklist** preference, but only for datasets of the type that appears in the preference name. These conditions do not exist by default, so must be created.

The definitions of these conditions depend entirely on your site's security requirements. The key requirements for a signature are that it should take a dataset object as input and the expression should include the check for `o != NULL` before accessing any field on the "o" object as shown in the earlier example for **Fnd0DatasetShouldFSCCacheThisDataset**.

3. Log on to Teamcenter as an administrator with dba privileges, using the rich client to set the following preferences:

- **FMS_Enable_Selective_File_Caching**

Determines if selective file caching is enabled. Set **FMS_Enable_Selective_File_Caching** to **true** to enable selective file caching. (This preference is visible and accessible only to Teamcenter administrators.)

- **Dataset_Selective_File_Caching_Rule_Condition_And_Blacklist**

Specifies the condition that handles files for any dataset type, unless overridden by other type-specific **{DatasetType}_Selective_File_Caching_Rule_Condition_And_Blacklist** preferences. By default, this preference is set to **Fnd0DatasetShouldFSCCacheThisDataset:***. If the COTS

condition name in this preference is changed, that condition must be defined in the Business Modeler IDE.

- **{DatasetType}_Selective_File_Caching_Rule_Condition_And_Blacklist**

Specify the conditions to be evaluated to determine the dataset type-specific instances that *are not cached* on the specified FSC servers.

Note:

These preferences should be created with system protection scope and their names hidden to be accessible only by Teamcenter administrators.

For each dataset type, set the following values:

- **Dataset Type**

Set to the type of data, for example, **Text** or **MSWord**.

- **Restricted FSCs**

In a comma-separated list, list the FSC ID of each server on which the dataset type *should not be cached*. The "*" wildcard can represent all FSCs. Limit the list of restricted sites to ten sites or fewer for each **{DatasetType}_Selective_File_Caching_Rule_Condition_And_Blacklist** preference.

- **Dataset Condition**

Set to the dataset type-specific condition for this dataset type, for example, **TextShouldFSCCacheThisDataset** or **MSWordShouldFSCCacheThisDataset**.

Example:

Preference: **Text_Selective_File_Caching_Rule_Condition_And_Blacklist**

Preference value:

TextShouldFSCCacheThisDataset:FSC_NYC1,FSC_NYC2,FSC_LA1,FSC_Lon1

Or

Preference: **Word_Selective_File_Caching_Rule_Condition_And_Blacklist**

Preference value: **WordShouldFSCCacheThisDataset:***

Note:

Recall that the FSC lists in the preferences are deny lists. Sensitive datasets of the type specified by the preference name are not cached on the specified FSCs.

4. Restart the pool server to deploy the changes to all Teamcenter server instances.

Administering FCCs

Introduction to administering FCCs

The FMS client cache (FCC) is a private user-level cache. It provides a high-performance cache for both downloaded and uploaded files. The FCC provides proxy interfaces to client programs and connectivity to the server caches and volumes.

Files captured by the FCC do not change for download or upload, for neither whole files nor partial files. All file copies and file segment copies are identical throughout the system and are never updated. New file versions are checked into the system with a new GUID, but a file with an existing GUID in the FMS system never changes. Therefore, there are no issues with file change or cache consistency.

FCCs provide access to the transient volume for the business server in a Teamcenter two-tier configuration. The business server writes or reads temporary files directly to a disk directory. The rich clients access those files using the standard FCC interfaces. This approach provides client independence from the system configuration and ensures that client programs operate the same in both two-tier and four-tier mode for file access functions.

TCCS considerations

As of Teamcenter 9.0, FCC runs within the Teamcenter client communication system (TCCS) container, which also contains the **TcServerProxy** and **TcModelEventManager** applications. Starting, stopping, and restarting any of these applications cause the entire TCCS service (including all applications within the container) to start, stop, or restart.

Stopping an FCC

It is important that you shut down TCCS/FCC in a prescribed manner. Not following one of several recommended methods can result in FCC and TCCS errors.

You are strongly advised not to use an operating system **kill** command to shut down a TCCS/FCC instance unless safer methods have failed.

Restarting an FCC

You may need to restart a TCCS/FCC instance when:

- You have changed FCC cache parameters that require a restart.

For information about which cache parameters require an FCC restart, see *Elements requiring a restart of an FCC*.

- A TCCS/FCC process executing in memory is not responding to pipe connection attempts. For example, when all of the following events occur:
 - The **FMS_HOME\bin\fccstat -status** command reports that the FCC is offline.
 - The **FMS_HOME\bin\fccstat -start** command cannot start the FCC.
 - The **FMS_HOME\startfcc.bat** reports that the FCC cache is locked.
- The TCCS container needs restarting for non-FCC reasons.

Reconfiguring an FCC

There are three ways to reconfigure an FCC. Each method involves a different level of user interaction and a different set of restrictions.

WAN acceleration

WAN acceleration improves file download performance across WAN assets by splitting larger files into smaller pieces, downloading the pieces simultaneously, and reconstructing the files in the cache before passing the file to the client. See [Enabling and tuning WAN acceleration](#) for instructions on implementing WAN acceleration.

Shutting down a TCCS/FCC instance

Whenever you stop a TCCS/FCC instance, it is important that the shutdown process is as clean as possible. You may want to stop a TCCS/FCC instance because:

- It is no longer needed.
- You need to stop and restart the instance to receive new configuration information.

Regardless of why you stop an FCC, remember that the FCC runs within the TCCS container process. Stopping an FCC also stops the **TcServerProxy** and **TcMEM** processes.

Before stopping a TCCS/FCC instance, close all client applications and wait 10 seconds.

Note:

Siemens Digital Industries Software does not recommend using the operating system **kill** command or the **Windows Task Manager** to stop an FCC unless safer methods have failed. Doing so can cause issues with the FMS fast cache, the FCC cache lock, the TCCS process lock, and any active FCC/**TcServerProxy**/**TcMEM** transactions.

Warning:

On Windows, JRE shutdown hooks are not honored, preventing the FCC from closing cleanly. If the TCCS/FCC instance remains running when users log off (or shut down) these operating systems, the FCC segment cache may be corrupted.

Siemens Digital Industries Software recommends you add the **fccstat -kill** command to all user logoff scripts and to any relevant Windows shutdown scripts for Teamcenter clients running on these operating systems.

The following methods for stopping an FCC are listed in the order by which they reduce the risk of corrupting the cache or creating a stuck lock file. If after stopping an FCC, it does not properly restart, **reset the user's FCC environment**.

- Method 1
 - Run **FMS_HOME\bin\fccstat -stop**.

This method stops a TCCS/FCC instance if it is idle.

The system confirms that the user has shut down all the attached client processes before stopping the TCCS.

If non-idle clients are connected to the FCC or other TCCS applications, a message appears and the FCC is not stopped. If you receive this message, confirm that all client applications are disconnected from FCC/TCCS, wait 10 seconds, and retry.

Note:

An FCC client is non-idle if it holds an open FCC file handle, an open segment cache handle, or an open pipe connection that has made a request within the past 5 to 10 seconds.

This method is effective 90 percent of the time and results in the cleanest possible shutdown. The FCC can be restarted afterward with no data loss.

- Method 2
 - Run **FMS_HOME\bin\fccstat -kill**.

This method stops a TCCS/FCC if it is idle or non-idle, as long as it is responsive to pipe commands.

The system does *not* confirm that the user has shut down all the attached client processes before stopping the FCC. The system does *not* display a message that other client applications are connected.

Any transactions in progress at the time the FCC is terminated may fail. This can have a negative effect on the connected client applications.

This method is effective 99 percent of the time. The FCC can typically be restarted afterward with no data loss.

- Method 3

If a TCCS/FCC instance is not responsive to FCC pipe commands, it may report FCC **Offline** though the TCCS/FCC process is running. In this situation, use the **tspsstat** or **tcmemstat** utility to stop the shared TCCS process.

- Method 4 (only if TCCS/FCC running in foreground)

- Press Ctrl+C in the FCC foreground window.

The foreground window is available from the interface *only* if TCCS/FCC is running in a visible command prompt window or shell. (This is not usually the case.)

- Alternatively, close the TCCS command prompt window (Windows only).

Note:

If the FCC is running in a hidden window, and you have system tools access, you can locate the hidden FCC window and send it a **WM_CLOSE** message (Windows only).

This method is highly effective and usually results in a clean shutdown of the FCC.

- Method 5 (Linux only)

- Run the operating system **kill** command without the **-9** option.

This method stops an FCC if it is idle or non-idle, even when it is not responsive to pipe commands.

This method stops the FCC as cleanly as possible. The effect is similar to method 2, but it is possible that file handles become stuck or that cache states are lost.

- Method 6 (Linux only)

- Run the operating system **kill** command with the **-9** option.

This method performs a hard stop on the FCC. Usually, the contents of the FCC fast cache (segment cache) are lost. Occasionally, the FCC lock file becomes stuck.

- Method 7 (Windows only)

- From the **Task Manager**, select the user's FCC process and click **End Process**.

This method performs a hard stop on the FCC. Usually, the contents of the FCC fast cache (segment cache) are lost. Occasionally, the FCC lock file becomes stuck.

Restarting an FCC

When do I need to restart the FCC?

Restart the FCC when:

- You changed FCC elements that require a restart.

For information about which FCC elements require a restart, see Elements requiring a restart of an FCC.

- A TCCS/FCC process executing in memory is not responding to pipe connection attempts. For example, when all of the following events occur:
 - The **\$FMS_HOME/bin/fccstat -status** command reports that the FCC is offline.
 - The **\$FMS_HOME/bin/fccstat -start** command cannot start the FCC.
 - The **\$FMS_HOME/startfcc.bat** reports that the FCC cache is locked.
- The TCCS container needs **restarting for non-FCC** reasons.

Restart an FCC manually

Any change to the FCC properties file requires a manual restart of the FCC.

Some changes to FCC elements cannot be automatically propagated to the FCC when you reconfigure the FMS primary configuration file (**fmsmaster_fscid.xml**) or local FCC file (**fcc.xml**). When you reconfigure these elements, you must manually stop and restart the FCC. All FCC applications and the FCC must be shut down and then restarted.

1. Close all client applications.
2. Wait 10 seconds.
3. Run **FMS_HOME\bin\fccstat -restart**.

If this method does not work, stop and restart the FCC:

1. **Stop the FCC.**
2. Run **FMS_HOME\bin\fccstat -start**.

Note:

Remember that the FCC runs within the TCCS container process. Stopping an FCC also stops the **TcServerProxy** and **TcMEM** processes.

Elements requiring a restart of an FCC

If any of the following parameters are changed, the FCC must be restarted to read the new settings:

- General elements

The following general FCC elements require a manual restart of the FCC:

FCC_ProxyPipeName
FCC_CacheLocation
FCC_StatusFrequency

- Logging elements

The following FCC logging elements require a manual restart of the FCC:

FCC_LogFile
FCC_LogLevel
FCC_TraceLevel

- Whole file cache elements

The following FCC whole file cache elements require a manual restart of the FCC:

FCC_WholeFileCacheSubdirectories
FCC_CacheTableHashSize
FCC_MaxWriteCacheSize
FCC_MaxReadCacheSize
FCC_MaximumReadCacheAge
FCC_MaximumWriteCacheAge
FCC_ReadCachePurgeSizePercentage
FCC_WriteCachePurgeSizePercentage

- Segment cache elements

The following FCC segment cache elements require a manual restart of the FCC:

FCC_MaximumNumberOfFilePages
FCC_MaximumNumberOfSegments
FCC_HashBlockPages
FCC_MaxExtentFiles

FCC_MaxExtentFileSizeMegabytes

Reset a user's environment

1. Stop all client applications that use FMS.
2. **Stop the FCC process.**
3. Reset the user's system environment using one of the following methods:
 - On a single-user machine, restart (or shut down and reboot) the operating system.
 - On a multiuser machine, log off and log back on.

If you cannot restart the FCC because of locked file handles, warn other users that the machine must be rebooted, then reboot the operating system.
4. Remove the following files from the user's FCC cache directory (for example, **Users\user-name\FCCCache\username**). Corruption in any of these files can prevent the FCC from restarting:
 - **fcc.lck**
 - **fms.hsh**
 - **fms.mf**
 - **fms.seg**
 - All files beginning with **fms.ext** (cache extent files)
5. Remove the TCCS lock file, stored in the **Users\.user-name_lock_host-name** directory.

The file name is **TCCS_CONFIG.lck**. By default, the **TCCS_CONFIG** environment variable is set to **Teamcenter**.

For example:

```
C:\Users\smith\.smith_lock_host123\Teamcenter.lck
```

Reconfiguring an FCC

There are three ways to reconfigure an FMS client cache (FCC). Each method involves a different level of user interaction and a different set of restrictions.

- Automatic reconfiguration of the FCC

Automatic reconfiguration of the FCC occurs when changes are made to the **fmsmaster_fscid.xml** file and the primary FMS file server cache (FSC) configuration server is reconfigured. Not all FCC changes are reconfigured automatically.

Many changes to FMS client cache (FCC) elements are automatically propagated to the FCC when you reconfigure the FMS primary configuration file (**fmsmaster_fscid.xml**).

For example, the FCC detects when changes are made to the following **FCCDefault** parameters in the primary configuration file and reconfigures itself accordingly, without interrupting FCC client application service:

- **FCC_TransientFileFSCSource**
- **FCC_EnableDirectFSCRouting**
- **FCC_WanThreshold**
- **FCC_MaxWANSources**
- **FCC_FSCConnectionRetryInterval**

The FCC also detects when changes are made to the following FCC configuration elements in the primary configuration file and reconfigures itself accordingly, without interrupting FCC client application service:

- **FCCDefaultssite**
- **parentfsc**
- **assignment**
- **assignedfsc** elements within the **clientmap** element
- **directfscroute** elements generated by the **parentfsc** file from resource elements in the primary configuration file
- **directtransientvolume** elements generated by the **parentfsc** file from resource elements in the primary configuration file
- Manual reconfiguration of the FCC

Manual reconfiguration of the FCC is required when changes are made to the local **fcc.xml** file when the FCC is processing commands for one or more applications.

The FMS client cache (FCC) responds to reconfiguration requests made through the **fccstat** utility, attempting to reconfigure without loss of service to the client workstation or to applications connected to the FCC.

However, there are situations in which the FCC cannot be reconfigured automatically. Changes made in the local **fcc.xml** file while the FCC is processing commands for one or more applications require a manual FCC reconfiguration.

- Run the **fccstat** utility with the **-reconfig** argument.

Changes are applied from your local **fcc.xml** files and from any **parentfsc** configurations.

- **Restart the FCC**

Restarting the FCC is required when an immutable parameter (such as cache size) is changed in the **fmsmaster_fscid.xml** file or the **fcc.xml** file. Clients receive changes to immutable parameters when their FCCs are restarted.

Using the FCC assignment mode element to override default client mapping behavior

A *client mapping* is a list of **assignedfsc** elements appropriate for a particular client's IP address and/or domain name. The **assignedfsc** element is found in the **fmsmaster_fscid.xml** file.

Static client mapping is not always appropriate, because it is possible for a client to change network contexts without changing its IP address or domain name.

Example:

A user takes a company laptop from the office to a hotel room. The FMS server cache (FSC) is inaccessible because the client has moved outside the company firewall. To access company data from the public side of the firewall, the user must use a different FSC server (or at least a different access address) than the server used in the office.

Example:

A user takes a company laptop from the office to a remote location. At the remote location, the client continues accessing data over a WAN, using the client mapped **assignedfsc** elements, even though the data is now more readily accessible from an FSC in a local satellite office. This is inefficient, as the WAN connection is slower than accessing the data from the FSC servers in the same office.

It is possible, but inconvenient, to reconfigure the **clientmap** elements in the **fmsmaster_fscid.xml** file for each client machine (and propagate dynamic changes to several **clientmap** settings throughout the entire FMS system) each time a Teamcenter client is relocated on the network.

It is more efficient to change the default **assignment mode** setting (in the **fcc.xml** file) from **clientmap** to **parentfsc**.

- **clientmap**

With this setting, FCC data requests are routed to the **assignedfsc** elements specified within the **clientmap** section of the **fmsmaster_fscid.xml** configuration file. The FCC downloads the list of **assignedfsc** elements from the **parentfsc** element when it starts.

An **assignedfsc** element represents an FMS server that distributes Teamcenter data to clients that have no direct access to the FSCs serving the origin volume.

Each **clientmap** section typically contains one to three **assignedfsc** elements. The **assignedfsc** elements represent FMS cache or data servers.

- **parentfsc**

Use this setting when the default client mapping setting is not efficient.

With this setting, the **parentfsc** list is used as the **assignedfsc** list. FCC data requests are routed to the list of **parentfsc** elements specified in the **fcc.xml** configuration file.

A **parentfsc** element represents an FSC server that distributes FMS configuration settings to clients upon request. Each **fcc.xml** file typically contains one to three **parentfsc** elements.

The following **fcc.xml** sample code illustrates an appropriate **assignment mode** setting for a client in a hotel room.

```
<parentfsc address="https://plm.vpnaccess.newyork.mycompany.net:4318"
transport="wan"/>
  <assignment mode="parentfsc"/>
```

The following **fcc.xml** sample code illustrates an appropriate **assignment mode** setting for a client in a remote satellite office.

```
<parentfsc address="http://plm1.product.india.mycompany.net:4545"
priority="0"/>
<parentfsc address="http://plm2.product.india.mycompany.net:4545"
priority="0"/>
<parentfsc address="http://backup.plm.product.india.mycompany.net:4545"
priority="1"/>
  <assignment mode="parentfsc"/>
```

Note:

For legacy client mapping, use the **iptoparent** setting in the **fcc.properties** file:

```
com.teamcenter.fms.clientcache.iptoparent=false
```

The default setting is **true**, which causes the FCC to send its IP to the **parentfsc** list. The FSC maps the FCC client by the IP as determined at the client.

Setting **iptoparent** to **false** inhibits the client IP on the **parentfsc** client mapping request. The FSC maps the FCC client by the IP address on which the request arrives at the **parentfsc**. This is often an IP router, firewall, or proxy server at the parent FSC's location.

Why am I getting errors after modifying the FCC configuration?

The following list discusses typical FCC modifications which result in errors:

- Modified a routing or connection element in the **fcc.xml** file, and then did not run the **\$FMS_HOME/bin/fccstat -reconfig** command.

Changes made in the local **fcc.xml** file while the FCC is processing commands for one or more applications require a manual FCC reconfiguration.

- Modified an FCC element that required an FCC restart. The effects of the change are not applied until the FCC is restarted.

For a list of the FCC elements that require a manual restart, see Elements requiring a restart of an FCC.

- Improperly shutting down an FCC may generate cache or lock errors upon restart.

Clear these errors by restarting the user's FCC environment.

Other FCC modifications incompletely applied may generate other errors. Most error messages provide suggestions for how to correct the problem. Check the FCC log and Teamcenter system logs for error messages.

Why am I getting errors after removing the FCC?

If you remove the cache files, the cache is empty when FCC restarts. This can generate errors in two areas:

- You cannot create new cache files on the local hard disk due to permission settings, lack of disk space, or configuration mistakes.

Fixing the underlying issue typically resolves the FCC error.

- You cannot repopulate the cache because the FMS system from which the data originated is not accessible on the network.

If the data is no longer available from the PLM volumes, you must find a new data source. Fixing the underlying network access problem typically resolves the FCC error.

Why am I getting errors after using the kill command to stop FCC processes?

Siemens Digital Industries Software does not recommend using the operating system **kill** command or the **Windows Task Manager** to stop an FCC unless safer methods have failed. Doing so can cause issues with the FMS fast cache, the FCC cache lock, and active FCC transactions.

For recommended methods of stopping an FCC, see [Shutting down a TCCS/FCC instance](#).

If you use one of these methods to stop an FCC and receive errors, you must [reset the user's FCC environment](#).

Set file buffer size to optimize file transfer rate

If you find that the transfer of large files is sluggish, you can optimize the file transfer rate by adding the following buffer parameters to the *fcc.properties* file:

- **buffSize**

Sets the internal buffer size which is used to read data from the file system and the segment caches. For peak I/O efficiency, this value should always be a multiple of 16KB and also always a multiple of the file system sector size.

Following is the commented text for this parameter in the *fcc.properties.template* file:

```
# FCC internal buffer size.
# Default value is 64K.
# Value should be in 16K increments.
# Minimum is 16K.
# (The setting name you see here is correct; this information is
internally
#   associated with the FMS server cache (FSC) connections.)
#
#com.teamcenter.fms.servercache.FSCConstants.buffSize=64K
```

- **sockBuffSize**

Controls the Transmission Control Protocol (TCP) window size in the Java TCP stack, which is used for reading and writing data from network connections. This size must exceed the BDP (bandwidth delay product) of the network connection in order to prevent acknowledgment (ACK) response throttling caused by the TCP window filling with unacknowledged data. To calculate the BDP of a network connection, multiply the round-trip latency (seconds) by the network bandwidth (bytes per second). A factor of about 2:1 is needed for reliable high-performance operation.

Following is the commented text for this parameter in the *fcc.properties.template* file:

```
# Socket buffer size override.
# Default value is (com.teamcenter.fms.servercache.FSCConstants.buffSize * 2) +
1024.
```

```
# The value 0 disables setting the socket buffer sizes (uses system default).
# Minimum value is 8K (excluding the 0 case).
# (The setting name you see here is correct; this information is internally
#   associated with the FMS server cache (FSC) connections.)
#
#com.teamcenter.fms.servercache.FSCConstants.sockBuffSize=129K
```

Tip:

To add these parameters to the *fcc.properties* file, open the *fcc.properties* file in the **FMS_HOME** directory. Copy the parameter to the file and remove the comment character (#) from the line containing the parameter. If the *fcc.properties* file does not exist, copy the *fcc.properties.template* file to the **FMS_HOME** directory and rename it as *fcc.properties*.

The **buffSize** and **sockBuffSize** parameters can also appear in the *fsc.properties* file, the *fsc.clientagent.properties* file, or the *fscadmin.properties.template* file.

When determining the socket buffer size, use the largest BDP of all of the other FMS modules (FSCs and FCCs) to which this FMS module connects. For example, an FSC server that connects to clients over a BDP=600MB LAN and other FSCs over a BDP=100KB WAN should use the 600MB LAN value to determine the socket buffer size.

The socket buffer size is typically set to twice the internal buffer size plus a little more. This is reflected in the default value. The internal buffer size should be a little less than half of the socket buffer size, but nonetheless, a convenient multiple of the local disk sector size. This value should definitely never be more than the socket buffer size. The minimum socket buffer size should be determined by the local hard disk rather than the network connection (which functions correctly when the TCP window size is too big).

Consider this example:

Round-trip latency = 750ms

Bandwidth = 400 Mbits/sec

$BDP = 750\text{ms} * (400 \text{ Mbits/sec}) * (1 \text{ sec}/1000 \text{ ms}) * (1 \text{ byte}/10 \text{ bits}) = 300000 / 10000 = 30 \text{ MBytes}$

If the BDP is 30 MBytes, make the socket buffer size a little larger, and the socket buffer size roughly half of that. But 15 MB (half of the BDP) is not a multiple of the hard disk sector size (4MB in this hypothetical case), so make the internal buffer size 16 MB. This makes the socket buffer size about (twice 16 is 32 MB, plus a little more), or say, 33 MB.

Note:

For the byte/bits setting, allow 10 bits per byte to account for 8 data bits plus start/stop bits on network media.

Auditing FSCs

Introduction to auditing FSCs

Teamcenter provides flexible and detailed **auditing of FSC** access and operations. The primary purpose of this auditing functionality is to track system access for security purposes. It also allows monitoring of the servers for operational purposes and can be used to debug or verify complex FMS system interactions.

After FSC audit logging is configured, the audit logs are written to `%USERPROFILE%\Siemens\logs\FSC\process\fsid_audit.log` files.

As server requests are processed, each request is identified by a transaction ID. The audit log output is generated as the requests are processed by the server. A single request/transaction can propagate across various FSC servers. However, they are easily identified and correlated in all of the participating FSC audit log files.

You can import the audit log information into standard text or word processors for correlation and examination.

Teamcenter provides configurable audit points for different types of processing:

- **Request**

Identified by **request** in the audit log. It is at the top of the request processing chain before any routing within the server. It can render HTTP request information, the transaction ID, and ticket information if it is provided with the request. Request information can include HTTP request headers, remote address, and so forth.

- **Primary operation start**

Identified by **priopstart** text in the audit log. This is the primary operation starting audit point. It signals a ticketed operation start event. It can render the same information as the request audit point. It includes a short description of the operation indicating how the request is being processed.

- **Primary operation stop**

Identified by **priopstop** text in the audit log. This is encountered once a primary request is finished processing. All request and operation start renderers are available as well as operation stop and response renderers.

- **Subordinate operation start**

Identified by **subopstart** text in the audit log. This is a subordinate operation starting audit point. It can render the same information as the request audit point. It includes a short description of the operation indicating how the request is being processed. Tickets in subordinate operations may differ from the tickets used in primary operation tickets.

- **Subordinate operation stop**

Identified by **subopstop** in the audit log. This is processed once a subordinate operation is finished processing. All request and operation start renderers are available as well as operation stop and response renderers.

- **Web operation start**

Identified by **webopstart** text in the audit log. This is a simple *web server-like* operation start audit point, such as a configuration download, **favicon** request, or other non-ticketed requests. It can render request information, such as a header, remote address, and the transaction ID. The operation indicates how the request is processed.

- **Web operation stop**

Identified by **webopstop** text in the audit log. This is processed after a simple web server-like operation has finished processing. All operation start renderers are available as well as operation stop and response renderers.

If all audit points are enabled, the simplest request generates at least three audit log outputs. Ticketed requests include **request**, **priopstart**, and **priopstop** audit points. Non-ticketed requests include **request**, **webopstart**, and **webopstop** audit points.

You can configure only the audit points desired. You can also configure only the information of interest for output. For example, for minimal output, all audit points can be disabled except for **priopstop** and **webopstop**. This provides information on each request without generating multiple output lines for each request. This does not show subordinate operations because they are not a concern.

Enable FSC audit logging

You must configure audit logging in the **fmsmaster** file and cycle the configurations for them to take effect. Audit logs can become huge very quickly; therefore, tuning the logging configuration and providing buffering after you enable logging can prevent disk space and performance issues.

1. Determine the audit points and fields/renderers that address your security or operational concerns.
2. **Define the format to use for each audit point** by **adding log properties** to the **fscdefaults** elements in the **fmsmaster** configuration file. The configurations must be cycled to take effect.
3. Enable the audit loggers using **fscadmin** commands, for example:

```
fscadmin -s http://myserver:4544 FSC_MYSERVER_user1
```

4. Inspect the logs to ensure the formats are parsed as expected.
5. Run some sample use cases to verify the output is sufficient.

6. Permanently set the audit logger level to **info** and tune the logging configuration if required.

FSC audit log properties

There is one **fscdefaults** element property used to configure the field delimiter in the audit file output, and seven **fscdefaults** properties used to configure each audit point output. Any audit point that does not contain a value does not generate output.

To allow the FSCs to consume the same tools, all FSCs in the system must share the same audit log configuration. Ensure that all FSCs use the same audit configuration by defining the **fscdefaults** elements at the **fmsenterprise** level in the **fmsmaster_fscid.xml** configuration file; then set the **overridable** attribute on the properties to **false**.

- **FSC_AuditLogDelimiter**

Specifies the delimiter used to separate audit field output in the audit log file. This can be a single or multicharacter value. The default value is the unique **|,|** character sequence. This is used because it is not found in any of the data and therefore is reliable when used to separate fields.

- **FSC_AuditLogRequestFormat**

Specifies a comma-separated list of format specifications (or renderers) for the **request** audit point.

- **FSC_AuditLogPrimaryOperationStartFormat**

Specifies a comma-separated list of format specifications for the *primary operation start* audit point. Primary relates to *ticketed* operations.

- **FSC_AuditLogPrimaryOperationStopFormat**

Specifies a comma-separated list of format specifications for the *primary operation stop* audit point.

- **FSC_AuditLogSubordinateOperationStartFormat**

Specifies a comma-separated list of format specifications for the *subordinate operation start* audit point.

- **FSC_AuditLogSubordinateOperationStopFormat**

Specifies a comma-separated list of format specifications for the *subordinate operation stop* audit point.

- **FSC_AuditLogWebOperationStartFormat**

Specifies a comma-separated list of format specifications for the *web operation start* audit point. These are non-ticketed requests, for example, FCC configuration downloads.

- **FSC_AuditLogWebOperationStopFormat**

Specifies a comma-separated list of format specifications for the *web operation stop* audit point.

The following example **fscdefaults** element shows the use of log properties:

fscdefault example:

```
<fscdefaults>
  <!-- audit configuration -->
  <property name="FSC_AuditLogDelimiter" value="|" overridable="false"/>
  <property name="FSC_AuditLogRequestFormat" value="Text(request),
PrimaryTransactionID,
  RequestRemoteAddr, RequestHeader(X-Route), RequestHeader(User-Agent),
  RequestLine, RequestHeader(Range)" overridable="false"/>
  <property name="FSC_AuditLogPrimaryOperationStartFormat" value="Text(priopstart),
PrimaryTransactionID, Operation, RequestMethod, RequestRemoteAddr,
  RequestHeader(X-Route), RequestHeader(User-Agent), RequestHeader(Range),
  TicketVersion, TicketAccessMethodNice, TicketIsBinaryNice,
TicketSignature,
  TicketExpiresTime, TicketUserID, TicketSiteID, TicketGUID,
  TicketFilestoreIDs, TicketRelativePath" overridable="false"/>
  <property name="FSC_AuditLogPrimaryOperationStopFormat" value="Text(priopstop),
PrimaryTransactionID, StatusCode, Message, ResponseHeader(Content-
Encoding),
  TargetBytes, ActualBytes, ResponseStreamStatus, DeltaMS"
  overridable="false"/>
  <property name="FSC_AuditLogSubordinateOperationStartFormat"
value="Text(subopstart),
  TransactionID, Operation, TicketVersion, TicketAccessMethodNice,
  TicketIsBinaryNice, TicketSignature, TicketExpiresTime, TicketUserID,
  TicketSiteID, TicketGUID, TicketFilestoreIDs, TicketRelativePath"
  overridable="false"/>
  <property name="FSC_AuditLogSubordinateOperationStopFormat" value="Text(subopstop),
TransactionID, StatusCode, Message, DeltaMS" overridable="false"/>
  <property name="FSC_AuditLogWebOperationStartFormat" value="Text(webopstart),
PrimaryTransactionID, Operation, RequestMethod, RequestRemoteAddr,
  RequestHeader(User-Agent), RequestLine" overridable="false"/>
  <property name="FSC_AuditLogWebOperationStopFormat" value="Text(webopstop),
PrimaryTransactionID, StatusCode, Message, ResponseHeader(Content-
Encoding),
  TargetBytes, ActualBytes, ResponseStreamStatus, DeltaMS"
  overridable="false"/>
</fscdefaults>
```

FSC audit log format specifications

Format specifications, also known as *field renderers*, determine the content of the log file. Some are simple and render a single value into the log. These values may come from transactional information, request or response headers, or even a string constant. Others are more complex and provide some analysis. For an example, see the **ResponseStreamStatus** field renderer. Any renderer can be specified in any audit point but may not be able to produce useful information. They are grouped depending on how they are intended to be used.

- General renderers

Available on all audit points.

Text(...) Renders the constant value provided between the parentheses. All white space is ignored. This is used to identify the audit point type. Examples are **priopstart**, **subopstop**, and so forth, but could be anything in your environment.

- Request related renderers

Available on all audit points.

RequestLine	Renders the HTTP request line as presented to the server.
RequestMethod	Renders the HTTP request method (PUT , GET , POST , and so forth).
RequestRemoteAddr	Renders the request (client) IP address.
RequestHeader(...)	Renders the value of any request header. The request name is provided between the parentheses.
PrimaryTransactionID	Shows the base (primary) transaction ID that can be used to track and correlate a request though the FMS system.

- Ticket related renderers

Available whenever a ticket is available at the given audit point.

TicketAccessMethod	Renders the numeric access the ticket provides (2 , 4 , and so forth; see TicketAccessMethodNice).
TicketAccessMethodNice	Renders the numeric access the ticket provides (see TicketAccessMethod) into easily understood access names: READ , WRITE , ADMINREAD , ADMINWRITE .
TicketExpiresTime	Renders the ticket expiry time in coordinated universal time (UTC).
TicketFileName	Renders the file name included in the ticket if there is one.
TicketFilestoreId	Renders the list of filestore IDs (volume IDs) referenced in the ticket.
TicketGUID	Renders the file GUID.
TicketIsBinary	Renders the binary flag for the ticket as T or F (see TicketIsBinaryNice).
TicketIsBinaryNice	Renders the binary flag (see TicketIsBinary) for the ticket in a string as TEXT or BINARY .
TicketRaw	Renders the entire content of the ticket.
TicketRawURLEncoded	Renders the entire content of the ticket in URL encoded form.
TicketRelativePath	Renders the relative path and file name included in the ticket based from the volume root.
TicketSignature	Renders the signature of the ticket.

TicketSiteID	Renders the site ID that generated the ticket. This is the same as the fmsenterprise ID.
TicketUserID	Renders the user ID (userid value) that generated the ticket.
TicketVersion	Renders the ticket version related to the encryption key as v100 , F100 , or M050 .

- General operation renderers

Available on stop and start audit points.

Operation	Renders a short description of the operation the FSC is performing.
TransactionID	For subordinate audit points, renders the transaction ID of the subordinate action with additional decoration to identify the <i>n</i> th subordinate call. For primary audit points, it is the same as the PrimaryTransactionID renderer.

- Operation stop renderers

Available on stop audit points.

DeltaMS	Renders the delta time in milliseconds from start to stop audit points.
StatusCode	Renders the resulting status code; it may be an HTTP status or an FSC error code.
Message	Renders the resulting message; it may be the HTTP status message or some form of error text.
TargetBytes	Renders the target bytes of the operation. If the value is not known, the output is -1 .
ActualBytes	Renders the actual bytes of the operation. If the value is not known, the output is -1 .

- Response related renderers that require a complete HTTP response

Available only on **priopstop** and **webopstop** audit points.

ResponseHeader(...)	Renders the value of any HTTP response header. The name is provided between the parentheses.
ResponseStreamStatus	Renders the status of the response stream. This renderer attempts to detect if a client's stream was downloaded completely or truncated. The possible outputs are UNKNOWN , COMPLETE , or TRUNCATED .

Any renderer can be included in any audit point output, although it may not be useful. Format errors, such as unknown renderer names (misspellings), do not cause configuration load errors, but the audit log output contains **FORMATERROR** in the problem fields.

Fields that do not have required information present, such as ticket-related renderers when no ticket is present, generally result in **null** in the output for that field in the audit log.

The first output to the audit log writes the current formatting for all enabled audit points. The formatting is also output whenever the audit configuration changes. It does not contain information about audit points that have no formatting configured and are therefore disabled.

The following is a sample audit log format output:

```
INFO - 2022/01/26-07:54:51,365 UTC - myhost123 - Active audit entry formats:

INFO - 2022/01/26-07:54:51,378 UTC - myhost123 - |,|Text(request)|,|PrimaryTransactionID|,|RequestRemoteAddr|,|RequestHeader(X-Route)|,|RequestHeader(User-Agent)|,|RequestLine|,|RequestHeader(Range)|,|

INFO - 2022/01/26-07:54:51,378 UTC - myhost123 - |,|Text(priopstart)|,|PrimaryTransactionID|,|Operation|,|RequestMethod|,|RequestRemoteAddr|,|RequestHeader(X-Route)|,|RequestHeader(User-Agent)|,|RequestHeader(Range)|,|TicketVersion|,|TicketAccessMethodNice|,|TicketIsBinaryNice|,|TicketSignature|,|TicketExpiresTime|,|TicketUserID|,|TicketSiteID|,|TicketGUID|,|TicketFilestoreIDs|,|TicketRelativePath|,|

INFO - 2022/01/26-07:54:51,378 UTC - myhost123 - |,|Text(priopstop)|,|PrimaryTransactionID|,|StatusCode|,|Message|,|ResponseHeader(Content-Encoding)|,|TargetBytes|,|ActualBytes|,|ResponseStreamStatus|,|DeltaMS|,|

INFO - 2022/01/26-07:54:51,378 UTC - myhost123 - |,|Text(subopstart)|,|TransactionID|,|Operation|,|TicketVersion|,|TicketAccessMethodNice|,|TicketIsBinaryNice|,|TicketSignature|,|TicketExpiresTime|,|TicketUserID|,|TicketSiteID|,|TicketGUID|,|TicketFilestoreIDs|,|TicketRelativePath|,|

INFO - 2022/01/26-07:54:51,378 UTC - myhost123 - |,|Text(subopstop)|,|TransactionID|,|StatusCode|,|Message|,|DeltaMS|,|

INFO - 2022/01/26-07:54:51,378 UTC - myhost123 - |,|Text(webopstart)|,|PrimaryTransactionID|,|Operation|,|RequestMethod|,|RequestRemoteAddr|,|RequestHeader(User-Agent)|,|RequestLine|,|

INFO - 2022/01/26-07:54:51,378 UTC - myhost123 - |,|Text(webopstop)|,|PrimaryTransactionID|,|StatusCode|,|Message|,|ResponseHeader(Content-Encoding)|,|TargetBytes|,|ActualBytes|,|ResponseStreamStatus|,|DeltaMS|,|
```

The following is sample audit log output based on the previous configuration:

```
INFO - 2022/01/26-07:54:51,379 UTC - myhost123 - |,|request|,|(-7316198962075068416)fsc_s6|,|127.0.0.1|,|null|,|FMS-FSCJavaClientProxy/8.2 (bd:20220119)|,|GET /mapClientIPToFSCs?client= HTTP/1.1|,|null|,|

INFO - 2022/01/26-07:54:51,380 UTC - myhost123 - |,|webopstart|,|(-7316198962075068416)fsc_s6|,|BootstrapHandler|,|GET|,|127.0.0.1|,|FMS-FSCJavaClientProxy/8.2 (bd:20220119)|,|GET /mapClientIPToFSCs?client= HTTP/1.1|,|

INFO - 2022/01/26-07:54:51,381 UTC - myhost123 - |,|webopstop|,|(-7316198962075068416)fsc_s6|,|200|,|OK|,|null|,|57|,|57|,|COMPLETE|,|1|,|

INFO - 2022/01/26-07:54:51,384 UTC - myhost123 - |,|request|,|(-7316198962075068415)fsc_s6|,|127.0.0.1|,|null|,|FMS-FSCAdmin/8.2 (bd:20220225) Java/1.5.0_11|,|GET / HTTP/1.1|,|null|,|

INFO - 2022/01/26-07:54:51,385 UTC - myhost123 - |,|priopstart|,|(-7316198962075068415)fsc_s6|,|CacheCommands$ClearCommand|,|GET|,|127.0.0.1|,|null|,|FMS-FSCAdmin/8.2 (bd:2022
```

```

0125) Java/1.5.0_11|,|null|,|v100|,|ADMINREAD|,|BINARY|,|739388a12ef48c3473e19bd78049661
6b989cf3b8bab1f5d5dfd0bb22a7d71db|,|2022/01/26 07:56:51|,|FSCAdmin|,|,|,|nougid
|,|[]|,|./clearcache|,|

INFO - 2022/01/26-07:54:51,388 UTC - myhost123 - |,|priopstop|,|(-7316198962075068415)f
sc_s6|,|200|,|OK|,|null|,|17|,|17|,|COMPLETE|,|3|,|

INFO - 2022/01/26-07:55:14,180 UTC - myhost123 - |,|request|,|(-362480191128027786)fsc_
s7[1]>fsc_s6|,|127.0.0.1|,|fms.teamcenter.com^fsc_s7,fms.teamcenter.com^fsc_s6|,|FMS-FSC
/8.2 (bd:20220225) Java/1.5.0_11|,|GET /tc/fms/fms.teamcenter.com/g2/fsc_s6 HTTP/1.1|,|n
ull|,|

INFO - 2022/01/26-07:55:14,180 UTC - myhost123 - |,|priopstart|,|(-362480191128027786)f
sc_s7[1]>fsc_s6|,|CoordinatorVolumeState|,|GET|,|127.0.0.1|,|fms.teamcenter.com^fsc_s7,f
ms.teamcenter.com^fsc_s6|,|FMS-FSC/8.2 (bd:20220225) Java/1.5.0_11|,|null|,|v100|,|ADMIN
READ|,|BINARY|,|ca124695734bb33ee6e65ba0fdb087587214de0b43d8da2c2eb8353a3d92e89|,|2022/
01/26 07:57:11|,|nouser|,|,|,|,|[]|,|fsc_s6/config/volum
estate/nvars/action=get;enterpriseid=fms.teamcenter.com|,|

INFO - 2022/01/26-07:55:14,181 UTC - myhost123 - |,|priopstop|,|(-362480191128027786)fs
c_s7[1]>fsc_s6|,|200|,|OK|,|null|,|6|,|6|,|COMPLETE|,|1|,|

```

The following figure shows a sample format specification for a primary start operation that can be used to track access to a dataset files by users, the associated portion of the format output, and the resulting portion in the log file output for a sample transaction.

```

<fscdefaults>
  <!-- audit configuration -->

  ↓

  <property name="FSC_AuditLogPrimaryOperationStartFormat" value="Text({priopstart},
    PrimaryTransactionID, Operation, RequestMethod, RequestRemoteAddr,
    TicketAccessMethod, TicketIsBinary, TicketSignature, TicketExpiresTime,
    TicketFileName, TicketUserID, TicketSiteID, TicketGUID,
    TicketFilestoreIDs, TicketRelativePath" overridable="false"/>

  ↓

</fscdefaults>
-----
INFO - 2022/01/26-07:54:51,365 UTC - myhost123 - Active audit entry formats:

INFO - 2022/01/26-07:54:51,378 UTC - myhost123 - |,|Text({priopstart})|,|PrimaryTransactio
nID|,|Operation|,|RequestMethod|,|RequestRemoteAddr|,|TicketAccessMethod|,|TicketIsBinar
y|,|TicketSignature|,|TicketExpiresTime|,|TicketFileName|,|TicketUserID|,|TicketSiteID|,
|TicketGUID|,|TicketFilestoreIDs|,|TicketRelativePath|,|

↓
-----
INFO - 2022/01/26-07:54:51,385 UTC - myhost123 - |,|priopstart|,|(-7316198962075068415)f
sc_s6|,|IMAN_export|,|GET|,|127.0.0.1|,|2012/01/26 07:56:51|,|2|,|T|,|739388a12ef48c3473e
19bd780496616b989cf3b8bab1f5d5dfd0bb22a7d71db|,|ToePlate.prt|,|gl_eng_1|,|,|,|nougid
|,|[]|,|fsc_s6/tv1/ToePlate.prt|,|
↓

```

The **TicketFileName** and **TicketUserID** renders are added to the format specification as shown in the top section. This results in the user ID and accessed file name values in the output file as shown in the bottom section. You may also notice the file name value appears in the output for

the **TicketRelativePath** render, making it unnecessary to include the **TicketFileName** render in this instance.

FSC audit log transaction IDs

Transaction IDs are the key to associating all the audit information together across the FMS network. Early in the request processing, the FSC looks for a transaction ID in the request headers. If it finds one, it appends the local FSC ID and uses that as its *base transaction ID*. If a previous one is not found, it increments an internal count and appends its FSC ID. The internal count starts based on a secure random number.

There is no guarantee duplicate IDs are not generated, but given the range of a 64-bit value collisions, duplicates are very rare and even more unlikely in close proximity in time. When you search for transactions, if a collision occurs, the timestamps can be used to identify the different transactions.

Any suboperation appends $[n+1]$ to the end of each transaction ID. A transaction ID supplies exact information on how a request is routed through the network. The following is an example of how a transaction ID propagates through a number of servers and shows the resulting transaction ID information.

Given the following FSCs:

```
FSC_emdbangfms_tcdba
FSC_spandfms_tcdba
FSC_flodup_tcdba
FSC_yagmlsp_tcdba
FSC_wndisxk_tcdba
```

1. The first FSC (**FSC_emdbangfms_tcdba**) generates a new ID.

```
(342342349932)FSC_emdbangfms_tcdba
```

It then performs a subordinate operation (subop), appends **[1]** to the transaction ID, and sends the request.

```
( 342342349932)FSC_emdbangfms_tcdba[1]
```

2. The next FSC (**FSC_spandfms_tcdba**) receives and joins the existing transaction ID.

```
( 342342349932)FSC_emdbangfms_tcdba[1]>FSC_spandfms_tcdba
```

It then performs a subop.

```
( 342342349932)FSC_emdbangfms_tcdba[1]>FSC_spandfms_tcdba[1]
```

3. A third FSC (**FSC_flodup_tcdba**) receives and joins the existing transaction ID.

```
(342342349932)FSC_emdbangfms_tcdba[1]>FSC_spandfms_tcdba[1]>FSC_flodup_tcdba
```

It then performs a subop.

```
(342342349932)FSC_emdbangfms_tcdba[1]>FSC_spandfms_tcdba[1]>FSC_flodup_tcdba[1]
```

And another subop.

```
(342342349932)FSC_emdbangfms_tcdba[1]>FSC_spandfms_tcdba[1]>FSC_flodup_tcdba[2]
```

4. The forth FSC (**FSC_yagmlsp_tcdba**) received the prior subop ([1]).

```
(342342349932)FSC_emdbangfms_tcdba[1]>FSC_spandfms_tcdba[1]>FSC_flodup_tcdba[1]>FSC_yagmlsp_tcdba
```

5. A fifth FSC (**FSC_wndisxk_tcdba**) received the prior subop ([2]).

```
(342342349932)FSC_emdbangfms_tcdba[1]>FSC_spandfms_tcdba[1]>FSC_flodup_tcdba[2]>FSC_wndisxk_tcdba
```

This shows that the transaction ID on any server provides an indication of the path through the network.

Even if intermediate FSCs do not have auditing enabled, transaction IDs are still generated or propagated along with the requests.

Configuring FMS

Configuring FMS for HTTPS

Introduction to configuring FMS for HTTPS

You can configure File Management System (FMS) to use either HTTP or HTTPS (a secure server layer protocol). Set the protocol during installation from Teamcenter Environment Manager (TEM), using the **FSC Non-Master Settings** panel and the **Proxy** tab in the **File Client Cache** panel.

HTTPS creates a secure channel over an insecure network. HTTPS protection is based on a major certificate authority guaranteeing that your web server is the entity it claims. This protection method uses verified and trusted server certificates, private keys (your keystore), and public keys (the certificate signing request).

Selecting HTTPS during Teamcenter installation

If you select HTTPS as your protocol during Teamcenter installation, you are prompted to supply the appropriate proxy, host, and port information. You are also asked whether you want to add the URL of the local host to the list of servers defined in the **Fms_BootStrap_Urls** preference. Your only postinstallation tasks are to:

- Use a trusted certificate to generate a keystore and key entry. Trusted certificates are purchased from third-party certificate vendors, such as VeriSign or Thawte.

Using untrusted (self-signed) certificates requires additional configuration to either import the certificate into the client certificate keystores, or to modify FMS properties to permit clients to connect to servers using self-signed certificates. Siemens Digital Industries Software does not recommend using self-signed certificates.

- Generate a certificate signing request. (If you upgrade from a previous version of Teamcenter, you must transfer all the HTTPS certificate information to the upgraded Teamcenter installation.)

Switching to HTTPS after selecting HTTP during Teamcenter installation

If you selected HTTP as your protocol during Teamcenter installation, but sometime *after* installation decide to configure FMS to use HTTPS, perform the following tasks:

- Use a trusted certificate to generate a keystore and key entry.
- Generate a certificate signing request (CSR).
- Modify the FMS primary configuration file to reflect the new HTTPS addresses.
- Add the URL of the local HTTPS host to the list of servers defined in the **Fms_BootStrap_Urls** preference.
- Update any installed FCCs.

Configure FMS for HTTPS

You can configure FMS to use either the hypertext transfer (HTTP) protocol or a secure server layer (HTTPS) protocol. You set the protocol during installation from the **FSC Non-Master Settings** and **FCC Settings** panels in Teamcenter Environment Manager (TEM).

Note:

If you chose the HTTPS protocol for FMS during installation, you are prompted to supply the appropriate proxy, host, and port information. You are also asked whether you want to add the URL of the local host to the list of servers defined in the **Fms_BootStrap_Urls** preference.

The only post installation tasks required to implement HTTPS are **generating a keystore and key entry** and **generating a certificate signing request**.

If you chose the HTTP protocol for FMS during installation and now want to use the HTTPS protocol, you must:

- Modify the FMS primary configuration file to reflect the new HTTPS addresses.

The **fmsmaster_fscid.xml** is located in the **TC_ROOT\fsc** directory.

Update the **fsc address** in the FMS primary configuration file as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fmsworld SYSTEM "fmsmasterconfig.dtd">
<fmsworld>
  <fmsenterprise id="471539747">
    <fscgroup id="mygroup">
      <fsc id="FSC_myhost_userid"
        address="https://myhost.mycompany.com:4545">
        <volume id="139747566d871c1b2023"
          root="/mnt/disk1/tcapps/tc/TC_VOL/volume1"/>
          <transientvolume id="ce8399515feada2dee4c3e79b955d8ba"
            root="/tmp/transientVolume_tc_userid"/>
          </fsc>
        <clientmap subnet="127.0.0.1" mask="0.0.0.0">
          <assignedfsc fscid="FSC_myhost_userid"/>
        </clientmap>
      </fscgroup>
    </fmsenterprise>
  </fmsworld>
```

- Add the URL of the local HTTPS host to the list of servers defined in the **Fms_BootStrap_Urls** preference.
- Update any installed FCCs.

FCC installations contain configuration files that point to FSC servers. When you change the FSC servers to support HTTPS, you must also revise the **parentfsc** address in the **fcc.xml** file to the new protocol. The **fcc.xml** file is located in the **FMS_HOME** directory (for example, **TC_ROOT\tccs**).

1. In the **fcc.xml** file, update the parent FSC address. For example:

```
<parentfsc address="https://myhost.mycompany.com:4545/tc/fms/471539747"/>
```

2. Restart the system.

Keystores and key entries

To implement HTTPS for FMS, you must generate a keystore and key entry, and then generate a certificate signing request (CSR) from the private key.

Siemens Digital Industries Software recommends using a trusted certificate, purchased from a third-party vendor, when configuring FMS to use HTTPS.

Note:

The certificate authority's root certificates for your purchased certificates must include standard distributions of Sun Microsystems' Java run-time environment. This is usually the case for certificates purchased from well-known certificate authorities.

Use the following elements to create a keystore.

Element	Description
<i>keystore</i>	Specifies the keystore file name. This is the Java-based storage standard. Public and private keys are stored in an encrypted keystore. Individual keys within this cryptographic storage may also have individual password protection.
<i>keystore.password</i>	Specifies the keystore password. The password is required to open or manage the keystore. The standard entry is changeit .
FSC_myhost	Specifies an alias name for the certificate. The certificate is bound to the host; name it accordingly. A similar convention FSC_host_userid is used by installers to name configuration files.
FSC_myhost.password	Specifies the certificate (alias) password required to retrieve the certificate.
FSC_myhost.csr	Specifies the name of the certificate signing request (CSR) file. The file contains the CSR information and is sent to the signing authority.
FSC_myhost.cer	Specifies the certificate file. This is the file returned by the signing authority.

Use the trusted certificate to create a keystore in your **FSC_HOME** directory.

Generate a keystore and key entry

1. Run the following command to create a keystore and private key in your *FSC_HOME* directory:

```
>keytool -genkey -keystore keystore -keyalg RSA -alias FSC_myhost
```

2. Complete each question prompted by the command. For example:

```
>keytool -genkey -keystore keystore -keyalg RSA -alias FSC_myhost
```

```
Enter keystore password: keystore.password
```

```
What is your first and last name?
```

```
[Unknown]: myhost.mydomain.com
```

```
What is the name of your organizational unit?
```

```
[Unknown]: mycompany
```

What is the name of your organization?

[Unknown]: *mycompany*

What is the name of your City or Locality?

[Unknown]: *mycity*

What is the name of your State or Province?

[Unknown]: *mystate*

What is the two-letter country code for this unit?

[Unknown]: *my*

Is CN=myhost.mydomain.com, OU=mycompany, O=mycompany, L=mycity, ST=mystate, C=my correct?

[no]: *yes*

Enter key password for <FSC_myhost>

(RETURN if same as keystore password): *FSC_myhost.password*

- Run the following command to confirm that you can read the file and view the key entry:

```
>keytool -list -keystore keystore
```

- Complete each question prompted by the command. For example:

```
>keytool -list -keystore keystore
```

Enter keystore password: *keystore.password*

Keystore type: *jks*

Keystore provider: *SUN*

Your keystore contains 1 entry

```
fsc_myhost, Nov 8, 2007, keyEntry,
Certificate fingerprint (MD5): 59:B6:2D:38:24:16:45:1B:47:2A:E9:06:55:80:B3:C6
```

- Create a copy of the keystore file and keep it in a secure location.

Caution:

The private key stored in the keystore is not recoverable if the file or passwords are lost.

Create a certificate signing request (CSR)

The certificate signing request (CSR) is the message sent from your site to a certificate authority in order to apply for a digital identity certificate. The CSR is the public key generated on your server that validates your web server/organization data when you request a certificate from your third-party certificate vendor.

After creating a CSR, follow the process required by your certificate signing authority to obtain your signed certificate.

1. Run the following command to create a CSR from your private key:

```
>keytool -certreq -keystore keystore -alias FSC_myhost -file FSC_myhost.csr
```

2. Complete each question prompted by the command. For example:

```
>keytool -certreq -keystore keystore -alias FSC_myhost -file FSC_myhost.csr
Enter keystore password: keystore.password
Enter key password for <FSC_myhost> FSC_myhost.password
```

3. Locate the CSR file. This is the file you must submit to the certificate signing authority. For example:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBtjCCAR8CAQAwZjELMAkGA1UEBhMCBkxkxEDA0BgNVBAgTB215c3RhZGUxDzANBgNVBACTBm15
Y2l0eTESMBAGA1UEChMJbXljY21wYW55MRIwEAYDVQQLEwltewNvbXBhbnkxHDAaBgNVBAMTE215
aG9zdC5teWRvbWpbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJ0h3iF8KBEN2UKw
hw1dw+RLxGwCsptLA3EI+6rAKa32dg/4FY89zBcUG02413X0BxQWcsRznYwFDJHLK4En7I2xeJNs
ORwJfBeF9yW6d4lzaWA6LATFr5T3DHafF6mSRNP1+739mpGuQr44AXBQWqZo0Mhecc+n/ErekMLZ
dgWTagMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQCQTqujL7GIXz0is0fUoAxtCydMiX1BeVHU+l/
IqcTh4BX8V3vJmm+kHwwKn3yeih+WJzYmDdNh/uaKx07txyFdPPDd1bdIosFc4XIZwys0jFKwGqf
MUjB9wgaKgHSRQTtCOPBE0/CLLjm8ocFNQBWysYVevAZQAmEMP90BxBt/Q==
-----END NEW CERTIFICATE REQUEST-----
```

Importing certificates into the FSC keystore

After obtaining the signed certificate from your certificate signing authority, you must import it into the keystore.

1. Run the following command to import the signed certificate:

```
>keytool -import -trustcacerts -keystore keystore -file FSC_myhost.cer -alias FSC_myhost
```

2. Complete each question prompted by the command. For example:

```
>keytool -import -trustcacerts -keystore keystore -file FSC_myhost.cer -alias FSC_myhost
Enter keystore password: keystore.password
Enter key password for FSC_myhost FSC_myhost.password
```

3. Verify the **keystore.FSC_host_userid** file in the **FSC_HOME** directory. The keystore must contain the private key and certificate for the local machine. For example:

```

myhost> keytool -list -v -keystore
keystore.FSC_myhost_userid.jks -storepass keystore.FSC_myhost_userid.password

Keystore type: jks
Keystore provider: SUN
Your keystore contains 1 entries

Alias name: myhost.mycompany.com
Creation date: Jan 23, 2008
Entry type: keyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=myhost.mycompany.com, OU=QA, O=YOUR Corp, L=Plano, ST=Texas, C=US
Issuer: EMAILADDRESS=premium-server@thawte.com, CN=Thawte Premium Server CA,
OU=Certification Services Division, O=Thawte Consulting cc, L=Cape Town, ST=Western Cape,
C=ZA Serial number: 485099dcc36d1ea9d773ba153022a951
Valid from: Thu Jan 10 16:44:38 CST 2008 until: Thu Mar 27 13:20:25 CDT 2008 Certificate
fingerprints:
MD5: 86:7E:16:59:99:E6:6F:B6:27:9B:92:19:E7:65:EB:A2
SHA1: 6A:D1:64:7A:0A:E1:CB:62:D3:EF:91:BF:E9:A0:CE:AF:A3:3D:E4:1E
Certificate[2]:
Owner: EMAILADDRESS=premium-server@thawte.com, CN=Thawte Premium Server CA,
OU=Certification Services Division, O=Thawte Consulting cc, L=Cape Town,
ST=Western Cape, C=ZA, Issuer: EMAILADDRESS=premium-server@thawte.com, CN=Thawte Premium
Server CA
OU=Certification Services Division, O=Thawte Consulting cc, L=Cape Town, ST=Western Cape,
C=ZA Serial number: 1
Valid from: Wed Jul 31 19:00:00 CDT 1996 until: Thu Dec 31 17:59:59 CST 2020 Certificate
fingerprints:
MD5: 06:9F:79:16:66:90:02:1B:8C:8C:A2:C3:07:6F:3A
SHA1: 62:7F:8D:78:27:65:63:99:D2:7D:7F:90:44:C9:FE:B3:F3:3E:FA:9A
*****
*****

```

4. Update the **fsc.FSC_host_userid.properties** file in the *FSC_HOME* directory with the keystore and key entry information. For example:

```

# fsc.FSC_myhost_userid.properties
com.teamcenter.fms.servercache.keystore.file=${FMS_HOME}/keystore.FSC_myhost_userid.jks
com.teamcenter.fms.servercache.keystore.password=keystore.FSC_myhost_userid.password
com.teamcenter.fms.servercache.keystore.ssl.certificate.password=
keystore.FSC_myhost_userid.password
# these are not needed for 1-way SSL
javax.net.ssl.keyStore=${FMS_HOME}/keystore.FSC_myhost_userid.jks
javax.net.ssl.keyStorePassword=keystore.FSC_myhost_userid.password
javax.net.ssl.trustStore=${FMS_HOME}/keystore.FSC_myhost_userid.jks
javax.net.ssl.trustStorePassword=keystore.FSC_myhost_userid.password

```

Configuring FMS ticket signing keys

Configuring symmetric keys

For improved security, you can move the FMS encryption key from a clear text file to an encrypted, password-protected keystore file. Use the **keygen** script to import the key file into a keystore and the **passwordtool** script to generate encrypted passwords based on a clear text password.

The keytool used in this example is from Java JDK 1.5.

1. Create the signing keystore to hold the FMS encryption key.

- a. Determine the key alias under which you want to store the FMS key.

In this example, the key alias is **ent123.tickets**.

- b. Determine the keystore name.

In this example, the keystore name is **trusted.jceks**.

- c. Determine the passwords used for the keystore.

In this example, the password is **trusted.jceks.lp7qZF.password**.

- d. Determine the password used for the FMS encryption key.

In this example, the password is **ent123.tickets.z3nYsY.password**.

2. Use the **keygen** script to create the keystore and key.

In this example, a new key is created. Alternatively, you can import an existing key.

```
> keygen 128
5706c8eebd67eb754544ab720f08d95b
```

3. Use the **keygen** script to import the key file into the keystore using the following form:

```
> keygen -importseckey -keystore keystorefilename -storepass keystore.password
-alias alias [-overwrite] [-keypass key.password] [[-k keyfile] | [-key asciihexkey]]
keystore filename must end in .jceks (SecretKeys can only be stored in jceks keystores)
[-keypass key.password] is optional (defaults to storepass value)
[-overwrite] is optional, by default will not allow overwriting an existing key
Either [-k keyfile] or [-key asciihexkey] are required
```

For example:

```
> keygen -importseckey -keystore trusted.jceks -storepass trusted.jceks.lp7qZF.password
-alias ent123.tickets -keypass ent123.tickets.z3nYsY.password -key
5706c8eebd67eb754544ab720f08d95b
```

No messages are displayed when the script succeeds.

4. Use the keytool to list the contents of the keystore. For example:

```
> keytool -storetype jceks -keystore trusted.jceks -storepass trusted.jceks.lp7qZF.password
-list -v
```

```
Keystore type: jceks
Keystore provider: SunJCE
```

```
Your keystore contains 2 entries
```

```
Alias name: ent123.tickets
Creation date: Jul 10, 2023
Entry type: keyEntry
```

```
*****
*****
```

```
Alias name: ent123.trustedadmin
Creation date: Jul 10, 2023
Entry type: keyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=FMS trusted admin policy site ent123, OU=org unit, O=org, L=c, ST=st, C=cc
Issuer: CN=FMS trusted admin policy site ent123, OU=org unit, O=org, L=c, ST=st, C=cc
Serial number: 4a578037
Valid from: Mon Jul 10 13:53:59 EDT 2023 until: Mon Nov 24 12:53:59 EST 2050
Certificate fingerprints:
    MD5: 66:6F:67:55:09:CA:04:69:52:76:C8:49:30:30:75:F0
    SHA1: E4:74:66:DD:54:C2:0D:4B:D2:AD:74:EA:65:69:89:C7:0F:16:71:49
```

```
*****
*****
```

5. Create and/or modify the FSC property files.

- a. Encrypt the keystore and key/alias password values using the **passwordtool** script. For example:

```
> passwordtool -encrypt trusted.jceks.lp7qZF.password
```

```
fcLxB/oeZ+IeNnP/vofAqFpDqmJdSyaU0y+EHU0ffRc=
```

```
> passwordtool -encrypt ent123.tickets.z3nYsY.password
```

```
vhTHTCLYz9BxE8TN4MpLFNIFkIrDMCfU7mh+pYbqfcw=
```

- b. Add the following properties to the **fsc.fscid.properties** file:

```
# signing keystore file and password
com.teamcenter.fms.signing.keystore.file=trusted.jceks
com.teamcenter.fms.signing.keystore.epassword=fcLxB/oeZ+IeNnP/vofAqFpDqmJdSyaU0y+EHU0ffRc=
# key password(s) property name form: com.teamcenter.fms.signing.<alias>.epassword
com.teamcenter.fms.signing.ent123.tickets.epassword=vhTHTCLYz9BxE8TN4MpLFNIFkIrDMCfU7mh+pYbqfcw=
```

6. In the **fmsmaster** configuration file, add the following elements under **fscadminpolicies** and before **fscdefaults**. For example:

```
<fmsworld>
...
<fmsenterprise id="ent123">
...
<!-- policy and fscadminpolicies elements would be here -->
...
<ticket version="N050" keystorealias="ent123.tickets" />
<ticket version="Z100" keystorealias="ent123.tickets" />
<ticket version="T100" keystorealias="ent123.tickets" />
```

The system uses the **keystorealias** attribute to retrieve the password from the properties file and access the key in the keystore.

7. Confirm the accuracy of the configuration by reloading the **fmsmaster** configuration file. The FSC cannot reload the configuration if the ticketing aliases or keys are unavailable for any reason. For example:

```
> fscadmin -s http://cii6w223:4544 ./config/reload

Initial configuration hash: efed77c0315fc5f9bf289cb4db0b164f
Configuration reload successful.
Final configuration hash: ac718b9778cbcfecabeca266b3c1155a
```

8. Restart the FSC.

The ticketing keys are applied.

9. Create and/or modify the **fscadmin** property file by adding the following properties. Use the same encrypted passwords as generated previously. For example:

```
# signing keystore file and password
com.teamcenter.fms.signing.keystore.file=trusted.jceks
com.teamcenter.fms.signing.keystore.epassword=fcLxB/oeZ+IeNnP/vofAqFpDqmJdSyaU0y+EHU0ffRc=
# default fms ticket signing alias
com.teamcenter.fms.signing.tickets.alias=ent123.tickets
# key password(s) property name form: com.teamcenter.fms.signing.<alias>.epassword
com.teamcenter.fms.signing.ent123.tickets.epassword=vhTHTCLYz9BxE8TN4MpLFNIFkIrDMCFU7mh+pYbqf
CW=
```

10. Use the **fscadmin** command to confirm that the keys/certificates are available. For example:

```
> fscadmin -s http://cii6w223:4544 ./keystoreinfo

Keystore info:
# of private keys: 1, aliases: [ent123.trustedadmin]
# of secret keys: 1, aliases: [ent123.tickets]
# of certificates: 1, aliases: [ent123.trustedadmin]
```

11. Check in the FSC log files. For example:

```
...
INFO - 2023/07/10-18:38:55,500 UTC - cii6w223 - Keystore info:
# of private keys: 1, aliases: [ent123.trustedadmin]
# of secret keys: 1, aliases: [ent123.tickets]
# of certificates: 1, aliases: [ent123.trustedadmin]
...
```

Any keys that cannot be loaded are listed in the FSC log files. For example:

```
java.security.UnrecoverableEntryException
    at java.security.KeyStoreSpi.engineGetEntry(KeyStoreSpi.java:455)
    at java.security.KeyStore.getEntry(KeyStore.java:1218)
    ...
```

12. Verify that the key has been moved or configured correctly by running the FSC. If the FSC runs normally, reports that the secret key is available, and the **fscadmin** command works, then the move is successful.

If the configuration is incorrect, either the FSC does not reload, or the secret key is not listed, or the **fscadmin** command gives the following error:

```
> fscadmin -s http://cii6w223:4544 ./keystoreinfo

Error, server returned status code: 400, status message: TICKET_VALIDATION_FAIL_0
```

13. If a previous plain text key configuration exists, delete the previous clear text key file after verifying the **fscadmin** command and FSC are successfully using the keystore.

After confirming the encryption key has been successfully moved to the keystore file, manage it by:

- Storing a backup of your keystores and passwords in a safe location.
- Creating a new encryption key and deploying new keystores if you suspect the encryption key is compromised.

Update the Teamcenter database with the new key

Once you have generated the new key, you must update the Teamcenter database with the key so that tickets generated from Teamcenter are signed with the new key before being sent to the FSC which then validates using the new key. Use the **install_encryptionkeys** utility with the **-f=modify** option to update the database. For example:

```
install_encryptionkeys -u=Tc-admin-user -p=password -g=group -f=modify
```

When prompted, supply the new key.

Configuring FMS to use public-private (asymmetric) key pairs

Asymmetric keys and FMS

By default, FMS uses symmetric keys for ticket validation. A typical symmetric key scenario has the Teamcenter server generating the ticket and signing it with the symmetric key. That ticket is then sent to FMS where it is validated using the same symmetric key deployed on the FMS side.

You can optionally configure your site to use more secure public-private (asymmetric) key pairs. When [setting up your sites and choosing to use asymmetric keys](#), Deployment Center generates the public-private key pair during deployment. The private key resides with the Teamcenter server, and the public key is deployed with the FSC. The Teamcenter server uses the private key to sign the ticket. The FSC then uses the public key to validate the signature on the ticket that is sent when uploading or downloading a file.

Be aware that FSC administrative commands continue to use a symmetric key. That key is included in the same keystore as the public key that is deployed on the FSC.

You can also [update sites that use symmetric keys to use asymmetric keys](#).

To use asymmetric keys with Multi-Site Collaboration, all sites in the Multi-Site federation must use asymmetric keys. See [Configure Multi-Site Collaboration to use asymmetric keys](#) for details.

Configure FMS to use asymmetric keys

Use Deployment Center to configure FMS to use asymmetric keys. Deployment Center generates and deploys the public-private key pair for use with Teamcenter and FMS. A public-private key pair and an administration symmetric key are generated corresponding to a specific Teamcenter site.

To use asymmetric keys at sites already using symmetric keys, see [Convert sites that use symmetric keys to use asymmetric keys](#).

To use asymmetric keys for sites using Multi-Site, see [Configure Multi-Site Collaboration to use asymmetric keys](#).

In Deployment Center, configure FMS to use asymmetric keys as follows:

1. On the **Components** tab, select the **FSC Keys** component.
2. Under **General Settings**, check **Generate New Keys**.
3. Enter and confirm the password to be used for the keystore (.p12) file that is created. Record this password for later use.
4. Under **FSC Keys Options**, check **Use Asymmetric Key (advanced)** and click **Save Component Settings**.

When you generate and run the Deployment Center installation scripts on the servers, the keys are installed.

Convert sites that use symmetric keys to use asymmetric keys

You can update an existing FMS deployment that uses symmetric keys to one that uses asymmetric keys. Perform the following tasks to update an existing FMS deployment:

1. Prepare for the update.
2. Set the Teamcenter preference to specify using asymmetric keys.
3. Generate a new public-private key pair in the Teamcenter deployment.
4. Export the public key from Teamcenter to a keystore file.
5. Generate and add the symmetric key to the asymmetric keystore.
6. Deploy the keystore file on the FSC and update the FSC configuration.
7. Restart all processes.

Run all commands from a Teamcenter command prompt. Commands requiring credentials must be run by a Teamcenter administrator with DBA privileges.

Prepare for the update

Shut down the FSCs, Pool Manager, and all TcServer processes using **the Teamcenter Management Console**.

Set the Teamcenter preference to specify using asymmetric keys

Use the **preferences_manager** utility to set the **FMS_USE_ASYMMETRIC_KEYS** preference as follows:

```
preferences_manager -u=Tc-admin-user -p=password -g=group
-mode=import -scope=SITE -action=OVERRIDE
-preference=FMS_USE_ASYMMETRIC_KEYS -values="true"
```

Generate a new public-private key pair in the Teamcenter deployment

Run the **install_fms_keys** utility to generate a new private key in the Teamcenter database as follows:

```
install_fms_keys -u=Tc-admin-user -p=password -g=group -f=install
```

The key is created and installed in the Teamcenter database.

Export the public key from Teamcenter to a keystore file

Once the private key is generated and updated in the database, use the **install_fms_keys** utility to export the public key corresponding to the private key as follows:

```
install_fms_keys -u=Tc-admin-user -p=password -g=group -f=exp_pubkey
-file_password=key_pass
```

Where *key_pass* is the password for the exported keystore file.

The public key is exported from the Teamcenter database and saved in the current directory as a keystore file named *FMSPublicKeys.p12*.

Generate and add the symmetric key to the asymmetric keystore

With asymmetric keys configured for ticket validation for file operations, FMS still uses a symmetric key for administrative commands. Use the *keygen.bat* utility (*keygen.sh* on Linux) to add this symmetric key to the keystore (*.p12* file) that holds the public key for file operations:

```
keygen -genadminkey -keystore p12_keystore_file -storepass key_pass
```

Where:

p12_keystore_file

The file name of the *.p12* file holding the public key (*FMSPublicKeys.p12*).

key_pass

The password of the *.p12* file holding the public key (*FMSPublicKeys.p12*).

The administrative key is generated and added to the *FMSPublicKeys.p12* keystore.

Deploy the keystore file on the FSC and update the FSC configuration

Use the following steps to update the FSC to use asymmetric keys and the new keystore file:

1. Copy the *.p12* keystore file to the *FSC_Home* directory.
2. Update the FMS primary configuration file (*FSC_HOME\fmsmaster_fscid.xml*) to specify using asymmetric keys by adding a **ticketkeys** element. For example:

```
<fmsworld>
  <fmsenterprise id="site_ID">
    <ticketdigest value="SHA-256"/>

    <ticketkeys keystore="p12_keystore_file">
      <publickeyalias name="pub_key_alias" enterpriseid="site_ID" />
      <adminalias name="admin"/>
    </ticketkeys>
  </fmsenterprise>
</fmsworld>
```

```

<fscgroup id="g1">
  <fsc id="fsc_ent_a" address="http://fscserver:5555">
    <volume id="vol1" root="D:\FMSVolumes\vol1" enterpriseid="site_ID" />
  </fsc>

  <clientmap subnet="127.0.0.1" mask="0.0.0.0">
    <assignedfsc fscid="fsc_ent_a" />
  </clientmap>
</fscgroup>
</fmsenterprise>
</fmsworld>

```

Where:

p12_keystore_file

The file name of the .p12 keystore file created earlier (*FMSPublicKeys.p12*). Include the path to the file if the file is not stored in *FSC_Home*.

site_ID

The ID of your site.

pub_key_alias

The public key (stored in the .p12 keystore file) alias. The alias is the same as *site_ID*.

3. Use the **passwordtool.bat** utility (**passwordtool.sh** on Linux) to generate an encrypted password for the keystore file. For example:

```
passwordtool -encrypt ekey_pass
```

Where *ekey_pass* is the encrypted password for the keystore file.

4. Add the following line to the FMS server configuration properties file (*FMS_HOME\fsc.properties* or *FSC_HOME\fsc_fscid.properties*) to specify using the new keystore password:

```
com.teamcenter.fms.public.keystore.epassword=key_pass
```

Where *key_pass* is the encrypted password generated in the previous step using **passwordtool**.

5. Add the following lines to the *FSC_HOME\fscadmin.properties*:

```
com.teamcenter.fms.signing.keystore.file=keystore_path
```

```
com.teamcenter.fms.signing.keystore.epassword=ekey_pass
```

```
# Alias name from entry added in fmsmaster xml
com.teamcenter.fms.signing.tickets.alias=admin
```

Where:

keystore_path

The name of the *.p12* keystore file created earlier (*FMSPublicKeys.p12*). If the keystore file is not in *FSC_HOME*, include the path to the file.

ekey_pass

The encrypted password for the keystore file.

Restart all processes

Restart the FSCs, Pool Manager, and all Teamcenter server processes shut down before the update using the Teamcenter Management Console.

Renew the public keys in the asymmetric keystore

You can renew your public keys manually or by using Deployment Center.

Use Deployment Center to renew the public key

Use the following steps to use Deployment Center to renew the public key in the keystore.

1. Open Deployment Center, go to the **Components** tab, and select the **FSC Keys** component.
2. Ensure **Generate New Keys** is selected under **General Settings**.
3. Ensure **Use Asymmetric Keys (advanced)** is selected under **FSC Key Options**.
4. Enter and confirm the password of the existing keystore (*.p12*) file containing the public key and the admin key for this site.

When you generate and run the Deployment Center installation scripts on the servers, the key pair and keystore file are updated with the new public key. The keystore file is also updated with a new administrative symmetric key.

Manually renew the public key

Perform the following steps from a Teamcenter command prompt to manually renew your public key. Commands requiring credentials must be run by a Teamcenter administrator with DBA privileges.

1. Use the **install_fms_keys** utility to generate a new private key in the Teamcenter database as follows:

```
install_fms_keys -u=Tc-admin-user -p=password -g=group -f=install
-overwrite
```

The private key is added to the Teamcenter database.

2. Use the **install_fms_keys** utility to export the public key corresponding to the private key in the Teamcenter database as follows:

```
install_fms_keys -u=Tc-admin-user -p=password -g=group -f=exp_pubkey
-file_password=key_pass
```

Where *key_pass* is the password for the exported keystore file.

The public key is saved as a new *.p12* keystore file.

3. Use the *keygen.bat* utility (*keygen.sh* on Linux) to add the new public key to the previously existing keystore (*FMSPublicKeys.p12*) overwriting the current public key in the keystore as follows:

```
keygen -importpubkey -keystore p12_keystore_file -storepass key_pass
-srckeystore src_key_file -srcstorepass src_key_pass -overwrite
```

Where:

p12_keystore_file

The file name of the *.p12* file holding the public key being replaced (*FMSPublicKeys.p12*).

key_pass

The password for the *.p12* file holding the public key being replaced (*FMSPublicKeys.p12*).

src_key_file

The file name of the *.p12* file holding the new public key.

src_key_pass

The password for the *.p12* file holding the new public key.

4. (Optional.) You can regenerate and import the administrative symmetric key into the *.p12* keystore file using the **-overwrite** option of the **keygen** utility. Doing so replaces the existing administrative key with a new key. For example:

```
keygen -genadminkey -keystore p12_keystore_file -storepass key_pass
-overwrite
```

Where:

p12_keystore_file

The file name of the *.p12* file holding the public key.

key_pass

The password of the *.p12* file holding the public key.

5. Deploy the new *.p12* keystore file to each of the FSCs.

Revoke key pairs

When revoking keys, the private key must be revoked from Teamcenter. The public key must be revoked from the keystore deployed with the FSC.

To deactivate only particular sites, maintain the private key in Teamcenter and only revoke the public key from the particular sites.

When working with a Multi-Site federation, revoke the public key at each site where the key is used.

Perform the following steps from a Teamcenter command prompt. Commands requiring credentials must be run by a Teamcenter administrator with DBA privileges.

Revoke the private key from Teamcenter

Use the **install_fms_keys** utility to revoke the private key from Teamcenter as follows:

```
install_fms_keys -u=Tc-admin-user -p=password -g=group -f=delete
```

Revoke the public key from the keystore deployed with the FSC

1. Use the **keygen** utility to revoke a public key from the keystore deployed with the FSC as follows:

```
keygen -deletekey -keystore keystore_name -storepass key_pass -alias  
pub_key_alias
```

Where:

keystore_name

The file name of the *.p12* keystore file created earlier.

key_pass

The password of the *.p12* file holding the public key.

pub_key_alias

The public key (stored in the *.p12* keystore file) alias. The alias is the same as *site_ID*.

If the alias name contains a dash (-) at its beginning, which is common for Teamcenter site IDs, exclude the dash when specifying the value for the **-alias** parameter. For example, if the alias is **-12345678**, specify the value as **12345678**.

2. Remove or update (as appropriate) the **ticketkeys** section of the FMS primary configuration file (*FSC_HOME\fmsmaster_fscid.xml*).

3. Remove (or update) the following line in the FMS server configuration properties file (*FMS_HOME\fsc.properties* or *FSC_HOME\fsc_fscid.properties*):

```
com.teamcenter.fms.public.keystore.epassword=epassword
```

4. Remove (or update) the following lines in the *FSC_HOME\fscadmin.properties*.

```
com.teamcenter.fms.signing.keystore.file=keystore_path
```

```
com.teamcenter.fms.signing.keystore.epassword=epassword
```

Configure Multi-Site Collaboration to use asymmetric keys

All Multi-Site Collaboration sites must use the same key type. Asymmetric keys cannot be used unless all sites use asymmetric keys. To use asymmetric keys with Multi-Site, all sites must be running Teamcenter 14.3 or later.

Each Multi-Site site has its own public-private key pair. Sites that communicate with each other must have their public keys deployed on the FSCs of both sites.

Use the following steps to configure Multi-Site sites to use asymmetric keys.

1. Export the public key from each site using the steps in *Export the public key from Teamcenter to a keystore file* in **Convert sites that use symmetric keys to use asymmetric keys**. Do this for each key needed for the sites to communicate.
2. Send the public keystore files and passwords to the other sites. Use a secure method such as encrypted email or an encrypted file transfer mechanism to send the files and passwords to the other sites.
3. At each site, use the following command from a Teamcenter command prompt to import each public key into the keystore file that is deployed with the FSC:

```
keygen -importpubkey -keystore p12_keystore_file -storepass key_pass  
-srckeystore src_key_file -srcstorepass src_key_pass
```

Where:

p12_keystore_file

The file name of the .p12 file holding the public key being replaced.

key_pass

The password for the .p12 file holding the public key being replaced.

src_key_file

The file name of the .p12 file holding the new public key.

`src_key_pass`

The password for the `.p12` file holding the new public key.

4. Update the FMS primary configuration file (`FSC_HOME/fmsmaster_fscid.xml`) to reference the alias names of each site. See the steps in *Deploy the keystore file on the FSC and update the FSC configuration* in [Convert sites that use symmetric keys to use asymmetric keys](#).

Following is an example of an updated configuration file.

```
<fmsworld>
  <multisiteimport siteid="87654321">
    <defaultfscimport fscid="fsc_ent_b" fscaddress="http://fscserverB:5556"
    priority="0"/>
  </multisiteimport>

  <fmsenterprise id="12345678">
    <ticketdigest value="SHA-256"/>

    <ticketkeys keystore="FMSPublicKeys.p12">
      <publickeyalias name="-12345678" enterpriseid="-12345678" />
      <publickeyalias name="-459292456" enterpriseid="-459292456" />
      <admindialias name="admin"/>
    </ticketkeys>

    <fscgroup id="g1">
      <fsc id="fsc_ent_a" address="http://fscserverA:5555">
        <volume id="vol1" root="D:\FMSVolumes\vol1" enterpriseid="12345678" />
      </fsc>

      <clientmap subnet="127.0.0.1" mask="0.0.0.0">
        <assignedfsc fscid="fsc_ent_a" />
      </clientmap>
    </fscgroup>
  </fmsenterprise>
</fmsworld>
```

Configuring PKI authentication for FMS

Best practices for configuring PKI authentication for FMS

You can configure public key infrastructure (PKI) authentication for FMS to authorize **fscadmin** commands. This authentication prevents offsite administrators (such as administrators at supplier sites) from performing unauthorized FSC administrative commands.

Use PKI authentication to specify which **fscadmin** commands require additional signing, allowing you to control the functionality available for specified servers and installations.

Use the following best practices for optimal security.

- Password conventions

Use strong passwords.

Do not use passwords vulnerable to dictionary attacks.

Do not use password patterns that can be easily guessed if one password is compromised.

Use different passwords for each keystore and key.

Use only encrypted passwords in property files.

Use only characters that can be reliably and repeatedly typed (or cut/pasted) into command shells; avoid characters that make this difficult.

- Keystore conventions

The keystore type must be **JCEKS**; the keystore file extension must be **.jceks**.

Use meaningful names, such as **trusted.jceks** and **supplier.jceks** and **fsc.fscid.signing.jceks**.

In each keystore, for each **keystorealias** element defined in the **fmsmaster** configuration file, place either the private key *or* the public certificate. Each private key requires that a password entry in the properties file is deployed along with the keystore.

Place only private keys in keystores you plan to deploy to trusted sites.

Consider the keystore and its associated properties file as a pair and name them accordingly, for example, **fsc.fscid.signing.jceks** and **fsc.fscid.properties**.

Siemens Digital Industries Software recommends using scripts to manage keystores, generate key pairs, export public certificates, and import the public certificates to other keystores. Keep scripts in a secure location.

- Naming conventions

Use no spaces, commas, equal signs, or colons in the names of keystore aliases.

Use no spaces or commas in the names of policy IDs.

Siemens Digital Industries Software recommends adding a system identifier to aliases, such as the system ID or site ID. Doing so ensures that over time, as signatures are passed between sites, the aliases continue to be unique and traceable to the owning site.

Restricting selected **fscadmin** commands for PKI authentication

Before selecting **fscadmin** commands for additional authentication, you must first:

- Determine which **fscadmin** commands you want to restrict.

This example requires additional authentication for the **filestoredetail** and **cachedetail** commands.

- Determine the policy names associated with the restricted commands.

This example defines a single policy (**trustedadmin**) for both commands. Policy names are arbitrary, but should be meaningful, such as **siteadmins** or **supplieradmins**.

- Determine the key/certificate aliases used to assert a policy.

This example uses **ent123.trustedadmin**. You can use different keys/certificates for each installation, though this requires significant keystore management.

The keytool used in this example is from Java JDK 1.5.

1. Create the *trusted* keystore to hold the private keys.

In this example, this is the keystore deployed to trusted servers and installations.

- a. Determine the keystore name.

In this example, the keystore name is **trusted.jceks**

- b. Determine the password used for the keystores.

In this example, the password is **trusted.jceks.lp7qZF.password**.

- c. Determine the password used for individual keys.

In this example, the password is **trustedadmin.5oDHfVV.password**.

- d. Use the keytool to create the keystore and key. For example:

```
> keytool -storetype jceks -keystore trusted.jceks -storepass trusted.jceks.lp7qZF.password
-genkey -v -keyalg RSA -alias "ent123.trustedadmin" -keypass trustedadmin.5oDHfVV.password
-validity 9999 -dname "CN=FMS trusted admin policy site ent123, OU=org unit, O=org, L=c,
ST=st, C=cc"
```

```
Generating 1,024 bit RSA key pair and self-signed certificate (MD5WithRSA)
for: CN=FMS trusted admin policy site ent123, OU=org unit, O=org, L=c, ST=st, C=cc
[Storing trusted.jceks]
```

- e. Use the keytool to export the public certificate. For example:

```
> keytool -storetype jceks -keystore trusted.jceks -storepass trusted.jceks.lp7qZF.password
-export -v -alias "ent123.trustedadmin" -keypass trustedadmin.5oDHfVV.password
-file ent123.trustedadmin.cer
```

Certificate stored in file <ent123.trustedadmin.cer>

- f. Use the keytool to list the contents of the keystore. For example:

```
> keytool -storetype jceks -keystore trusted.jceks -storepass trusted.jceks.lp7qZF.password
-list -v
```

Keystore type: jceks

Keystore provider: SunJCE

Your keystore contains 1 entry

Alias name: ent123.trustedadmin

Creation date: Jul 10, 2022

Entry type: keyEntry

Certificate chain length: 1

Certificate[1]:

Owner: CN=FMS trusted admin policy site ent123, OU=org unit, O=org, L=c, ST=st, C=cc

Issuer: CN=FMS trusted admin policy site ent123, OU=org unit, O=org, L=c, ST=st, C=cc

Serial number: 4a578037

Valid from: Fri Jul 10 13:53:59 EDT 2022 until: Mon Nov 24 12:53:59 EST 2036

Certificate fingerprints:

MD5: 66:6F:67:55:09:CA:04:69:52:76:C8:49:30:30:75:F0

SHA1: E4:74:66:DD:54:C2:0D:4B:D2:AD:74:EA:65:69:89:C7:0F:16:71:49

```
*****
*****
```

2. Create the *untrusted* keystore to contain only public certificates.

In this example, this is the keystore deployed to untrusted servers and installations (such as supplier sites).

- a. Determine the keystore name.

In this example, the keystore name is **untrusted.jceks**

- b. Determine the passwords used for the keystores.

In this example, the password is **untrusted.jceks.2TLiFD.password**.

- c. Use the keytool to create and import the public certificate. For example:

```
> keytool -storetype jceks -keystore untrusted.jceks -storepass
untrusted.jceks.2TLiFD.password
-import -v -noprompt -trustcacerts -alias "ent123.trustedadmin" -file ent123.trustedadmin.cer
```

Certificate was added to keystore
[Storing untrusted.jceks]

- d. Use the keytool to list the contents of the keystore. For example:

```
> keytool -storetype jceks -keystore untrusted.jceks -storepass
untrusted.jceks.2TLiFD.password -list -v
```

Keystore type: jceks
Keystore provider: SunJCE

Your keystore contains 1 entry

Alias name: ent123.trustedadmin
Creation date: Jul 10, 2022
Entry type: trustedCertEntry

Owner: CN=FMS trusted admin policy site ent123, OU=org unit, O=org, L=c, ST=st, C=cc
Issuer: CN=FMS trusted admin policy site ent123, OU=org unit, O=org, L=c, ST=st, C=cc
Serial number: 4a578037
Valid from: Fri Jul 10 13:53:59 EDT 2022 until: Mon Nov 24 12:53:59 EST 2036
Certificate fingerprints:
MD5: 66:6F:67:55:09:CA:04:69:52:76:C8:49:30:30:75:F0
SHA1: E4:74:66:DD:54:C2:0D:4B:D2:AD:74:EA:65:69:89:C7:0F:16:71:49

3. Create and/or modify the FSC property files.

- a. Encrypt the keystore and key/alias password values using the **passwordtool** script. For example:

```
> passwordtool -encrypt trusted.jceks.lp7qZF.password
fcLxB/oeZ+IeNnP/vofAqFpDqmJdSyaU0y+EHU0ffRc=
> passwordtool -encrypt trustedadmin.5oDHfVV.password
Bpg2TLiFDni3bT4xS4kyIaLvz5TWGLQ/GPTJN2r5T3s=
> passwordtool -encrypt untrusted.jceks.2TLiFD.password
J168VL0QG4bTbJpcIs57lqwc3P42GhTnXwfogeoVWs0=
```

- b. Add the following properties to the **fsc.fscid.properties** file for trusted installations:

```
# signing keystore file and password
com.teamcenter.fms.signing.keystore.file=trusted.jceks
com.teamcenter.fms.signing.keystore.epassword=fcLxB/oeZ+IeNnP/vofAqFpDqmJdSyaU0y+EHU0ffRc=
# key password(s) property name form: com.teamcenter.fms.signing.<alias>.epassword
com.teamcenter.fms.signing.ent123.trustedadmin.epassword=Bpg2TLiFDni3bT4xS4kyIaLvz5TWGLQ/
GPTJN2r5T3s=
```

- c. Add the following properties to the **fsc.fscid.properties** file for untrusted installations:

```
# signing keystore file and password
com.teamcenter.fms.signing.keystore.file=untrusted.jceks
com.teamcenter.fms.signing.keystore.epassword=J168VL0QG4bTbJpcIs57lqwc3P42GhTnXwfogeoVWs0=
# key password(s) property name form: com.teamcenter.fms.signing.<alias>.epassword
# none...
```

4. Create and/or modify the **fscadmin** property files by adding the following properties to the **fscadmin.properties** file for trusted installations. (No properties need be added for untrusted installations.)

```
# signing keystore file and password
com.teamcenter.fms.signing.keystore.file=trusted.jceks
com.teamcenter.fms.signing.keystore.epassword=fcLxB/oeZ+IeNnP/vofAqFpDqmJdSyaU0y+EHU0ffRc=
# default admin ticket signing alias
com.teamcenter.fms.signing.fscadmin.default.alias=ent123.trustedadmin
# key password(s) property name form: com.teamcenter.fms.signing.<alias>.epassword
com.teamcenter.fms.signing.ent123.trustedadmin.epassword=Bpg2TLiFDni3bT4xS4kyIaLvz5TWGLQ/
GPTJN2r5T3s=
```

5. Modify the **fmsmaster** configuration file.

- a. Add the following elements under the last **fmsenterprise** element in the file (or after the final **multisiteexport** element, if any exist).

In this example, the **fscadminpolicies** element maps **fscadmin** commands to policies. The **policy** element maps policies to the keystore aliases (in this case, to the keys/certificates).

```
<fmsworld>
...
<fmsenterprise id="ent123">
...
<!-- "policy" elements map a policy "id" to one or more "keystorealiases" that refer to a
PrivateKey
and/or a Certificate in the signing keystore -->
<policy id="trustedadmin" keystorealiases="ent123.trustedadmin"/>
<!-- "fscadminpolicies" map fscadmin "cmd" names to "policyids" (many to many) -->
<fscadminpolicies cmds="filestoredetail,cachedetail" policyids="trustedadmin"/>
...
```

- b. Reload the **fmsmaster** configuration file by stopping and starting the FSC service or by issuing an **fscadmin config reload** command. For example:

```
> fscadmin -s http://cii6w223:7168 ./config/reload

Initial configuration hash: 9a727fb3215fc5f9bf289cb4db0b164f
Configuration reload successful.
Final configuration hash: efed77c0315fc5f9bf289cb4db0b164f
```

The **fmsmaster** configuration file, FSC properties, and signing keystores are read each time the configuration file is reloaded.

- c. Verify the available keys/certificates. Trusted installations should have access to private keys and public certificates. Untrusted installations should only have access to public certificates. Use the **fscadmin** command to perform the verification. For example:

```
> fscadmin -s http://cii6w223:7168 ./keystoreinfo

Keystore info:
# of private keys: 1, aliases: [ent123.trustedadmin]
```

```
# of secret keys: 0, aliases: []
# of certificates: 1, aliases: [ent123.trustedadmin]
```

- d. Check in the FSC log files. For example:

```
...
INFO - 2022/07/10-18:38:55,500 UTC - cii6w223 - Keystore info:
# of private keys: 1, aliases: [ent123.trustedadmin]
# of secret keys: 0, aliases: []
# of certificates: 1, aliases: [ent123.trustedadmin]
...
```

6. Test to confirm the selected FSC commands are restricted. The FSC allows an **fscadmin** command when a required signature for *any* certificate for *any* policy associated with the **fscadmin** command is present.

If a required signature is not present, or cannot be validated, the **fscadmin** command is denied.

- a. Use a trusted **fscadmin** command in a trusted installation. For example:

```
> fscadmin -s http://cii6w223:7168 ./filestoredetail

*** volume filestores:
*** transient volume filestores:
*** accession filestores:
Filestore Details: testvolarh---sy2a---bHA, root: e:\workdir\FMSShare\FMSTestExplodedWar
\cr.txt, Len: 771999, Last modified: Mar 18 17:07 EST 2019, Last access: Jul 09 17:17 EDT
2021
\crlf.txt, Len: 776010, Last modified: Mar 18 17:06 EST 2019, Last access: Jul 09 17:17 EDT
2021
\FMS_User_Doc_Java.doc, Len: 403456, Last modified: Mar 12 09:23 EDT 2020,
Last access: Mar 19 14:25 EDT 2021
\index.html, Len: 18372, Last modified: Dec 10 15:35:22 EST 2020, Last access: Jul 09
09:09:55 EDT 2021
...
\testfiles - empty
\testvol - empty
Dirs: 270, Files: 9009, Bytes: 19719498212
```

- b. Use a trusted **fscadmin** command in an untrusted installation. For example:

```
> fscadmin -s http://cii6w223:7168 ./filestoredetail

Error, server returned status code: 400, status message:
ERROR_SIGNATURE_MISSING_1{filestoredetail}
```

After confirming the selected **fscadmin** commands are restricted, manage PKI authorization by:

- Storing a backup of your keystores and passwords in a safe location.
- Creating new keys/certificates and deploying the new keys if you suspect a private key is compromised.

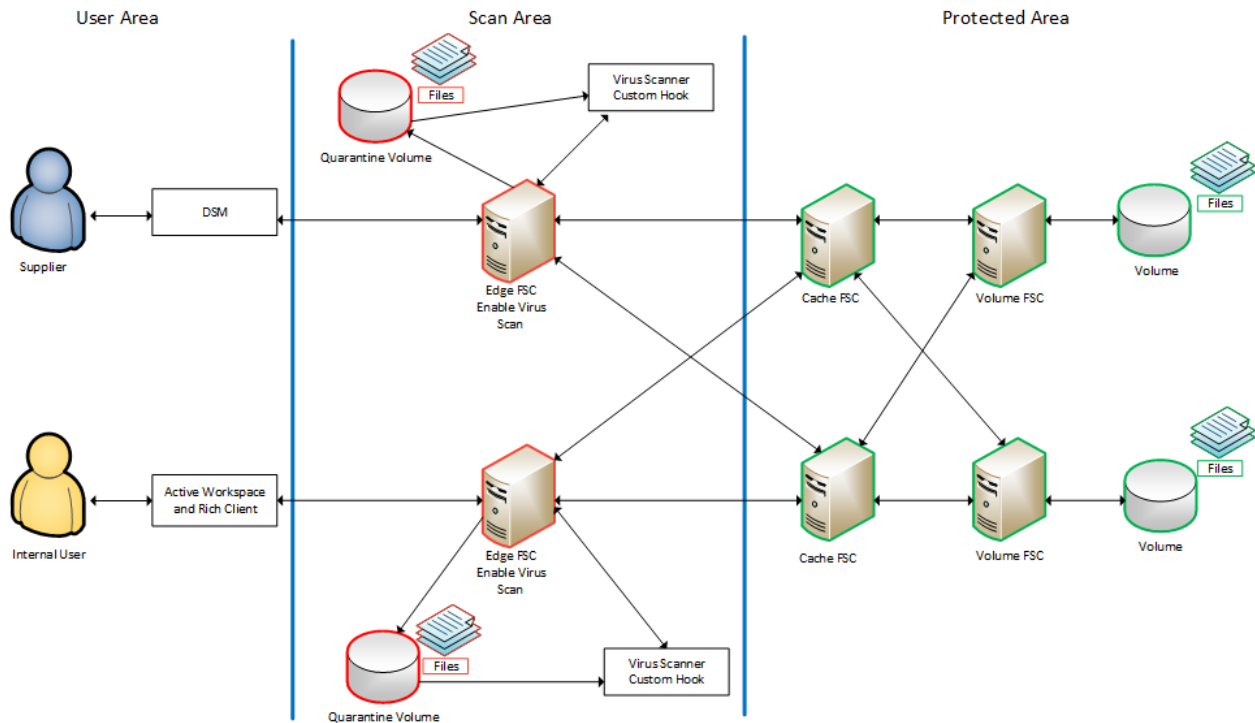
- Creating new keystores, creating new keys/certificates, using new passwords, and deploying the new keystores and property values if you suspect a password is compromised. This causes all previous keys and passwords to cease working.

Configuring file scanning during upload

File scanning process

You may allow suppliers to connect to existing File Management System (FMS) networks and to upload files into FMS volumes. As suppliers connect to your network from other companies and networks, the Teamcenter destination site loses control of the safety of uploaded files. Configuring FMS to scan files for viruses as they are being uploaded, and to reject uploads that fail the check, protects your internal system from potential viruses.

The following shows a basic topology for a virus scan configuration. (This is not a required configuration but represents a common configuration that many sites may employ.)



You can configure a quarantine volume on an FMS server cache (FSC) server to invoke a third party virus scanner on files being uploaded through it. This configuration allows uploaded files to be scanned before the file is available for download. Siemens Digital Industries Software recommends that this be a separate managed location away from FMS volumes.

When a file is routed through an FSC that has a quarantine volume defined, it is scanned for viruses. Only uploads performed after you configure the scanner and the quarantine volume can be scanned for viruses. Any file uploaded previously cannot be scanned by FMS.

1. A user starts the upload of a file using the rich client, Active Workspace, or the Data Share Manager.
2. If the file is routed through an FSC with a quarantine volume, a virus scan is initiated. The file is uploaded to the quarantine volume temporary storage location.
3. When the upload is complete, FMS runs the virus scan script file. If the script fails to operate properly, a virus scan failure is reported.
4. The file is scanned according to the individual scanner and configuration. Results are processed by the script file returned to the calling FSC.
5. If no virus is found, the file is forwarded to the destination FSC according to the routing. FMS commits the file to the destination volume and informs the client that the upload was successful.

If a virus is found, FMS returns an error to the client and displays an error to the user. The upload fails. Files that fail the scan are removed from the quarantine volume.

Scanner requirements

If you currently have a virus scanner in use in your environment, you must ensure it meets the following requirements for use with File Management System (FMS). Otherwise, you must choose and install a scanner that meets the requirements.

FMS connects to the scanner through a script or executable that you must supply. This approach lets you use the scanner of your choice. When FMS attempts to scan a file, it calls the scanning script or executable you've provided. The scanner software must meet the following requirements:

- Support for command line execution or API integration. Command line support allows the software to be called directly from FMS or directly from a custom script. API integration allows you to supply a custom executable to make the appropriate vendor calls.
- Support for unattended operation. The FMS virus scanning is done on the server without user attention. If the scanner requires user interaction, FMS cannot complete the upload. Some tools have a command line option to run without user interaction.
- Acceptable performance. A poorly performing scanner can have a significant impact on file upload times.
- The ability to target specific files or directories. The scanner must be able to accept input specifying an individual file or directory to scan. File and directory locations and names change with each call.
- Usable result output. If the FSC is calling the scanner from a script, the script is responsible for converting the scanner output to a 0 or non-0 return value. Some scanners place the results in an output file. In these cases, the calling script must parse that file to determine if the scan succeeded. Other scanners may send the same result to the console.

- Support for multiple execution runs. A single FSC can handle multiple uploads in parallel, so the FSC may be making parallel calls to the same scanner. If the scanner places its output in a single file that cannot be configured, the scan result will not be associated with a specific file. If a scanner sends its results to an output file only, you must be able to configure the name and location of the file with each call.

File virus scanning impact on upload performance

Several factors can impact performance when scanning a file for viruses. Consider these factors carefully to ensure your environment provides acceptable upload times.

Scanning an uploaded file causes a file to pass through at least one additional server, the quarantine volume, before the file can be stored on the destination volume. This process can double the upload time. The performance of your network and the scanning software is important in limiting this impact, so test the performance of the scanning software prior to deployment.

If you define multiple quarantine volumes in your File Management System network, define the routing carefully. If an uploaded file routes across multiple quarantine volumes, it is scanned multiple times resulting in a significant decrease in upload performance.

FMS quarantine volume substitutions and attributes

The FMS (File Management System) can recognize the following parameters and make substitutions for them to supply the correct values for a specific call to the virus scanner. Each time the scanner is called, these values, if they exist in the command, are repopulated before being sent to the script.

- **\$SCAN_FILE**

This is the full path to the file to scan.

Some third-party scanners accept an individual file as input. Each time a file is set to scan, it is within the **quarantinevolume** root using this parameter's value.

- **\$SCAN_DIR**

This is the full path of the directory to scan. Some scanners can scan a directory only. Use this parameter in those cases. Each time a file is set to scan, it is placed in a temporary directory within the **quarantinevolume** root using this parameter's value

- **\$SCAN_UNIQUE**

This is a randomly generated UID that can be used when forming the scanner output file name. Some scanners send their output to a temporary file that must be parsed by the scanning script. Use this parameter to generate a unique output file name. This avoids file name collisions.

The **quarantineVolume** XML element has the following required and optional attributes:

- **root** (required)

Specifies a directory where the files are temporarily placed for scanning. The virus scanner and script must have access to this location. The location cannot be automatically scanned by other virus scanners.

- **command** (required)

Specifies a command line string that is used to run the scan script. This includes the script itself and any input parameters.

Use the full path to the command. You must also supply either the file to scan or the directory containing the file to scan. Use either **\$SCAN_FILE** or **\$SCAN_DIR** substitution to supply this value.

- **timeout** (optional)

Specifies the maximum amount of time in seconds that FMS waits for a return value from the virus scanner. The default value is 20.

Configure the file virus scan

When configuring virus scanning, each assigned FSC (specified using **assignedfsc** elements in the *fmsmaster.xml* configuration file) and each FMS bootstrap FSC (specified using the **Fms_Bootstrap_Urls** preference) should have a quarantine volume. A common configuration is for assigned FSCs and FMS bootstrap FSCs to be the same set of servers.

When configuring assigned FSCs in the *fmsmaster.xml* configuration file, assigned FSCs should be placed in a different FSC Group (**fscgroup** element) than the protected area FSC servers. Doing so prevents features such as direct FSC routing from allowing files to bypass the assigned FSC scanning.

1. Create a new File Management System (FMS) server cache (FSC) where the scanning is done or configure an existing FSC. For better performance, you can install and configure the virus scanner on the same host. The scanner software and quarantine volume must be accessible by the FSC.
2. (Optional) Configure the FSC as an entry FSC.
3. If you create a new entry FSC, update the **FMSBootstrapUrls** preference at supplier sites to include the new entry FSC URL.
4. If you are not using an existing virus scanner, install the selected third-party virus scanner.
5. Add a **quarantinevolume** element to the FMS primary configuration file for the scan FSC. This specifies the location of the temporary directory where the scanning is done. Exempt this location from automatic scanning by any other virus scanner.

Note:

All FMS volumes should be exempted from automatic virus scans.

6. Ensure the virus scanner is installed and configured to be accessible to FMS.
7. Write a script or executable to call the scanner. The contents of the calling script or executable depends on the scanner interface. The script or executable acts as an intermediary between FMS and the virus scanner. It must handle the inputs and process the output to determine if the scan was a success or not. The script or executable must:
 - Remove any temporary files it creates.
 - Return a **0** (zero) value if the file passes the scan or a non **0** value if it fails.
 - (Optional) Remove the files or directory that it scans. This is not required.

The following example is a simple script for calling the freeware ClamAV virus scanner:

```

    A simple example of a scanner script that
    talks to the ClamAV freeware 3rd party scanner.
@echo off
@REM This is a batch file that will scan for viruses

IF "%1" ==
    "" (EXIT /B 1)
IF "%2" ==
    "" (EXIT /B 2)

"C:\Program
    Files\ClamAV-x64\clamscan.exe" -r %1 >> %2 2>&1

findstr
    /c:"Scanned files: 1" %2 >nul
IF NOT
    "%ERRORLEVEL%" == "0" (
del %2 >nul
EXIT /B 3
)

findstr
    /c:"Infected files: 0" %2 >nul
IF NOT
    "%ERRORLEVEL%" == "0" (
del %2 >nul
EXIT /B 4
)

```

```
del %2 >nul
```

```
EXIT /B 0
ENDLOCAL
```

This example takes two input values: the directory to scan and the path to a temporary output file. The third-party scanner is called directly and the output is directed to a temporary file. The temporary file is scanned for keywords indicating a scan success. The temporary file is deleted, and 0 is returned.

8. Create the quarantine volume root directory.

The following example shows a sample **quarantinevolume** volume entry added to the **FSC** element.

```
<fsc id="myscanfsc" address="http://127.0.0.1:4445">
  <volume id="myvol" enterpriseid="fms.teamcenter.com"
  root="D:\apps\volumes\myvol"/>
  <quarantinevolume root="E:\scanvolume" command="$HOME/scanner/
  scan.bat $SCAN_FILE $HOME/scanner/temp/$SCAN_UNIQUE.txt" timeout="20"/>
</fsc>
```

Where:

root

Specifies the location where the file is quarantined until it is scanned. Once the file is determined to be free of threats, the file is moved to the main volume in the FSC.

command

Specifies the script invoked by the FSC to have the virus scanner evaluate the file.

\$SCAN_FILE

Represents the file to be scanned. **\$SCAN_FILE** is automatically set to the name of file currently being uploaded to FMS.

\$SCAN_UNIQUE.txt

The file containing the scanning results for each file evaluated.

timeout

Specifies the number of seconds the FSC waits to receive the result of the scan. (20 seconds in this example.) If a result is not received within this time period, the FSC fails the operation.

Configuring enhanced ticket authentication

FMS permits file data access only when the requester presents a valid security ticket. You can enhance this security by configuring FMS to use Teamcenter Security Services (TCSS) to authenticate tickets only when the requester presenting the valid security ticket is the user who generated the ticket.

Configure enhanced ticket authentication using one of the following scenarios.

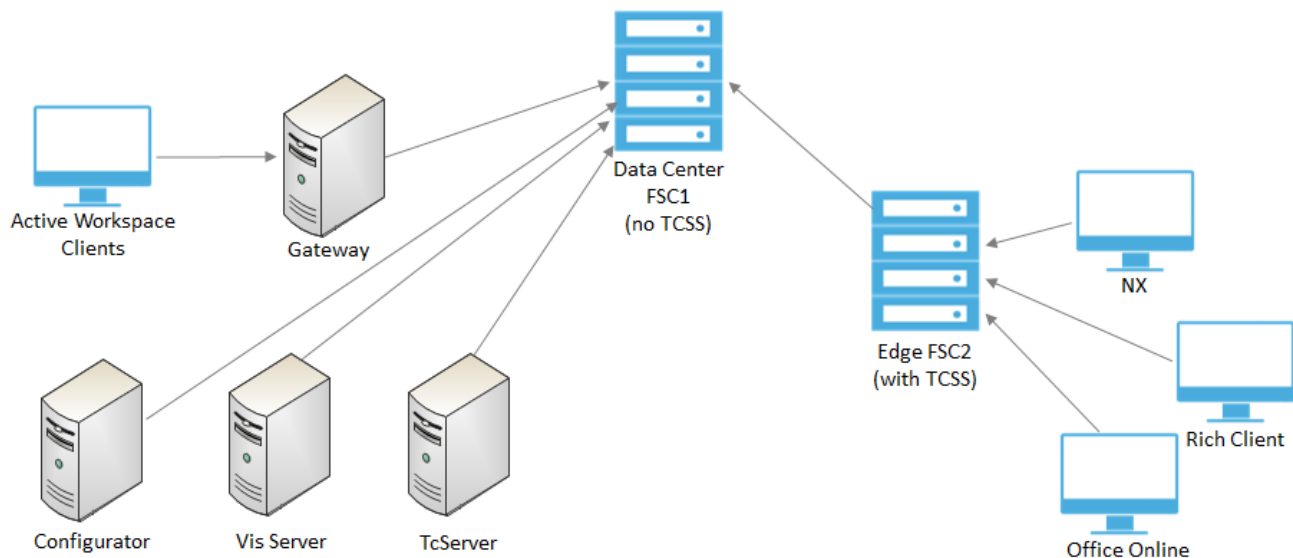
- Configure an FSC server to act as an edge server with which TCSS clients interact.
- Configure a single FSC to handle both authenticated and non-authenticated ticket access in addition to uploading and downloading files to other FSCs and FCCs.

Configure an FSC server as an edge server

This approach designates an FSC server with TCSS installed as an edge server with which TCSS clients interact. The FSC edge server authenticates each ticket, verifying that the requester presenting the valid security ticket is the user who generated the ticket, before passing it to a data center FSC.

Non-interactive clients (for example, Visualization Server and Teamcenter Server) not using TCSS are mapped to FCS servers not configured with TCSS authentication. Active Workspace does not use TCSS, so is also mapped to FCS servers not using TCSS authentication.

Example of an enhanced ticket authentication configuration:



Configure enhanced ticket authentication as follows:

Verify TCSS configuration

Perform the following steps:

- Ensure TCSS and Security Services (SSO) are installed and configured.
- Ensure that you can log on to the rich client. Verify you logged on using SSO.
- Record the following values for later use. See [Find Security Services settings for use in TCCS](#).
 - **SSO AppID**
 - SSO Identity Service URL
 - Login Service URL

Configure TCCS on the edge FSC

Enable ticket authentication for all requests on the edge FSC by ensuring that the following lines are included (uncommented) in the edge FMS server configuration file (**fsc.xml**). (Here *SOOappId*, *identity_service_url*, and *login_service_url* are the values you recorded when verifying your TCCS configuration.)

```
<fscdefaults>
...
<!-- ticket validation parameters -->
<property name="FSC_TcSSAppId" value="SOOappId" overridable="true" />
<property name="FSC_TcSSIdentityServiceUrl" value="identity_service_url"
  overridable="true" />
<property name="FSC_TcSSLoginServiceUrl" value="login_service_url"
  overridable="true" />
</fscdefaults>

<fccdefaults>
...
<!-- ticket validation parameters -->
<property name="FCC_TcSSAppId" value="SOOappId" overridable="false" />
<property name="FCC_TcSSLoginServiceUrl" value="login_service_url"
  overridable="false" />
</fccdefaults>
```

Example:

```
<fscdefaults>
...
<!-- ticket validation parameters -->
<property name="FSC_TcSSAppId" value="Teamcenter1" overridable="true" />
<property name="FSC_TcSSIdentityServiceUrl" value="identity_service_url"
  overridable="true" />
<property name="FSC_TcSSLoginServiceUrl" value="login_service_url"
  overridable="true" />
</fscdefaults>

<fccdefaults>
...
<!-- ticket validation parameters -->
<property name="FCC_TcSSAppId" value="SOOappId" overridable="false" />
<property name="FCC_TcSSLoginServiceUrl" value="login_service_url"
  overridable="false" />
```

```
<property name="FSC_TcSSSessionCookieTimeout" value="180" overridable="false"/>
</fccdefaults>
```

By default, the session cookie on the FSC is valid for 1800 seconds (30 minutes) when SSO is configured. After this time, a new cookie is created on the next request sent to the FSC. Use the **FSC_TcSSSessionCookieTimeout** parameter as shown to adjust this timeout value (in seconds) if required.

Configure the FMS primary

For each rich client and other interactive client assigned to use the edge FSC, configure the **clientmap** element of the FMS primary configuration file (*fmsmaster_fscid.xml*) to use the edge FSC as the assigned FSC. (See [Subnet/mask attributes in a client map.](#))

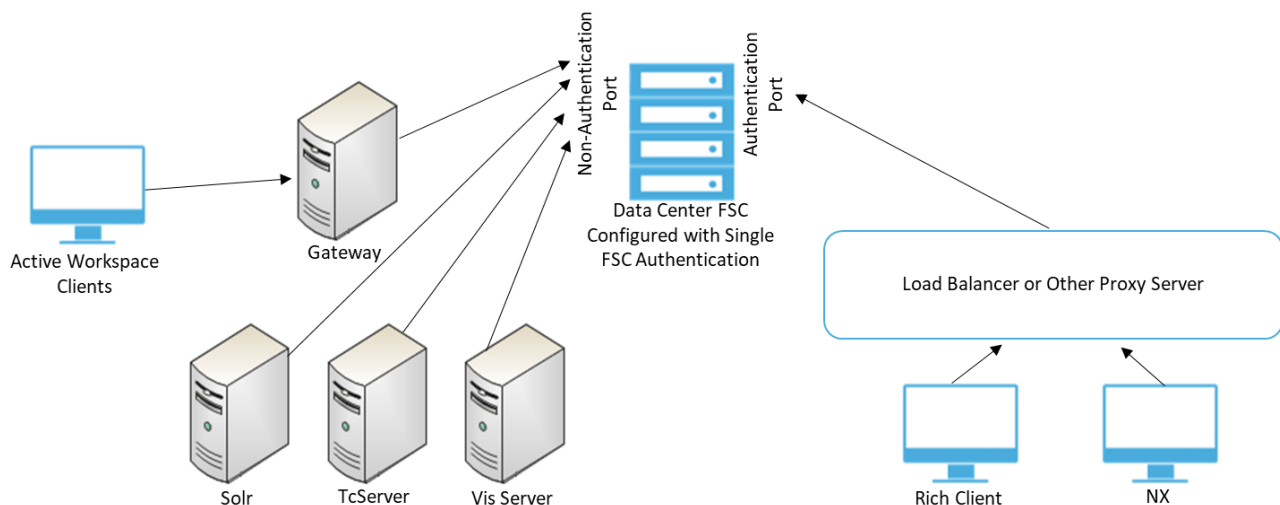
Example:

```
<clientmap subnet="100.194.53.0" mask="255.255.255.000">
  <assignedfsc fscid="fsc_edge1" priority="0"/>
</clientmap>
<clientmap subnet="100.214.36.0" mask="255.255.252.000">
  <assignedfsc fscid="fsc_edge1" priority="0"/>
</clientmap>
```

Configure a single FSC for all ticket authentication

You can configure a single FSC to handle both authenticated and non-authenticated ticket access in addition to its responsibilities of uploading and downloading files to other FSCs and FCCs. With an FSC configured for single FSC authentication, the FSC authenticates the TCSS clients requiring ticket authentication and the non-interactive clients (such as Visualization (Vis) Server and Teamcenter Server (TcServer)) not using TCSS authentication. Using this single FSC authentication simplifies your system configuration and processes requests more quickly than configurations using a secondary server for authentication.

Example of a single FSC ticket authentication configuration:



Configure an FSC for single FSC authentication as follows.

- Ensure TCSS and Security Services (SSO) are installed and configured.
- Ensure that you can log on to the rich client. Verify you logged on using SSO.
- Open Deployment Center and perform the following steps.
 1. Go to the **Components** task, and select the **FSC** component.
 2. Check **Enable Enhanced Ticket Authentication?. Enable Single Authentication FSC?** is displayed. Check **Enable Single Authentication FSC?**.
 3. Confirm that **Authenticated Port** is set to the value of an open port on the FSC. The default port value is **4545**.

For reference, **FSC Authenticated Url** provides the full URL to the authenticated port on the FSC.

4. Generate and run the Deployment Center installation script.

Configuring native FSC client proxy in TcServer

The native implementation uses cURL and OpenSSL and requires a compatible trusted certificate file. This trusted certificate file must be in privacy enhanced mail (PEM) format (which is not the same format as the Java **cacerts** file).

Use one of the following methods to generate a **cacerts.pem** file that contains the required certificates.

- Download the trusted certificate file.
- Generate a trusted certificate file based on the installed Java **cacerts** file.
- Generate a trusted certificate file with only the certificates required by your CA.

Download the trusted certificate file

If your certificate authority (CA) signer is well known, the certificates may be available from the Internet. Siemens Digital Industries Software recommends you contact your internal security team.

1. Download a current **ca-bundle.crt** file from the Internet, or get the file from your internal security team. The file must be compatible with cURL and OpenSSL.

The file must contain the certificate chain that can validate the FSC certificate.

2. Name the file **cacerts.pem** and save it in the *FSC_HOME* directory.

3. Create or modify the **FSC_HOME/fsc.clientagent.properties** file with the **com.teamcenter.fms.curl.cacerts.file** property with a value that is the absolute path and file name to the **cacerts.pem** file. For example:

```
d:\path\to\tc\install\fsc\fsc.clientagent.properties
#
#
com.teamcenter.fms.curl.cacerts.file=d:\path\to\tc\install\fsc\cacerts.pem
```

Be aware that any other host that requires FMS access (for example when running **tcserver** or a utility such as **review_volumes**) must also have a local copy of the *cacerts.pem* file specified in its *fsc.clientagent.properties* file.

Generate a trusted certificate file based on the installed Java cacerts file

If your certificate authority (CA) signer certificates are in the Java **cacerts** file, you can extract them for cURL and OpenSSL.

1. Extract the trusted certificates in the Java **cacerts** file in the PEM format that cURL and OpenSSL requires, for example (Windows):

```
rem cd to the dir the java cacerts file is in
cd /d %JAVA_HOME%\jre\lib\security
rem cleanup leftover files from this script, and any existing cacerts.pem as we are building
a new one
del /q cacerts.pem cacerts.list export.pem
rem get the aliases for all trusted certificates in the cacerts file
keytool -keystore cacerts -storepass changeit -list | find "trustedCertEntry" | sort >
cacerts.list
rem for each alias export the certificate and append to the cacerts.pem file
for /f "delims=, tokens=1" %f in (cacerts.list) do keytool -export -rfc -alias %f -keystore
cacerts
-storepass changeit -file export.pem & echo %f >> cacerts.pem & type export.pem >>
cacerts.pem
rem cleanup leftover files from this script
del /q cacerts.list export.pem
rem cacerts.pem contains all the trusted certificates in pem format
```

2. Save the file in the *FSC_HOME* directory.
3. Create or modify the **FSC_HOME/fsc.clientagent.properties** file with the **com.teamcenter.fms.curl.cacerts.file** property with a value that is the absolute path and file name to the **cacerts.pem** file. For example:

```
d:\path\to\tc\install\fsc\fsc.clientagent.properties
#
#
com.teamcenter.fms.curl.cacerts.file=d:\path\to\tc\install\fsc\cacerts.pem
```


Generate a trusted certificate file with only the certificates required by your CA

If your CA used new certificates, you must also acquire the signer certificates from your CA. These must be contained within the **cacerts.pem** file.

1. Acquire the signer certificates, in PEM format, from your CA.
2. Append the signer certificates to a **cacerts.pem** file.
3. Save the file in the *FSC_HOME* directory.
4. Create or modify the **FSC_HOME/fsc.clientagent.properties** file with the **com.teamcenter.fms.curl.cacerts.file** property with a value that is the absolute path and file name to the **cacerts.pem** file. For example:

```
d:\path\to\tc\install\fsc\fsc.clientagent.properties
#
#
com.teamcenter.fms.curl.cacerts.file=d:\path\to\tc\install\fsc\cacerts.pem
```

The client (native implementation in **TcServer**) is now able to communicate with the FSC.

Enabling and tuning WAN acceleration

Enable WAN acceleration across WAN assets to improve file transfers between FSCs and file download performance to FMS clients. WAN acceleration splits larger files into smaller pieces, transfers file pieces simultaneously, and reconstructs the files before passing the files to the client. WAN acceleration is not enabled by default.

Enable WAN acceleration by setting the FSC and FCC transport modes as follows:

Configure FMS Server Cache Transport mode

Enable WAN acceleration between FSCs:

- In the FMS primary configuration file (**fmsmaster_fscid.xml**), ensure that the FSCs are defined in separate **fscgroup** structures. For example:

```
<fscgroup id="myGroup">
  <fsc id="fsc_SAMPLE_DB"...>...</fsc>
</fscgroup>
<fscgroup id="cacheGroup">
  <fsc id="fsc_CACHE_DB".../>...
</fscgroup>
```

- In the FMS primary configuration file, update or add **linkparameter** elements with **transport="wan"** to specify that WAN acceleration is used between the groups. For example:

```
<linkparameters fromgroup="myGroup" togroup="cacheGroup"
transport="wan">
<linkparameters fromgroup="cacheGroup" togroup="myGroup"
transport="wan">
```

- Optionally adjust the **FSC_WANThreshold**, **FSC_WANChunkSize**, **FSC_DoNotCompressExtensions**, and **Maxpipes** WAN acceleration parameters for your site's operation as described in **FSC WAN acceleration tuning parameters**.
- The **FSC_MaximumWANDownloadSockets** environment variable controls the number of sockets used for WAN acceleration downloads. The default value of **255** allows for a maximum of 25 concurrent WAN downloads (10 sockets per download) to be running at one time. **255** is also the default value of the **FSC_MaximumThreads** parameter specified in the FMS server configuration file.

If you adjust the value of **FSC_MaximumWANDownloadSockets** to a value greater than the value set for **FSC_MaximumThreads**, set **FSC_MaximumThreads** to the same value. If you set **FSC_MaximumWANDownloadSockets** to a value greater than that of **FSC_MaximumThreads**, **FSC_MaximumWANDownloadSockets** is automatically reduced to the same value as **FSC_MaximumThreads**.

Configuring the FMS Client Cache Transport mode

Enable WAN acceleration between FSCs and FCCs:

- In the FMS client configuration file (**FMS_HOME/fcc.xml**), locate the parent FSC to which the FCC connects and set **transport="wan"**. For example

```
<parentfsc address="civ6s437:22470" priority="0" transport="wan"/>
```

Optionally adjust the **FCC_WANThreshold**, **FCC_MaxWANSources**, and **FCC_WANChunkSize** WAN acceleration parameters for your site's operation as described in **General FCC configuration parameters**.

- In the FMS primary configuration file (**fmsmaster/fscid.xml**), ensure that the parent FSC is included in the client map settings and set **transport="wan"** for that FSC. For example:

```
<clientmap subnet="127.0.0.1" mask="0.0.0.0">
  <assignedfsc fscid="fsc_cache_DB" priority="0" transport="wan"/>
</clientmap>
```

- Optionally adjust the **FSC_WANThreshold**, **FSC_WANChunkSize**, **FSC_DoNotCompressExtensions**, and **Maxpipes** WAN acceleration parameters for your site's operation as described in **FSC WAN acceleration tuning parameters**.

Configure Cloud Data Storage Service (DSS)

In addition to standard network storage of volumes, File Management System supports S3 cloud volumes through the Cloud Data Storage Service (DSS). DSS lets you create volumes that are stored on Amazon S3 cloud-based servers. Configuration and use of S3 cloud storage is currently supported with Siemens Managed Services accounts only.

Siemens Managed Services has partnered with industry leaders to deliver world-class cloud solutions. These partners include Smartronix Inc. for system monitoring and administration of the delivery of Amazon web services (AWS) for cloud infrastructure. The partnering of Smartronix and AWS with the Siemens Professional Services and Cloud Services teams allows for the delivery of complete turnkey cloud deployments tailored to customer business needs. These partners form a broad solution team with Siemens Managed Services acting as your single point of contact for cloud-related solutions.

Configuring Cloud Data Storage Service during FMS installation

Chose to enable DSS when installing FMS as follows:

- When installing with Deployment Center, select the FSC component, check **Enable Cloud Data Storage Service (DSS)**.
- When installing using TEM, on the **File System Cache Service (FSC)** panel, click **Advanced**, open the **Cloud DSS** tab, and check **Enable Cloud Data Storage Service (DSS)**.

Supply the following details when using either TEM or Deployment Center:

Setting	Description
Keystore Password	A password you specify to protect the keystore file containing the Cloud Data Storage Service credentials.
Account ID	The ID of the Cloud Data Storage Service account that is to store Teamcenter volumes.
User ID	The user ID for the Cloud Data Storage Service account that stores Teamcenter volumes.
Access Key ID	The ID of the key used to access Teamcenter volumes stored in the Cloud Data Storage Service account.
Secret Access Key	The secret string of the key used to access Teamcenter volumes stored in Cloud Data Storage Service account.

Optimizing FMS startup memory

When enabling cloud volumes, Siemens Digital Industries Software recommends you increase the FMS and FCC startup memory values from their default values of 256MB to at least 1GB. To increase the values, edit the values of **FSC_MEM**, **TCCS_MEM**, and **TCCS_MEM_MAX** in the file for your platform as follows:

startfsc.bat (Windows)

```
set FSC_MEM=1024M

set TCCS_MEM=1024M

set FSC_MEM_MAX=1024M
```

startfsc.sh (Linux)

```
setenv FSC_MEM=1024M

setenv TCCS_MEM=1024M

setenv TCCS_MEM_MAX=1024M
```

Creating cloud volumes

A cloud volume named **DefaultCloudVolume** is created when Cloud Data Storage Service is installed. If you want to use this cloud volume, use the Organization application to grant member access to the volume and to otherwise manage the volume.

You can also create new cloud volumes as described in [Create a volume](#).

Manage DSS keys

Rotate (change) your DSS keys to limit exposure in case of a security compromise and to limit the amount of information protected by any particular key. Use the following steps to rotate DSS keys.

1. Ensure cloud services are enabled and your environment is configured as specified in *Manually configure the Teamcenter environment*. and your files are configured as specified in *Log files produced by Teamcenter*.
2. Open a Teamcenter command shell:
 - (Windows) Click **Start** and choose **All Programs**→**Teamcenter x.x**→*configuration-name* **Command Prompt**.
 - (Linux) Open a command shell, change to the *TC_ROOT/tc_menu* directory, and run the *configuration-name.sh* script.

- Use the following **fscadmin** command to rotate the DSS keys:

```
fscadmin -s FSC-URL ./external/keys/rotate/nvars/storagetype="dss"
```

For example:

```
fscadmin -s http://localhost:4544 ./external/keys/rotate/nvars/storagetype="dss"
```

Messages of the following form are returned:

```
DSS Keys rotated successfully and key store updated
ACCESS_KEY_ID=e83a4c2f98ff4290a098774dcc24258a
SECRET_ACCESS_KEY=9lCCIQ+wRF/evDKCFgKN8PU1T0+md1lpvAHxpn+Yesc=
```

- Use the following **fscadmin** command to verify FMS access for your Teamcenter site:

```
fscadmin -s FSC-URL -k common-key-file ./status
```

For example:

```
fscadmin -s http://localhost:4544 -k FMS.key ./status
```

Resolving ticket expiration errors

Ticket expiration errors (such as **TICKET_EXPIRED_0**) are noted in the FSC log or can be displayed in other parts of the system. These errors can be caused by time zone configuration issues or by poor clock synchronization. This can hinder administrative FMS tasks, such as creating volumes, or, in severe cases, cause file access issues. Resolve ticket expiration errors by verifying proper time zone configuration, and then synchronize your local clock with local or public time servers.

If your company has local time servers, synchronize with them based on your company's policies. To synchronize with public time servers:

- To synchronize your local clock on Windows, run the following operating system command:

```
net time /setsntp:pool.ntp.org
```

- To synchronize your local clock on Linux, run the following operating system command:

```
ntpdate -u pool.ntp.org
```

Configure FMS to work with multiple versions of Java

If you are running multiple versions of Teamcenter on your system and they are compatible only with different versions of Java, you must configure your FMS client caches (FCCs) to use the Java run-time environments (JREs) with which the caches were installed.

1. Open the **FMS_HOME/starttccs.sh** (Linux systems) file or the **FMS_HOME\starttccs.bat** (Windows systems) file in a plain text editor.
2. Set the **TCCS_JAVA** environment variable to the JRE supplied with the Teamcenter version with which the FCC was installed.

For information about versions of operating systems, third-party software, Teamcenter software, and system hardware certified for your platform, see the Hardware and Software Certifications knowledge base article on Support Center.

Configure FMS multiuser support

On Windows systems, the pipe connection to the FMS client cache (FCC) uses a single value by default, allowing only one user to connect to the FCC per Windows workstation. If you want to enable multiple users to start FCC client applications on the same machine at the same time, set the **FMS_WINDOWS_MULTIUSER** environment variable to **true**.

A multiuser situation occurs, for example, when multiple users employ remote access applications such as Citrix or Remote Desktop to access Teamcenter applications on a common Windows machine. FCC separation is achieved because the environment variable setting causes the pipe names used for the client-to-FCC communication to include the user ID, ensuring that each user connects with a unique pipe to the appropriate FCC.

If the **FMS_WINDOWS_MULTIUSER** variable is set to **true**, user names should not end in numerals. Names ending in numerals confuse the automated pipe name generation. For example, the pipe named **{RootPipeName}_User121** could be the first data pipe belonging to **User12**, or the 21st pipe belonging to **User1**.

Use the following steps to configure FMS running on a Windows system to accept multiple users:

1. In the system environment (**System Properties>Advanced>Environment Variables>System Variables**), set the **FMS_WINDOWS_MULTIUSER** environment variable to **true**.

To set a global environment variable, local administration privileges are required.

2. To propagate this setting to all users and applications, reboot the system.

Siemens Digital Industries Software customer support will assist in working through issues. Any issues that are Siemens Digital Industries Software code related can be addressed as problem reports. Issues arising from operating system or third party tools must be addressed with that vendor.

FMS considerations for running multiple Teamcenter environments

If for a testing or development purpose you have created multiple Teamcenter environments, either on a single machine or multiple machines, such that there are multiple Teamcenter sites with the same site ID that also have the same volume ID for each of their volumes, then the sites look the same to FMS

because they have the same site ID. This is different from regular Teamcenter deployments, which are typically single instance installations on a given server infrastructure.

In this testing or development scenario, it is not necessary to move volumes to distinct volume IDs. You can maintain the integrity of the data as long as you keep the sites apart.

To keep the sites apart, comply with the following:

- Do not use Multi-Site. There can be only one primary site. If you use Multi-Site in this scenario, then there is a risk of replicating owned data.
- Do not share FSCs. There is a risk of checking in data to the wrong system.

Warning:

Production environments must not have duplicate site IDs.

Administering transient volumes

Introduction to administering transient volumes

Transient volumes are the mechanism used to transfer data either generated by or required by the business logic server (**TcServer**). For example, transient volumes are used during PLM XML export. The file is generated by the **TcServer** process and written to a temporary area, the transient volume. A ticket for the file in the transient volume is sent to the client that uses the ticket to retrieve the file.

In a four-tier configuration, each business logic or enterprise tier server requires a transient volume. It is best practice to have a local FSC on each server to service the local transient volume.

Client access to files within a transient volume is dependent on whether clients are running in two-tier or four-tier. The tickets generated for client file access specify either two-tier, or four-tier depending on how the client is connected to the **TcServer**.

- Two-tier transient volume file access

Clients in a two-tier environment can read and write to the same physical location as the **TcServer** since they are on the same machine. In this case, the FCC process reads files directly from the transient volume location.

File operations need no special **fmsmaster_fscid.xml** configuration because no FSCs are required to transport the files, and therefore no **fmsmaster_fscid.xml** configuration is required.

- Four-tier transient volume file access

Clients in a four-tier environment do not have direct access to the transient volume location and must use the FMS system to retrieve files. Therefore, four-tier transient file operations require transient

volumes to be configured in the **fmsmaster_fscid.xml** configuration file. Transient volumes are declared under an FSC with direct access to the files, much like the **TcServer** process. The FSCs that host transient volumes usually run on the same hosts, under the same user IDs, as the **TcServer** process.

Note:

The four-tier FSC transient volume server must run with the same user credentials as the pool server and **TcServer** processes that own the transient volume and all its files. The OS user that FSC and **TcServer** run with requires read, create, write, and delete permissions in all directories and subdirectories within the transient volume. No other users should have access to this directory.

On the server side, two-tier or four-tier configuration makes no difference to how the **TcServer** process assesses the transient volume.

Transient volume configuration components

In a two-tier environment the value used for the **Transient_Volume_RootDir** preference is always overridden by a user-specific directory; the values set for any transient volume-related preferences or environment variables are ignored.

Caution:

Do not define the directory path as a UNC path, for example, `\\server\shared-transient-folder`. Use a direct path location. This requirement is because some ZIP archive utilities do not accept UNC paths, resulting in failed exports to Microsoft Excel or Word.

If the **TC_TMP_DIR** environment variable is set, the value used for the **Transient_Volume_RootDir** preference is:

- **\${TC_TMP_DIR}/\${USER}2TierTransientVolume** on Linux
- **TC_TMP_DIR\%USERNAME%2TierTransientVolume** on Windows

If the **TC_TMP_DIR** environment variable is not set, the value used for the **Transient_Volume_RootDir** preference is:

- **/tmp/\${USER}2TierTransientVolume** on Linux
- **c:\temp\%USERNAME%2TierTransientVolume** on Windows

In a four-tier environment, three elements control transient volumes:

- The site ID defines the database ID for the system.

- The **Transient_Volume_RootDir** preference is a multi-value variable used to provide Linux and Windows path name of the transient volume on the server. When set as a preference, the values must be valid for all machines of each platform type. When set as an environment variable (which overrides any preference values) only a single platform path can be specified.

The preference is designed to accept multi-values to support multiple platforms. However, if you set this preference with identical values, the following error message appears:

```
Duplicate transientvolume configuration elements were found for attribute id
with value attribute_value
```

If you are using only one platform, you can delete the other value to ensure you do not receive this error message.

- The **Transient_Volume_Installation_Location** preference specifies the node name of the transient volume. It is a location-based logical identifier and is generally set to the host name of the local machine by the `TC_DATA\tc_profilevars` script.

The transient volume location is read/write/verify checked during **TcServer** startup. The location value and any warning messages about the transient volumes are printed to the **TcServer** system log for diagnostic purposes.

Configure transient volume elements in the primary configuration file

For four-tier client file access, transient volumes must be declared within the **fmsmaster_fscid.xml** configuration file. They must be declared with the FSCs that host the transient volumes (these are the FSCs capable of file input/output directly to the root path of the transient volume).

The declaration must include the transient volume ID as defined by the **backup_xmlinfo** utility, and the path to the root of the transient volume. For example:

```
<fsc id="FSC_cinsun4_v10002a1" address="http://cinsun4:44021">
  <transientvolume id="f4986a4ab9b155d11da1afceb35f55ac" root="/m/v10002/
  transientVolume_v10002a" />
</fsc>
```

Any change to the **fmsmaster_fscid.xml** configuration file requires all FSCs to be configured as configuration primaries to have their configuration files updated. Then the configuration must be reloaded across the FMS network to propagate the changes to all FSCs.

To modify the transient volume ID or path in the primary configuration file:

1. Run the **backup_xmlinfo** utility and examine the **backup.xml** file created by the utility.
2. Edit the FMS primary configuration file(s) to reflect the new ID of the transient volume(s) and/or the changed paths.

3. Either reload the configuration by stopping and restarting the FSCs that use this configuration information, or reload the configuration using the **fscadmin** utility.

Reload the configuration across the entire FMS network by running the following command:

```
fscadmin -s http://fsc host:fscport ./config/reload/all
```

Modify the transient volume ID for the current server context

Although a given **TcServer** only knows about a single logical transient volume, other parts of the system (such as FMS) may need to work with more than one transient volume and therefore transient volumes must be identifiable. A transient volume's identity is a unique value that is based on the site ID and several of the transient volume preferences. This identity is called the transient volume ID.

The transient volume ID listed in the FMS primary configuration file is determined by a combination of the site ID, the value of the **Transient_Volume_RootDir** preference, and the value of the **Transient_Volume_Installation_Location** preference. Modifying any of these values changes the transient volume ID, in which case you must manually modify the transient volume ID listed in the primary configuration file.

Note:

The `TC_DATA\tc_profilevars` file sets the **Transient_Volume_Installation_Location** preference to the computer/host name.

You can obtain a list of current volume definitions in the database by running the **backup_xmlinfo** utility to generate the **backup.xml** file. The utility generates transient volume information for the context in which the utility is being run (the current **TcServer** context). All other server pools or hosts with transient volumes are not identified.

The procedure for setting the **TcServer** context differs, depending on whether you are on a Windows and Linux platform.

- To set the **TcServer** context on Windows and run the utility:
 1. Run the Teamcenter command prompt to open a command window by choosing **Start→Programs→Teamcenter version→configuration-name Command Prompt**.
 2. In the command prompt, type **backup_xmlinfo -u=tc-admin-user -p=password -g=group** where *tc-admin-user*, *password*, and *group* are your site's administrative user ID, password and group ID.

The **backup.xml** file is generated, listing the transient volumes.

- To set the **TcServer** context on Linux and run the utility:
 1. Set the **TC_DATA** environment variable to your *TC_DATA* directory.

2. Set the **TC_ROOT** environment variable to your **TC_ROOT** directory.
3. Run **source tc_profilevars**.
4. Run **backup_xmlinfo -u=tc-admin-user -p=password -g=group** where *tc-admin-user*, *password*, and *group* are your site's administrative user ID, password and group ID.

The **backup.xml** file is generated, listing the transient volumes. For example:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<backupInfo>
  <enterpriseId>468532367</enterpriseId>
  <bootstrapInfo>
    <fscUrl>http://cili6s08:7131</fscUrl>
  </bootstrapInfo>
  <volumeInfo>
    <volumeName>public_vol</volumeName>
    <volumeUid>1b604728a74d1bed3c8f</volumeUid>
    <nodeName>cili6s08</nodeName>
    <unixPath>/tc_volumes/tc200712</unixPath>
  </volumeInfo>
  <transientVolumeInfo>
    <transVolId>875acf876dfccbc2dc76e9bf1e807d85</transVolId>
    <unixPath>/tmp/transientvolume</unixPath>
  </transientVolumeInfo>
  <transientVolumeInfo>
    <transVolId>8900c6b23daca8dcf231dbbf4331b703</transVolId>
    <wntPath>c:\temp</wntPath>
  </transientVolumeInfo>
</backupInfo>
```

Note:

The transient volume ID and path are platform dependent. Use the ID from either the **unixPath** or the **wntPath**.

Define the transient volume within FSC elements in the **fmsmaster_fscid.xml** configuration file.

Determining the transient volume ID for a different server context

Although a given **TcServer** only knows about a single logical transient volume, other parts of the system (such as FMS) may need to work with more than one transient volume and therefore transient volumes must be identifiable. A transient volume's identity is a unique value that is based on the **SiteID** and several of the transient volume preferences. This identity is called the *transient volume ID*.

The transient volume ID listed in the FMS primary configuration file is determined by a combination of the site ID, the value of the **Transient_Volume_RootDir** preference, and the value of the **Transient_Volume_Installation_Location** preference. Modifying any of these values changes the transient volume ID, in which case you must manually modify the transient volume ID listed in the primary configuration file.

You can obtain a list of current volume definitions in the database by running the **backup_xmlinfo** utility to generate the **backup.xml** file. You must define which server context for which you are generating transient volume information.

To generate transient volume information for other **TcServers** within the same system, set the **Transient_Volume_Installation_Location** preference to the value used for the desired **TcServer**, then complete the steps listed in *Modify the transient volume ID for the current server context*.

Note:

The **Transient_Volume_RootDir** preference allows the configuration of transient volume locations for multiple platforms. Typically, one value represents the location of a transient volume on a Linux system and the other a transient volume on a Windows system. Teamcenter distinguishes the values based on the path name separator (/ for Linux, \ for Windows). If you use only one platform, you can delete the other value. Multiple values can be specified but are not required.

If the values specified for the **Transient_Volume_RootDir** preference have identical values, Teamcenter displays the following error message:

```
Duplicate transientvolume configuration elements were found for
attribute id with value attribute_value
```

Modifying transient volume ID components

Modifying either the site ID or the **Transient_Volume_RootDir** preference has far reaching impact, affecting multiple transient volumes defined in the system.

Warning:

Siemens Digital Industries Software strongly discourages changing these values.

Changing the site ID invalidates all existing FMS configured transient volumes because the site ID is used directly to generate transient volume IDs.

Changing the **Transient_Volume_RootDir** preference value invalidates all transient volumes on the platform (Windows/Linux) that was modified because this preference defines the path to the transient volumes for the specified platform.

To modify the **Transient_Volume_RootDir** preference:

1. Change the value of the preference to the new path name of the transient volume on the **Transient_Volume_RootDir**.

Caution:

Do not define the path as a UNC path, for example, `\\server\shared-transient-folder`. Use a direct path location. This requirement is because some ZIP archive utilities do not accept UNC paths, resulting in failed exports to Microsoft Excel or Word.

2. Generate the new transient volume IDs one server context at a time by completing the steps for *Determining the transient volume ID for a different server context*.
3. Use the Organization application to reload the FMS configuration.

Administering volumes

Introduction to administering volumes

Use the Organization application to create new volumes and manage volume locations and properties. After adding new volumes, use Organization to reload the File Management System (FMS) configuration throughout the entire network, generate FMS reports, and display the FMS primary configuration file.

You can also create, manage, review, purge, and move volumes using various Teamcenter utilities.

Default volumes

Default volumes specify the default final destination volume for Teamcenter files. *Default volumes* differ from *default local volumes* in that default volumes are the first and final destination for Teamcenter files. Default local volumes are temporary storage locations.

When users upload a new file, the default volume for the file is determined by the volume attribute of either the user or the user's group. The system determines the volume by working through the following values in order (the first value to have a valid volume is the destination volume):

- The default volume set by the user
- The default volume set by a group to which the user belongs
- The default volume set by a parent group of a group to which the user belongs
- Any volume to which the user has access
- Any volume to which a group the user belongs to has access

Create the volumes you want to use as default volumes in the Organization application. Creating volumes in the Organization application adds the volumes to the **List of Defined Volumes** dialog box. To view this list, select a user or a group and click the **Default Volume** button.

For performance reasons, you cannot set a cloud-based volume as a default volume.

After you create the volumes, the option to define a default volume is available while creating a new user, modifying an existing user, creating a new group, and modifying an existing group.

You can also define a default volume for a user by choosing **Edit→User Setting** and selecting a default volume from the **Volume** list.

Default local volumes

Introduction to default local volumes

Default local volumes are temporary local volumes that allow files to be stored locally before they are automatically transferred to the final destination volume. This functionality improves end-user file upload times from clients by uploading files to a temporary volume. Users can continue to work on their files from the temporary location. The system moves the files to their final destination according to administer-defined criteria. Files are accessible to FMS at all times. This behavior is also referred to as the *store and forward* of files.

By default, store and forward functionality moves files from the local volume one at a time. If you prefer, you can **move files in batch** by using the **store_and_forward** Dispatcher translator.

The purpose of temporary, local storage volumes is to provide upload capability to a volume that is local to remote users. This is useful in situations when an FMS volume does not exist on the LAN with the remote user. In these situations, the closer the initial volume is to the user uploading the file, the faster the upload.

Note:

Default *local* volumes differ from default volumes in that default local volumes are temporary volumes. Default volumes are the final destination for Teamcenter files.

When users upload a new file, the default local volume destination for the file is determined by the volume attributes of either the user or the user's group. The system determines the initial upload volume destination by working through the following values in order. The first value to have a valid volume is the destination volume.

- The default local volume defined for the user
- The default local volume defined for the group under which the user is currently logged on
- The default local volume defined for the user's default group
- The default local volume defined for the parent group of:
 - The default local volume defined for the group under which the user is currently logged on.
 - The default local volume defined for the user's default group, if the previous value is not set.

Note:

The **TC_Store_and_Forward** preference must be set to enable store and forward functionality. If this preference is set, any of the users' accessible volumes can be defined as the session local volume using the **Local Volume** box on the **User Settings** dialog box in the rich client. The session local volume setting overrides the default local volume setting.

If there are no default local volumes defined for the previous values, then store and forward functionality is not used for the file. The normal upload volume location is used. The system determines the volume destination by working through the following values in order. The first value to have a valid volume is the destination volume.

- The default volume defined for the user
- The default volume defined for the group under which the user is currently logged on
- The default volume set for the parent group of the group under which the user is currently logged on
- Any volume to which the user has access
- Any volume to which a group the user belongs to has access

Enable default local volumes

Default local volumes are temporary local volumes that allow files to be stored locally before they are automatically transferred to the final destination volume. This functionality improves end-user file upload times from clients by uploading files to a temporary volume. This functionality is referred to as *store and forward* functionality.

1. Set the **TC_Store_and_Forward** preference to **true**.

This preference can be used as either a site, user, or group preference. Users and administrators can set the preference by choosing **Edit→Options** to open the **Options** dialog box. Use the **Search** tab in this dialog box to locate the preference and ensure it is set to **true**. (The default value is **false**.)

2. (Optional) Set the **TC_Store_and_Forward_Transfer_Delay** preference to how many minutes file transfer between the initial volume and the destination volume is delayed.

Delaying the transfer to the final destination volume can improve performance if your site has a high volume of revisions and the delay is long enough to allow a purge of file revisions before the transfer to the final destination volume.

3. (Optional) Set the **TC_allow_inherited_group_volume_access** preference to allow subgroups to inherit access to a Teamcenter volume from its parent group. If a group is explicitly granted volume

access, and this preference is set to a nonzero number, that group's subgroups (and the subgroup's children) are implicitly granted access to that volume.

Note:

Inherited access applies to all volumes including default local volumes, also known as *store and forward volumes*.

4. Create the volumes you want to use as default local volumes in the Organization application. Creating volumes in the Organization application adds the volumes to the **List of Defined Volumes** dialog box. To view this list, select a user or a group and click the **Default Volume** button.

After you create the volumes, the option to define a default local volume is available while creating a new user, modifying an existing user, creating a new group, and modifying an existing group.

You can also define a default local volume for a user by choosing **Edit→User Setting** and selecting a default local volume from the **Local Volume** list.

Configuring default local volumes

The FMS Transfer Dispatcher Server module is used to transfer uploaded files in the default local volume to the default destination volume. There are no location requirements for this module. It can be placed on the corporate server, the server on which you deploy other Dispatcher Server modules, or by itself at the remote site.

The module only acts as a trigger for the transfer, it does not actively participate in it. Therefore, there is no requirement that it have a fast connection with the FCCs participating in the transfer. The only requirement is that it is connected with its bootstrap FSC and with the Dispatcher Scheduler. This connection allows placement at remote data centers.

A single module can handle multiple default local volumes. The number of modules required depends on the expected store and forward usage, and your environment.

The **Ticket_Expiration_Interval** preference determines the length of time tickets are available for Teamcenter functions such as reading a file or viewing a file. In the context of store and forward functionality, this preference determines the amount of delay between the initial transfer of the file and the cleanup of the file. For example, if you have cleanup tasks in your Dispatcher Server queue, this value determines how long they remain in the queue. Store and forward functionality delays the cleanup to ensure the file is available for any read ticket requests against the file's initial volume location.

Move files in batch for default local volumes

Default local volumes are temporary local volumes that allow files to be stored locally before they are automatically transferred to the final destination volume. This functionality is also known as *store and forward*.

By default, the store and forward functionality moves files from the local volume one at a time using the **fmstranslator** Dispatcher translator. You can use the **store_and_forward** Dispatcher translator to upload files in batch. This is useful in situations when an FMS volume does not exist on the LAN with the remote user. In these situations, the closer the initial volume is to the user uploading the file, the faster the upload.

Perform the following steps to configure moving files in batch for default local volumes:

1. Install and configure the translator. Choose **Enterprise Knowledge Foundation > Dispatcher Server**. Then, in the **Select Translators** panel, select the **StoreAndForward** translator.

If you have questions about setup, see the instructions in the **\Module\Translators\store_and_forward\Readme.txt** file.

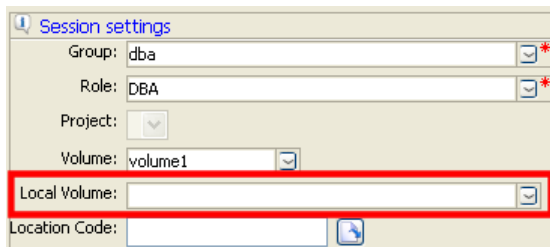
2. In the rich client, set the **TC_Store_and_Forward** preference to **true**.

To set the preference, choose **Edit > Options** to open the **Options** dialog box, and then use the **Search** tab in the dialog box to locate the preference.

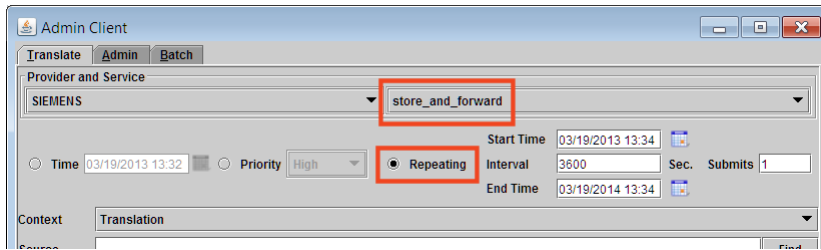
3. Set the **FMS_SAF_Batch_Transfer_Enabled** preference to **true**.

This switches the transfer mode from moving files one at a time (using the **fmstranslator** translator) to moving files in batches (using the **store_and_forward** translator).

4. In the Organization application, choose **Edit→User Setting** to set the local volume for the user in the **User Settings** dialog box.



5. Optionally, to view the files in local volumes to be moved to default volumes, run the **move_volume_files** utility with the **-listaf** argument.
6. Run the **store_and_forward** translator in the Dispatcher Admin Client as a repeating job.



This translator in turn runs the **move_volume_files** utility with the **-transferaf** argument, which queries the database for the user files in local volumes and transfers them to their respective default volumes. It also automatically schedules a task to clean up the files from the local volumes.

You can also run the **move_volume_files** utility as a **cron** job or as a scheduled task using **process_move_file_volumes** as a **.sh** or **.bat** script, respectively.

Note:

If you choose to use the Dispatcher Admin Client to schedule repeating jobs, ensure that the module service is started by an OS user who is a member of the **dba** group. This is mandatory because the **-transferaf** argument of the **move_volume_files** utility must have DBA privileges with bypass authority to commit the database records belonging to different users. However, if you use the **cron** job to schedule repeating jobs, you do not have this restriction.

7. After the translator setup is complete, verify the translator installation:
 - a. Create datasets in the local volume using the rich client.
 - b. Import files into the datasets.
 - c. Verify the files are created under the local volume.
 - d. Schedule the repeating **store_and_forward** translation job using the Admin Client.
 - e. After the translation is complete, verify that the files are moved to the default volume.

Using store and forward after upgrading

After you upgrade from a pre-10.1 version of Teamcenter, and you subsequently use the **batch store and forward process** to move files, you may discover that the process moves files back from the default volume to the default local volume.

This problem occurs when you transition from the old store and forward method introduced in Teamcenter 8.0 to the new batch store and forward method introduced in Teamcenter 10.1. Perform the following steps to remedy the problem.

1. Ensure that all **FMSTransfer** dispatcher tasks are completed and that no new ones start up.

2. Ensure that the current value of the **FMS_SAF_Batch_Transfer_Enabled** preference is set to **false**.
3. Run the following command:

```
move_volume_files -u=admin-username -p=password -g=dba
-migrate_to_batch_saf
```

If the command completes successfully, the following message is displayed:

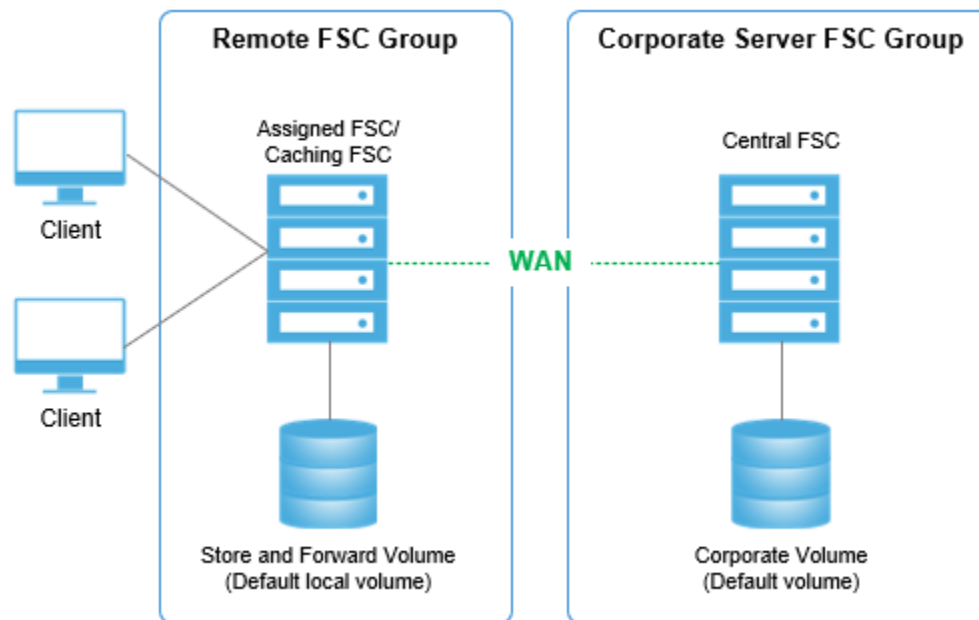
You can now switch over to using the batch Store and Forward functionality.

4. After the **move_volume_files** utility is successfully executed, switch to the batch store and forward process by setting the **FMS_SAF_Batch_Transfer_Enabled** preference to **true**.

Using a default local volume with a single FSC

The simplest configuration of store and forward functionality is to add a single default local volume to an existing caching FSC (FSC). This solution is appropriate when you have a small number of users at the remote site.

The following graphic illustrates the addition of a default local volume to a remote caching FSC.

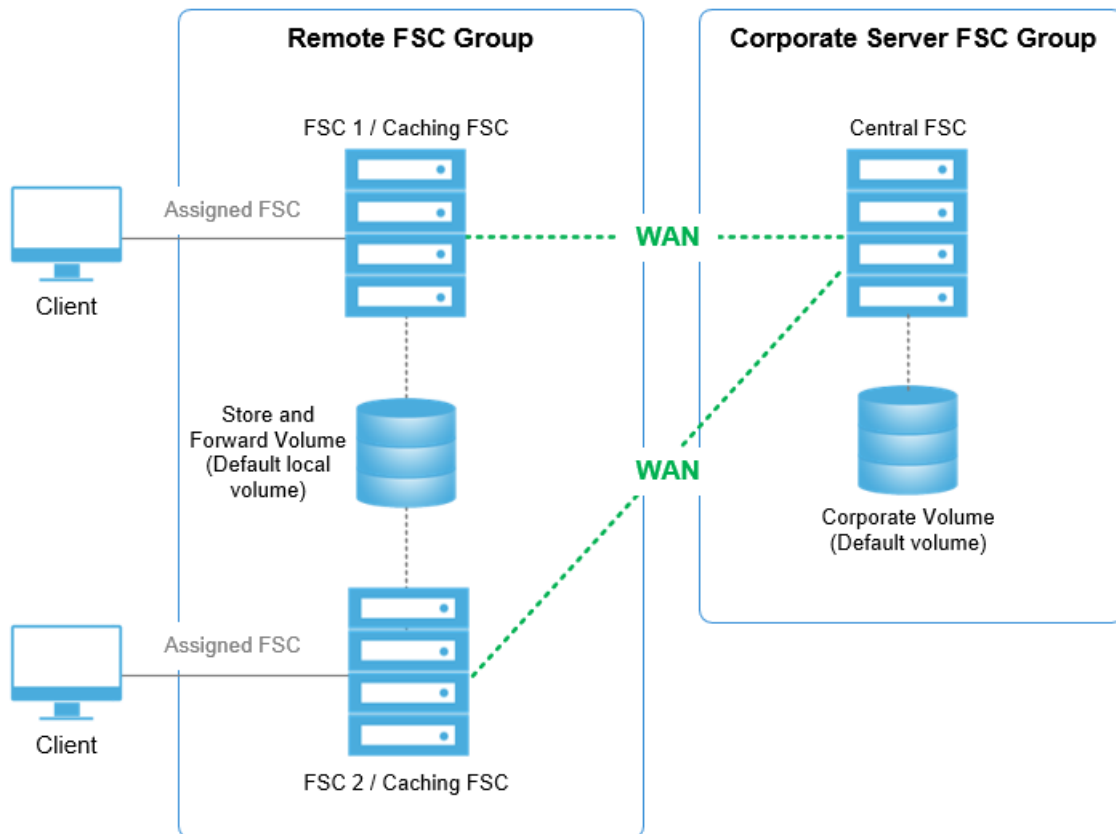


Add a default local volume to an existing caching FSC using the Organization application.

Using a default local volume with multiple FSCs

Remote sites with many users, or high-usage users, may have multiple caching FSCs sharing the load across multiple clients. In such situations, a default local volume can be cross-mounted on the FSCs.

The following graphic illustrates the addition of a default local volume cross-mounted on two FSCs.



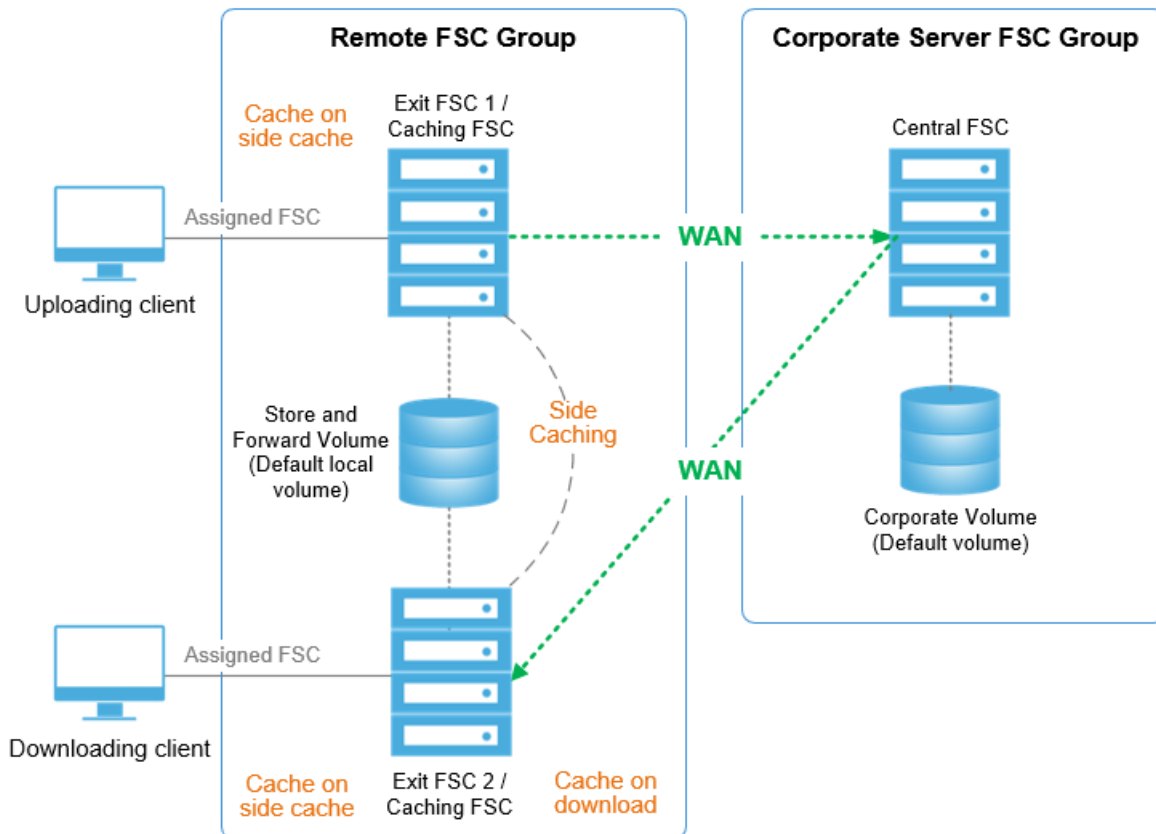
Use the **filestoregroup** element in the **fmsmaster_fscid.xml** file to add a default local volume, cross-mounted on two FSCs. See the **FSC_HOME\fmsmaster.xml.template** file for an example of the **filestoregroup** element.

Using a default local volume with side caching

After a file is transferred to the final destination volume, it is prepopulated into the exit cache on the remote FSC group if side caching is enabled. Enable side caching by explicitly defining exit FSCs. Entry FSCs and FSCs not configured in the FMS primary configuration file (**fmsmaster_fscid.xml**) as exit FSCs do not perform side caching. Normal caching behavior applies in these circumstances.

Side caching is performed only for the highest priority exits defined in the primary configuration file. For example, if three exits are defined, two set to priority 0 and one set to priority 1, only the two exits with priority 0 are side cached.

The following graphic illustrates a caching configuration designed to support a large number of users. In the example, one remote user uploads a file. Another remote user requests a download of the same file. Without side caching configured, the file is only cached after the initial download. With side caching enabled, after the file is transferred to the final destination volume, it is automatically side cached to the defined exit FSCs in the remote FSC group. In this example, the remote user requesting the download receives the file from **Exit FSC 2**.



Note:

The transfer to the final destination volume may not occur immediately. Until the transfer occurs, remote users requesting download of the file receive it directly from the default local volume on their LAN.

Best practices for configuring default local volumes

The default local volume (also referred to as the store and forward volume) is a real volume, storing production data. The production data is transferred to the destination volume relatively soon (depending on the load, and how you have configured the volume), but you must still consider upload requirements. The volume must have the expected availability and reliability you require from a production data volume. Siemens Digital Industries Software recommends RAID 5 or better.

It is best to ensure that once a file is uploaded to the destination volume, any remote requests for that file are retrieved from the cache, not the destination volume. Keep in mind that once a file is transferred to the destination volume, any download request for that file is delivered from the destination volume. If the request comes from a remote user who has just uploaded the file, this represents a WAN hop, producing the WAN penalty just avoided by using store and forward functionality.

Note:

Normal cache flushing and expiration rules apply to local default volume files.

Volume failover

Configuration failover versus volume failover

You can use File Management System (FMS) to manage different types of failover behavior.

Configuration failover features are configured for each client API, either in the **Fms_Bootstrap_Urls** preference or the **parentfsc** list in the **fcc.xml** configuration file. (The **fcc.xml** file is located in the **FMS_HOME** directory, for example, **TC_ROOT\tccs**.)

Volume failover features are configured with the **priority=** attributes of a number of elements in the **fmsmaster_fscid.xml** file. (This file is located in the **TC_ROOT\fsc** directory.)

Once configuration failover delivers the **assignedfsc** list (and/or the **directfscroutes** list) to the FSC or FCC client, the client uses this information to select the initial FSC for each data request, which is the initial aspect of volume failover for Teamcenter data access.

Much of the volume failover implementation is in the intermediate FSC server routing logic, though the client selection of the initial FSC server to receive each data request is also important.

Warning:

When configuring an FMS system for volume failover, you should also consider configuring for configuration failover.

Volume failover is not particularly effective if clients cannot obtain an **assignedfsc** list because the one and only configuration server FSC is offline. Larger deployments encompassing many remote sites and several FSC servers may choose to set up a couple of FSC servers at each site exclusively for use as configuration servers. Smaller deployments (using only a few FSC servers) should consider configuring configuration failover similarly to volume failover.

Overview of configuration failover

FMS configuration is controlled through a **primary configuration file** (**fmsmaster_fscid.xml**) that is hosted by an FSC referred to as the primary or configuration FSC. You can also configure more than one primary or configuration FSC so that the additional FSCs act as failover servers.

Configuration failover allows the client or the FMS network to fail over from one FSC configuration server to another, based on the **priority** value of the FSC set in the *fmsmaster* configuration file. Similar to other failovers in FMS configuration, the **priority** attribute determines the failover configuration with **0** being highest priority, followed by **1**, **2** and so on.

Fail over priority is configured in the **assignedfsc** element in the **clientmap** element. For example:

```
<clientmap subnet="146.122.40.1" mask="255.255.255.0">
  <assignedfsc fscid="fscPrimary1" priority="0" />
  <assignedfsc fscid="fscPrimary2" priority="1" />
</clientmap>
```

Note that in this context, the configuration FSC does not need not be the same as the bootstrap FSC specified using **Fms_Bootstrap_Urls or similar mechanisms**.

- If the bootstrap URL is indeed the same as the primary configuration FSC URL, to then achieve failover, you must also specify *all* of the relevant Primary FSC URLs in the bootstrap.
- If the bootstrap URL is not the Primary FSC URL (for example, if it is a reverse proxy URL), the failover will then work with the **assignedfsc** priority settings described here.

Overview of volume failover

Unlike configuration failover, *volume failover* is implemented in many forms and in many places throughout FMS.

- Implemented within FSC clients

FSC clients use only the **assignedfsc** list in the **fmsmaster_fscid.xml** file to determine the initial FSC destination of data requests from the client. This list is derived exclusively from the **priority=** attributes of the elements in the client's **assignedfsc** list.

- Implemented within FCC clients

FCC clients use a variety of information from the FSC configuration to determine the initial FSC destination of a data request. These include, in order of precedence:

- If **assignment mode="parentfsc"** is specified in the **fcc.xml** file:
 - The **parentfsc** list in the **fcc.xml** file is treated as the actual **assignedfsc** list.
 - The **assignedfsc** list returned by the configuration server FSC is required but largely ignored.
 - The **FCC_EnableDirectFSCRouting** element is automatically set to **false**, regardless of any actual settings in the **fcc.xml** or **fmsmaster_fscid.xml** files.
 - The **directfscroutes** list returned by the configuration server FSC is also ignored.

In this mode, the configuration failover settings are also used for volume failover at the FCC client (by design). The **assignment mode="parentfsc"** setting only affects FCC selection of the initial FSC server; it does not affect volume failover features at intermediate FSCs.

- If the **FCC_TransientFileFSCSource** parameter is set to **ticketuri** (the default setting), the URI in a transient file ticket takes precedence. FMS tickets for resource types¹ other than **transientvolume** do not have a URI embedded in the ticket.
- If the **FCC_EnableDirectFSCRouting** parameter is set to **true** (the default setting), and the **resource** element is served within the same **fscgroup** element as the client's primary **assignedfsc** element, the request is routed to the direct FSCs that serve it in order of the **priority=** attributes of the direct FSC routes.

The priorities of direct FSC routes are determined by the **priority=** attributes of the **resource** and **filestoregroup** elements of the **fsc** and **loadbalancer** elements in the **fmsmaster_fscid.xml** file and delivered to the FCC with the **assignedfsc** list on FCC initialization.

- If the **SiteID** element (*FMS-enterprise-ID*) of the FMS ticket is listed in the **site** list of the **fcc.xml** configuration file, the request is sent to the **assignedfsc** list specific to that client, obtained from the **parentfsc** list specified in the corresponding site element of the **fcc.xml** file (using configuration failover).
- If none of these situations apply, or all of the attempts fail in a manner indicating appropriate failover, the request is sent to the **assignedfsc** list specific to that client, obtained from the default **parentfsc** list (not in a **site** element) listed in the **fcc.xml** file. The priority order of the **assignedfsc** list is determined by the **priority=** attributes of the **assignedfsc** elements of **clientmap** elements in the **fmsmaster_fscid.xml** configuration file obtained from a default **parentfsc**.
- Implemented within intermediate FSCs

The bulk of the volume failover implementation is in the routing code of intermediate FSCs, which route FMS requests to other FSC servers. Intermediate FSCs use **priority=** attributes on a number of other elements in the **fmsmaster_fscid.xml** file to implement volume failover as a part of FMS message routing. These elements include:

- The **resource** and **filestoregroup** elements of **fsc** and **loadbalancer** elements, which apply most directly to the selection of target resource servers for FMS routing.
- The **entryfsc** elements of **fscgroup** elements, particularly when the **entryfsc** element refers directly to volume servers.
- The **exitfsc** elements of **fscgroup** elements, particularly when the **exitfsc** element refers directly to volume servers.

1 The list of **resource** elements includes **volume**, **transientvolume**, **logvolume**, and **accession**.

- The **fscimport** elements of **defaultfscimport** elements, particularly when the **fscimport** elements refer directly to volume servers in the **fmsmaster_fscid.xml** configuration file of the external **fmsenterprise** element.

FMS uses client-proximity routing. Requests are routed by the first FSC to receive a request for a resource that it can neither fulfill from cache nor serve directly from volume. This initial FSC routes the request through the FMS network to another FSC that serves the requested resource.

Warning:

It is very important that all FSC servers within the same FMS system share identical **fmsmaster_fscid.xml** file information. If an FSC server has an inaccurate configuration, it is prone to routing errors, which can cause user transaction failure.

Implement volume failover

Implement volume failover by specifying backup volume servers using the **priority=** attributes of **resource** and **filestoregroup** elements of the **fsc** and **loadbalancer** elements in the **fmsmaster_fscid.xml** file.

The **priority=** attributes on a number of other elements in the **fmsmaster_fscid.xml** file affect the failover characteristics of request routing, including:

- The **assignedfsc** elements of **clientmap** elements, particularly when the **assignedfsc** element refers directly to volume servers.
- The **resource** and **filestoregroup** elements of **fsc** and **loadbalancer** elements, which apply most directly to the selection of target resource servers for FMS routing.
- The **entryfsc** elements of **fscgroup** elements, particularly when the **entryfsc** element refers directly to volume servers.
- The **exitfsc** elements of **fscgroup** elements, particularly when the **exitfsc** element refers directly to volume servers.
- The **fscimport** elements of **defaultfscimport** elements, particularly when the **fscimport** elements refer directly to volume servers in the **fmsmaster_fscid.xml** configuration file of the external **fmsenterprise** element.

Configuring volume failover using multiple FSCs

You can provide volume failover by assigning multiple FSCs to serve the same storage resource. If the primary FSC fails, the secondary FSC continues to serve files from the volume. Configure this behavior by mounting the same volume on multiple FSCs, giving each FSC a different priority level. The volume is always served by the available (alive) FSC with the lowest assigned priority.

In the following example, two FSCs mount the **testvol** volume. The primary FSC is **fsc3**. It is assigned priority level **0**. FSC **fsc4** acts as a backup server. It is assigned priority level **1**.

Look at the configuration for **fscgroup g3**, at the bottom of the code example. Both of the FSCs mount the **testvol** volume, each with a different volume priority value. If **fsc3** is alive it will serve **testvol**, **fsc4** will only serve **testvol** if **fsc3** is down.

```
<fmworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      ...
    </fscdefaults>
    <fscgroup id="g1">
      <fsc id="fsc1" address="http://localhost:5551">
        <volume id="myvol" root="data/myvol"/>
      </fsc>
      <grouproute destination="g3">
        <routethrough groups="g2" priority="0" />
      </grouproute>
    </fscgroup>
    <fscgroup id="g2">
      <fsc id="fsc2" address="http://localhost:5552">
      </fsc>
      <clientmap subnet="127.0.0.1" mask="0.0.0.0">
        <assignedfsc fscid="fsc1" priority="0" />
      </clientmap>
    </fscgroup>
    <fscgroup id="g3">
      <fsc id="fsc3" address="http://localhost:5553">
        <volume id="testvol" root="//nfserver1/datashare/testvol" priority="0"/>
      </fsc>
      <fsc id="fsc4" address="http://localhost:5554">
        <volume id="testvol" root="//nfserver1/datashare/testvol" priority="1"/>
      </fsc>
    </fscgroup>
  </fmsenterprise>
</fmworld>
```

Configure volume failover during file import

You can provide volume failover during file import by defining a failover volume for importing files. If this behavior is configured, the system checks the target volume before import. If the target volume is filled to a specified capacity, the file is directed to a specified failover volume.

Volume failover during file import proceeds as follows:

1. A user initiates a file import. The targeted volume is determined by the default volume specified for the user's group. If store and forward functionality is configured, the importing volume is the specified default local volume.

Note:

Default volumes and default local volumes are defined using the Organization application. These settings are typically defined when the user account is created.

Generally, default volumes are defined at the group level, not the user level. Siemens Digital Industries Software recommends that you do not define a default volume for each user; such granular assignments are time-consuming to maintain. When the user's default volume is not specified, the group's default volume information is used.

2. The system searches for the cached value of the percent full of the targeted volume.
 - If a cached value is found, the value's age is checked. If it exceeds the time specified by the **TC_Volume_Status_Resync_Interval** preference, a fresh value is requested by an FSC admin call. A new percent full value is retrieved and cached.

The percent full values are cached to prevent excessive FSC requests. The **TC_Volume_Status_Resync_Interval** preference specifies the minimum amount of time that can pass before the percent-full value of a volume is retrieved from an FSC.

- If a cached value is not found, the value is requested by a FSC admin call. The percent-full value is retrieved and cached.
3. The system compares the percent-full value with the percent full specified by the **TC_Volume_Failover_Trigger** preference.
 - If the trigger point is not met, the system imports the file to the original target volume.
 - If the trigger point is met, the system imports the file to the failover volume defined by the **TC_Volume_Failover_Name** preference.

The same behavior applies to default local volumes if store and forward functionality is configured.

Specify alternate failover volumes

When a volume size is bigger than a certain threshold, you may decide to set the volume as read-only and create a new volume with the same assigned users and groups. All subsequent file operations should then use this new volume. There are conditions under which file write operations from the Dispatcher Server fail because it cannot write back to the original volume.

To overcome this problem, the **TC_Volume_Failover_Volume_Map** preference allows specifying failover (alternate) volumes for one or more volumes in the system. Use this when a volume is marked as read-only or otherwise inaccessible for new write operations. Specifying the alternate volume ensures that future writes are done to this alternate volume rather than the currently set volume. Each value set in the preference should be of the form: *source-volume:destination-volume*. The mappings can be cascaded, for example: **VolA:VolB and VolB:VolC**. This indicates that the system should use **VolC** when **VolA** is the current volume.

The Teamcenter administrator typically follows these steps to enable this mechanism:

1. Create the new volume in Teamcenter.
2. Grant permissions for the required users and groups to access the new volume.
3. Set the values in the **TC_Volume_Failover_Volume_Map** preference in Teamcenter, for example, *original-volume:new-volume*.
4. Set the original volume as read-only.

Note:

If permissions are not granted to a certain user to access the new volume, the system does not error out. Instead, it defaults to the volume currently set for that user and group.

Volume data

Volume allocation rules

You can move files from one volume to another using allocation rules. Use the **move_volume_files** utility with the **-rulesfile** argument to retrieve an XML file containing the volume allocation rules template, edit the rules as desired, and then use the utility to move the volume files.

You can write volume allocation rules based on various dataset, item, item revision, and volume criteria.

Example:

Consider a site using both CAD and JT files. Because JT files are volatile and can be recovered from the CAD file if lost, they are on a different backup schedule. The administrator decides to store all JT files in a different volume than the CAD files.

Rules can be written in the XML file specifying different target volumes for the JT and CAD files. Each time the utility is run, JT and CAD files not already stored in the respective target volume are moved to the appropriate destination.

The utility can be run manually, as a **cron** job, or as a scheduled task.

The volume allocation rules accept the following criteria.

User criteria	Sample rules
ID	<code><usercriteria name="id" value="tcdba" /></code>
Group	<code><usercriteria name="group" value="dba" /></code>
Project	<code><usercriteria name="project" value="FMS" /></code>

Item criteria	Sample rules
Item ID	<code><itemcriteria name="id" value="ABC" /></code>
Owning user	<code><itemcriteria name="user" value="tcdba" /></code>
Owning group	<code><itemcriteria name="group" value="dba" /></code>
Owning project	<code><itemcriteria name="project" value="FMS" /></code>
Attached dataset type	<code><itemcriteria name="type" value="UGMaster" /></code>

Item revision criteria	Sample rules
Owning user	<code><itemrevisioncriteria name="user" value="tcdba" /></code>
Owning group	<code><itemrevisioncriteria name="group" value="dba" /></code>
Owning project	<code><itemrevisioncriteria name="project" value="FMS" /></code>
Attached dataset type	<code><itemcriteria name="type" value="UGMaster" /></code>

Dataset criteria	Sample rules
Dataset type	<code><datasetcriteria name="type" value="UGMaster" /></code>
Owning user	<code><datasetcriteria name="user" value="tcdba" /></code>
Owning group	<code><datasetcriteria name="group" value="dba" /></code>
Owning project	<code><datasetcriteria name="project" value="FMS" /></code>
Volume ID	<code><datasetcriteria name="volume" value="vol1" /></code>
Created before	<code><datasetcriteria name="createdbefore" value="20-Oct-2020 12:12" /></code>
Created after	<code><datasetcriteria name="createdafter" value="10-Oct-2020 16:24" /></code>

Volume data criteria	Sample rules
Volume free space	<code><volumecriteria name="availablespace" value="50GB" /></code>

Allocate volume data

1. Access the volume allocation rules XML template file by running the **move_volume_files** utility with the **-outrulesfile** argument set to the desired name of the new XML template. For example, the following command creates an XML template named **VolumeSelectionRules.xml**:

```
move_volume_files -u=tc-admin-user -p=password -g=group
-outrulesfile=VolumeSelectionRules.xml
```

The XML file is stored in the current directory.

2. Edit the volume allocation rules template to define volume allocation rules for your site.
3. Evaluate the rules and store the results by running the **move_volume_files** utility with the **-rulesfile** argument set to the name of the XML file and the **-f** argument set to **list**. For example:

```
move_volume_files -u=tc-admin-user -p=password -g=group
-rulesfile=VolumeSelectionRules.xml -f=list
```

4. Evaluate the rules and move the specified files by running the **move_volume_files** utility with the **-rulesfile** argument set to the name of the XML file and the **-f** argument set to **move**. For example:

```
move_volume_files -u=tc-admin-user -p=password -g=group
-rulesfile=VolumeSelectionRules.xml -f=move
```

5. (Optional) Exclude specific volumes from all listing and transfer actions using the **-excludedvollist** argument to process a file containing the list of volumes to be excluded. This argument is typically used to list default local volumes (store and forward volumes) to ensure the files stored in these temporary volume locations are not transferred.

Use either the full path to the file, or use the partial path/file name, in which case the utility searches for the file name in the current directory.

Any number of volumes can be specified in this file. Each entry must be a valid volume name, listed on its own row in the file.

You can run this utility as a **cron** job or as a scheduled task, using **process_move_file_volumes** as a **.sh** or **.bat** script, respectively.

DTD file of volume allocation rules

Reference the **volumereallocation.dtd** file for the elements and references used in the volume allocation rules XML file and their syntax. The DTD file is stored in the **TC_DATA** directory.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- main structure volumereallocation -->
<!ELEMENT volumereallocation (reallocationrule+)>
<!-- Volume Reallocation Rules -->
<!ELEMENT reallocationrule (usercriteria*, itemcriteria*, itemrevisioncriteria*,
datasetcriteria*, volumecriteria*)>
<!ATTLIST reallocationrule id CDATA #REQUIRED>
<!ATTLIST reallocationrule destination CDATA #REQUIRED>
<!ELEMENT usercriteria EMPTY>
<!ATTLIST usercriteria name (id|group|project) #REQUIRED>
<!ATTLIST usercriteria value CDATA #IMPLIED>
<!ELEMENT itemcriteria EMPTY>
<!ATTLIST itemcriteria name (user|id|group|project|type) #REQUIRED>
<!ATTLIST itemcriteria value CDATA #IMPLIED>
<!ELEMENT itemrevisioncriteria EMPTY>
<!ATTLIST itemrevisioncriteria name (user|group|project|type) #REQUIRED>
<!ATTLIST itemrevisioncriteria value CDATA #IMPLIED>
<!ELEMENT datasetcriteria EMPTY>
<!ATTLIST datasetcriteria name (user|type|group|project|volume|createdbefore|
createdafter) #REQUIRED>
<!ATTLIST datasetcriteria value CDATA #IMPLIED>
<!ELEMENT volumecriteria EMPTY>
<!ATTLIST volumecriteria name (availablespace) #REQUIRED>
<!ATTLIST volumecriteria value CDATA #IMPLIED>
```

Sample volume allocation rules XML file

Retrieve the volume allocation rules XML template by running the **move_volume_files** utility with the **-outrulesfile** argument set to the desired name of the new XML template.

The utility generates the XML template and stores it in the current directory. For example:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE volumereallocation SYSTEM "volumereallocation.dtd">
<volumereallocation>
  <!-- Volume Reallocation Rules -->
  <reallocationrule id="rule1" destination="volume1">
    <usercriteria name="id" value="tcdba" />
    <usercriteria name="group" value="dba" />
    <usercriteria name="project" value="Aircraft" />
    <volumecriteria name="availablespace" value="1GB" />
  </reallocationrule>
  <reallocationrule id="rule2" destination="volume1">
    <itemcriteria name="id" value="myitem" />
    <itemcriteria name="project" value="Aircraft" />
    <datasetcriteria name="type" value="UGMaster" />
  </reallocationrule>
  <reallocationrule id="rule3" destination="volume2">
    <usercriteria name="project" value="CarPart" />
    <datasetcriteria name="type" value="DirectModel" />
  </reallocationrule>
</volumereallocation>
```

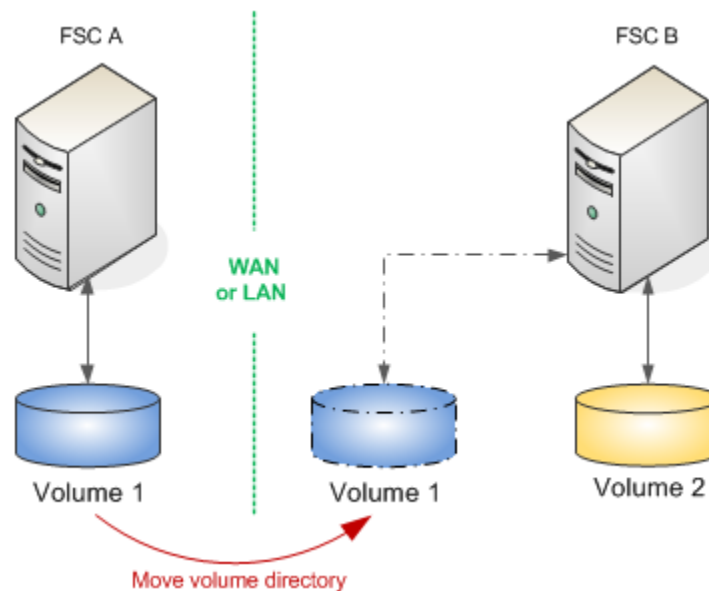
Edit the rules template to define volume allocation rules for your site. For example:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE VolumeReallocationInfo SYSTEM "volumereallocation.dtd">
<volumereallocation>
  <!-- Volume Reallocation Rules -->
  <reallocationrule id="VM_0" destination="DV">
    <usercriteria name="id" value="tcdba" />
    <usercriteria name="group" eqvalue="dba" />
    <usercriteria name="role" nevalue="Designer" />
    <usercriteria name="project" value="11234556" />
    <volumecriteria name="availablespace" value="10000000" />
  </reallocationrule>
  <reallocationrule id="VM_1" destination="vol1">
    <itemcriteria name="id" value="ABC" />
    <itemcriteria name="project" value="11234556" />
    <datasetcriteria name="type" value="UGMaster" />
  </reallocationrule>
  <reallocationrule id="rule2" destination="volume1">
    <datasetcriteria name="type" value="JT" />
    <usercriteria name="project" value="CarPart" />
  </reallocationrule>
</volumereallocation>
```


Move volumes within an enterprise

You can move volumes across a WAN or LAN from one FSC to another FSC within a single enterprise using the **fscadmin** utility.

For example, as shown in the following graphic, you can move **Volume 1** from **FSC A** to **FSC B**. To do so, you must remove the volume UID definition from **FSC A** and add it to **FSC B**. Because the volume UID is not changed during the move, there is no impact to file access.



Read/write tickets may not be honored during the move operation. This has a significant impact on a production system. To move a volume within an enterprise with minimum impact:

1. Use the **backup_modes** utility to put Teamcenter in read-only mode. For example:

```
backup_modes -m=ronly -u=tc-admin-user -p=password -g=group
```

2. Use your operating system copy tools to copy the entire volume directory across a LAN or WAN from its source location in **FSC A** to its destination location in **FSC B**.
3. Use the **fscadmin** utility (stored in the *FSC_HOME* directory) to retrieve the FMS primary configuration of **FSC A**. Note the volume UID and the enterprise ID of **FSC A**. For example:

```
fscadmin -s http://FSCA-address:port ./config
```

4. Use the **fscadmin** utility to remove the volume UID from the FMS primary configuration file for **FSC A**. For example.

```
fscadmin -s http://FSCA-address:port ./config/removevolume/nvars/  
volumeid=volume-UID;enterpriseid=site-ID;fmsconfigentryonly=1
```

5. Use the **fscadmin** utility to add the destination location and volume UID to the FMS primary configuration file of **FSC B** using one of the following identifiers:

- Using FSC ID. For example, type the following all on one line:

```
fscadmin -s http://FSCB-address:port ./config/newaddvolume/nvars/
root=destination-location;volumeid=volume-UID;
fscid=FSC-B;enterpriseid=site-ID
```

- Using the file store group. For example, type the following all on one line:

```
fscadmin -s http://FSCB-address:port ./config/newaddvolume/nvars/
root=destination-location;volumeid=volume-UID;
filestoregroupid=Group;enterpriseid=site-ID
```

- Using the load balancer ID. For example, type the following all on one line:

```
fscadmin -s http://FSCB-address:port ./config/newaddvolume/nvars/
root=destination-location;volumeid=volume-UID;
loadbalancerid=load-balancer;enterpriseid=site-ID
```

6. Use the **fscadmin** utility to reload configuration settings on the FSC primaries and secondaries. For example,

```
fscadmin ./reload/all
```

7. Using your operating system tools, delete the volume under **FSC A**.
8. Use the **backup_modes** utility to put Teamcenter in read-only mode. For example:

```
backup_modes -m=normal -u=tc-admin-user -p=password -g=group
```

9. In Teamcenter, using the Organization application, select the volume, modify the volume's path name, and click **Modify**.

In the **Move Volume** dialog box, at the prompt **Do you want to move files?**, select **No**. Select **Yes** to change the volume location without moving the volume data.

Note:

Use the Organization application to move a volume from one location to another *on the same host*.

Load balancing FMS

Configuration load balancing

FMS configuration is controlled through a **primary configuration file** (*fmsmaster_fscid.xml*) that is hosted by an FSC referred to as the primary or configuration FSC. You can also configure more than one primary or configuration FSC so that the additional FSCs load balance each other.

Configuration load balancing allows the client or the FMS network to use any one of the FSC configuration servers in a rotating manner.

Similar to other load balancing configurations in FMS, the **priority** attribute determines the load balancing configuration, such that all FSCs with the same priority value (such as **0**) are considered balanced.

Load balancing priority is configured in the **assignedfsc** element in the **clientmap** element. For example:

```
<clientmap subnet="146.122.40.1" mask="255.255.255.0">
  <assignedfsc fscid="fscPrimary1" priority="0" />
  <assignedfsc fscid="fscPrimary2" priority="0" />
</clientmap>
```

Note that in this context, the configuration FSC does not need not be the same as the bootstrap FSC specified using **Fms_Bootstrap_Urls or similar mechanisms**.

- If the bootstrap URL is indeed the same as the primary configuration FSC URL, to then achieve load balancing, you must also specify *all* of the relevant Primary FSC URLs in the bootstrap.
- If the bootstrap URL is not the Primary FSC URL (for example, if it is a reverse proxy URL), the load balancing will then work with the **assignedfsc** priority settings described here.

Introduction to load balancing FMS data

You can load balance FMS data by distributing the network access load among same-priority elements. Do this by setting selected XML elements to equal priority.

When elements of equal priority are encountered, FMS attempts to distribute the network access load among the same-priority items. Lower-priority (numerically larger) elements are processed only after all of the higher-priority (numerically smaller) elements are depleted without successful fulfillment of the request.

You can assign same-priority values to the following elements:

Element	Location
accesson	Within the filestoregroup and fsc elements in the fmsmaster_fscid.xml file
assignedfsc	Within the clientmap elements in the fmsmaster_fscid.xml file
defaultfsc	Within the multisiteimport elements within the fmsenterprise elements in the fmsmaster_fscid.xml file
entryfsc	Within the fscgroup elements in the fmsmaster_fscid.xml file
exitfsc	Within the fscgroup elements in the fmsmaster_fscid.xml file
filestore	Within the fsc elements in the fmsmaster_fscid.xml file
fscmaster	The fsc.xml file
logvolume	Within the filestoregroup and fsc elements in the fmsmaster_fscid.xml file
parentfsc	The fcc.xml file
transientvolume	Within the filestoregroup and fsc elements in the fmsmaster_fscid.xml file
volume	Within the filestoregroup and fsc elements in the fmsmaster_fscid.xml file

Note:

The **routethrough** elements within **grouproute** elements of the **fmsmaster_fscid.xml** file cannot be load balanced.

This functionality is triggered entirely by the values of existing XML priority attributes. Load balancing FMS elements requires no DSD or structural configuration changes. Load balancing is fully backward compatible with existing and previous FMS configurations where unique priorities were rigidly enforced. Existing configurations using unique priorities will experience no behavioral effect from the implementation of the load balancing capabilities.

Examples of load balancing FMS data

The following examples illustrate how to load balance FMS data by assigning equal priorities to various elements.

- Parent FSCs

The **parentfsc** element selects the FSC server(s) from which an FCC attempts to download its configuration information. An FCC randomly selects from among elements of equal priority.

In the following example, the **parentfsc** setting results in an FCC first attempting to download its configuration from either **fsc1** or **fsc2** approximately half of the time. When considered as an overall effect among all clients, the requests for FCC configuration information are distributed approximately equally between **fsc1** and **fsc2**.

```
<parentfsc address="fsc1:4444" priority="0"/>
<parentfsc address="fsc2:4444" priority="0"/>
```

- Direct FSC routes

If the **fscgroup** element contains cross-mounted resources, you can load balance direct FSC routing. An FCC randomly selects from among elements of equal priority.

Note:

This feature applies only when direct FSC routing is enabled on the FCC.

In the following example, if the primary configuration file contains the following elements, the FCC directs requests for data on the volume **vol1** approximately equally among FMS servers **fsc1** and **fsc2**.

```
<filestoregroup id="fsgroup1">
  <volume id="vol1" root="/data/vol1"/>
</filestoregroup>

<fsc id="fsc1" address="http://fsc1:4444">
  <filestore groupid="fsgroup1" priority="0"/>
</fsc>

<fsc id="fsc2" address="http://fsc2:4444">
  <filestore groupid="fsgroup1" priority="0"/>
</fsc>
```

- Entry FSCs

The **entryfsc** attribute within the **fscgroup** element of the **fmsmaster_fscid.xml** file defines which of the FSCs in the FSC group are preferred for access from outside the FSC group. An FSC randomly selects from among elements of equal priority.

In the following example, the presence of the following elements cause requests coming from other FSC groups to be sent to **router1** and **router2** each approximately half of the time:

```
<entryfsc fscid="router1" priority="1"/>
<entryfsc fscid="router2" priority="1"/>
```

Using external hardware devices for load balancing

Introduction to using external hardware devices for load balancing

You can load balance FMS data using external hardware load balancers. These devices exist within the network topology, but are not part of the FMS system. This load balancing method allows you to route requests from multiple clients among a number of FSCs. You must configure the devices so that each target server receives an equal load. Your goal is to minimize overall response time at the requesting clients by routing requests to the most available server.

Configure FMS to recognize these third-party hardware devices using the **loadbalancer** XML element in the **fmsmaster_fscid.xml** file. Identify which FSCs are within the scope of a load balancer using the **loadbalancerid** attribute of the **fsc** element. Each **loadbalancer** XML element that contains one or more resources (**volume**, **logvolume**, **transientvolume**, or **accession**) or one or more **filestore** entries must have at least one FSC in its scope.

FSCs within the scope of the specified load balancer serve all resources declared in the **loadbalancer** element, and perform the routing behaviors (**entryfsc**, **exitfsc**, **assignedfsc**) attributed to the load balancer. This prevents FSCs within the scope of a load balancer from forwarding requests back to the load balancer.

To clients and FSCs not within the scope of the specified load balancer, the load balancer appears as a single FSC which serves all of the balanced resources and exhibits the load balancer's routing behaviors. Direct FSC routing, intergroup routing, and intragroup routing all function on this basis. Thus clients request the balanced resources only from the load balancer.

Note:

The **loadbalancer** XML element is similar to the **fsc** XML element. The significant difference is that this element cannot be used as the ID of the FSC element in a local FSC configuration file. This is because the **loadbalancer** XML element must always represent external hardware, never an actual FSC.

This element can only be used in the primary configuration file (**fmsmaster_fscid.xml**). You can use the **loadbalancer** ID as a value for any of the following attributes within the FSC group in which it is declared:

```
entryfsc fscid
exitfsc fscid
assignedfsc fscid
fsc loadbalancerid
```

The **loadbalancer** element can contain any of the following child elements:

- **connection**
- **fccdefaults**

- **fscdefaults**
- **filestore**
- Any type of resource, such as **volume**, **logvolume**, **transientvolume**, **accession**

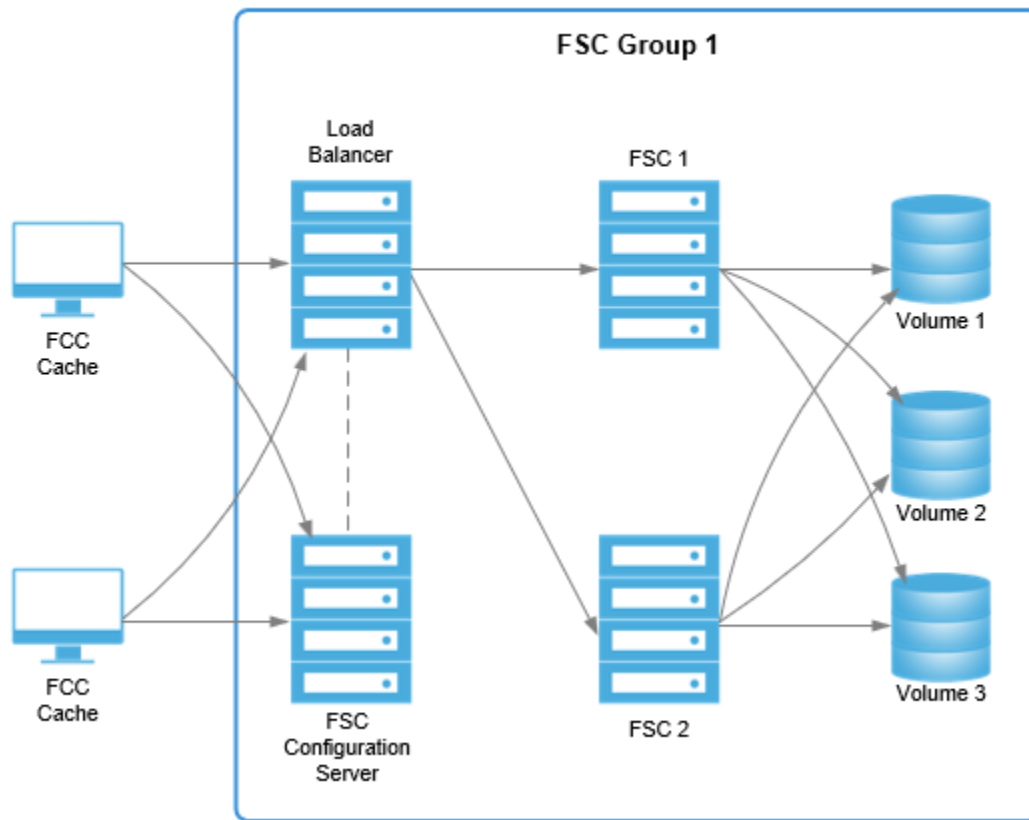
Load balancers support health checking for their servers. Health checking is generally configured to use a particular URL within the back-end servers.

Because FSC servers support few requests that do not require tickets, many default health checks return **HTTP 400** errors that load balancers may interpret to mean service is unavailable. To avoid these errors, Siemens Digital Industries Software recommends using **HTTP HEAD** or **GET** requests for **/Ping** or **/favicon.ico** resources. These requests return an **HTTP 200** status and verify the FSC is alive.

For WebSEAL, this configuration is controlled using the **ping-uri** configuration element, documented in the WebSEAL product documentation.

Local load balancing example

In the following example, three volumes are cross-mounted and served by two FSCs (**FSC1** and **FSC2**), which are serviced by an external load balancer. A third FSC services configuration information. Direct FSC routing enables all of the clients to access volume data through the load balancer, which forwards the requests to **FSC1** and **FSC2**. These two FSCs share equal responsibility for providing volume data access to the clients.



This implementation of local external load balancing is configured using the following XML code. Any FSC containing a **loadbalancerid** element lies within the scope of the specified load balancer. In this example, **FSC1** and **FSC2** are within the scope of **loadbal**. These servers must reference data through the load balancer. All servers within the scope of the same load balancer provide equivalent service in regard to load balanced resources.

```
<fscgroup id="fscGroup1">
  <!-- The load balancer and the volumes it references -->
  <filestoregroup id="vols">
    <volume id="vol1" root="/data/vol1"/>
    <volume id="vol2" root="/data/vol2"/>
    <volume id="vol3" root="/data/vol3"/>
  </filestoregroup>

  <loadbalancer id="loadbal" address="http://lb.ugs.com:4444">
    <filestore groupid="vols" />
  </loadbalancer>

  <!-- FSCs fronted by the load balancer -->
```



```

<fsc id="fsc1" address="http://fsc1.ugs.com:4444"
  loadbalancerid="loadbal">
  <!-- it is invalid to specify load balanced volumes here -->

</fsc>

<fsc id="fsc2" address="http://fsc2.ugs.com:4444"
  loadbalancerid="loadbal">
  <!-- it is invalid to specify load balanced volumes here -->

</fsc>

<!-- FSC configuration server -->

<fsc id="fscConfig" address="http://fscConfig.ugs.com:4444" />

<clientmap subnet="146.122.40.1" mask="255.0.0.0">
  <!-- it is invalid to specify the load balanced FSCs here -->
  <assignedfsc fscid="fscConfig" />

</clientmap>

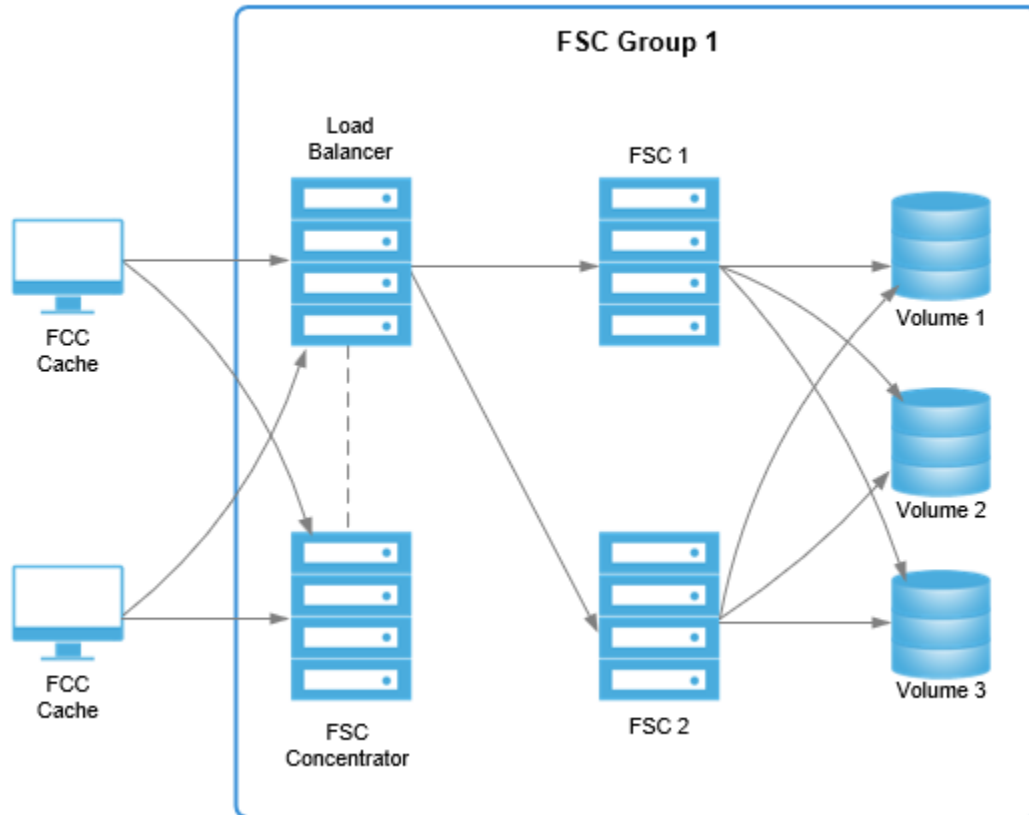
</fscgroup>

```

Remote load balancing example

In the following example, three volumes are cross-mounted and served by two FSCs (**FSC1** and **FSC2**) which are serviced by an external load balancer. Clients are assigned to a remote FSC cache server, which provides configuration information and caching at the remote site. As in the previous example, direct FSC routing enables all of the clients to access volume data through the load balancer, which forwards the requests to **FSC1** and **FSC2**. These two FSCs share equal responsibility for providing volume data access to the clients.

Additionally, an FSC concentrator can be used as a local cache concentrator and/or a router node for requests to additional FSC groups.



This implementation of remote external load balancing is configured using the following XML. Adding the **fscConcentrator** element to an FSC group in the **fmsmaster_fscid.xml** file allows it to act as a router and cache concentrator for data in this FSC group and/or other FSC groups. Defining the **loadbalancer id** attribute assigns the load balancer preference for requests for data it serves (even though it is a lower priority). Without this preferential treatment, all incoming requests are routed through the FSC concentrator.

```
<fscgroup id="fscGroup1">
  <!-- The load balancer and the volumes it references -->
  <filestoregroup id="vols">
    <volume id="vol1" root="/data/vol1"/>
    <volume id="vol2" root="/data/vol2"/>
    <volume id="vol3" root="/data/vol3"/>
  </filestoregroup >
  <loadbalancer id="loadbal" address="http://lb.ugs.com:4444">
  <filestore groupid="vols" />
</loadbalancer>
```

```

<filestoregroup id="fsgroup1">
  <volume id="vol1" root="/data/vol1"/>
  <volume id="vol2" root="/data/vol2"/>
  <volume id="vol3" root="/data/vol3"/>
</filestoregroup>

<!-- FSCs fronted by the load balancer -->

<fsc id="fsc1" address="http://fsc1.ugs.com:4444"
loadbalancerid="loadbal">

<!-- it is invalid to specify load balanced volumes here -->

</fsc>

<fsc id="fsc2" address="http://fsc2.ugs.com:4444"
loadbalancerid="loadbal">

<!-- it is invalid to specify load balanced volumes here -->

</fsc>

<!-- FSC concentrator -->

<fsc id="fscConcentrator"
address="http://fscConcentrator.ugs.com:4444" />

<!-- Routing information -->

<!--
Declaring fscConcentrator as an entry to the fscgroup allows it to act as a router and cache
concentrator for data in this and/or other groups.
-->

<fscentry fscid="fscConcentrator" />

<!--
Declaring the load balancer as an entry gives it the preference for requests for data it
serves
(even though it's a lower priority). Otherwise, all requests would be routed through the
other
fscentry items (fscConcentrator).
-->

<fscentry fscid="loadbal" priority="1"/>

</fscgroup>

<fscgroup id="fscRemoteGroup">

<fsc id="fscRemote" address="http://fscRemote.ugs.com:4444">

</fsc>

<clientmap subnet="162.0.0.1" mask="255.0.0.0">

```

```
<!-- it is invalid to specify the load balanced FSCs here -->
<assignedfsc fscid="fscRemote" />
</clientmap>
</fscgroup>
```

FMS client maps

Introduction to client maps

The client map element in the primary configuration file (**fmsmaster_fscid.xml**) allows FCCs to access the FMS network.

To log on to the FMS network, an FCC must locate its assigned FSC server. To do so, it sends a message to its default FSC server. The default FSC server retrieves the FCC IP address from the message protocol. It performs an **AND** operation on the mask value and IP address, then compares the results with the results of the same operation performed on the subnet/mask values of available servers. The FCC is directed to the FSC server with the matching client map.

You can configure the client map element using either subnet/mask attributes, DNS-based attributes, or a combination of both.

Subnet/mask attributes in a client map

A *subnet* is a range of logical addresses within the address space assigned to an organization. Within the realm of FMS client map functionality, the subnet attribute is the base IP address.

The mask attribute masks a requesting FCC IP address, then compares the results with the subnet to determine whether the requesting FCC address is assigned to a particular client map. The client map contains a set of FSCs to be assigned to the FCC. This mapping technique works if your clients can group their FCCs to numerically adjacent IP addresses, for example:

```
<clientmap subnet="216.068.063.000" mask="255.255.255.000">
  <assignedfsc fscid="fsc_engC" priority="0"/>
</clientmap>
```

When using the subnet/mask method, you can create a default client map by setting the mask attribute value to **0.0.0.0**. This creates a client map that matches all FCC addresses, for example:

```
<clientmap subnet="127.0.0.1" mask="0.0.0.0">
  <assignedfsc fscid="fsc_engC" priority="0"/>
</clientmap>
```

Warning:

This technique cannot be used in conjunction with **DNS-based client map attributes**. If you use a mask value of **0.0.0.0** in conjunction with DNS-based client map attributes, the process fails.

CIDR attributes in a client map

You can use Classless Inter-Domain Routing (CIDR) notation to specify subnetted IPv4 or IPv6 addresses. (A *subnet* is a range of logical addresses within the address space assigned to an organization.) Within the realm of FMS client map functionality, the subnet attribute is the base IP address.

CIDR notation uses the following format:

IPv4-or-IPv6-address / prefix-length

The prefix length specifies how many leftmost bits of the address specify the prefix. This is an alternate way of using the subnet and mask attributes in a **clientmap** element in the **fmsmaster_fscid.xml** file.

The prefix length masks a requesting FCC IP address and then compares the results with the prefix of the IP address to determine whether the requesting FCC address is assigned to a particular client map. The client map must contain a set of FSCs to be assigned to FSC clients.

Note:

The maximum prefix length is 32 for IPv4 addresses and 128 for IPv6 addresses.

- The following example illustrates a default client map using the CIDR method with an IPv4 address:

```
<clientmap cidr="216.068.063.000/24">
  <assignedfsc fscid="fsc_engC" priority="0"/>
</clientmap>
```

- The following example illustrates a client map using the CIDR method with an IPv6 address:

```
<clientmap cidr="fe80::7a5c:6199:766a:812e/64">
  <assignedfsc fscid="fsc_engC" priority="0"/>
</clientmap>
```

When using the CIDR method, you can create a default client map by setting the address and prefix length to zeroes.

- The following example illustrates a default client map using the CIDR method with an IPv4 address:

```
<clientmap cidr="0.0.0.0/0">
  <assignedfsc fscid="fsc_engC" priority="0"/>
</clientmap>
```

- The following example illustrates a default client map using the CIDR method with an IPv6 address:

```
<clientmap cidr="0::0/0">
  <assignedfsc fscid="fsc_engC" priority="0"/>
</clientmap>
```

Warning:

This technique cannot be used in conjunction with **DNS-based client map attributes**. If you use a prefix length value of **0** in conjunction with DNS-based client map attributes, DNS client attributes are not considered.

Domain name client maps

Domain name client maps allow you to administer client map functionality based on domain name servers. There are four DNS-based client map attributes. Only one of the attributes can be specified per **clientmap** element in the **fmsmaster_fscid.xml** file.

Attribute	Use
dnszone	Assigns all the FCCs within the specified DNS zone to an FSC or set of FSCs, for example: <pre><clientmap dnszone="engA.company.com"> <assignedfsc fscid="fsc_engA" priority="0"/> </clientmap></pre>
dnshostname	Assigns a specific FCC to an FSC or set of FSCs, for example: <pre>#clientmap dnshostname="wkst1.engA.company.com"\$ #assignedfsc fscid="fsc_engB" priority="0"/\$ #/clientmap\$</pre>
default	Defines the FSCs to be assigned when no other client map matches the FCC address. This default client map applies to both subnet/mask and domain name client map matches. If the default is not defined and the client map match fails, then the FCC assignment fails, for example: <pre><clientmap default="true"> <assignedfsc fscid="fsc_engC" priority="0"/> </clientmap></pre>
dns_not_defined	Use this attribute if you expect a reverse DNS lookup for an FCCs domain name address cannot be performed. (For example, if it is an unregistered address.) If this attribute is not defined and the reverse DNS lookup fails then FCC assignment fails, for example: <pre><clientmap dns_not_defined="true"> <assignedfsc fscid="fsc_engC" priority="0"/> </clientmap></pre>

How the system processes client maps

Whenever an FCC requests its assigned FSCs from the FSC, the system performs the following actions in the following order:

1. A subnet/mask IP match is attempted. If a match is found, its assigned FSCs are returned.
2. If there is no subnet/mask IP match and there are no domain name client maps defined:
 - If the default IP client map was configured, its assigned FSCs are returned.
 - If the default client map is not configured, a null is returned.
3. If there is no IP subnet/mask match, but there is at least one **dnszone** or **dnshostname** attribute defined, a reverse DNS lookup of the FCC's domain name address is performed.
 - If the reverse DNS lookup fails and the **dns_not_defined** attribute is defined, its assigned FSCs are returned.
 - If the reverse DNS lookup fails, but the **dns_not_defined** attribute is not defined, an error message is logged and a null value is returned.
 - If the reverse DNS lookup succeeds, the domain name client maps are searched for a match. If one is found, its assigned FSCs are returned.
4. If no domain name client map match is found and the **default** attribute is defined, its assigned FSCs are returned.
5. If no domain name client map is matched and no default attribute is defined, the query fails and a null is returned.

Client map specificity

The subnet/mask client maps and the domain name client maps are applied in order of specificity. Specificity is implemented by the following algorithms on the two separate lists.

Subnet/mask client maps are sorted based on the size of their mask values. Client maps with larger mask values specify a bigger mask over the FCC IP address, so they are more specific. Client maps with larger masks are searched first for a match.

In the following example, the "255.255.255.192" mask is more specific than the "255.255.255.000" mask, so it is sorted first in the subnet/mask clientmap list.

```
subnet="192.168.000.128" mask="255.255.255.192"
subnet="192.168.000.000" mask="255.255.255.000"
```

Domain name client maps are sorted based on the length of the string in their **dnszone** or **dnshostname** attributes. In the following example, the **support.ugs.com dnszone** is the most specific so it is sorted first in the domain name client map list. The **com dnszone** is the least specific client map and is sorted last.

```
dnszone="support.ugs.com"
dnszone="ugs.com"
dnszone="com"
```

Accessing multiple databases through a single FCC

You can access multiple databases through a single File Management System (FMS) client cache (FCC). Teamcenter supports multiple installations on the same machine to access multiple site IDs (databases).

For example, consider a part supplier with multiple customers, requiring access to each customer's PLM data, where the customers are in direct competition with each other and each customer has a different security key. The part supplier can modify the local FCC file or the primary configuration file at the primary site to allow the FCC to determine from which FSCs to request additional site data.

Implement this functionality by configuring the FCC's local **fcc.xml** file or the primary configuration file (**fmsmaster_fscid.xml**) at the primary site to provide the configuration data required so the FCC can determine from which FSCs to request additional site data. The FCC receives this configuration data upon startup when it loads data from the **fcc.xml** file and downloads configuration data from the **fmsmaster_fscid.xml** file. Do this by modifying the **fcc.xml** file or **fmsmaster_fscid.xml** file to configure your FCCs to route each FMS request to a different FSC (or set of FSCs) based on the site ID contained in each request ticket. Each FCC continues to have one or more default FSC destination to which it routes requests associated with unknown site IDs.

To implement this functionality:

- Each database must have a unique site ID.
- Each database may be assigned its own security key to prevent unauthorized access from other PLM systems. Each competitor's PLM data should be controlled by a separate database.
- Configure either your **fmsmaster_fscid.xml** file or **fcc.xml** files to support multiple databases by including valid configuration elements referencing at least one parent FSC that supports each of the other databases to which the client may connect.
- The proper multisite FCC client configuration must be present in the **FMS_HOME** directory from which the FCC is started. Siemens Digital Industries Software recommends that you use a single **FMS_HOME** environment variable setting to point to the combined configuration.
- FSC servers for all databases must be online, properly configured, of the proper version, and functional.

Following are configuration examples:

- **fmsmaster_fscid.xml** configuration example

Modify your **fmsmaster_fscid.xml** file to support multiple databases based on the following example. The code defining multiple databases is in *italic*.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fmsworld SYSTEM "fmsmasterconfig.dtd">
...

<fmsworld>
  <fmsenterprise id="myenterprise">
    ...
    <fccdefaults>
      <!-- All clients in this fmsenterprise will be able to access "globalsite" -->
      <site id="globalsite" overridable="true">
        <parentfsc address="192.0.2.1:4220" priority="0"/>
      </site>
    </fccdefaults>
    <fscgroup id="mygroup">
      <fccdefaults>
        <!-- All clients whose primary assignedfsc is in the fscgroup "mygroup"
        will be able to access "groupsite" (as well as "globalsite") -->
        <site id="groupsite" overridable="true">
          <parentfsc address="192.0.5.1:4220" priority="0"/>
          <parentfsc address="192.0.5.2:4220" priority="1"/>
        </site>
      </fccdefaults>
      <fsc id="myfsc" address="127.0.0.1:4444">
        <!-- ----->
        <!-- fccdefaults from the myfsc.xml file (if any) should be moved HERE. -->
        <!-- ----->
        <fccdefaults>
          <!-- All clients with "myfsc" as their primary assignedfsc will be able
          to access "specialsite" (as well as "globalsite" and "groupsite") -->
          <site id="specialsite" overridable="true">
            <parentfsc address="192.0.3.1:4220" priority="0"/>
            <parentfsc address="192.0.3.2:4220" priority="0"/>
            <assignment mode="parentfsc"/>
          </site>
        </fccdefaults>
        <volume id="myvol" root="e:/FMSShare/FMSTestExplodedWar"/>
        <volume id="testvol" root="d:/temp/testvol"/>

        <transientvolume id="mytransvol" root="e:/FMSShare/Temp"/>
        <transientvolume id="testtransvol" root="d:/temp/testtempvol"/>

      </fsc>
      ...
    </fscgroup>
  </fmsenterprise>
</fmsworld>
```

- **fcc.xml** file configuration example

Modify your **fcc.xml** file to support multiple databases based on the following example. The code defining multiple databases is in *italic*.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">
...
<fccconfig>
  <fccdefaults>
    <property name="FCC_CacheLocation" value="~/FMSCache"/>
    <property name="FCC_MaxWriteCacheSize" value="10M"/>
    <property name="FCC_MaxReadCacheSize" value="30M"/>
    ...
    <!-- All clients on this local machine will be able to access "othersite"
      (as well as any sites passed down from fmsmaster.xml) -->
    <site id="othersite" overridable="true">
      <!-- These parentfscs and assignment mode apply only to "othersite." -->
      <parentfsc address="192.0.4.1:4220" priority="0"/>
      <parentfsc address="192.0.4.2:4220" priority="0"/>
      <parentfsc address="192.0.4.3:4220" priority="1"/>
      <assignment mode="clientmap"/>
    </site>
  </fccdefaults>
  <!-- The default site's parentfscs and assignment mode are defined here. -->
  <parentfsc address="192.0.1.1:4444" priority="1"/>
  <parentfsc address="192.1.1.1:4444" priority="3"/>
  <parentfsc address="192.0.1.2:4321" priority="0"/>
  <parentfsc address="192.1.1.2:4321" priority="2"/>
  <assignment mode="parentfsc"/>
</fccconfig>
```

- **fcc.xml** file personal use configuration example

Modify your **fcc.johndoe.xml** file to allow the user (**johndoe**) to access a specific site (*personalsite*) based on the following example. The **fcc.johndoe.xml** file should be owned by the user with the name indicated in its file name (**johndoe**) and given owner-only access. The code defining the access to the specific site is in *italic*.

Note:

The file name must include the user name, as indicated. Do not use this method in a **fcc.xml** file shared by multiple users on the same machine, specifically, at sites using user-specific **fcc.xml** files.

```
<fccconfig>
  <fccdefaults>
    <!-- This enables the user "johndoe" to access "personalsite"
      from this machine (as well as "othersite" found in the local
      fcc.xml and "globalsite" and any other sites passed down from
      the fmsmaster.xml) -->
    <site id="personalsite" overridable="true">
      <parentfsc address="192.0.4.1:4220" priority="0"/>
      <parentfsc address="192.0.4.2:4220" priority="85"/>
    </site>
  </fccdefaults>
</fccconfig>
```

```

    </site>
  </fccdefaults>
</fccconfig>

```

Compressing FMS files

Compress files for multisite transfer

You can compress File Management System (FMS) files before transferring them between FMS server caches (FSCs), increasing performance and reducing network traffic. File compression is available for FSC to FSC transfers across groups and across sites.

File compression is controlled with two **fscdefault** elements (**FSC_DoNotCompressExtensions** and **FSC_WANThreshold**) and the **compression** attribute, available in the **defaultfsc** and **linkparameters** elements.

To configure file compression for multisite transfer:

1. Specify the file extensions you *do not* want compressed by adding the extension names to the **FSC_DoNotCompressExtensions** element, located within the **fscdefault** element in the **fmsmaster_fscid.xml** file.

Enter values as a comma-separated list. FSCs do not send these file types as compressed files, nor request compressed content for these file types.

The default value for this element is:

```

<property-name="FSC_DoNotCompressExtensions"
value="bz,bz2,cab,deb,docm,docx,ear,gif,gz,jar,jpeg,jpg,jt,lha,lzh,lzo,mp3,
mp4,mpg,rar,rpm,sit,taz,tgz,war,xlsm,xlsx,z,zip" overridable="true"/>

```

2. Specify the minimum file size threshold that must be reached before files are compressed by setting the **FSC_WANThreshold** element located within the **fscdefault** element in the **fmsmaster_fscid.xml** file. Files smaller than this value are not compressed. This value also determines the threshold file size that must be reached before WAN acceleration is used. The default setting is 256K.

This value also determines the threshold file size that must be reached before WAN acceleration is used. The default setting is 32K.

3. For multisite transfer, add the **compression** attribute to the **defaultfsc** element and set it to **gzip**. For example:

```

<defaultfsc fscaddress="http://localhost:5551" transport="wan" compression="gzip"
maxpipes="4" />

```

For group-to-group transfer, add the **compression** attribute to the **linkparameters** element for each group and set each instance to **gzip**. For example:

```
<linkparameters fromgroup="g2" togroup="g4" transport="wan" compression="gzip"
maxpipes="4" />
<linkparameters fromgroup="g4" togroup="g2" transport="wan" compression="gzip"
maxpipes="4" />
```

This configuration causes FSCs acting as servers to compress content for all clients that indicate they can accept **gzip** compressed responses. It allows all FSCs acting as clients to request compressed content for whole file transfers across sites and across groups.

File compression example

The following example illustrates how to compress FMS files for transfer across sites and across groups, increasing performance, and reducing network traffic:

```
<fmsworld>
  <multisiteimport siteid="s1">
    <fscgroupimport localgroupid="g2">
      <defaultfsc fscaddress="http://localhost:5551" transport="wan" compression="gzip"
maxpipes="4" />
    </fscgroupimport>
    <fscgroupimport localgroupid="g4">
      <defaultfsc fscaddress="http://localhost:5551" transport="wan" compression="gzip"
maxpipes="4" />
    </fscgroupimport>
  </multisiteimport>

  <fmsenterprise id="fms.teamcenter.com">

    <fscdefaults>

      <!-- these are what we expect to be install options -->
      <property name="FSC_ReadCacheLocation" value="$HOME/FSCCache|/scratch/$USER/
FSCCache" overridable="true" />
      <property name="FSC_WriteCacheLocation" value="$HOME/FSCCache|/scratch/$USER/
FSCCache" overridable="true" />

      <!-- these are what we expect to be maintenance options -->
      <property name="FSC_LogLevel" value="WARN" overridable="true" />
      <property name="FSC_TraceLevel" value="2" overridable="true" />

      <!-- segment cache sizing, read cache -->
      <property name="FSC_MaximumReadCacheFilePages" value="4096" overridable="true" />
      <property name="FSC_MaximumReadCacheSegments" value="5000" overridable="true" />
      <property name="FSC_ReadCacheHashBlockPages" value="2048" overridable="true" />
      <property name="FSC_MaximumReadCacheExtentFiles" value="3" overridable="true" />
      <property name="FSC_MaximumReadCacheExtentFileSizeMegabytes" value="64"
overridable="true" />

      <!-- segment cache sizing, write cache -->
      <property name="FSC_MaximumWriteCacheFilePages" value="4096" overridable="true" />
      <property name="FSC_MaximumWriteCacheSegments" value="5000" overridable="true" />
      <property name="FSC_WriteCacheHashBlockPages" value="2048" overridable="true" />
      <property name="FSC_MaximumWriteCacheExtentFiles" value="3" overridable="true" />
      <property name="FSC_MaximumWriteCacheExtentFileSizeMegabytes" value="64"
overridable="true" />

      <property name="FSC_DoNotCompressExtensions"
```

```

value="bz,bz2,cab,deb,ear,gif,gz,jar,jpeg,jpg,lha,
    lzh,lzo,mp3,mp4,mpg,rar,rpm,sit,taz,tgz,war,z" overridable="true"/>

</fscdefaults>

<fscgroup id="g2">
  <fsc id="fsc2" address="http://localhost:5552"/>
  <clientmap subnet="127.0.0.1" mask="0.0.0.0">
    <assignedfsc fscid="fsc2" />
  </clientmap>
</fscgroup>

<fscgroup id="g4">
  <fsc id="fsc4" address="http://localhost:5554">
    <volume id="testvol" root="$FMS_TESTFILE_DIR" />

    <transientvolume id="transvol" root="$FMS_TESTFILE_DIR" />

    <accession id="myvolAearh---sy2a---bHA" root="$FMS_TESTFILE_DIR" username="mrd"
      textfileencoding="UTF-8" lineterminationstyle="LF" />
    <accession id="testvolarh---sy2a---bHA" root="$FMS_TESTFILE_DIR" username="mrd"
      textfileencoding="ISO-8859-1"
      lineterminationstyle="CRLF" />
  </fsc>
</fscgroup>

  <linkparameters fromgroup="g2" togroup="g4" transport="wan" compression="gzip"
maxpipes="4" />
  <linkparameters fromgroup="g4" togroup="g2" transport="wan" compression="gzip"
maxpipes="4" />

</fmsenterprise>
</fmsworld>

```

Transport methods for compressed files

The transport method that FMS uses to send compressed files is determined by how you configure the **transport** and **compression** elements, file size, file extension, and network configuration.

Transport method	Description
Standard LAN download	Simple, single-stream download. Supports whole files and ranges. Best for high bandwidth/low latency networks.
Compressed LAN download	Single compressed stream. Supports whole files only. Best for compressed data.
WAN download	Multistream download. Supports whole files and ranges. Best for low bandwidth/high latency networks. Relies on HTTP range functionality.

The following table lists the transport method used to transport files based the following factors: compression parameter configuration, file type, and whether the file size is greater than the threshold value set for the **FCC_WANThreshold** element.

Transport value	Compression value	File can be compressed	File size greater than threshold	Transport method
Unset	Unset	N/A	N/A	Standard LAN download
lan	Unset	N/A	N/A	Standard LAN download
lan	gzip	Yes	No	Standard LAN download
lan	gzip	Yes	Yes	Compressed LAN download
lan	gzip	No	N/A	Standard LAN download
wan	Unset	N/A	No	Standard LAN download
wan	Unset	N/A	Yes	WAN download
wan	gzip	Yes	No	Standard LAN download
wan	gzip	Yes	Yes	Compressed LAN download
wan	gzip	No	No	Standard LAN download
wan	gzip	No	Yes	WAN download

Installing and configuring Data Share Manager

Data Share Manager installation overview

Teamcenter users can use Data Share Manager to asynchronously upload and download files, managing the processes by pausing, resuming, or canceling them. Data Share Manager is supported on the rich client, and is a separate program with its own user interface.

Data Share Manager can be installed on Windows, Linux, and Mac OS clients. It can be used to manage file uploads and downloads for the rich client. Data Share Manager must be installed if you require external site file virus scanning.

Note:

If a client has a previous version of Data Share Manager installed, **uninstall the earlier version** before installing the new version of Data Share Manager.

Follow this process to install Data Share Manager:

1. Data Share Manager must be installed with a public key that grants access to the Teamcenter database for end users to use Data Share Manager. Provide public keys using either of the following methods:
 - Provide separate key files to end users using the instructions in **Export a public key for the data share manager**. (This is the only way to distribute keys on non-Windows platforms.)
 - Bundle the key files with the Windows-based Data Share Manager stand-alone installer so the keys are included when end users **run the Windows stand-alone installer**. Refer to **Bundle keys with the Windows stand-alone installer for the Data Share Manager** for instructions on bundling key files with the installer.
2. Set the **Use_DataShare_Manager** preference to **true** on the server. Doing so activates Data Share Manager for file uploading and downloading.

Tip:

If users prefer to the synchronous import and export mechanism for file upload and download instead of using Data Share Manager, they can set the **Use_Datashare_Manager** preference to **false** with a **User** protection scope.

3. If the client has a previous version of Data Share Manager installed, **uninstall the earlier version** before installing the new version of Data Share Manager.

Install Data Share Manager using the **stand-alone installer for your platform**.

Export a public key for the data share manager

Data Share Manager must be installed with a public key that grants access to the Teamcenter database. If this key has not been installed, users will receive the an error message stating "The Task File "*file-name.plmd*" required the key file "*key-file-name.x509*", which was missing".

By default, each Teamcenter database contains a private/public key pair that validates Data Share Manager clients accessing the database. Export and install the public key as follows.

1. Export the public key using the utility:

```
install_datasharekeys -u=username -p=password -g=dba -f=exp_pubkey
```

This exports a ***.pem** file and a ***.x509** key file.

2. Zip the resulting ***.x509** key file into a **keys.zip** file. (The **.pem** file is not needed in the ZIP file.)

Ensure the ***.x509** file is in a **\keys** directory in the **keys.zip** file so it can be properly read by installers.

3. If users need access to multiple databases from their client, export a public key for each database and add it to the **keys.zip** file.
4. Distribute the public keys to individuals who need to install the Data Share Manager in either of the following ways:

- **keys.zip** file

Distribute the **keys.zip** file to users and tell them to point to this file when they install the Data Share Manager using the stand-alone installer.

- Individual key files

Distribute the ***.x509** key files directly to end users and tell them to place the files into their **keys** directory after installation using the stand-alone installer. For example, on a Windows machine that uses the stand-alone installation, end users copy the key files to the **C:\Users\username\Siemens\dsmgr\dm\keys** directory.

Bundle keys with the Windows stand-alone installer for the Data Share Manager

Rather than **provide a separate key file to end users**, an administrator can bundle the key file with the Windows-based Data Share Manager stand-alone installer. Then the keys are already included when end users **run the stand-alone installer**.

1. The system administrator exports the public key from the database with the **install_datasharekeys** utility by running the following from a Teamcenter command prompt:

```
install_datasharekeys -u=username -p=password -g=dba -f=exp_pubkey
```

The administrator zips the resulting ***.x509** key file from the database into a **keys.zip** file. If users need access to multiple databases from their client, the administrator must export a public key for each database and add it to the **keys.zip** file.

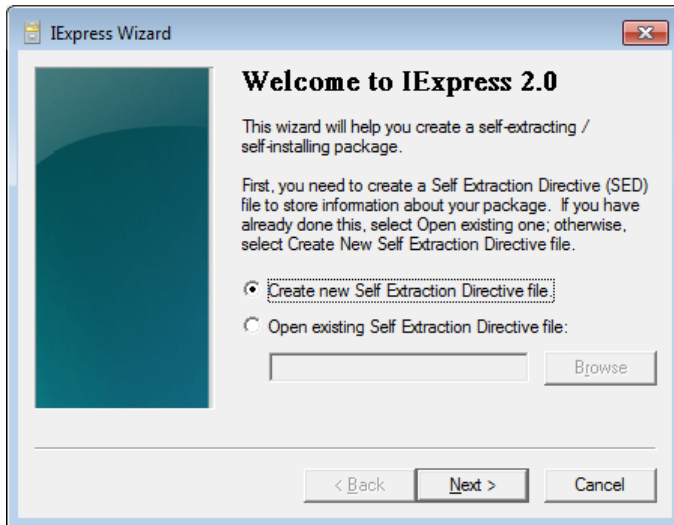
Tip:

The ***.x509** files must be included in the **keys.zip** file in a **\keys** directory so they can be properly read by installers.

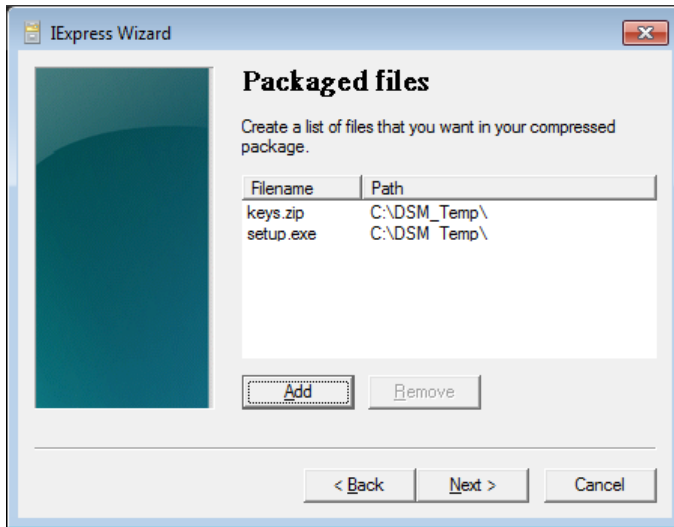
2. Copy the Data Share Manager stand-alone installer **setup.exe** file from the directory where it is located in the Teamcenter installation source (**additional_applications\dsm_install**).

3. Place the **setup.exe** file and the **keys.zip** file into a temporary directory.
4. On a Windows system, run the **iexpress** command from a command prompt.

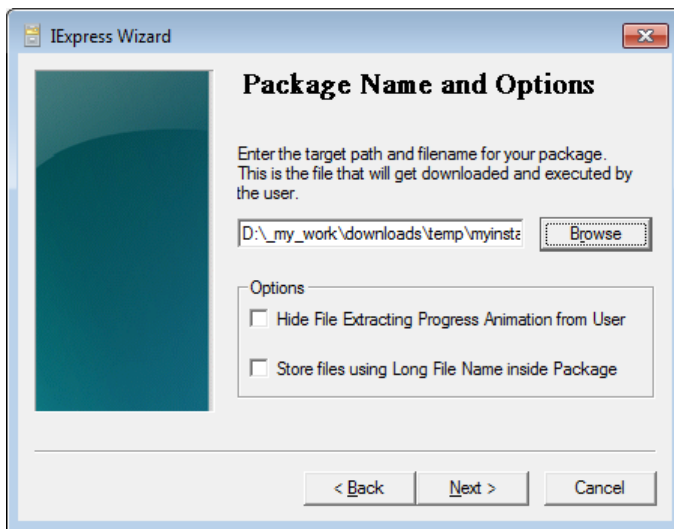
The IExpress wizard runs.



5. In the **Welcome to IExpress 2.0** dialog box, select **Create new Self Extraction Directive file** and click **Next**.
6. In the **Package purpose** dialog box, select **Extract files and run an installation command** and click **Next**.
7. In the **Package title** dialog box, type a title for the installer (such as **Datashare Manager Installation**) and click **Next**.
8. In the **Confirmation prompt** dialog box, select **No prompt** and click **Next**.
9. In the **License agreement** dialog box, select **Do not display a license** and click **Next**.
10. In the **Packaged files** dialog box, click **Add** button and select the **setup.exe** file and the **keys.zip** file from the temporary directory location. Click **Next**.



11. In the **Install Program to Launch** dialog box, select **setup.exe** and click **Next**.
12. In the **Show window** dialog box, select **Default** and click **Next**.
13. In the **Finished message** dialog box, select **No message** and click **Next**.
14. In the **Package Name and Options** dialog box, click the **Browse** button and select a temporary directory where to place the installer. Name the installer file with a temporary name such as **myinstaller.exe**. This is to differentiate it from the original **setup.exe** file. Click **Next**.



15. In the **Configure restart** dialog box, select **No restart** and click **Next**.
16. In the **Save Self Extraction Directive** dialog box, click the **Browse** button and select a temporary directory where to place the installer directive file. Click **Next**.

17. In the **Create package** dialog box, click **Next**.

The install package file is created.

18. When you look in the temporary directory where the output files were placed, you should see two new files, for example, **myinstaller.EXE** and **myinstaller.SED**.

If the original **setup.exe** file is in this directory, rename it to something different, such as **setup_original.exe**. Then rename the packaged installer file (for example, **myinstaller.EXE**) to **setup.exe**. This new **setup.exe** file contains the **keys.zip** file.

19. Distribute the new **setup.exe** file to end users. When they run the file to install the Data Share Manager, the bundled **keys.zip** file is unzipped into the **\keys** directory.

Install the Data Share Manager with the stand-alone installer

Use the stand-alone installer to install the Data Share Manager on individual client machines.

1. Ensure the **Use_DataShare_Manager** preference is set to **true** on the server or locally as described in [Data Share Manager installation overview](#).
2. If your client has a previous version of the Data Share Manager installed, **uninstall the earlier version** before installing the new version of the Data Share Manager.
3. Browse to the **additional_applications\dsm_install** folder in the Teamcenter installation source.
4. Run the installation file for your platform.
 - Windows: *setup.exe*

By default, Data Share Manager is installed to be available to only the user installing Data Share Manager. To have Data Share Manager available for all users on the workstation, run *setup.exe* with the **/v** option and specify an installation directory. For example:

```
setup.exe /v"INSTALLDIR=c:\apps\dsm_all_users"
```

- Linux: *install.bin*
 - Mac OS: *install.bin*
5. Follow the prompts in the installation wizard to install the Data Share Manager.
 - On Windows, if your administrator has provided you with a **public key file (keys.zip)**, the wizard will use this key file automatically if it is stored in the same directory as *setup.exe*. If the key file is not found in the same directory as *setup.exe*, the wizard will prompt you to choose a security key file. If you know the key file location, check **Enter Security Key File**, browse to the key file, and

select the file. Leave **Enter Security Key File** unchecked if you do not know the key file location. The key file can be installed after Data Share Manager is installed.

On Linux and Mac OS systems, you have the option to choose either a **Typical** or a **Custom** setup type. If your administrator has provided you with a public key file (*keys.zip*), the wizard will use this key file automatically if it is stored in the same directory as *install.bin* when you select a **Typical** install set.

- If your administrator has provided you with individual ***.x509** public key files, choose the **Complete** setup type. (If choosing the **Custom** setup type, leave **Enter Security Key File** unchecked.)

Copy the ***.x509** public key files into the **keys** directory after installing Data Share Manager. For example, on a Windows workstation, copy the key files to the **C:\Users\username\Siemens\dsmgr\dm\keys** directory.

6. Once the installation wizard finishes, log off and log on again to complete installation.

After logon, the Data Share Manager icon  is displayed in the system tray.

Handling possible installation errors

If a previous version of Data Share Manager was installed, you may encounter an error stating the older version of Data Share Manager cannot be uninstalled. This is a known issue with InstallAnywhere, the underlying installation program. Use the following steps as a workaround:

1. Navigate to the following directory:

(Windows): *ProgramFiles\Zero G Registry*

(Linux) *\$HOME/*

2. In that directory, locate the file *.com.zerog.registry.xml*. You may need to enable viewing hidden files.
3. Create a backup copy of *.com.zerog.registry.xml*. The file may be required by other InstallAnywhere-based installations, such as Teamcenter client communication system (TCCS).
4. In *.com.zerog.registry.xml*, use an ASCII editor to locate and remove the entries for Data Share Manager and its components:

```
<product name="Data_Share_Manager" id="8ca639e8-1f16-11b2-8d72-974fa857b767"
  upgrade_id="8ca639e8-1f16-11b2-8d72-974fa857b767" version="11.2.0.3"
  copyright="2014" info_url="" support_url="" location="C:\Users\<userid>\
  Siemens\dsmgr"
  last_modified="2015-07-30 10:53:00">
```

```

<![CDATA[]]>
  <vendor name="Siemens PLM Solutions" id="54cad4f2-1f16-11b2-
bbc2-974fa857b767"
    home_page="" email=""/>
  <feature short_name="Application" name="Application"
    last_modified="2015-07-30 10:53:00">
    <![CDATA[This installs the application feature.]]>
    <component ref_id="54cad522-1f16-11b2-bbef-974fa857b767"
version="1.0.0.0"
    location="C:\Users\<userid>\Siemens\dsmgr\
    _Data_Share_Manager_installation\
    Change Data_Share_Manager Installation.exe"/>
    <component ref_id="54cad521-1f16-11b2-bbed-974fa857b767"
version="1.0.0.0"

location="C:\Users\<userid>\Siemens\dsmgr\dm\com\teamcenter\fms\dlmgr\
    ui\controls\dialogs.css"/>
    </feature>
    <feature short_name="Help" name="Help" last_modified="2015-07-30
10:53:00">
    <![CDATA[This installs the help feature.]]>
    <component ref_id="54cad522-1f16-11b2-bbef-974fa857b767"
version="1.0.0.0"

location="C:\Users\<userid>\Siemens\dsmgr\_Data_Share_Manager_installation\
    Change Data_Share_Manager Installation.exe"/>
    <component ref_id="54cad521-1f16-11b2-bbed-974fa857b767"
version="1.0.0.0"

location="C:\Users\<userid>\Siemens\dsmgr\dm\com\teamcenter\fms\dlmgr\
    ui\controls\dialogs.css"/>
    </feature>
  </product>

<component id="54cad522-1f16-11b2-bbef-974fa857b767" version="1.0.0.0"
  name="InstallAnywhere Uninstall Component"
  location="C:\Users\<userid>\Siemens\dsmgr\_Data_Share_Manager_installation\
  Change Data_Share_Manager Installation.exe" vendor="Siemens PLM
  Solutions"/>

<component id="54cad521-1f16-11b2-bbed-974fa857b767" version="1.0.0.0"
  name="Common" location="C:\Users\<userid>\Siemens\dsmgr\dm\com\teamcenter
  \fms\dlmgr\ui\controls\dialogs.css" vendor="Siemens PLM Solutions"/>

```

5. Save *.com.zerog.registry.xml*.
6. Restart the installation file for your platform.

Uninstall Data Share Manager

Note:

You must first close Data Share Manager to stop the Java process before uninstalling. For example, on a Windows system, right-click the Data Share Manager icon  and choose **Exit Data Share Manager**. On a Linux system, click the close button (X) in the upper-right corner of the Data Share Manager user interface and click **Yes** in the confirmation dialog box.

The way you uninstall Data Share Manager depends on how it was originally installed.

- Teamcenter Environment Manager (TEM)

- **Windows systems**

1. Use the Windows Task Manager to terminate the **dsm.exe** process.
2. Remove the **Data Share Manager** item from the Windows Start menu.
3. Delete the `TC_ROOT\dm` directory.

- **Linux systems**

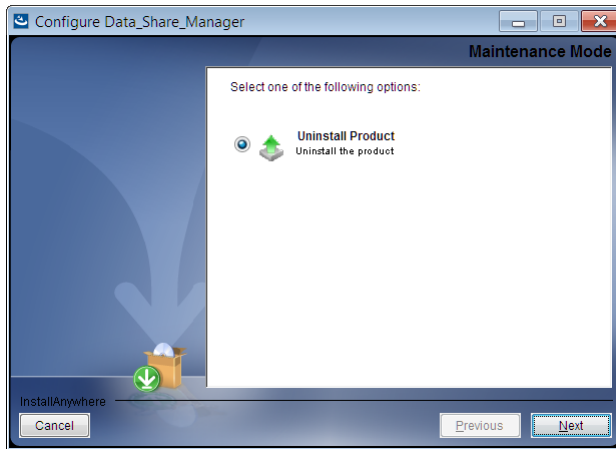
1. Use the **kill -9** command to stop the **dsm** process.
2. Remove the **DSM** entries from `$HOME/.mime` types and `$HOME/.mailcap`.
3. Delete the `TC_ROOT\dm` directory.

- Stand-alone installer

Use this method for stand-alone installations.

1. Browse to the location where Data Share Manager is installed. For example, on a Windows machine, browse to `C:\Users\username\Siemens\dsmgr\` directory.
2. In the `_Data_Share_Manager_installation` directory, run the **Change Data_Share_Manager Installation** file.

The uninstall dialog box is displayed.



3. Click **Next**.

Data Share Manager is uninstalled.

4. Log off and log back on to complete the uninstallation.

Configure Data Share Manager

Import a Data Share Manager private key into an existing database

There is only one active private key in the system that is used by Teamcenter. To import a private data share key to an already installed database, use the **install_datasharekeys** utility. This utility is initially called by the Teamcenter installation process to install the predefined private key.

To use the same key for more than one database (for example, after installing test environments), perform the following steps:

1. Export the private key from a database that already uses it (for example, the database that generated it):

```
install_datasharekeys -u=username p=password -f=exp_pvtkey
```

This generates the *private_keypairID.pem* file.

Caution:

Always keep the private key in a safe and secure place, inaccessible to unauthorized parties, preferably away from the public key files.

2. Import (seed) the private key into each of the other already-installed databases:

```
install_datasharekeys -u=username -p=password -f=seed -f=private_keypairID.pem
```

private_keypairID.pem is the private key file exported from the database in step 1.

Caution:

Always keep the private key in a safe and secure place, inaccessible to unauthorized parties, preferably away from the public key files.

3. Export the corresponding public key for distribution to the clients from any database that already has the private key installed (after step 1 or 2):

```
install_datasharekeys -u=username -p=password -f=exp_pubkey
```

This generates the *public_keypairID.pem* and *public_keypairID.x509* public key files. The *public_keypairID.x509* public key file can be freely distributed to Data Share Manager clients.

Enable SHA-256 digest algorithm for Datashare Manager

By default, the Datashare Manager uses the **MD5** digest algorithm for validating file contents. If the procedure Enable SHA-256 digest algorithm for the ticket digest has been performed on the Teamcenter environment, you can configure the Datashare Manager to use the more secure **SHA-256** digest algorithm.

In the *dm.properties* file, set the following:

```
com.teamcenter.fms.digest.dlmgr.upload=SHA-256
```

Configure PKI for the Data Share Manager

You can configure public key infrastructure (PKI) authentication for Data Share Manager to authenticate all uploads and downloads from Data Share Manager to FMS volumes. Authentication relies on two-way encrypted communication between the client (Data Share Manager) and the server (FMS), and requires the FMS system to be fronted by a reverse proxy configured to use two-way encrypted communications.

To enable PKI authentication for Data Share Manager, open the *Data-Share-Manager-installed-location\dm\dm.properties* file and set the **com.teamcenter.fms.servercache.ssl.enabled** parameter to **true**. Setting only this value to true without setting the keystore and truststore properties specifies that the certificates should be read from the browser store. This is the recommended setting. (Setting the value to **false** disables the authentication.) Setting the **com.teamcenter.fms.servercache.ssl.enabled** parameter to **false** (or letting it default to **false**) may cause PKI-authenticated reverse proxy connections to fail.

After PKI authentication is enabled, configure the following properties to use file-based keystore or truststore:

- **com.teamcenter.fms.servercache.ssl.keystore=keystore.jks**
- **com.teamcenter.fms.servercache.ssl.keystorepassword=password**

- `com.teamcenter.fms.servercache.ssl.keystoretype=jks|pkcs12`
- `com.teamcenter.fms.servercache.ssl.truststore=truststore.jks`
- `com.teamcenter.fms.servercache.ssl.truststorepassword=password`
- `com.teamcenter.fms.servercache.ssl.truststoretype=jks`

To allow self-signed certificates from untrusted parties, set the value of the `com.teamcenter.fms.servercache.ssl.allowuntrusted` parameter to **true**.

Warning:

Siemens Digital Industries Software does not recommend using self-signed (untrusted) certificates. This configuration is provided only for test purposes and should only be used if you are confident in the identity of the FMS (reverse proxy) URL and the integrity of your connection.

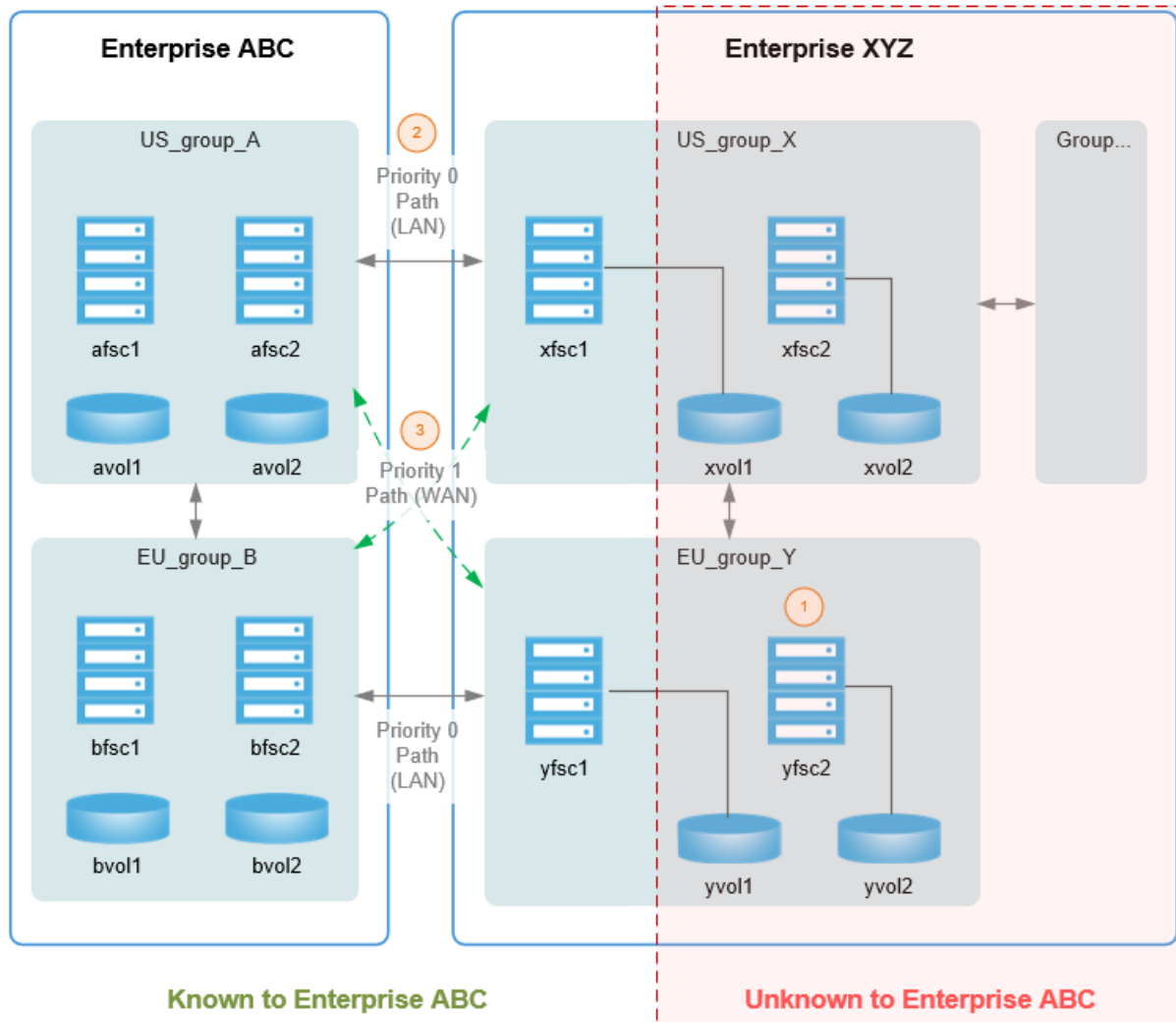
Routing FSCs between sites

You can define routes to a remote site based on the originating local **fscgroup** using WAN or LAN transport modes. This allows you to express multisite routing configuration information without exposing your entire network topology. Only the gateway FSCs in the remote site need to be known to the local site. Use this method to define routes to a geographically close FSC in a remote site.

In the following example, routes between geographically close groups are expressed using a priority scheme. (WAN routes can be used between geographically distant sites.)

Note:

1. Not all FSCs or volumes need to be known by Enterprise ABC.
2. multisiteimport for localfscgroup groupA transport LAN priority 0
3. multisiteimport for localfscgroup groupA transport WAN priority 1



Use the **fscgroupimport** XML element (in the **fmsmaster_fscid.xml** file) to define routes to a remote site based on your local FSC group information. Within this XML element, use the **defaultfsc** attribute to define the remote FSC address, ID, transport mode and priority for the route.

In the following example code, **fscgroupimport** defines routes to FSCs in a remote site for FSCs coming from a locally defined **fscgroup**.

```
<fmworld>
  <multisiteimport siteid="XYZ">
    <defaultfscimport fscid="xfsc1" address="http://127.100.0.1:5101" priority="0"/>
    <defaultfscimport fscid="yfsc1" address="http://127.100.0.1:5102" priority="1"/>
    <fscgroupimport groupid="US_group_A">
      <!-- only address or fscid is needed not both -->
      <!-- defaultfsc [ address="http://127.100.0.1:5101" | fscid="xfsc1" ]
```

```

priority="0"/ -->
    <defaultfsc address="http://127.100.0.1:5101" priority="0"/>
    <defaultfsc fscid="yfsc1" priority="1" transport="wan" maxpipes="4"/>
</fscgroupimport>
<fscgroupimport groupid="EU_group_B">
    <defaultfsc fscid="yfsc1" priority="0"/>
    <defaultfsc fscid="xfsc1" priority="1" transport="wan" maxpipes="4"/>
</fscgroupimport>
</multisiteimport>
<fmsenterprise id="ABC">
    <fscgroup id="US_group_A">
        <fsc id="afsc1" address="http://127.0.0.1:4101">
            <volume id="avol1" root="/avol1"/>
        </fsc>
        <fsc id="afsc2" address="http://127.0.0.1:4102">
            <volume id="avol2" root="/avol2"/>
        </fsc>
    </fscgroup>
    <fscgroup id="EU_group_B">
        <fsc id="bfsc1" address="http://127.0.0.1:4201">
            <volume id="bvol1" root="/bvol1"/>
        </fsc>
        <fsc id="bfsc2" address="http://127.0.0.1:4202">
            <volume id="bvol2" root="/bvol2"/>
        </fsc>
    </fscgroup>
</fmsenterprise>

</fmsworld>

```

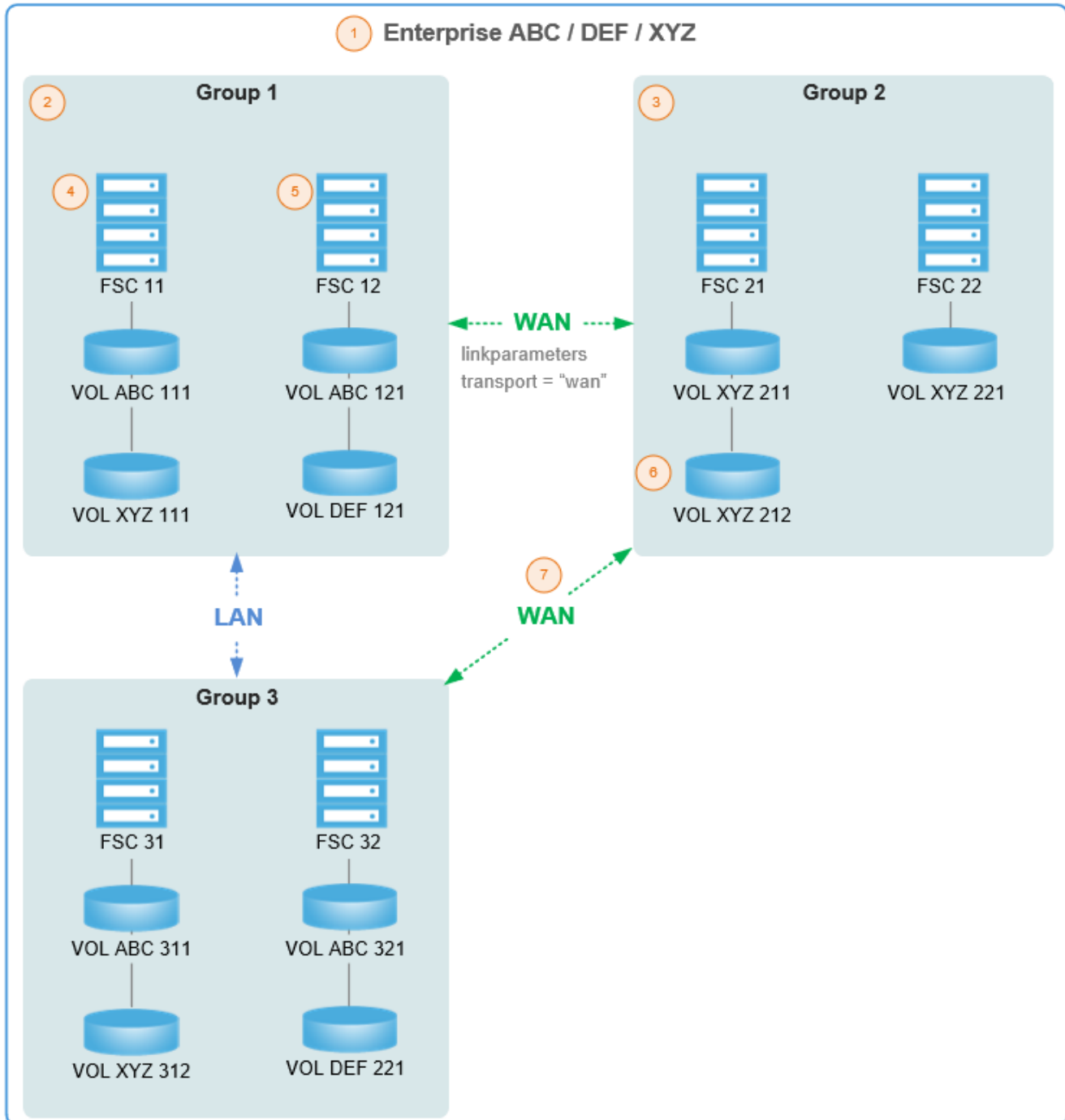
Accessing remote volumes using aliases (shared network)

You can configure FSCs at your local site to access volumes managed by another site. In this case, the FSC essentially becomes capable of managing volumes owned by the remote site. In this configuration, FMS can take advantage of your local configuration, including your WAN transport capability.

In this shared network example, the **fmsenterprise** element (in the **fmsmaster_fscid.xml** file) is used to map other sites to the local site. In this scenario, certain FSCs are shared and capable of managing volumes owned by any defined site. Multiple sites and databases can be supported by a single FMS configuration.

Note:

1. Multiple enterprises share a common FMS configuration.
2. Some groups or FSCs may have volumes for more than one enterprise.
3. Other groups or FSCs may have volumes for only one enterprise.
4. An FSC can mount volumes from any enterprise.
5. Any FSC can handle requests for any enterprise.
6. Any volume is managed (owned) by one of the enterprises.
7. All caching, transport configuration, and routing is common.



In the following example code, additional sites are added to the configuration using the **fmsenterprise** element (**DEF** and **XYZ**):

```
<fmsworld>
  <fmsenterprise id="ABC" >
    <fmsenterprise id="DEF"/>
    <fmsenterprise id="XYZ"/>

    <fscgroup id="group1">
      <fsc id="fsc11" address="http://fsc11:4544">
        <volume id="volABD111" root="c:/volumes/volABD111"/>
        <volume id="volXYZ111" root="c:/volumes/volXYZ111"/>
      </fsc>
      <fsc id="fsc12" address="http://fsc12:4544">
        <volume id="volABD121" root="c:/volumes/volABD121"/>
        <volume id="volDEF121" root="c:/volumes/volDEF121"/>
      </fsc>
    </fscgroup>

    <fscgroup id="group2">
      <fsc id="fsc21" address="http://fsc21:4544">
        <volume id="volXYZ211" root="c:/volumes/volXYZ211"/>
        <volume id="volXYZ212" root="c:/volumes/volXYZ212"/>
      </fsc>
      <fsc id="fsc22" address="http://fsc12:4544">
        <volume id="volXYZ221" root="c:/volumes/volXYZ221"/>
      </fsc>
    </fscgroup>

    <fscgroup id="group3">
      <fsc id="fsc31" address="http://fsc31:4544">
        <volume id="volABD311" root="c:/volumes/volABD311"/>
        <volume id="volABC312" root="c:/volumes/volABC312"/>
      </fsc>
      <fsc id="fsc32" address="http://fsc32:4544">
        <volume id="volABD321" root="c:/volumes/volABD321"/>
        <volume id="volDEF321" root="c:/volumes/volDEF321"/>
      </fsc>
    </fscgroup>

    <linkparameters fromgroup="group1" togroup="group2" transport="wan"
maxpipes="4"/>
    <linkparameters fromgroup="group2" togroup="group1" transport="wan"
maxpipes="4"/>
    <linkparameters fromgroup="group2" togroup="group3" transport="wan"
maxpipes="8"/>
    <linkparameters fromgroup="group3" togroup="group2" transport="wan"
maxpipes="8"/>
  </fmsenterprise>
</fmsworld>
```

FMS monitoring

Introduction to File Management System monitoring

For File Management System (FMS) events, you can configure the following metrics to provide specified levels of monitoring of specified events levels. Optionally, you can receive email notification when specified metrics cross specified thresholds.

Configure FMS monitoring using either:

- `FSC_HOME/monitoring/fscMonitorConfig.xml`
- **FMS monitoring administrative interface**

Metric	Description
All Routes Failed	All routes to resources in the FMS topology are inaccessible.
Expired Ticket	The FSC Server encountered a expired ticket.
Generic Error	The FSC server threw a general error.
Invalid Ticket	The FSC server encountered a invalid ticket.
Memory Collection Threshold Exceeded	The Java virtual machine has detected that the memory usage of a memory pool is exceeding the collection usage threshold.
Memory Usage Threshold Exceeded	The Java virtual machine detects that the memory usage of a memory pool is exceeding the usage threshold.
No Route Error	A client or FCC is connector to the wrong FSC server process.
Periodic Checks	The periodic FSC network, local volume, performance, and configuration checks has detected an issue.

Metric	Description
Quarantined Dead Link	A link between two resources in the FSC network topology was quarantined.
Remote Admin Not Supported	The Fms_BootStrap_Urls preference value is incorrect.

For each metric the following information is collected:

- Date and time
- FSCID
- Message
- Log

Each alert notification contains:

- Date
- Message
- Possible causes
- Recommended actions

The MLD holder is used for the **All Routes Failed** metric and the alert notification trigger is a single event occurrence. The countable MLD holder is used for all other metrics and the alert notification is triggered when the number of events is greater than threshold values in a specified time period.

The **FSC_Critical_Events_Monitoring_Summary** MBean consolidates the event metrics and their corresponding values of the FSC process for display in one window of the Java EE administrative interface. The **FSC_Critical_Events_Monitoring_Configuration** MBean contains the **Health_monitoring_mode** attribute that you use to enable or disable the monitoring system. The **FSCHealthDiagnostics** MBean performs periodic health checks and critical event reasoning.

Tip:

You should review all monitoring settings, ensuring the thresholds are set correctly for your site.

If you do not know the optimum monitoring setting for any given critical event, set the value to **COLLECT**. Collect the data and review to determine normal activity levels. Then set notification values slightly higher than normal activity levels.

Tip:

The contents of the email notifications are generated from the *TC_ROOT/fsc/monitoring/fscMonitorConfigInfo.xml* file. (This is a companion file to the *fscMonitorConfig.xml* file.) For a complete list of possible causes and recommended actions for server manager monitoring, see this file.

Configure FMS monitoring with the *fscMonitorConfig.xml* file

This procedure describes how to configure FMS monitoring with a configuration file. You can also [configure FMS monitoring with an administrative interface](#).

1. Open the *FSC_HOME/monitoring/fscMonitorConfig.xml* file.
2. At the top of the file, set **mode** to one of the following:
 - **Normal**
Enables monitoring of all the metrics listed in the file.
 - **Disable_Alerts**
Enables monitoring of all the metrics listed in the file, but disables all notifications of critical events, regardless of individual notification settings on any metric.
 - **Off**
Disables monitoring of all the metrics listed in the file.
3. (Optional) To be notified when criteria reaches the specified threshold, specify from whom, to whom, and how frequently email notification of critical events are sent by setting the following **EmailResponder** values.

You can specify more than one **EmailResponder id**.

All **EmailResponder id** values in all subsequent monitoring metrics in this file must match one of the **EmailResponder id** values set here.

- **EmailResponder id**

Specify an identification for this email responder. Multiple email responders can be configured, in which case, the identifiers must be unique.

- **protocol**

Specify the email protocol by which notifications are sent. The only supported protocol is SMTP.

- **hostAddress**

Specify the server host from which the email notifications are sent. In a Multi-Site environment, the host address identifies the location of the critical events.

- **fromAddress**

Specify the address from which the notification emails are sent.

- **toAddress**

Specify the address to which the notification emails are sent.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress email notification of critical events.

- **emailFormat**

Specify the format in which the email is delivered. Valid values are **html** and **text**.

4. (Optional) To be notified when criteria reaches the specified threshold, specify to whom, and to which file, critical events are logged by setting the following **LoggerResponder** values.

All **LoggerResponder** values in all subsequent monitoring metrics in this file must match the **LoggerResponder id** value set here.

- **LoggerResponder id**

Specify an identification for this logger responder. Multiple logger responders can be configured, in which case, the identifiers must be unique.

- **logFileName**

Specify the name of the file to which critical events are logged.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress logging of critical events to the log file.

5. Configure the criteria for a critical event for any of the metrics in the file by:

- a. Specifying a particular **EmailResponder**, if desired.
- b. Specifying a particular **LoggerResponder**, if desired.
- c. Setting the metric's monitoring mode to one of the following:

- **Collect**

Collect metric data and display results in the MBean view (within the FMS administrative interface) for this metric.

This is the default setting.

- **Alert**

When critical events occur, collect metric data, send email notifications (set up with **EmailResponder** values), write messages to log files (set up with **LoggerResponder** values), and display results in the FMS administrative interfaces for this metric. Log file messages and email notifications are only issued when monitoring mode is set to **Alert**.

- **Off**

No metric data is collected.

- d. Setting the remaining values to specify criteria that must be met to initiate a critical event for the metric.

6. Save the file.

FMS monitoring is enabled for the metrics you configured.

Sample fscMonitorConfig.xml code

In the following example, the email notifications are sent to **admin1@company.com**.

```
<ApplicationConfig mode="Off" version="1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="healthMonitorV1.0.xsd">
<RespondersConfig>
  <EmailResponder id="EmailResponder1">
    <protocol value="smtp"/>
    <hostAddress value="svc1smtp.company.com"/>
    <fromAddress value="tcsys@company.com" />
    <toAddress value="admin1@company.com" />
    <suppressionPeriod value="4200"/>
    <emailFormat value="html"/>
  </EmailResponder>
  <LoggerResponder id="LoggerResponder1">
    <logFileName value="FSCMonitoring.log" />
  </LoggerResponder>
</RespondersConfig>
</ApplicationConfig>
```

```

        <suppressionPeriod value="60" />
    </LoggerResponder>
</RespondersConfig>
- <MetricsConfig>
- <Metric name="Quarantined Dead Link" id="QuarantinedDeadLink" maxEntries="100"
mode="Collect"
    metricType="integer" description="A link between two resources in FSC network topology was
    quarantined.">
- <ThresholdWithPeriod>
    <ThresholdValue name="NumberOfQuarantinedDeadLinks" value="2" description="If number of
timeouts
    exceeds this limit, notify the administrator." />
    <ThresholdPeriod name="TimePeriodInSec" value="6000" description="Periods for which
timeouts
    will be monitored." />
</ThresholdWithPeriod>
- <Responders>
    <ResponderRef id="EmailResponder1" />
    <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
- <Metric name="All Routes Failed" id="AllRoutesFailed" maxEntries="100" mode="Collect"
    metricType="grave" description="All Routes to resources in FMS Network topology is down.
    Something very bad has happened causing the FSC process to malfunction.">
- <Responders>
    <ResponderRef id="EmailResponder1" />
    <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
- <Metric name="No Route Error" id="NoRouteError" maxEntries="100" mode="Collect"
    metricType="integer" description="Client or FCC is connected to wrong FSC Server
process.">
- <ThresholdWithPeriod>
    <ThresholdValue name="NumberOfNoRouteError" value="2" description="If number of timeouts
exceeds
    this limit, notify the administrator." />
    <ThresholdPeriod name="TimePeriodInSec" value="6000" description="Periods for which
timeouts
    will be monitored." />
</ThresholdWithPeriod>
- <Responders>
    <ResponderRef id="EmailResponder1" />
    <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
- <Metric name="Remote Admin Not Supported" id="RemoteAdminNotSupported" maxEntries="100"
    mode="Collect" metricType="integer" description="The Fms_BootStrap_Urls value is
incorrect.">
- <ThresholdWithPeriod>
    <ThresholdValue name="NumberOfRemoteAdminNotSupported" value="2" description="If number of
timeouts exceeds this limit, notify the administrator." />
    <ThresholdPeriod name="TimePeriodInSec" value="6000" description="Periods for which
timeouts will be monitored." />
</ThresholdWithPeriod>
- <Responders>
    <ResponderRef id="EmailResponder1" />
    <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
- <Metric name="Memory Collection Threshold Exceeded" id="MemoryCollection" maxEntries="100"

```

```

    mode="Collect" metricType="integer" description="Java virtual machine detects that the
memory usage
    of a memory pool is exceeding collection usage threshold.">
- <ThresholdWithPeriod>
  <ThresholdValue name="NumberOfMemoryCollection" value="1" description="If memory collection
    exceeds this limit, notify the administrator." />
  <ThresholdPeriod name="TimePeriodInSec" value="6000" description="Periods for which
timeouts
    will be monitored." />
</ThresholdWithPeriod>
- <Responders>
  <ResponderRef id="EmailResponder1" />
  <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
- <Metric name="Memory Usage Threshold Exceeded" id="MemoryUsage" maxEntries="100"
mode="Collect"
  metricType="integer" description="Java virtual machine detects that the memory usage of a
memory
  pool is exceeding usage threshold.">
- <ThresholdWithPeriod>
  <ThresholdValue name="NumberOfMemoryUsage" value="1" description="If memory collection
exceeds
  this limit, notify the administrator." />
  <ThresholdPeriod name="TimePeriodInSec" value="6000" description="Periods for which
timeouts
    will be monitored." />
</ThresholdWithPeriod>
- <Responders>
  <ResponderRef id="EmailResponder1" />
  <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
- <Metric name="Generic Error" id="GenericError" maxEntries="100" mode="Collect"
metricType="integer"
  description="A general error was thrown from FSC Server.">
- <ThresholdWithPeriod>
  <ThresholdValue name="NumberOfGenericError" value="10" description="If number of timeouts
exceeds
  this limit, notify the administrator." />
  <ThresholdPeriod name="TimePeriodInSec" value="6000" description="Periods for which
timeouts
    will be monitored." />
</ThresholdWithPeriod>
- <Responders>
  <ResponderRef id="EmailResponder1" />
  <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
- <Metric name="Expired Ticket" id="ExpiredTicket" maxEntries="100" mode="Collect"
metricType="integer"
  description="FSC Server encountered a expired ticket.">
- <ThresholdWithPeriod>
  <ThresholdValue name="NumberOfExpiredTicket" value="2" description="If number of timeouts
exceeds
  this limit, notify the administrator." />
  <ThresholdPeriod name="TimePeriodInSec" value="6000" description="Periods for which
timeouts
    will be monitored." />
</ThresholdWithPeriod>

```

```

- <Responders>
  <ResponderRef id="EmailResponder1" />
  <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
- <Metric name="Invalid Ticket" id="InvalidTicket" maxEntries="100" mode="Collect"
  metricType="integer" description="FSC Server encountered a invalid ticket.">
- <ThresholdWithPeriod>
  <ThresholdValue name="NumberOfInvalidTicket" value="1" description="If number of timeouts
exceeds
  this limit, notify the administrator." />
  <ThresholdPeriod name="TimePeriodInSec" value="6000" description="Periods for which
timeouts
  will be monitored." />
</ThresholdWithPeriod>
- <Responders>
  <ResponderRef id="EmailResponder1" />
  <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
- <Metric name="Periodic Checks" id="PeriodicChecks" maxEntries="100" mode="Collect"
  metricType="integer" description="Periodic FSC Server health checks has detected an
issue.">
- <ThresholdWithPeriod>
  <ThresholdValue name="NumberOfPeriodicChecks" value="2" description="If number of timeouts
exceeds
  this limit, notify the administrator." />
  <ThresholdPeriod name="TimePeriodInSec" value="6000" description="Periods for which
timeouts
  will be monitored." />
</ThresholdWithPeriod>
- <Responders>
  <ResponderRef id="EmailResponder1" />
  <ResponderRef id="LoggerResponder1" />
</Responders>
</Metric>
</MetricsConfig>
</ApplicationConfig>

```

Configure FMS monitoring with the administrative interface

You can use a web browser interface to manage FMS operations. This procedure describes how to configure FMS monitoring with the administrative interface.

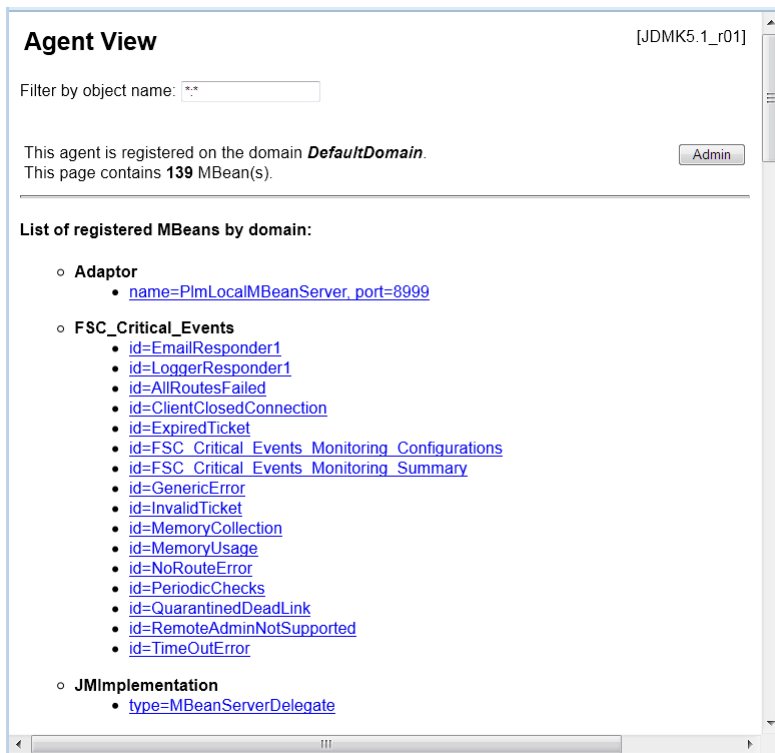
You can also [configure FMS monitoring with a configuration file](#).

Tip:

Because this functionality is provided through JMX Beans, any JMX client can access the FSC monitoring data. Java provides free JMX clients such as JConsole and JVisualVM that can be used for this purpose.

1. Start the FMS monitoring administrative interface.

By default, the address is **http://host-name:8999**.



2. Under the **FSC_Critical_Events** heading, click **id=FSC_Critical_Events_Monitoring_Configurations**.

This view lists the monitoring mode and all the metrics available for monitoring.

3. Set the **Health_monitoring_mode** value to one of the following:

- **Normal**

Enables monitoring of all the metrics listed in the file.

- **Disable_Alerts**

Enables monitoring of all the metrics listed in the file, but disables all notifications of critical events, regardless of individual notification settings on any metric.

- **Off**

Disables monitoring of all the metrics listed in the file.

4. Click **Apply** and click **Back to Agent View**.
5. In the **Agent View** under the **FSC_Critical_Events** heading, click the **id=EmailResponder1** value.

6. (Optional) To be notified when criteria reaches the specified threshold, specify from whom, to whom, and how frequently email notification of critical events are sent by setting the following **EmailResponder1** values.

All **EmailResponder1** values in all child monitoring metrics must match the values set here.

- **From_address**

Specify the address from which the notification emails are sent.

- **Host_address**

Specify the server host from which the email notifications are sent. In a Multi-Site environment, the host address identifies the location of the critical events.

- **Suppression_period**

Specify the amount of time (in seconds) to suppress email notification of critical events.

- **To_address**

Specify the address to which the notification emails are sent. You can specify multiple email addresses, separated by a semicolon (;).

7. Click **Apply**.
8. Click **Back to Agent View**.
9. Under the **FSC_Critical_Events** heading, click **id=LoggerResponder1**.
10. (Optional) To be notified when criteria reaches the specified threshold, specify to whom, and to which file, critical events are logged by setting the following **LoggerResponder** values.

All **LoggerResponder** values in all child monitoring metrics must match the **LoggerResponder** values set here.

- **Log_filename**

Specify the name of the file to which critical events are logged.

- **Suppression_period**

Specify the amount of time (in minutes) to suppress logging of critical events to the log file.

11. Click **Apply**.

Start the FMS monitoring administrative interface

The web application collects a variety of metrics on FMS events that are relevant to performance and the health of the FMS environment.

1. Ensure the web tier is installed (using Deployment Center or TEM).
2. Open the **pref_export.xml.template** file in the **FSC_HOME** directory.
3. Locate the **RunHtmlAdapter** key entry and ensure its value is set to **true**.
4. Add the following elements to the file following the **RunHtmlAdapter** setting:

```
<entry key="HtmlServerLogin" value="plmmonitor" />
<entry key="HtmlServerPassword" value="localhost" />
```

5. Save the modified file as **pref_export.xml**.
6. Open the **startfsc** file in the **FSC_HOME** directory.
7. Add the following to the VM parameters after the **-Dcom.sun.management.jmxremote** parameter and save the file.

```
-Dcom.teamcenter.mld.runadapter=yes -Dcom.teamcenter.mld.jmx.
HtmlServerLogin=plmmonitor -Dcom.teamcenter.mld.jmx.HtmlServerPassword=localhost
-Dcom.teamcenter.mld.jmx.HtmlServerPort=8999
```

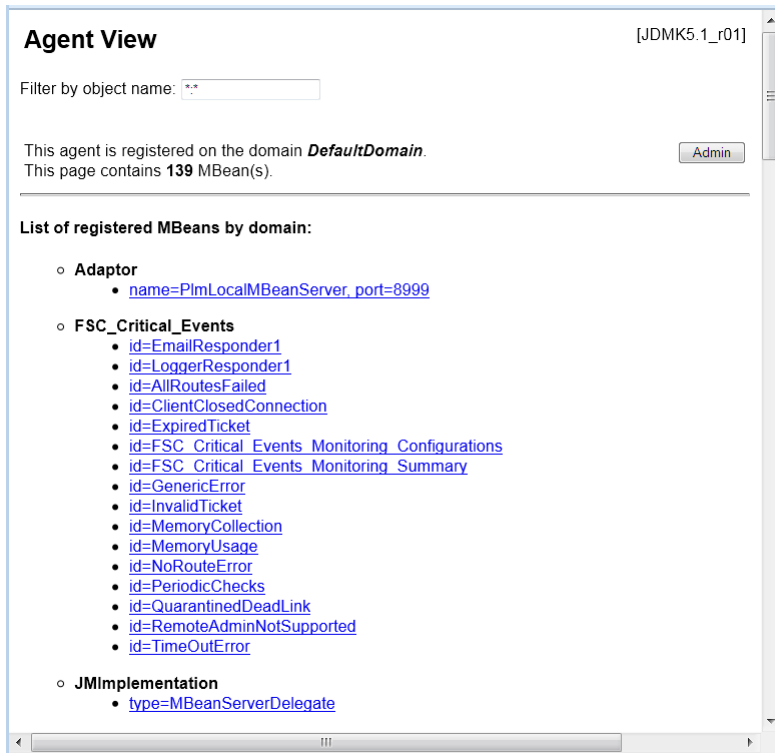
8. Restart the FSC and ensure there are no errors.
9. Open a web browser and type **http://application-server-host:8999**.

Note:

You can change the port number by changing the **HtmlAdapterPort** value in the **pref_export.xml** file and the **HtmlServerPort** value in the VM parameters you added to the **startfsc** file if necessary.

10. Log on using the default user name and password **plmmonitor** and **localhost**, respectively.

The web application metrics are displayed as in the following example:



11. Click any of the links for the listed metric MBean.

Verifying that files are not corrupted during transmission

You can verify that files are not corrupted during transmission to, from, and within Teamcenter by setting parameters in File Management System (FMS). This verification process computes message digests (complex checksums) of files before and after transmission, which are then compared to verify file integrity.

Content verification must be done in FMS between any two endpoints that transmit files. For example, the system could verify a file's content when uploading a file from a Teamcenter client to a Teamcenter volume. Endpoints can include FMS clients like the rich client, Lifecycle Visualization, NX, and volume server. There may even be more intermediate points like intermediate file cache servers.

To enable file content verification, you must set parameters in FMS configuration files such as the *fmsmaster_fscid.xml* file and the *fsc.properties* file.

Tip:

The *fmsmaster_fscid.xml* file and *fsc.properties* file are located in the *FSC_HOME* directory (for example, *TC_ROOT\fsc*). All the available parameters can be found in the corresponding template files, for example, *fmsmaster.xml.template*.

To generate new file digests, run the **generate_file_digests** utility.

Following are file content verification properties used by FMS server cache (FSC). These properties are set typically in the *fmsmaster_fscid.xml* file and possibly in the *fsc.xml* file for special cases. You can find these properties listed in the **fsc.xml.template** file located in the *FSC_HOME* directory (for example, *TC_ROOT\fsc*).

Name	Valid values	Description
FSC_EnableUploadValidation	true	Enables upload data validation. Uses whatever supported digest the client supplies.
	false	Disables upload data validation. Digests sent by clients are ignored. This is the default value.
FSC_EnableDownloadValidation	true	Enables download data validation. Uses the digest stored in the volume or cache.
	false	Disables download data validation. Any digest stored in the volume or cache is ignored. This is the default value.
FSC_TransferDigestAlgorithm	None	Disables transfer data validation. This is the default value.
	MD5	Requests MD5 digests on file transfers. Uses whatever digest is returned from file origin.
	SHA-1	Requests SHA-1 digests on file transfers. Uses whatever digest is returned from file origin.
	SHA-256	Requests SHA-256 digests on file transfers. Uses whatever digest is returned from file origin.
FSC_TransferRemoteMetadataFile	true	Enables transferring metadata files used in the validation process. This is the default value.
	false	Disables transferring metadata files.

Following is an example of these properties used in an *fmsmaster_fscid.xml* file:

```
<fscdefaults>
...
<property name="FSC_EnableUploadValidation" value="true" overridable="true" />
<property name="FSC_EnableDownloadValidation" value="true" overridable="true" />
<property name="FSC_TransferDigestAlgorithm" value="MD5" overridable="true" />
<property name="FSC_TransferRemoteMetadataFile" value="true" overridable="true"/>
...
</fscdefaults>
```

Following are file content verification properties used by the FMS client cache (FCC). These properties are set typically in the *fmsmaster_fscid.xml* file and possibly in the *fcc.xml* file for special cases. You can find these properties listed in the **fcc.xml.template** file located in the *FCC_HOME* directory (for example, *TC_ROOT\fcc*).

Name	Valid values	Description
FCC_UploadDigestAlgorithm	None	Disables upload data validation. This is the default value.
	MD5	Computes and sends MD5 digests on file uploads.
	SHA-1	Computes and sends SHA-1 digests on file uploads.
	SHA-256	Computes and sends SHA-256 digests on file uploads.
FCC_DownloadDigestAlgorithm	None	Disables download data validation. This is the default value.
	MD5	Requests MD5 digests on file downloads. (Uses whatever supported digest the FSC supplies.)
	SHA-1	Requests SHA-1 digests on file downloads. (Uses whatever supported digest the FSC supplies.)
	SHA-256	Requests SHA-256 digests on file downloads. (Uses whatever supported digest the FSC supplies.)

Following is an example of these properties used in an *fmsmaster_fscid.xml* file:

```
<fccdefaults>
...
  <property name="FCC_UploadDigestAlgorithm" value="MD5" overridable="true" />
  <property name="FCC_DownloadDigestAlgorithm" value="MD5" overridable="true" />
...
</fccdefaults>
```

Following are file content verification properties used by **FSCClientProxy**, **FSCJavaClientProxy** and **FSCNativeClientProxy**. You can set these properties in the *fsc.properties* file (for **JNI FSCClientProxy**), the *fsc.clientagent.properties* file (for **FSCNativeClientProxy** or **JNI FSCClientProxy**), and in the **System.getProperties()** call (for **FSCJavaClientProxy**). (For **JNI FSCClientProxy**, the *fsc.clientagent.properties* file settings override those found in the *fsc.properties* file.)

Name	Valid values	Description
com.teamcenter.fms.digest.upload	None	Disables upload data validation. This is the default value.
	MD5	Computes and sends MD5 digests on file uploads.
	SHA-1	Computes and sends SHA-1 digests on file uploads.
	SHA-256	Computes and sends SHA-256 digests on file uploads.
com.teamcenter.fms.digest.download	None	Disables download data validation. This is the default value.

Name	Valid values	Description
	MD5	Requests MD5 digests on file downloads. (Uses whatever supported digest the FSC supplies.)
	SHA-1	Requests SHA-1 digests on file downloads. (Uses whatever supported digest the FSC supplies.)
	SHA-256	Requests SHA-256 digests on file downloads. (Uses whatever supported digest the FSC supplies.)
com.teamcenter.fms.digest.transfer	None	Disables transfer data validation. This is the default value.
	MD5	Requests MD5 digests on file transfers. Uses whatever digest is returned from file origin.
	SHA-1	Requests SHA-1 digests on file transfers. Uses whatever digest is returned from file origin.
		Requests SHA-256 digests on file transfers. Uses whatever digest is returned from file origin.

Following is an example of these properties used in an *fsc.properties* file:

```
com.teamcenter.fms.digest.upload=MD5
com.teamcenter.fms.digest.download=MD5
com.teamcenter.fms.digest.transfer=MD5
```

Following are file content verification properties used by FSCNetClientProxy and set in the *FSCNetClientProxynn.dll.config* file. *NET-version* represents a .NET version number. This file resides in the *FSC_HOME\lib* directory.

Name	Valid values	Description
com.teamcenter.fms.digest.upload	None	Disables upload data validation. This is the default value.
	MD5	Computes and sends MD5 digests on file uploads.
	SHA-1	Computes and sends SHA-1 digests on file uploads.
	SHA-256	Computes and sends SHA-256 digests on file uploads.
com.teamcenter.fms.digest.download	None	Disables download data validation. This is the default value.
	MD5	Requests MD5 digests on file downloads. (Uses whatever supported digest the FSC supplies.)

Name	Valid values	Description
	SHA-1	Requests SHA-1 digests on file downloads. (Uses whatever supported digest the FSC supplies.)
	SHA-256	Requests SHA-256 digests on file downloads. (Uses whatever supported digest the FSC supplies.)

Following is an example of these properties used in an *appSettings* file section of the configuration file:

```
<configuration>
  <appSettings>
    <add key="com.teamcenter.fms.digest.upload" value="MD5" />
    <add key="com.teamcenter.fms.digest.download" value="MD5" />
  </appSettings>
</configuration>
```

Enable or disable SHA-256 digest algorithm for the ticket digest

By default, Teamcenter and FMS use the **SHA256** digest algorithm for the ticket digest. Siemens Digital Industries Software recommends this setting. You can configure the Teamcenter system and FMS to use the less secure MD5 digest algorithm, although we do not recommend using this algorithm.

Note:

Sites that communicate with each other in a Multi-Site federation must use the same digest algorithm, one of MD5 or SHA-256.

SHA-256 ticket support was introduced in Teamcenter 13.1. For Multi-Site environments where even one of the environments is running a version of Teamcenter earlier than 13.1, SHA-256 tickets are not supported. Siemens Digital Industries Software recommends upgrading such environments to Teamcenter 13.1 or later, so they can use SHA-256 tickets.

Disable SHA-256 tickets

1. Set the Teamcenter preference **FMS_Use_SHA256_Tickets** to **false**.
2. In the **FMS primary xml file**, after the **fmsenterprise** element, remove the **ticketdigest** element or set its value to "MD5" as follows:

```
<fmsenterprise id="-1807608066" volumestate="normal">
  <ticketdigest value="MD5" />
```

3. In the **\$FSC_HOME/fscadmin.properties** file, remove the following line or set its value to "MD5" as follows:

```
com.teamcenter.fms.fscadmin.default.ticketdigest=MD5
```

Re-enable SHA-256 tickets

If you previously disabled SHA-256 tickets, you can re-enable them using the following steps:

1. Set the Teamcenter preference **FMS_Use_SHA256_Tickets** to **true**.
2. In the **FMS primary xml file**, after the **fmsenterprise** element, add or update the **ticketdigest** element and set its value to "SHA-256" as follows:

```
<fmsenterprise id="-1807608066" volumestate="normal">
<ticketdigest value="SHA-256" />
```

3. In the `$FSC_HOME/fscadmin.properties` file, add or update the following line, setting its value to "SHA-256" as follows:

```
com.teamcenter.fms.fscadmin.default.ticketdigest=SHA-256
```

Enable use of FIPS 140-2 cryptography in FMS

You can use a cipher list on FSC servers to enforce FIPS manually. If all the ciphers in the list (and therefore enabled for use by FSC) are FIPS-certified, the FSC is FIPS-compliant. For example, the following configuration forces all FSCs in the scope of this setting to use the **TLS_RSA_WITH_AES_128_CBC_SHA** cipher for their HTTPS connections:

```
<fscdefaults>
...
<property name="FSC_SSLEnabledCiphers" value="TLS_RSA_WITH_AES_128_CBC_SHA"
overridable="false"/>
...
</fscdefaults>
```

To specify more than one cipher, separate each cipher name with a comma. This allows negotiation of any of the listed ciphers for client connections. For example:

```
<fscdefaults>
<property name="FSC_SSLEnabledCiphers" value="TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_RC4_128_SHA,
TLS_ECDHE_RSA_WITH_RC4_128_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_RC4_128_MD5,
TLS_RSA_WITH_RC4_128_SHA" overridable="false"/>
</fscdefaults>
```

Caution:

It has not been determined whether any of the ciphers listed in the preceding examples are actually FIPS compliant.

The list of ciphers available and supported by any given JRE are not determined by Siemens Digital Industries Software. Refer to published Oracle documentation to determine which ciphers are available in various JRE versions.

The list of FIPS-compliant ciphers is not determined by Siemens PLM or any of its software. You must refer to published FIPS documentation to determine which of the supported ciphers are FIPS-compliant.

Improving FSC cache performance

You can improve Teamcenter file management performance by prepopulating your FSC caches with a given set of files. This is useful if your site regularly has a large number of users all accessing the same set of files simultaneously.

For example, if you have 50 designers all arriving at the same time in the morning, and they are all simultaneously accessing the same large CAD assemblies, prepopulating your FSC caches with the assembly files improves Teamcenter performance.

Prepopulate FSC caches using the **plmxml_export** and **load_fscache** utilities.

1. Extract the file information for cache prepopulation by running the **plmxml_export** utility, using the **justDatasetsOut** transfer mode to create a PLM XML file containing all external file references associated with the top-level item selected. For example:

```
plmxml_export -u=tc-admin-user -p=password -g=group -item=item-id -export_bom=yes
-transfermode=justDatasetsOut -xml_file=tickets.xml
```

2. Run the **load_fscache** utility to target the **fsc_ids** within the PLM XML file and prepopulate the cache. For example:

```
load_fscache -u=tc-admin-user -p=password -g=group -f=load
-plmxml=tickets.xml -fscargets=FSC-ID
```

Sample FMS configurations

About the sample FMS configurations

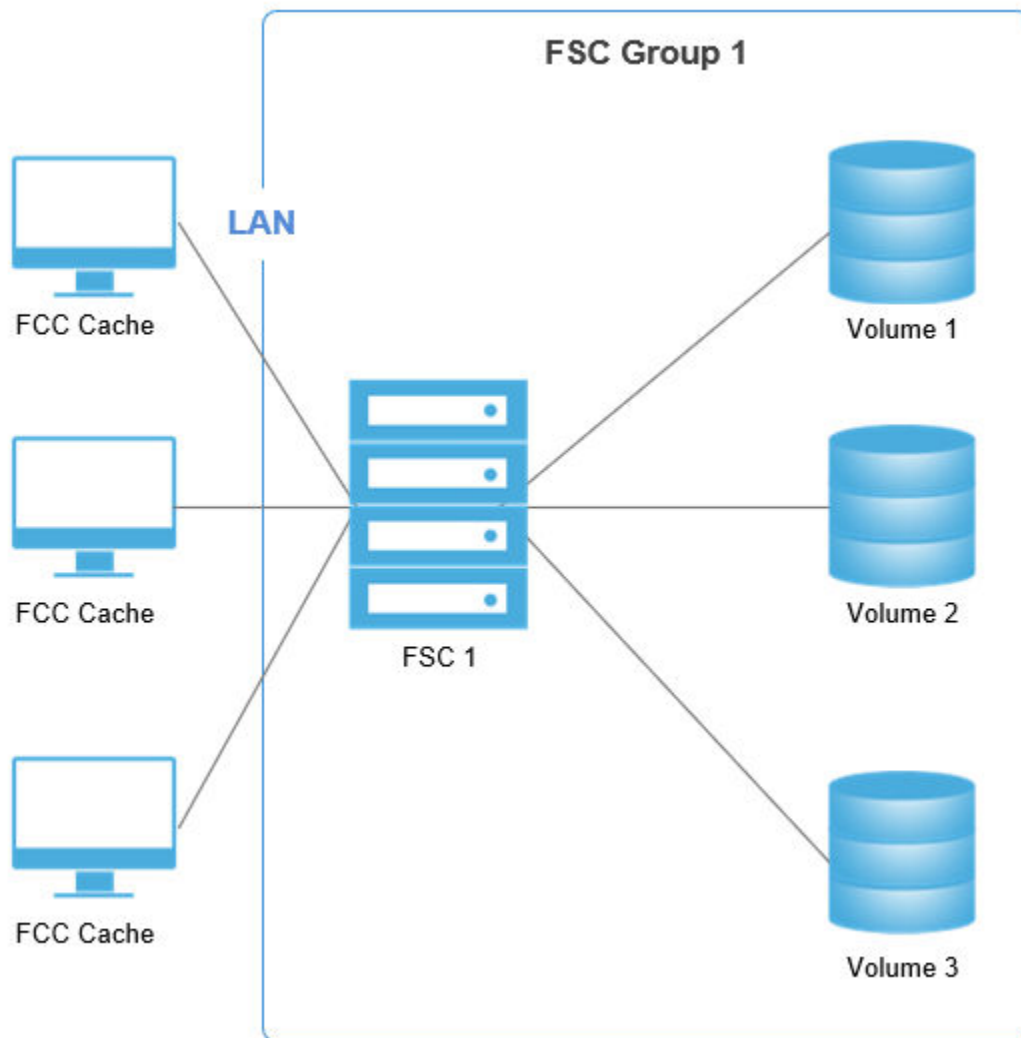
The sample Teamcenter rich client FMS configurations are fairly simple FMS deployments illustrating a single FMS capability and addressing basic FMS configuration solutions.

These sample configurations do not represent full configuration files. Rather, the configuration files include the key FMS configuration file statements and structures that are being illustrated, while leaving out default FSC and FCC property values to reduce the size of the file and to focus on the subject.

Sample LAN configurations

Single FSC configuration

A Teamcenter Environment Manager installation of Teamcenter provides a single FSC that mounts a single volume, a typical use for small deployments. Adding two additional volumes creates the basic two-tier Teamcenter FMS configuration shown in the following figure.



- FSC roles

The single FSC in this diagram fulfills the volume server and configuration server roles. The FSC does not provide file caching, as it has direct access to all the volumes. It also does not provide transient volumes as this function is performed by the FCC in two-tier mode.

- FCC roles

The FCC provides file caching (as always in both two-tier and four-tier configurations). It also provides a transient volume so that clients need not be aware of two-tier or four-tier operation.

- Client configuration

All clients are configured to retrieve the initial bootstrap configuration information from FSC1, which assigns all clients to FSC1 for file access.

- FMS primary configuration file (*FSC_HOME/fmsmaster_fsc-ID.xml*)

This file is served by FSC1. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. Key points of this file include:

- **fmsenterprise** element

This statement contains a definition of FSC defaults, FCC defaults, FSC1 and its assigned volumes. This statement also includes an ID attribute which must match the unique identifier for the database.

- **fscdefaults** element

Specifies where the FSC read and write cache files are stored. These defaults are used by any additional FSCs installed in the future.

- **fccdefaults** element

Specifies the default location of the user's cache.

- **fscGroup** element

Defines the FSC group. All FSCs must belong to one, and only one, FSC group.

- **fsc** element

Defines the FSC identifier and configures the FSC. In the following example, the address of FSC1 is **146.122.40.99:4444**. Dot notation is used in this example for simplicity. DNS names may also be used, such as **csun17.ugs.com:4444**.

- **volume** element

Several volumes are defined for FSC1. Each volume must have an entry under FSC1. The volume statements also specify the root directory for each volume. The volumes can be either local disk volumes, or mounted volumes on the network. While paths specified for the volume typically match the traditional Teamcenter path for the volume for a small deployment, FSCs can be installed on any box as needed and file paths may depend on mount points provided on that box. Thus the path need not match the traditional Teamcenter volume path.

You can use the **backup_xmlinfo** utility to generate the volume IDs.

- **clientmap** element

Specifies that all clients on 146 prefixed network addresses are assigned to FSC1.

A sample of the primary configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
    </fccdefaults>
    <fscGroup id="fscGroup1">
      <fsc id="fsc1" address="http://146.122.40.99:4444">
        <volume id="vol1" root="//csun17/vol1"/>
        <volume id="vol2" root="//csun17/vol2"/>
        <volume id="vol3" root="//csun17/vol3"/>
      </fsc>

      <clientmap subnet="146.0.0.1" mask="255.0.0.0">
        <assignedfsc fscid="fsc1"/>
      </clientmap>
    </fscGroup>
  </fmsenterprise>
</fmsworld>
```

- FSC configuration file (*FSC_HOME/fsc.xml*)

This file is small in this example, containing the minimum required statements for an **fsc.xml** file, and does not define or override any FSC or FCC defaults. See the *FSC_HOME/fsc.xml.template* file for all available elements. Key points of this file in this configuration are:

- **fscmaster** element

Specifies that the FSC reads the file directly from disk out of the FSC launch (working) directory. Note that the file name may be specified in the launch command as the **fms.config** property.

- **fsc** element

Specifies the FSC ID for this installed FSC. This FSC ID is used to refer to the FSC definition provided in the FMS primary configuration file. For example:

```
FSC1

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="true"/>
  <fsc id="fsc1"/>
</fscconfig>
```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This file is small in this example, containing only the bootstrap address for the FSC configuration server which allows the FCC configuration to be downloaded. See the *FMS_HOME/fcc.xml.template* file for all available elements. Key points of this file in this configuration are:

- **parentfsc** element

This statement identifies which parent FSC to use for the initial configuration download.

- FCC cache location

All windows clients contain an FCC cache located as specified by the **FCC_CacheLocation** XML attribute. All users with cache size parameters are based on the default coded values since there were no default settings in the FMS primary or FSC configuration files.

- Assigned FSC

The assigned FSC is FSC1, as specified in the FMS primary configuration file.

A sample of the FCC configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>

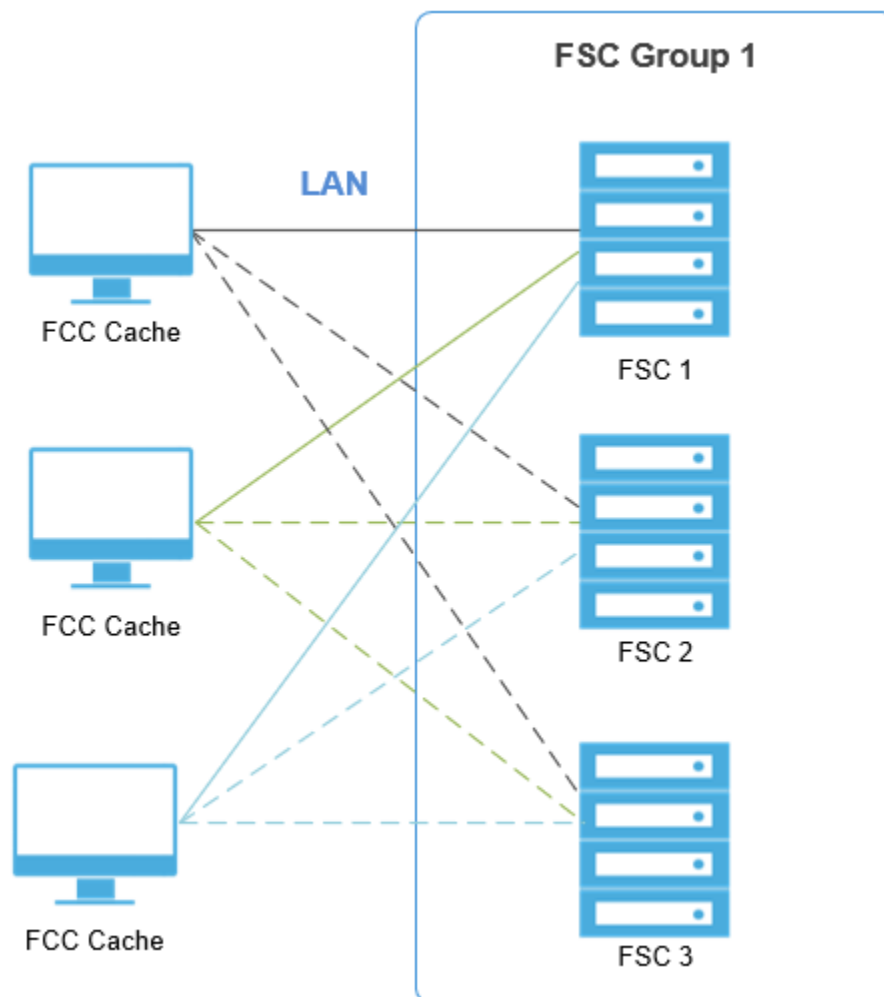
<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

<fccconfig>
  <parentfsc address="http://146.122.40.99:4444" priority="0"/>
```

```
</fccconfig>
```

FCC direct connect configuration

This configuration provides a standard multiple FSC configuration for small or medium deployments. This configuration contains a single FSC group, with an FSC deployed for each volume. Clients download files directly from the appropriate volume server. This configuration may handle approximately three times the file transfer volume of a single FSC deployment, since there are now three (essentially) independent FSC servers.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the previous example for the single FSC configuration. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. Key points of this file in this configuration are:

- Multiple FSCs

This configuration contains three FSCs deployed on different boxes, each containing a single volume.

- Direct user access

By default, a multiple FSC deployment enables all clients to directly download files from any FSC in the group that contains the clients assigned FSC.

- **assignedfsc** element

All users have an assigned FSC even though direct routing is enabled. In this configuration, the assigned FSC is only used to determine the group and the list of FSCs to which the client may directly connect. More complex deployments described in later sections describe additional roles for the assigned FSC.

- Separation of configuration and data paths

All FCCs and FSCs bootstrap the configuration file from FSC1, but users access files via all three FSC volume servers.

A sample of the primary configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
    </fccdefaults>
    <fscGroup id="fscGroup1">
      <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1"
          root="/data/vol1"/>
      </fsc>
      <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2"
          root="/data/vol2"/>
      </fsc>
      <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3"
          root="/data/vol3"/>
      </fsc>
    </fscGroup>
  </fmsenterprise>
</fmsworld>
```

```

        <clientmap subnet="146.0.0.1" mask="255.0.0.0">
            <assignedfsc fscid="fsc1"/>
        </clientmap>
    </fscGroup>
</fmsenterprise>

</fmsworld>

```

- FSC configuration files (*FSC_HOME/fsc.xml*)

In this example, each FSC has a separate **fsc.xml** file. See the *FSC_HOME/fsc.xml.template* file for all available elements. Key points of this file in this configuration are:

- Multiple FSC configuration files

Each FSC has a separate configuration file.

- **fmsmaster** element

Specifies that FSC1 is the primary configuration server in the FSC1config file. The FSC2 and FSC3 configuration files download the primary configuration file from FSC1.

A sample FSC configuration file for this configuration is:

```

FSC1

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="true"/>
    <fsc id="fsc1"/>
</fscconfig>
FSC2

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
    <fsc id="fsc2"/>
</fscconfig>
FSC3

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="false" address="http://csun17.ugs.com:4444" />

```

```
<fsc id="fsc3"/>

</fscconfig>
```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This example of the FCC configuration file is similar to previous examples:

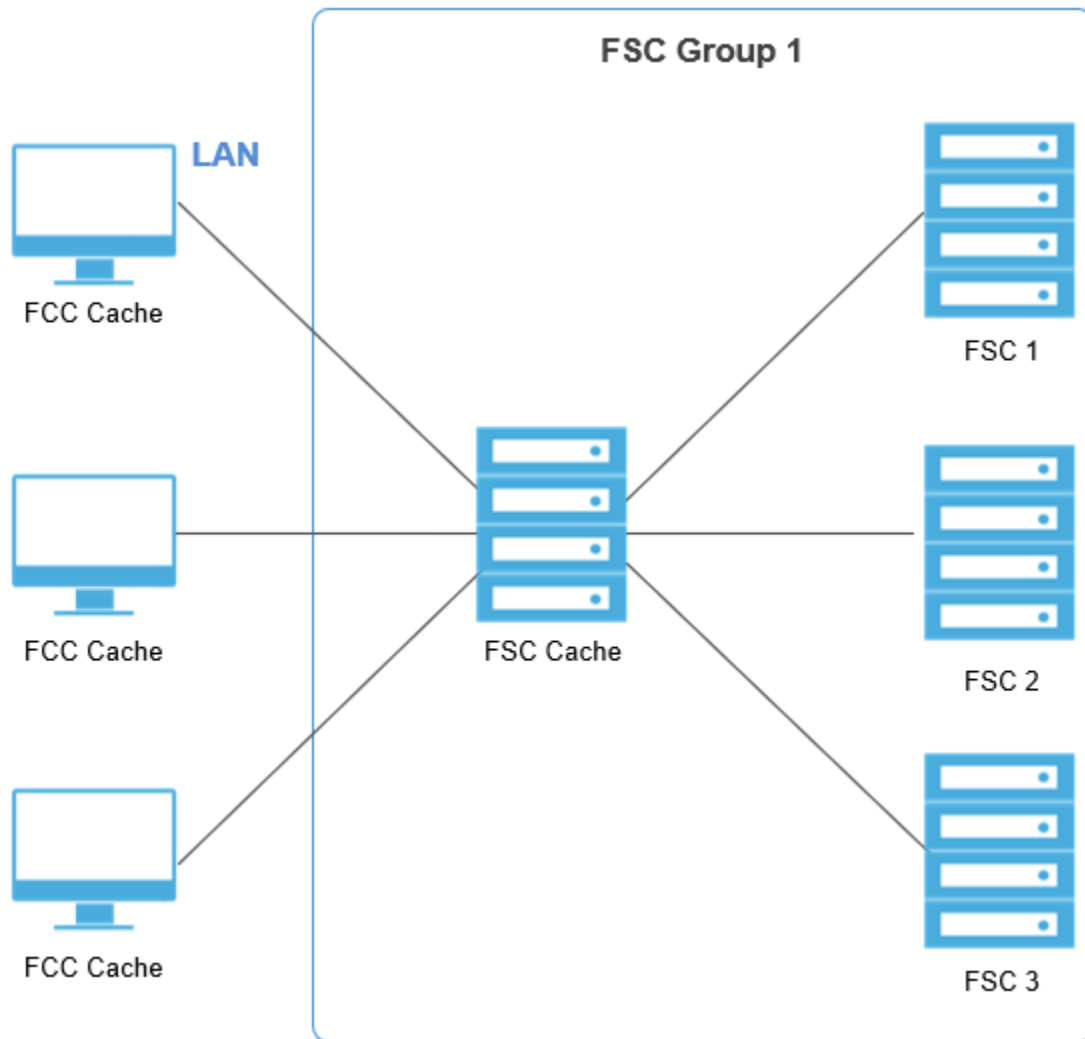
```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

<fccconfig>
  <parentfsc address="csun17.ugs.com:4444" priority="0"/>
</fccconfig>
```

FSC cached configuration

This configuration provides a high-speed cache server for improved client performance. When users upload new files to the volume, the files are cached in the FSC cache but streamed to the volume. This results in two copies of the file, but should not affect performance. Downloaded files are also cached. Typically, the end result is that the FSC cache server holds a small percentage of the most commonly accessed files, and the number of file accesses to the volume servers is substantially reduced. A benefit of this configuration is that clients may continue to download commonly accessed files with brief outages of the volume server network by downloading files from the FSC cache.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the example for the FCC direct connect configuration. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. Key points of this file in this configuration are:

- FSC cache

This configuration contains an FSC cache. The FSC cache acts as both a cache server and a configuration server. It does not provide volume or transient volume server functions.

- **FCC_EnableDirectFSCRouting** element

Set this configuration parameter to **false** to disable direct routing. This causes all FCC data access to be provided by the FCC's assigned FSC. If this parameter is not set, the client FSCs directly access the FSC volume servers.

- FSC cache sizing

The FSC uses the default read/write partial cache configuration and sizes.

A sample of the primary configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
    </fccdefaults>
    <property name="FCC_EnableDirectFSCRouting"
      Value="false"
      Overridable="false" />
    </fccdefaults>
    <fscGroup id="fscGroup1">
      <fsc id="fscCache" address="http://csun16.ugs.com:4444" />
      <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1"/>
      </fsc>
      <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2"/>
      </fsc>
      <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3"/>
      </fsc>

      <clientmap subnet="146.0.0.1" mask="255.0.0.0">
        <assignedfsc fscid="fscCache"/>
      </clientmap>
    </fscGroup>
  </fmsenterprise>
</fmsworld>
```

- FSC configuration files (*FSC_HOME/fsc.xml*)

This FSC configuration file is similar to the FSC direct connect configuration file. Each FSC has a separate **fsc.xml** file. See the *FSC_HOME/fsc.xml.template* file for all available elements. In this configuration, additional key points of the FSC configuration file are:

- Separation of configuration and data paths

All clients and all FSCs bootstrap the configuration from *csun16.ugs.com:4444*.

- FSC cache

This configuration includes an FSC cache. This configuration also requires a FSC configuration file.

A sample of the FSC configuration file for this configuration is:

```
FSCCACHE

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id=" fscCache"/>

</fscconfig>
FSC1

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="true" />
  <fsc id="fsc1"/>

</fscconfig>
FSC2

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fsc2"/>

</fscconfig>
FSC3

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fsc3"/>

</fscconfig>
```

- FCC configuration file (*FMS_HOME/fcc.xml*)

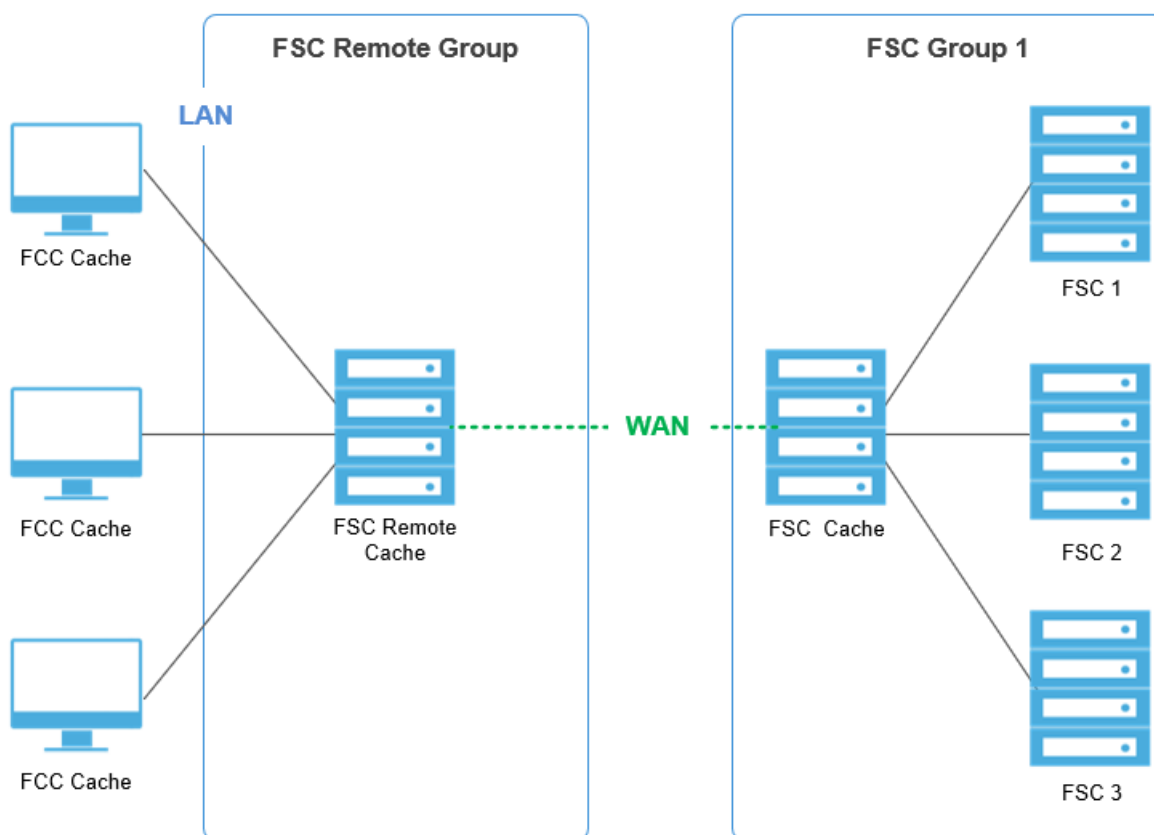
This example of the FCC configuration file is similar to previous examples:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">
<fccconfig>
  <parentfsc address="http:// csun16.ugs.com:4444" priority="0"/>
</fccconfig>
```

Remote user WAN configurations

FSC cached remote office configuration

This configuration provides a shared cache at the remote office so users have shared local LAN access to recently downloaded or uploaded files.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the example for the FSC cached configuration. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. Key points of this file in this configuration are:

- FSC remote group

This configuration contains a second FSC remote group, which contains the FSC remote cache. This is required because FSCs within an FSC group must be on a local LAN, not configured over a WAN.

- Assigned FSC

Clients are assigned to the FSC remote cache server. This provides a local shared cache for the remote user group. Direct routing is not required for the FSC remote cache, since there are no volumes in the FSC remote group.

- entryfsc** parameter

This parameter ensures that all incoming requests to FSC Group 1 are sent to the FSC cache server. If the entry FSC is not specified, requests are sent directly to the FSC volume servers.

- FCC_EnableDirectFSCRouting** parameter

By default, this parameter is set to **true**. In this configuration, the value has no effect, as there are no volumes in the FSC remote group.

- Link parameters

WAN acceleration is enabled from the remote office to the central office using the **linkparameters fromgroup** statement.

A sample of the primary configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">
<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
    </fccdefaults>
    <fscGroup id="fscRemoteGroup">
      <fsc id="fscRemoteCache"
        address="http://rsun10.ugs.com:4444" />
      <clientmap subnet="146.0.0.1" mask="255.0.0.0">
        <assignedfsc fscid="fscRemoteCache"/>
      </clientmap>
    </fscGroup>
```

```

    <fscGroup id="fscGroup1">
    <fsc id="fscCache" address="http://csun16.ugs.com:4444" />
      <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1"/>
      </fsc>
      <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2"/>
      </fsc>
      <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3"/>
      </fsc>
    <entryfsc fscid="fscache" priority="0"/>
    </fscGroup>
    <linkparameters fromgroup="fscRemoteGroup"
      togroup="fscGroup1"
      transport="wan" />
  </fmsenterprise>

</fmsworld>

```

- FSC configuration files (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

This FSC configuration file for this configuration has the same key points as the FSC cached configuration.

FSCREMOTECACHE

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fscRemoteCache" />
</fscconfig>
FSCCACHE

```

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fscCache"/>
</fscconfig>
FSC1

```

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="true" />
  <fsc id="fsc1"/>

```

```

</fscconfig>
FSC2

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fsc2"/>
</fscconfig>
FSC3

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fsc3"/>
</fscconfig>

```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This example of the FCC configuration file is similar to previous examples:

```

<?xml version="1.0" encoding="UTF-8"?>

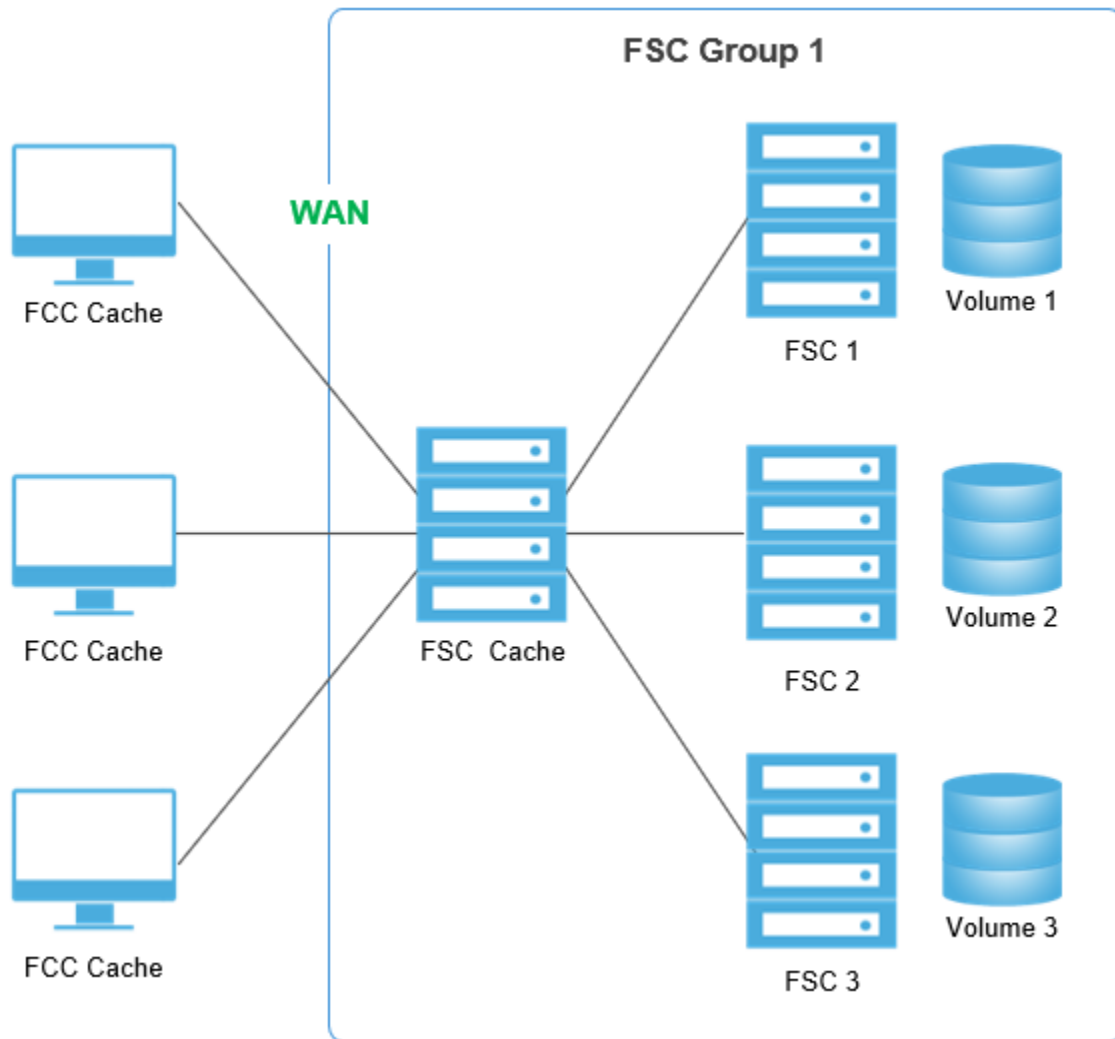
<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

<fccconfig>
  <parentfsc address="http://rsun10.ugs.com:4444" priority="0"/>
</fccconfig>

```

Remote FSC without caching configuration

This configuration services WAN users, but users must download their own file copies; there is no shared cache at the remote site.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the example for the FSC cached configuration. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key point of this configuration's configuration file is:

- FSC remote group

This configuration does not contain a remote cache. Therefore, a file may be downloaded multiple times as each user separately downloads files without the benefit of a shared cache.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
```

```

        <property name="FSC_ReadCacheLocation"
            value="$HOME/FscCache|/tmp/FscCache"
            overridable="true"/>
        <property name="FSC_WriteCacheLocation"
            value="$HOME/FscCache|/tmp/FscCache"
            overridable="true"/>
    </fscdefaults>
    <fccdefaults>
        <property name="FCC_CacheLocation"
            value="$HOME/FCCCache|/tmp/$USER/FCCCache"
            overridable="true"/>

    <property name="FCC_EnableDirectFSCRouting"
        Value="false"
        Overridable="false" />
    </fccdefaults>
    <fscGroup id="fscGroup1">
        <fsc id="fscCache" address="http://csun16.ugs.com:4444" />
            <fsc id="fsc1" address="http://csun17.ugs.com:4444">
                <volume id="vol1" root="/data/vol1"/>
            </fsc>
            <fsc id="fsc2" address="http://csun18.ugs.com:4444">
                <volume id="vol2" root="/data/vol2"/>
            </fsc>
            <fsc id="fsc3" address="http://csun19.ugs.com:4444">
                <volume id="vol3" root="/data/vol3"/>
            </fsc>
        <clientmap subnet="146.0.0.1" mask="255.0.0.0">
            <assignedfsc fscid="fscCache" transport="wan"/>
        </clientmap>
    </fscGroup>
</fmsenterprise>

</fmsworld>

```

- FSC configuration files (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

This FSC configuration file for this configuration has the same key points as the FSC cached remote office configuration.

FSC1

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="true"/>
    <fsc id="fsc1"/>

```

</fscconfig>

FSC2

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

```



```

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fsc2"/>

</fscconfig>
FSC3

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fsc3"/>

</fscconfig>
FscCache

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fscCache"/>

</fscconfig>

```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This example of the FCC configuration file is similar to previous examples:

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

<fccconfig>
  <parentfsc address="http://csun17.ugs.com:4444" priority="0"/>

</fccconfig>

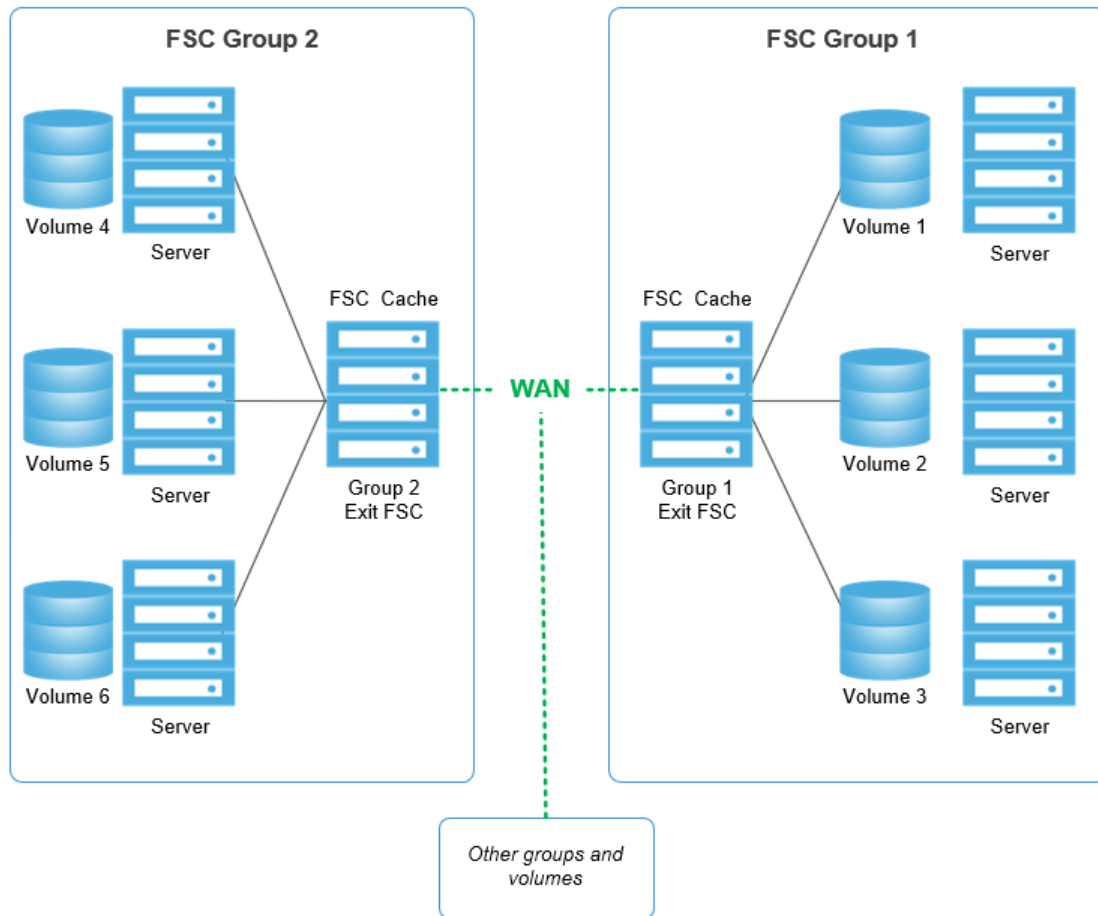
```

Exit FSC cache configuration

This configuration illustrates a shared cache at a remote site with volumes. Users have shared local LAN access to recently accessed files from other FSC groups.

- FSC servers route outbound requests to an exit FSC, which routes the requests to FSCs outside the group.
- The exit FSC routes the requests either to the outside group's entry FSC (if configured) or to an FSC that serves the requested resource. Outbound requests are requests for data on a resource outside the group. Outbound requests typically originate within the group, but this is not always the case.

- The exit FSC is primarily a cache server. Its purpose is to populate and serve data originating outside the group from its cache whenever possible. The data in its cache is high value. By serving the data from its cache, a WAN trip is prevented.
- Maintain the high value of the data in the exit FSC cache by setting the **FSC_CachePolicy** element to **CachelfNotInLocalFSCGroup**. This prevents the exit FSC from caching group data.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the example for the FSC cached configuration. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. Key points of this file in this configuration are:

- **ExitFSC** element

This configuration includes an exit FSC that caches data from outside groups. It also includes a second FSC remote group that contains local volumes.

- **FCC_EnableDirectFSCRouting** element

The **FCC_EnableDirectFSCRouting** element is set to **true**. FCCs in each FSC group route requests for locally served volumes directly to the FSCs serving the volume. Therefore, local requests from rich clients are not routed through the **AssignedFSC** or **ExitFSC** elements.

- **AssignedFSC** element

The **AssignedFSC** element for each group is **ExitFSC**. Therefore rich clients route requests for data served outside the group to the exit FSC first.

- **FSC_Cache Policy** element

The **FSC_CachePolicy** element is set to **CacheIfNotInLocalFSCGroup** in **FSCGroup1** and **FSCGroup2**, preventing the FSC caches within the groups from storing data served within the local FSC group.

- Link parameters

WAN acceleration is enabled from the remote office to the central office using the **linkparameters fromgroup** element.

An example of the primary configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">
<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache/tmp/$USER/FCCCache"
        overridable="true"/>
      <property name="FCC_EnableDirectFSCRouting"
        value="true"
        overridable="true"/>
    </fccdefaults>
    <fscGroup id="fscRemoteGroup">
      <fscdefaults>
        <property name="FSC_CachePolicy"
          value="CacheIfNotInLocalFSCGroup"
          overridable="true"/>
      </fscdefaults>
      <fsc id="fscRemoteExit"
        address="http://rlnx03.ugs.com:4444" />
      <fsc id="fsc4" address="http://rsun44.ugs.com:4444">
        <volume id="vol4" root="/data/vol1"/>
      </fsc>
      <fsc id="fsc5" address="http://rsun45.ugs.com:4444">
```

```

        <volume id="vol5" root="/data/vol2"/>
      </fsc>
      <fsc id="fsc6" address="http://rsun45.ugs.com:4448">
        <volume id="vol6" root="/data/vol3"/>
      </fsc>
      <clientmap subnet="147.0.0.1" mask="255.0.0.0">
        <assignedfsc fscid="fscRemoteExit"/>
      </clientmap>
      <ExitFSC fscid="fscRemoteExit" priority = "0"/>
    </fscGroup>
    <fscGroup id="fscGroup1">
    <fscdefaults>
      <property name="FSC_CachePolicy"
        value="CacheIfNotInLocalFSCGroup"
        overridable="true"/>
    </fscdefaults>
    <fsc id="fscExit"
      address="http://csun16.ugs.com:4444" />
      <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1"/>
      </fsc>
      <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2"/>
      </fsc>
      <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3"/>
      </fsc>
    <entryfsc fscid="fscExit" priority="0"/>
    <clientmap subnet="146.0.0.1" mask="255.0.0.0">
      <assignedfsc fscid="fscExit"/>
    </clientmap>
    <ExitFSC fscid="fscExit" priority = "0"/>
  </fscGroup>
  <linkparameters fromgroup="fscRemoteGroup"
    togroup="fscGroup1"
    transport="wan" />
  <linkparameters fromgroup="fscGroup1"
    togroup="fscRemoteGroup"
    transport="wan" />
</fmsenterprise>
</fmsworld>

```

- FSC configuration files (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

This FSC configuration file for this configuration has the same key points as the FSC cached configuration.

```

FSCREMOTECACHE
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fscRemoteExit"/>
</fscconfig>
FSCEXIT
<?xml version="1.0" encoding="UTF-8"?>

```

```

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fscExit"/>
</fscconfig>
FSC1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="true" />
  <fsc id="fsc1"/>
</fscconfig>
FSC2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fsc2"/>
</fscconfig>
...
FSC6
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" />
  <fsc id="fsc6"/>
</fscconfig>

```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This example of the FCC configuration file is similar to previous examples:

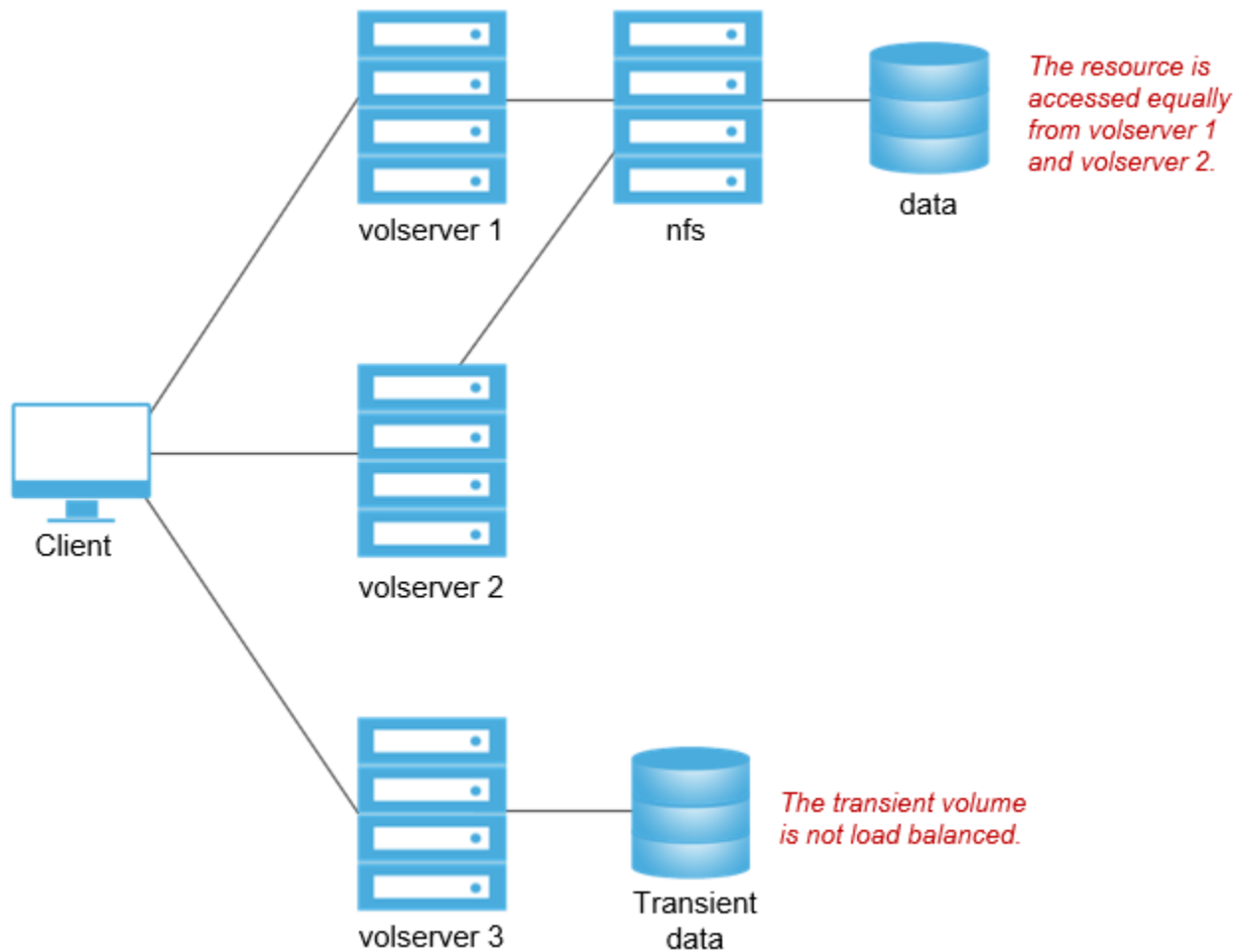
```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">
<fccconfig>
  <parentfsc address="rlx03.ugs.com:4444" priority="0"/>
</fccconfig>
FSCGROUP1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">
<fccconfig>
  <parentfsc address="http://csun16.ugs.com:4444" priority="0"/>
</fccconfig>

```

FSC primary server load balancing

This configuration illustrates how to configure multiple primary FSC servers for load balancing.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key points of this configuration's configuration file are:

- **filestoregroup** element

The **filestoregroup** element defines a volume (or group of volumes) to be load balanced across several FSCs.

- **filestore** element

The **filestore** element in an FSC definition, within an **fscgroup**, identifies the **filestore** group to serve. The priority attribute defines its load balancing priority.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">
```

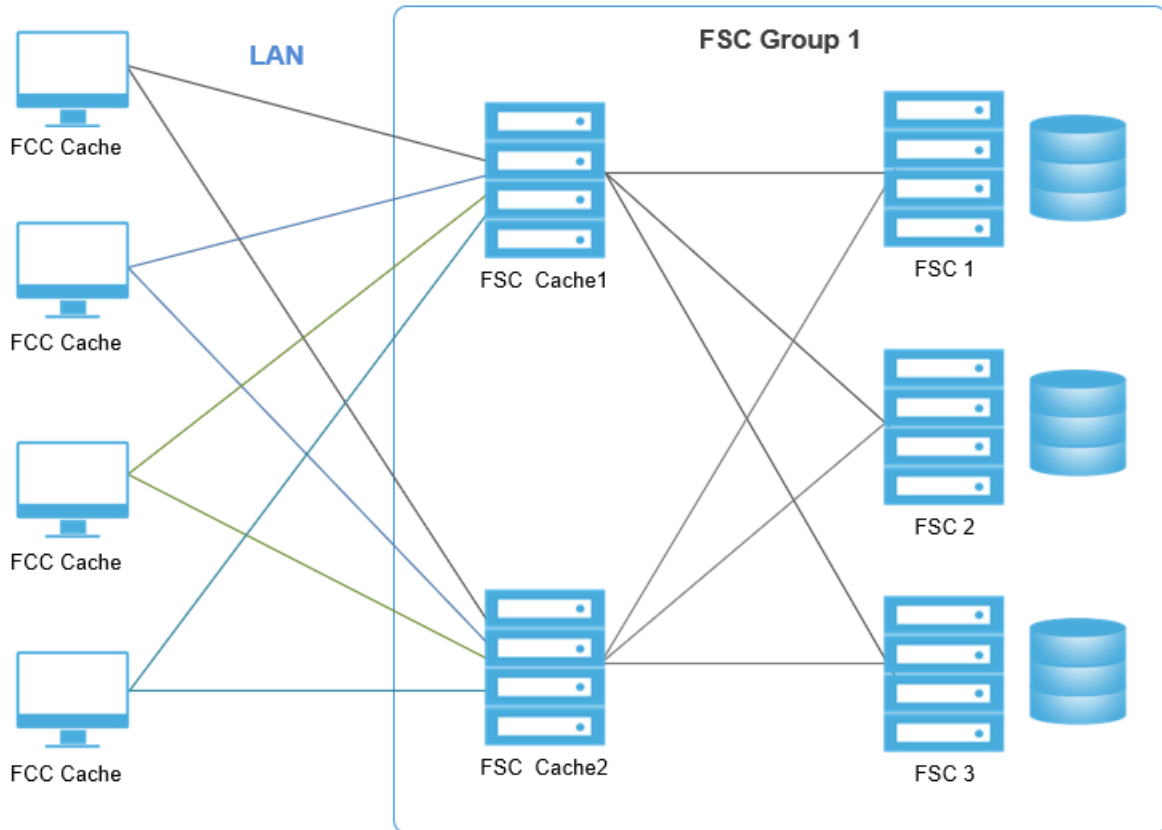
```

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
      <property name="FCC_EnableDirectFSCRouting"
        Value="false"
        Overridable="false" />
    </fccdefaults>
    <filestoregroup id="fsgroup1">
      <volume id="data" root="/nfs/data"/>
    </filestoregroup>
    <fscGroup id="fscGroup1">
      <fsc id="volserver1" address="http://fsc1:4444">
        <filestore groupid="fsgroup1" priority="0"/>
      </fsc>
      <fsc id="volserver2" address="http://fsc2:4444">
        <filestore groupid="fsgroup1" priority="0"/>
      </fsc>
      <fsc id="volserver3" address="http://fsc3:4444">
        <transientvolume id="transientdata"
          root="/PLM/volumes/transientdata"/>
      </fsc>
      <clientmap subnet="146.122.40.1" mask="255.255.255.0">
        <assignedfsc fscid="volserver1" />
      </clientmap>
      <clientmap subnet="146.122.41.1" mask="255.255.255.0">
        <assignedfsc fscid="fscCache1" />
      </clientmap>
    </fscGroup>
  </fmsenterprise>
</fmsworld>

```

FCC LAN client failover configuration

This configuration provides a hot local LAN failover configuration.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key point of this configuration's configuration file is:

- User assigned groups

Half of the users are assigned to FSC Cache 1 as the primary FSC, half are assigned to FSC Cache 2. If either FSC cache machine fails, the FCCs fail over to the other FSC.

- Hot failover

Both of the FSC cache machines are caching files. Therefore, if one FSC cache machine fails, the other takes up the additional traffic. There is potential performance degradation.

- Full system failure

If both cache machines fail, no file access is available for clients. The volume servers can be designated as failover machines to cover the unlikely event of a three-box failure scenario.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">
```



```

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
      <property name="FCC_EnableDirectFSCRouting"
        Value="false"
        Overridable="false" />
    </fccdefaults>
    <fscGroup id="fscGroup1">
      <fsc id="fscCache1" address="http://csun15.ugs.com:4444" />
      <fsc id="fscCache2" address="http://csun16.ugs.com:4444" />
      <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1"/>
      </fsc>
      <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2"/>
      </fsc>
      <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3"/>
      </fsc>
    <clientmap subnet="146.122.40.1"
      mask="255.255.255.0">
      <assignedfsc fscid="fscCache1" priority="0" />
      <assignedfsc fscid="fscCache2" priority="1" />
    </clientmap>
    <clientmap subnet="146.122.41.1"
      mask="255.255.255.0">
      <assignedfsc fscid="fscCache2" priority="0" />
      <assignedfsc fscid="fscCache1" priority="1" />
    </clientmap>
  </fscGroup>
</fmsenterprise>
</fmsworld>

```

- FSC configuration files (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

This FSC configuration file for this configuration has the following key points:

- Redundant configuration servers

FSC1 and FSC2 are designated as configuration servers. The other FSC servers specify FSC1 and FSC2 as the primary and failover configuration server, respectively.

```

FSC1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="true"/>
  <fsc id="fsc1"/>
</fscconfig>
FSC2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="true" />
  <fsc id="fsc2"/>
</fscconfig>
FSC3
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fsc3"/>
</fscconfig>
FscCache1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fscCache1"/>
</fscconfig>
FscCache2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fscCache2"/>
</fscconfig>

```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This example of the FCC configuration file is similar to previous examples. The key point is:

- Configuration failover

Two different FSCs are identified as configuration servers. This insures that the FCC can initialize by downloading the configuration file in case one FSC cache machine has failed or is taken down for maintenance.

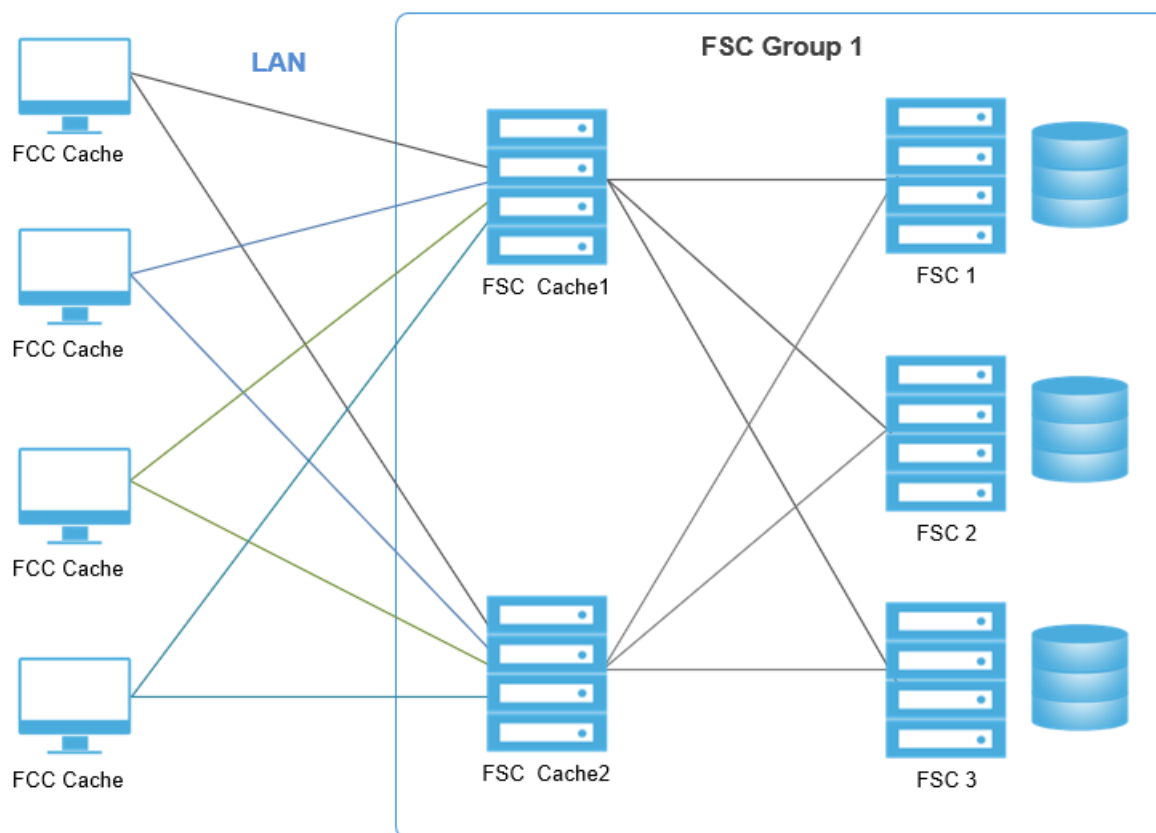
```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">
<fccconfig>
  <parentfsc address="http://csun17.ugs.com:4444" priority="0"/>
  <parentfsc address="http://csun18.ugs.com:4444" priority="1"/>
</fccconfig>

```

FSC clientmap DNS suffix configuration

This configuration illustrates how to use DNS names to map an FCC to the parent FSCs.



- Primary configuration file (*FSC_HOME/fmsmaster_fsc-ID.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key points of this configuration's configuration file are:

- **dnszone** client map attribute

The **dnszone** attribute can be used instead of the **subnet** and **mask** attributes to map FSCs.

- **dnshostname** client map attribute

The **dnshostname** attribute can be used instead of the **subnet** and **mask** attributes to map a specific host to FSCs.

- **default** client map attribute

The **default** attribute can be used to define default FSCs whenever **subnet/mask**, **dnszone**, or **dnshostname** client maps fail to map an FCC. This default attribute replaces the legacy **mask="0.0.0.0"** technique previously used for **subnet/mask** maps.

- **dns_not_defined** client map attribute

The **dns_not_defined** attribute can be used to define an FSC map whenever a requesting FCC's IP address cannot be converted to a DNS name.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">
<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache/tmp/$USER/FCCCache"
        overridable="true"/>
      <property name="FCC_EnableDirectFSCRouting"
        Value="false"
        Overridable="false" />
    </fccdefaults>
    <fscGroup id="fscGroup1">
      <fsc id="fscCache1" address="http://csun15.ugs.com:4444" />
      <fsc id="fscCache2" address="http://csun16.ugs.com:4444" />
      <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1"/>
      </fsc>
      <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2"/>
      </fsc>
      <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3"/>
      </fsc>
    <clientmap dnszone="yoyodyne.com">
      <assignedfsc fscid="fscCache2" priority="0" />
    </clientmap>
    <clientmap dnszone="eng.yoyodyne.com">
      <assignedfsc fscid="fscCache1" priority="0" />
      <assignedfsc fscid="fscCache2" priority="1" />
    </clientmap>
    <clientmap dnszone="mfg.yoyodyne.com">
      <assignedfsc fscid="fscCache2" priority="0" />
      <assignedfsc fscid="fscCache1" priority="1" />
    </clientmap>
    <clientmap dnshostname="supervisor.mfg.yoyodyne.com">
      <assignedfsc fscid="fscCache2" priority="0" />
      <assignedfsc fscid="fscCache1" priority="1" />
    </clientmap>
    <clientmap default="true">
      <assignedfsc fscid="fscCache1" priority="0" />
    </clientmap>
  </fmsenterprise>
</fmsworld>
```

```

        <assignedfsc fscid="fscCache2" priority="1" />
    </clientmap>
    <clientmap dns_not_defined="true">
        <assignedfsc fscid="fscCache1" priority="0" />
        <assignedfsc fscid="fscCache2" priority="1" />
    </clientmap>
</fscGroup>
</fmsenterprise>
</fmsworld>

```

- FSC configuration files (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

This FSC configuration file for this configuration has the following key points:

- Redundant configuration servers

FSC1 and FSC2 are designated as configuration servers. The other FSC servers specify FSC1 and FSC2 as the primary and failover configuration server, respectively.

```

FSC1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
    <fscmaster serves="true"/>
    <fsc id="fsc1"/>
</fscconfig>
FSC2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
    <fscmaster serves="true" />
    <fsc id="fsc2"/>
</fscconfig>
FSC3
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
    <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
    <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
    <fsc id="fsc3"/>
</fscconfig>
FscCache1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
    <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
    <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
    <fsc id="fscCache1"/>
</fscconfig>
FscCache2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
    <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />

```

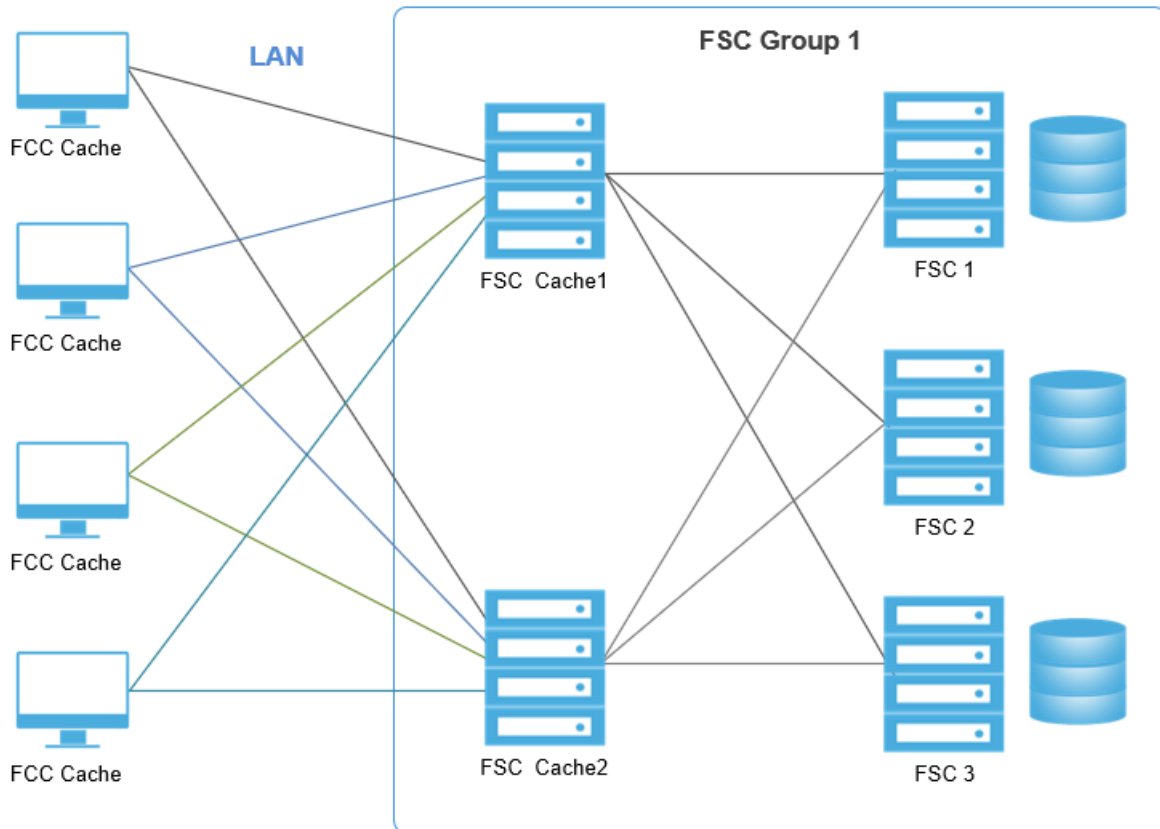
```

<fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
<fsc id="fscCache2"/>
</fscconfig>

```

FCC external load balancing configuration

This configuration illustrates how to load balance FMS data.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key points of this configuration's configuration file are:

- **assignedfsc** priority attribute

Within the primary configuration file, priority attributes may have priority levels of the same value. When this condition is found, the FCC returns these parallel priorities upon initialization. The FCC then load balances its requests to these parallel FSCs.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

```

```

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
      <property name="FCC_EnableDirectFSCRouting"
        Value="false"
        Overridable="false" />
    </fccdefaults>
    <fscGroup id="fscGroup1">
      <fsc id="fscCache1" address="http://csun15.ugs.com:4444" />
      <fsc id="fscCache2" address="http://csun16.ugs.com:4444" />
      <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1"/>
      </fsc>
      <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2"/>
      </fsc>
      <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3"/>
      </fsc>
      <clientmap subnet="146.122.40.1" mask="255.255.255.0">
        <assignedfsc fscid="fscCache1" priority="0" />
        <assignedfsc fscid="fscCache2" priority="0" />
        <assignedfsc fscid="fsc3" priority="1" />
      </clientmap>
      <clientmap subnet="146.122.41.1" mask="255.255.255.0">
        <assignedfsc fscid="fscCache1" priority="0" />
        <assignedfsc fscid="fscCache2" priority="0" />
        <assignedfsc fscid="fsc3" priority="1" />
      </clientmap>
    </fscGroup>
  </fmsenterprise>
</fmsworld>

```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This FCC configuration file for this configuration has the following key point:

- **parentfsc** priority attribute

If two or more **parentfscs** are assigned the same priority level, then when the FCC is initialized it randomly selects one of these parallel **parentfscs** and requests its configuration data. This feature is useful when a large number of FCCs use the same FSCs as configuration servers. If all the FCCs initialize at the same time (for example, after a building power failure) then the FCCs can be distributed across a number FSCs if the priorities are set to 0.

```

<?xml version="1.0" encoding="UTF-8"?>

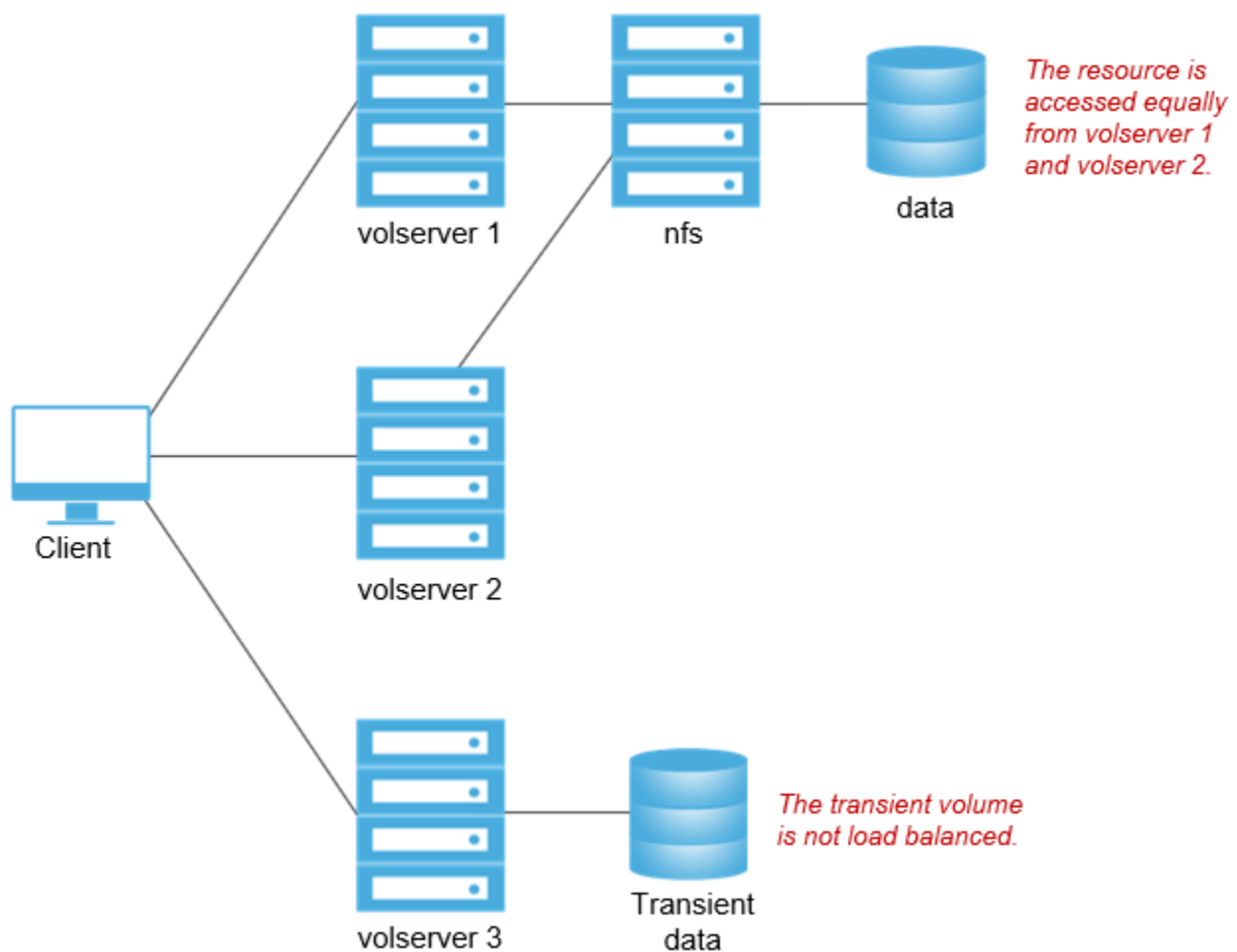
<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

<fccconfig>
  <parentfsc address="http://csun17.ugs.com:4444" priority="0"/>
  <parentfsc address="http://csun18.ugs.com:4444" priority="0"/>
</fccconfig>

```

FSC volume load balancing of FMS data configuration

This configuration illustrates how to load balance FMS data by distributing the network access load among same-priority elements.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key points of this configuration's configuration file are:

- **filestoregroup** element

The **filestoregroup** element defines a volume (or group of volumes) to be load balanced across several FSCs.

- **filestore** element

The **filestore** element in an FSC definition, within an **fscgroup**, identifies the **filestore** group to serve. The priority attribute defines its load balancing priority.

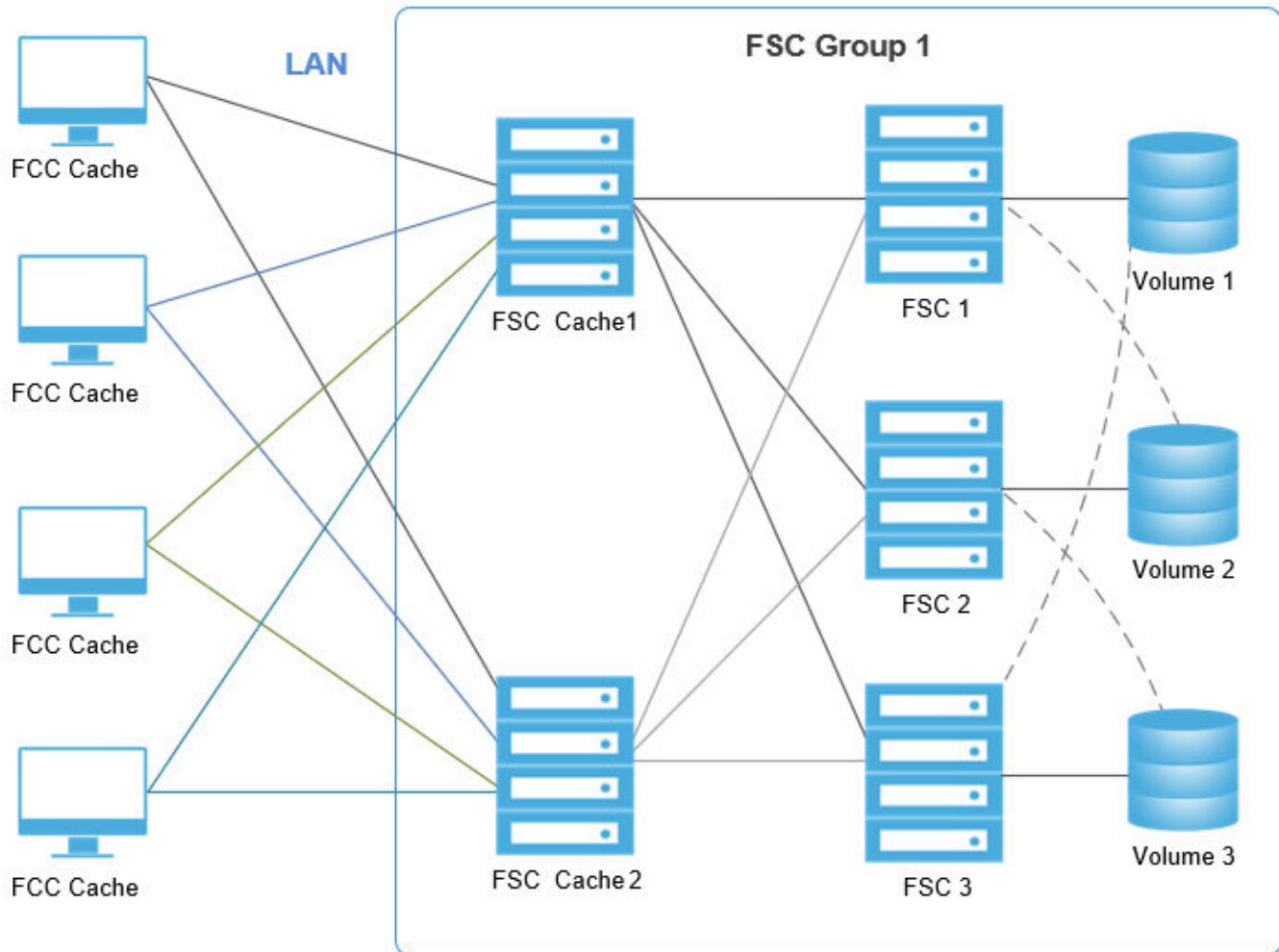
```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
      <property name="FCC_EnableDirectFSCRouting"
        Value="false"
        Overridable="false" />
    </fccdefaults>
    <filestoregroup id="fsgroup1">
      <volume id="data" root="/nfs/data"/>
    </filestoregroup>
    <fscGroup id="fscGroup1">
      <fsc id="volserver1" address="http://fsc1:4444">
        <filestore groupid="fsgroup1" priority="0"/>
      </fsc>
      <fsc id=" volserver2" address="http://fsc2:4444">
        <filestore groupid="fsgroup1" priority="0"/>
      </fsc>
      <fsc id=" volserver3" address="http://fsc3:4444">
        <transientvolume id="transientdata"
          root="/PLM/volumes/transientdata"/>
      </fsc>
      <clientmap subnet="146.122.40.1" mask="255.255.255.0">
        <assignedfsc fscid="volserver1" />
      </clientmap>
      <clientmap subnet="146.122.41.1" mask="255.255.255.0">
        <assignedfsc fscid="fscCache1" />
      </clientmap>
    </fscGroup>
  </fmsenterprise>
</fmsworld>
```

FSC volume failover configuration

This configuration illustrates how to configure an FSC volume failover.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key point of this configuration is:

- **volume** priority attribute

Assigning a priority to a volume assigned to an FSC defines what priority the FSC serves the volume. Notice in the configuration below that each of the three volumes are assigned to two of the three serving FSCs, one at **priority="0"** and one at **priority="1"**. The priority 0 FSC normally serves the volume but if one of the FSCs is down, the FSC with the priority 1 serves the offline volume.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

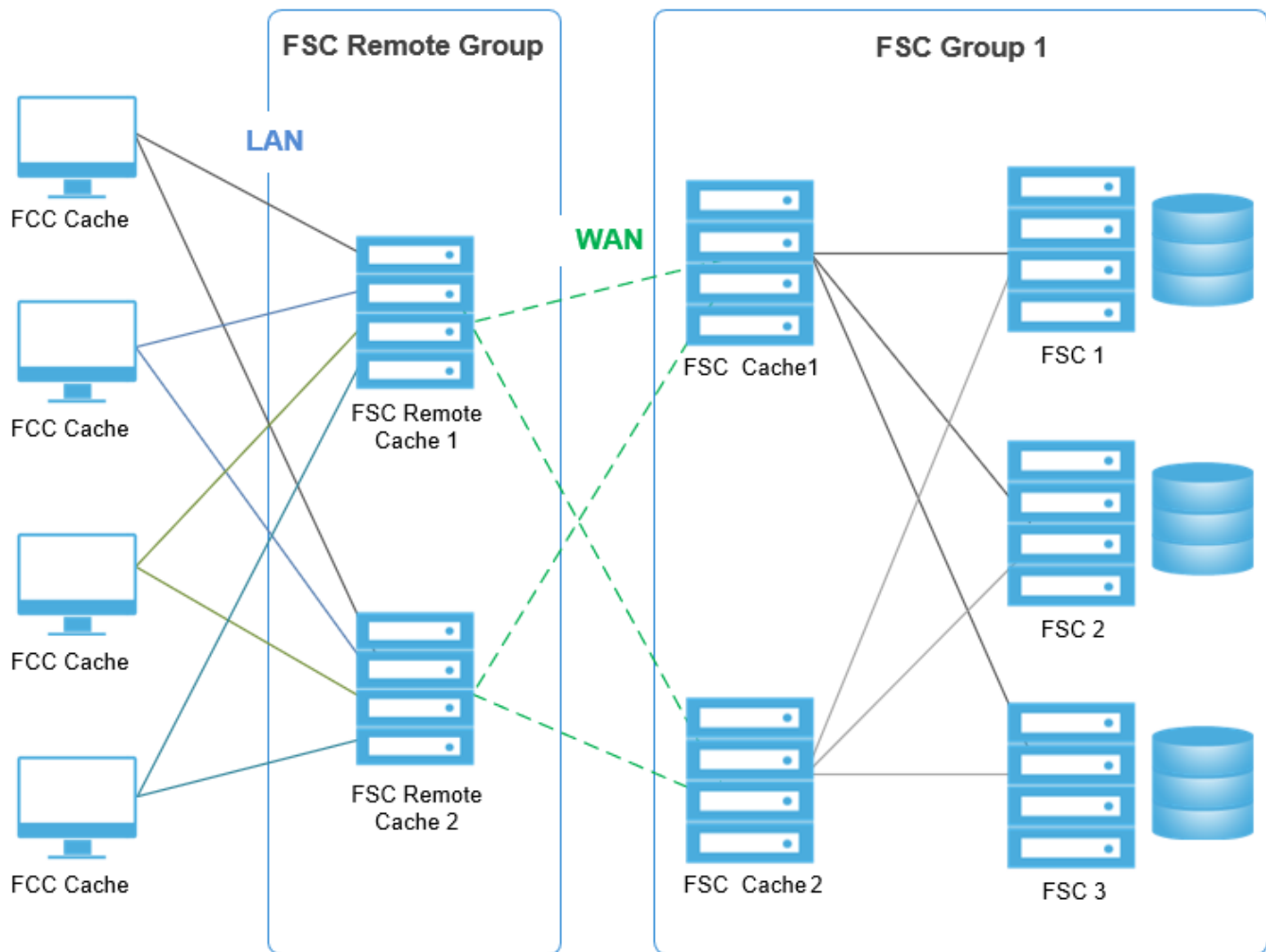
<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
      <property name="FCC_EnableDirectFSCRouting"
        Value="false"
        Overridable="false" />
    </fccdefaults>
    <fscGroup id="fscGroup1">
      <fsc id="fscCache1" address="http://csun15.ugs.com:4444" />
      <fsc id="fscCache2" address="http://csun16.ugs.com:4444" />
      <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1/" priority="0">
        <volume id="vol2" root="/data/vol2/" priority="1">
      </fsc>
      <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2/" priority="0">
        <volume id="vol3" root="/data/vol3/" priority="1">
      </fsc>
      <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3/" priority="0">
        <volume id="vol1" root="/data/vol1/" priority="1">
      </fsc>
      <clientmap subnet="146.122.40.1" mask="255.255.255.0">
        <assignedfsc fscid="fscCache1" priority="0" />
        <assignedfsc fscid="fscCache2" priority="1" />
      </clientmap>
      <clientmap subnet="146.122.41.1" mask="255.255.255.0">
        <assignedfsc fscid="fscCache2" priority="0" />
        <assignedfsc fscid="fscCache1" priority="1" />
      </clientmap>
    </fscGroup>
  </fmsenterprise>
</fmsworld>

```

FSC remote cache failover configuration

This configuration provides a hot remote cache configuration, and a idle central cache configuration.



- Primary configuration file (*FSC_HOME/fmsmaster_fsc-ID.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key points of this configuration's configuration file are:

- User assigned groups

Half of the users are assigned to FSC Cache 1 as the primary FSC, half are assigned to FSC Cache 2. If either FSC cache machine fails, the FCCs fail over to the other FSC.

- Hot remote failover

Both of the FSC remote cache machines are caching files. Therefore, if one fails, the other machine takes up the additional traffic. There is potential performance degradation.

- More remote FSC caches

Additional FSC remote cache machines can be added to divide users over more machines. Additional machines decrease performance degradation of a single FSC machine failure.

- Cold failover

All requests from the FSC remote group go through the FSC Cache 1 machine. The FSC Cache 2 machine is idle until there is a failure, in which case the FSC remote cache machines fail to FSC Cache 2. FSC Cache 2 can be assigned local LAN access to utilize this spare capacity.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
      <property name="FCC_EnableDirectFSCRouting"
        Value="false"
        Overridable="false" />
    </fccdefaults>
    <fscGroup id="fscRemoteGroup">
      <fsc id="fscRemoteCache1"
        address="http://rsun1.ugs.com:4444" />
      <fsc id="fscRemoteCache2"
        address="http://rsun2.ugs.com:4444" />
      <clientmap subnet="146.122.40.1"
        mask="255.255.255.0">
        <assignedfsc fscid="fscCache1" priority="0" />
        <assignedfsc fscid="fscCache2" priority="1" />
      </clientmap>
      <clientmap subnet="146.122.41.1"
        mask="255.255.255.0">
        <assignedfsc fscid="fscCache2" priority="0" />
        <assignedfsc fscid="fscCache1" priority="1" />
      </clientmap>
    </fscGroup>
    <fscGroup id="fscGroup1">
      <fsc id="fscCache1" address="http://csun15.ugs.com:4444" />
      <fsc id="fscCache2" address="http://csun16.ugs.com:4444" />
      <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1"/>
      </fsc>
      <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2"/>
      </fsc>
      <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3"/>
      </fsc>
    </fscGroup>
  </fmsenterprise>
</fmsworld>
```

```

        </fsc>
        <entryfsc fscid="fscache1" priority="0" />
        <entryfsc fscid="fscache2" priority="1" />
    </fscGroup>
    <linkparameters fromgroup="fscRemoteGroup"
                    togroup="fscGroup1"
                    transport="wan">
</fmsenterprise>

</fmsworld>

```

- FSC configuration files (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

This FSC configuration file for this configuration is as follows:

```

FSC1

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="true"/>
    <fsc id="fsc1"/>

</fscconfig>
FSC2

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="true" />
    <fsc id="fsc2"/>

</fscconfig>
FSC3

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
    <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="0" />
    <fsc id="fsc3"/>

</fscconfig>
FscCache1

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>

```

```

        <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
        <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="0" />
        <fsc id="fscCache1"/>

</fscconfig>
FscCache2

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
    <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="0" />
    <fsc id="fscCache2"/>

</fscconfig>
FSCRemoteCache1

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="false" address="http://csun15.ugs.com:4444" priority="0" />
    <fscmaster serves="false" address="http://csun16.ugs.com:4444" priority="1" />
    <fsc id="fscRemoteCache1"/>

</fscconfig>
FSCRemoteCache2

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
    <fscmaster serves="false" address="http://csun15.ugs.com:4444" priority="0" />
    <fscmaster serves="false" address="http://csun16.ugs.com:4444" priority="1" />
    <fsc id="fscRemoteCache2"/>

</fscconfig>

```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This example of the FCC configuration file is similar to previous examples.

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

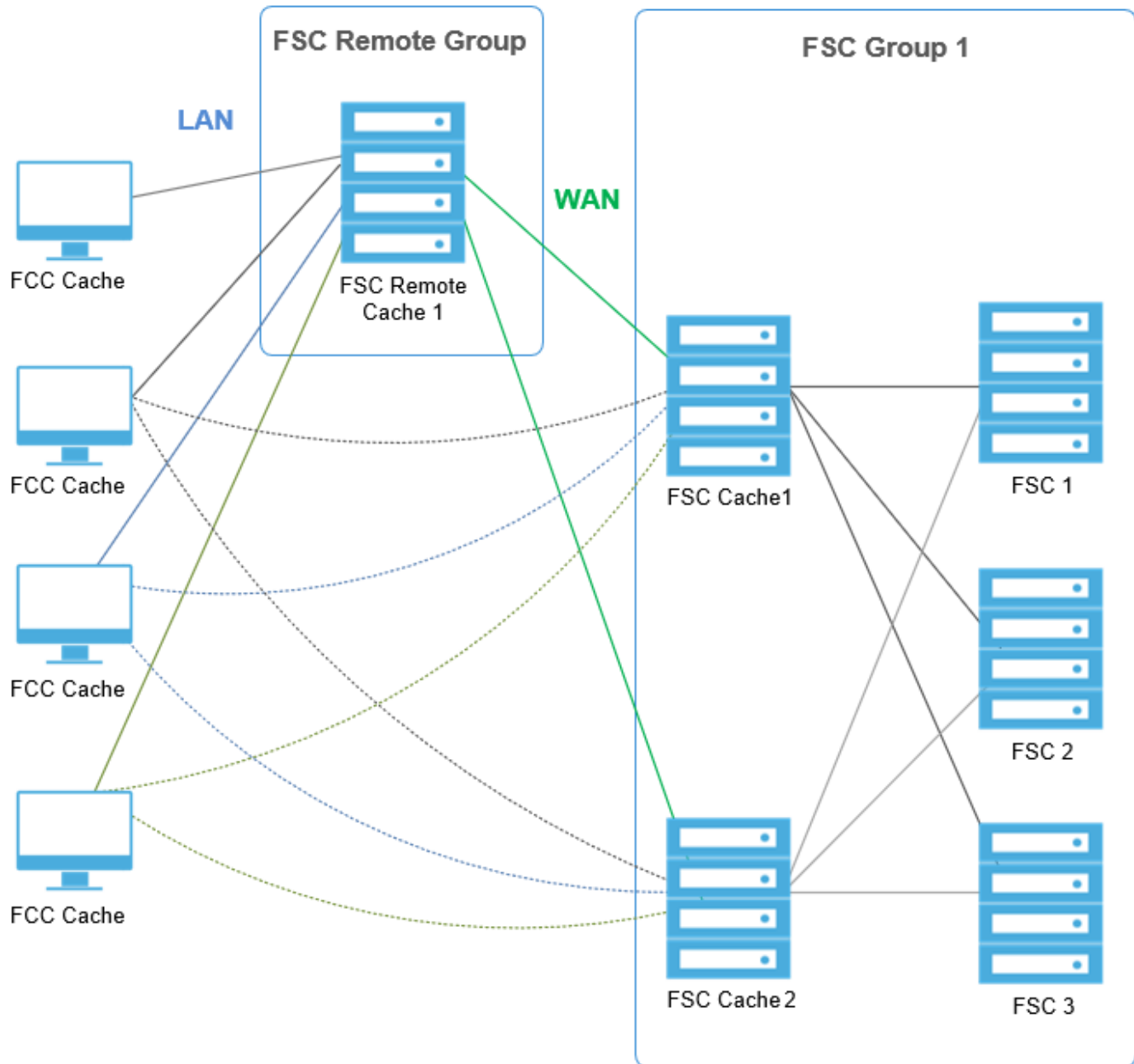
<fccconfig>
    <parentfsc address="http://rsun1.ugs.com:4444" priority="0"/>
    <parentfsc address="http://rsun2.ugs.com:4444" priority="1"/>

</fccconfig>

```

Alternate FSC remote cache failover configuration

This configuration provides a failover configuration with a local cache at the remote location, but this configuration results in files being loaded over the WAN multiple times if the remote location cache fails.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key points of this configuration's configuration file are:

- User assigned groups

All users are assigned to the shared FSC remote Cache 1

- FCC failover

When the FSC remote Cache 1 machine fails, all remote users' FCCs failover to FSC Cache1. If that machine also fails, all remote users fail over to FSC Cache 2. As a result, FSC Cache 2 is normally an idle machine.

- FCC configuration failover

FCCs receive configuration download data from the FSC remote Cache 1 machine. If this machine is down, the FCCs receive the configuration download data from the FSC Cache 1 machine. If this machine is also down, the FCCs receive the configuration download data from the FSC Cache 2 machine.

- **entryfsc** parameter

This parameter specifies the routing during normal operations when file requests flow through the FSC Remote Group during normal operations. If the remote cache fails, these statements have no effect. When the remote cache fails, users are assigned FSC Cache 1 or FSC Cache 2, according to the client masks.

- **FCC_EnableDirectFSCRouting** parameter

This parameter does not need to be set; there is only one FSC in the FSC Remote group, which is the primary group for all users. This parameter applies only to the primary assigned group, it does not apply to secondary groups if the remote cache server fails. Thus, this parameter setting has no impact on this configuration.

- WAN settings

During normal operations, traffic between the groups is based on the link parameters which specify WAN acceleration. During failover operations the assigned FSCs in the masks specify that the FCC should use WAN acceleration for access to the FSC Group 1 cache servers.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
```

```

        <property name="FCC_CacheLocation"
                  value="$HOME/FCCCache|/tmp/$USER/FCCCache"
                  overridable="true"/>
    </fccdefaults>
    <fscGroup id="fscRemoteGroup">
    <fsc id="fscRemoteCache1"
        address="http://rsun1.ugs.com:4444" />
    <clientmap subnet="146.122.40.1"
        mask="255.255.255.0">
        <assignedfsc fscid="fscRemoteCache1"
            priority="0" />
        <assignedfsc fscid="fscCache1"
            transport="wan"
            priority="1" />
        <assignedfsc fscid="fscCache2"
            transport="wan"
            priority="2" />
    </clientmap>
    <clientmap subnet="146.122.41.1"
        mask="255.255.255.0">
        <assignedfsc fscid="fscRemoteCache1"
            priority="0" />
        <assignedfsc fscid="fscCache2"
            transport="wan"
            priority="1" />
        <assignedfsc fscid="fscCache1"
            transport="wan"
            priority="2" />
    </clientmap>
    </fscGroup>
    <fscGroup id="fscGroup1">
        <fsc id="fscCache1" address="http://csun15.ugs.com:4444" />
        <fsc id="fscCache2" address="http://csun16.ugs.com:4444" />
        <fsc id="fsc1" address="http://csun17.ugs.com:4444">
            <volume id="vol1" root="/data/vol1"/>
        </fsc>
        <fsc id="fsc2" address="http://csun18.ugs.com:4444">
            <volume id="vol2" root="/data/vol2"/>
        </fsc>
        <fsc id="fsc3" address="http://csun19.ugs.com:4444">
            <volume id="vol3" root="/data/vol3"/>
        </fsc>
        <entryfsc fscid="fscache1" priority="0" />
        <entryfsc fscid="fscache2" priority="1" />
    </fscGroup>
    <linkparameters fromgroup="fscRemoteGroup"
        togroup="fscGroup1"
        transport="wan" />
</fmsenterprise>

</fmsworld>

```

- FSC configuration files (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

This FSC configuration file for this configuration is as follows:

FSC1

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="true"/>
  <fsc id="fsc1"/>

```

```
</fscconfig>
FSC2
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="true" />
  <fsc id="fsc2"/>

```

```
</fscconfig>
FSC3
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fsc3"/>

```

```
</fscconfig>
FscCache1
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fscCache1"/>

```

```
</fscconfig>
FscCache2
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fscCache2"/>

```

```
</fscconfig>
FSCRemoteCache1
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun15.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun16.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache1"/>
</fscconfig>
```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This example of the FCC configuration file is similar to previous examples.

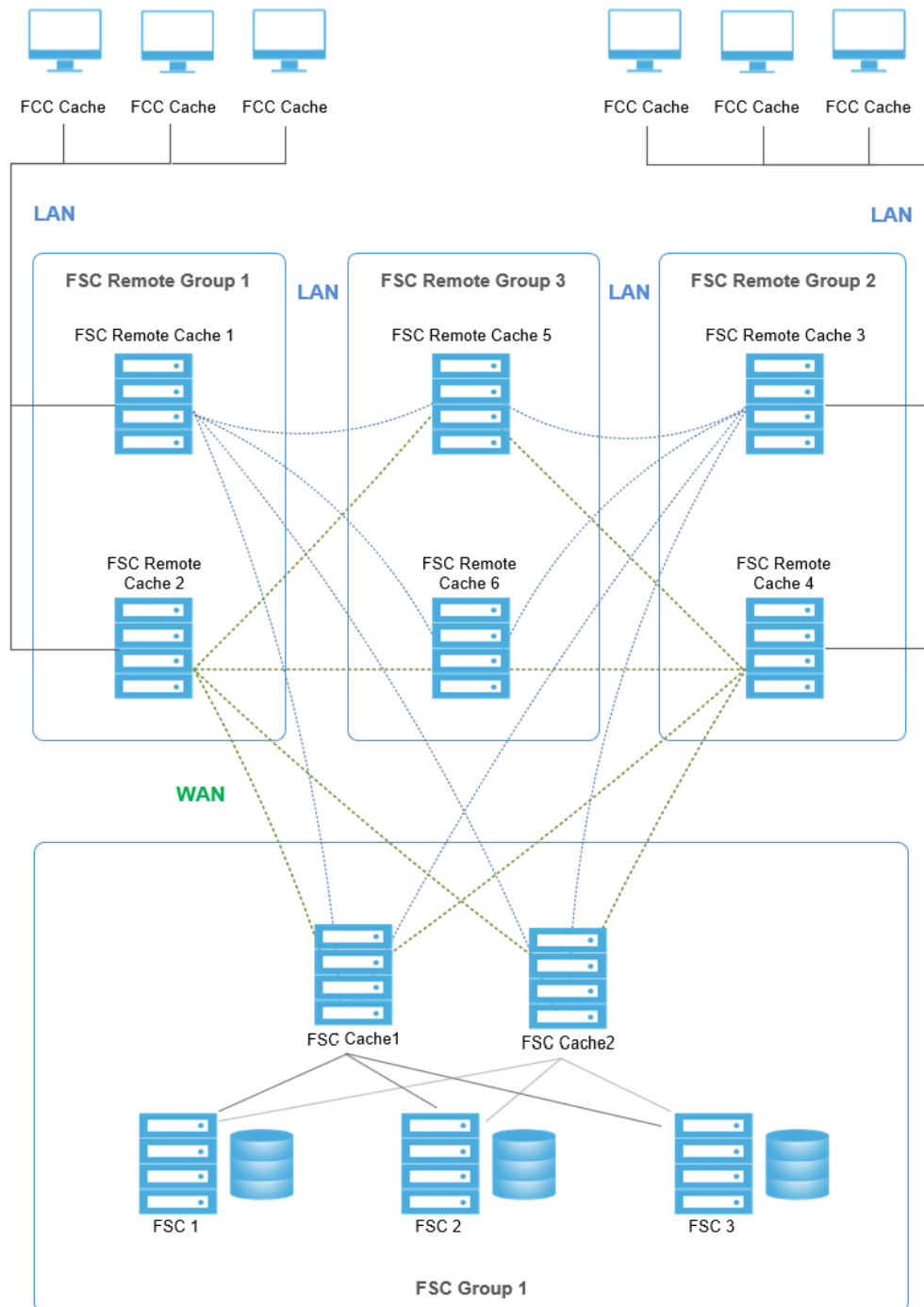
```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

<fccconfig>
  <parentfsc address="http://rsun1.ugs.com:4444" priority="0"/>
  <parentfsc address="http://csun15.ugs.com:4444" priority="1"/>
</fccconfig>
```

FSC remote multiple-level cache failover configuration

This configuration provides failover for either a single point of failure, or failover if both of the FSC Remote Group 3 cache machines fail.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key points of this configuration's configuration file are:

- User assigned groups

Half of the users are assigned to FSC Cache 1 as the primary FSC, half are assigned to FSC Cache 2. If either FSC cache machine fails, the FCCs fail over to the other FSC.

- Hot remote failover

Both of the FSC remote cache machines are caching files. Therefore, if one fails, the other machine takes up the additional traffic. There is potential performance degradation.

- More remote FSC caches

Additional FSC remote cache machines can be added to divide users over more machines. Additional machines decrease performance degradation of a single FSC machine failure.

- Cold failover

All requests from the FSC remote group go through the FSC Cache 1 machine. The FSC Cache 2 machine is idle until there is a failure, in which case the FSC remote cache machines fail to FSC Cache 2. FSC Cache 2 can be assigned local LAN access to utilize this spare capacity.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
    </fccdefaults>
    <fscGroup id="fscRemoteGroup1">
      <fsc id="fscRemoteCache1"
        address="http://rsun1.ugs.com:4444" />
      <fsc id="fscCache2"
        address="http://rsun2.ugs.com:4444" />
    <clientmap subnet="146.122.40.1"
      mask="255.255.255.0">
      <assignedfsc fscid="fscRemoteCache1" priority="0" />
      <assignedfsc fscid="fscRemoteCache2" priority="1" />
    </clientmap>
    <clientmap subnet="146.122.41.1"
      mask="255.255.255.0">
      <assignedfsc fscid="fscRemoteCache2" priority="0" />
      <assignedfsc fscid="fscRemoteCache1" priority="1" />
    </clientmap>
    <grouproute destination="fscGroup1">
      <routethrough groups="fscRemoteGroup3">
```

```

        priority="0" />
        <routethrough groups=""
        priority="1" />
    </grouproute>
</fscGroup>
<fscGroup id="fscRemoteGroup2">
<fsc id="fscRemoteCache1"
    address="http://rsun3.ugs.com:4444" />
    <fsc id="fscCache2"
    address="http://rsun4.ugs.com:4444" />
<clientmap subnet="146.122.42.1"
    mask="255.255.255.0">
    <assignedfsc fscid="fscRemoteCache3" priority="0" />
    <assignedfsc fscid="fscRemoteCache4" priority="1" />
</clientmap>
<clientmap subnet="146.122.43.1"
    mask="255.255.255.0">
    <assignedfsc fscid="fscRemoteCache4" priority="0" />
    <assignedfsc fscid="fscRemoteCache3" priority="1" />
</clientmap>
    <grouproute destination="fscGroup1">
        <routethrough groups="fscRemoteGroup3"
        priority="0" />
        <routethrough groups=""
        priority="1" />
    </grouproute>
</fscGroup>
<fscGroup id="fscRemoteGroup3">
<fsc id="fscRemoteCache5"
    address="http://rsun5.ugs.com:4444" />
    <fsc id="fscRemoteCache6"
    address="http://rsun6.ugs.com:4444" />
</fscGroup>
<fscGroup id="fscGroup1">
    <fsc id="fscCache1" address="http://csun15.ugs.com:4444" />
    <fsc id="fscCache2" address="http://csun16.ugs.com:4444" />
    <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1"/>
    </fsc>
    <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2"/>
    </fsc>
    <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3"/>
    </fsc>
    <entryfsc fscid="fscache1" priority="0" />
    <entryfsc fscid="fscache2" priority="1" />
</fscGroup>
<linkparameters fromgroup="fscRemoteGroup1"
    togroup="fscGroup1"
    transport="wan">
<linkparameters fromgroup="fscRemoteGroup2"
    togroup="fscGroup1"
    transport="wan">
<linkparameters fromgroup="fscRemoteGroup3"
    togroup="fscGroup1"
    transport="wan">
</fmsenterprise>

</fmsworld>

```

- FSC configuration files (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

This FSC configuration file for this configuration is as follows:

FSC1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="true"/>
  <fsc id="fsc1"/>
</fscconfig>
```

FSC2

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="true"/>
  <fsc id="fsc2"/>
</fscconfig>
```

FSC3

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fsc3"/>
</fscconfig>
```

FscCache1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fscCache1"/>
</fscconfig>
```

FscCache2

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
```



```

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fscCache2"/>

</fscconfig>
FSCRemoteCache1

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://rsun5.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://rsun6.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache1"/>

</fscconfig>
FSCRemoteCache2

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://rsun5.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://rsun6.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache2"/>

</fscconfig>
FSCRemoteCache3

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://rsun5.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://rsun6.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache2"/>

</fscconfig>
FSCRemoteCache4

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://rsun5.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://rsun6.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache2"/>

</fscconfig>
FSCRemoteCache5

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

```

```

<fscconfig>
  <fscmaster serves="false" address="http://csun15.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun16.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache5"/>

</fscconfig>
FSCRemoteCache6

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun15.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun16.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache6"/>

</fscconfig>

```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This example of the FCC configuration file is similar to previous examples. The following is the configuration file for FSC Remote Group 1:

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

<fccconfig>
  <parentfsc address="http://rsun1.ugs.com:4444" priority="0"/>
  <parentfsc address="http://rsun2.ugs.com:4444" priority="1"/>

</fccconfig>
Configuration file for remote group 2 clients

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

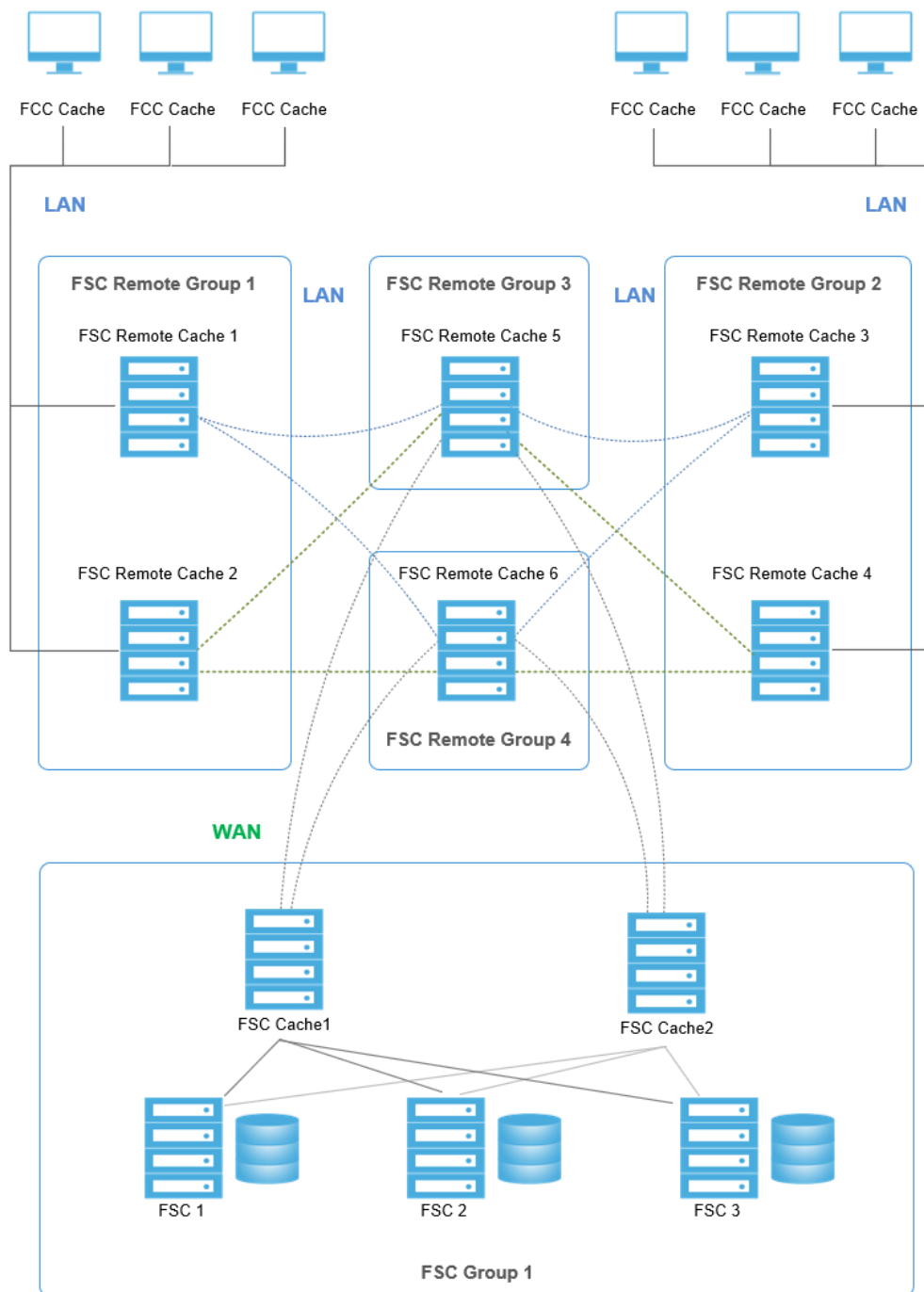
<fccconfig>
  <parentfsc address="http://rsun3.ugs.com:4444" priority="0"/>
  <parentfsc address="http://rsun4.ugs.com:4444" priority="1"/>

</fccconfig>

```

FSC remote multiple-level hot cache failover configuration

This configuration provides active remote cache servers at all remote groups and idle cache servers at the central site.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key points of this configuration's configuration file are:

- User assigned groups

Half of the users are assigned to FSC Cache 1 as the primary FSC, half are assigned to FSC Cache 2. If either FSC cache machine fails, the FCCs fail over to the other FSC.

- Hot remote failover

Both of the FSC remote cache machines are caching files. Therefore, if one fails, the other machine takes up the additional traffic. There is potential performance degradation.

- More remote FSC caches

Additional FSC remote cache machines can be added to divide users over more machines. Additional machines decrease performance degradation of a single FSC machine failure.

- Cold failover

All requests from the FSC remote group go through the FSC Cache 1 machine. The FSC Cache 2 machine is idle until there is a failure, in which case the FSC remote cache machines fail to FSC Cache 2. FSC Cache 2 can be assigned local LAN access to utilize this spare capacity.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <fmsenterprise id="fms.teamcenter.com">
    <fscdefaults>
      <property name="FSC_ReadCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
      <property name="FSC_WriteCacheLocation"
        value="$HOME/FscCache|/tmp/FscCache"
        overridable="true"/>
    </fscdefaults>
    <fccdefaults>
      <property name="FCC_CacheLocation"
        value="$HOME/FCCCache|/tmp/$USER/FCCCache"
        overridable="true"/>
    </fccdefaults>
    <fscGroup id="fscRemoteGroup1">
      <fsc id="fscRemoteCache1"
        address="http://rsun1.ugs.com:4444" />
      <fsc id="fscCache2"
        address="http://rsun2.ugs.com:4444" />
      <clientmap subnet="146.122.40.1"
        mask="255.255.255.0">
        <assignedfsc fscid="fscRemoteCache1" priority="0" />
        <assignedfsc fscid="fscRemoteCache2" priority="1" />
      </clientmap>

      <clientmap subnet="146.122.41.1"
        mask="255.255.255.0">
        <assignedfsc fscid="fscRemoteCache2" priority="0" />
        <assignedfsc fscid="fscRemoteCache1" priority="1" />
      </clientmap>
      <grouproute destination="fscGroup1">
```

```

        <routethrough groups="fscRemoteGroup3"
            priority="0" />
        <routethrough groups="fscRemoteGroup4"
            priority="1" />
    </grouproute>
</fscGroup>
<fscGroup id="fscRemoteGroup2">
    <fsc id="fscRemoteCache1"
        address="http://rsun3.ugs.com:4444" />
    <fsc id="fscCache2"
        address="http://rsun4.ugs.com:4444" />
    <clientmap subnet="146.122.42.1"
        mask="255.255.255.0">
        <assignedfsc fscid="fscRemoteCache3" priority="0" />
        <assignedfsc fscid="fscRemoteCache4" priority="1" />
    </clientmap>
    <clientmap subnet="146.122.43.1"
        mask="255.255.255.0">
        <assignedfsc fscid="fscRemoteCache4" priority="0" />
        <assignedfsc fscid="fscRemoteCache3" priority="1" />
    </clientmap>
    <grouproute destination="fscGroup1">
        <routethrough groups="fscRemoteGroup4"
            priority="0" />
        <routethrough groups="fscRemoteGroup3"
            priority="1" />
    </grouproute>
</fscGroup>
<fscGroup id="fscRemoteGroup3">
<fsc id="fscRemoteCache5"
    address="http://rsun5.ugs.com:4444" />

</fscGroup>
<fscGroup id="fscRemoteGroup4">
<fsc id="fscRemoteCache6"
    address="http://rsun6.ugs.com:4444" />

</fscGroup>
<fscGroup id="fscGroup1">
    <fsc id="fscCache1" address="http://csun15.ugs.com:4444" />
    <fsc id="fscCache2" address="http://csun16.ugs.com:4444" />
    <fsc id="fsc1" address="http://csun17.ugs.com:4444">
        <volume id="vol1" root="/data/vol1"/>
    </fsc>
    <fsc id="fsc2" address="http://csun18.ugs.com:4444">
        <volume id="vol2" root="/data/vol2"/>
    </fsc>
    <fsc id="fsc3" address="http://csun19.ugs.com:4444">
        <volume id="vol3" root="/data/vol3"/>
    </fsc>
    <entryfsc fscid="fscache1" priority="0" />
    <entryfsc fscid="fscache2" priority="1" />
</fscGroup>
<linkparameters fromgroup="fscRemoteGroup3"
    togroup="fscGroup"
    transport="wan">
    <linkparameters fromgroup="fscRemoteGroup4"
        togroup="fscGroup"
        transport="wan">
</fmsenterprise>

```

```
</fmsworld>
```

- FSC configuration files (*FSC_HOME/fsc.xml*)

See the *FSC_HOME/fsc.xml.template* file for all available elements.

This FSC configuration file for this configuration is as follows:

FSC1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="true"/>
  <fsc id="fsc1"/>
</fscconfig>
```

FSC2

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" />
  <fsc id="fsc2"/>
</fscconfig>
```

FSC3

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fsc3"/>
</fscconfig>
```

FscCache1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fscCache1"/>
</fscconfig>
```

FscCache2

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun17.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun18.ugs.com:4444" priority="1" />
  <fsc id="fscCache2"/>

</fscconfig>
FSCRemoteCache1

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun15.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun16.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache1"/>

</fscconfig>
FSCRemoteCache2

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun15.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun16.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache2"/>

</fscconfig>
FSCRemoteCache3

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://rsun5.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://rsun6.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache3"/>

</fscconfig>
FSCRemoteCache4

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://rsun5.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://rsun6.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache4"/>

</fscconfig>
FSCRemoteCache5

<?xml version="1.0" encoding="UTF-8"?>

```

```

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun15.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun16.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache5"/>

</fscconfig>
FSCRemoteCache6

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">

<fscconfig>
  <fscmaster serves="false" address="http://csun15.ugs.com:4444" priority="0" />
  <fscmaster serves="false" address="http://csun16.ugs.com:4444" priority="1" />
  <fsc id="fscRemoteCache6"/>

</fscconfig>

```

- FCC configuration file (*FMS_HOME/fcc.xml*)

This example of the FCC configuration file is similar to previous examples.

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">

<fccconfig>
  <parentfsc address="http://rsun1.ugs.com:4444" priority="0"/>
  <parentfsc address="http://rsun2.ugs.com:4444" priority="1"/>
</fccconfig>

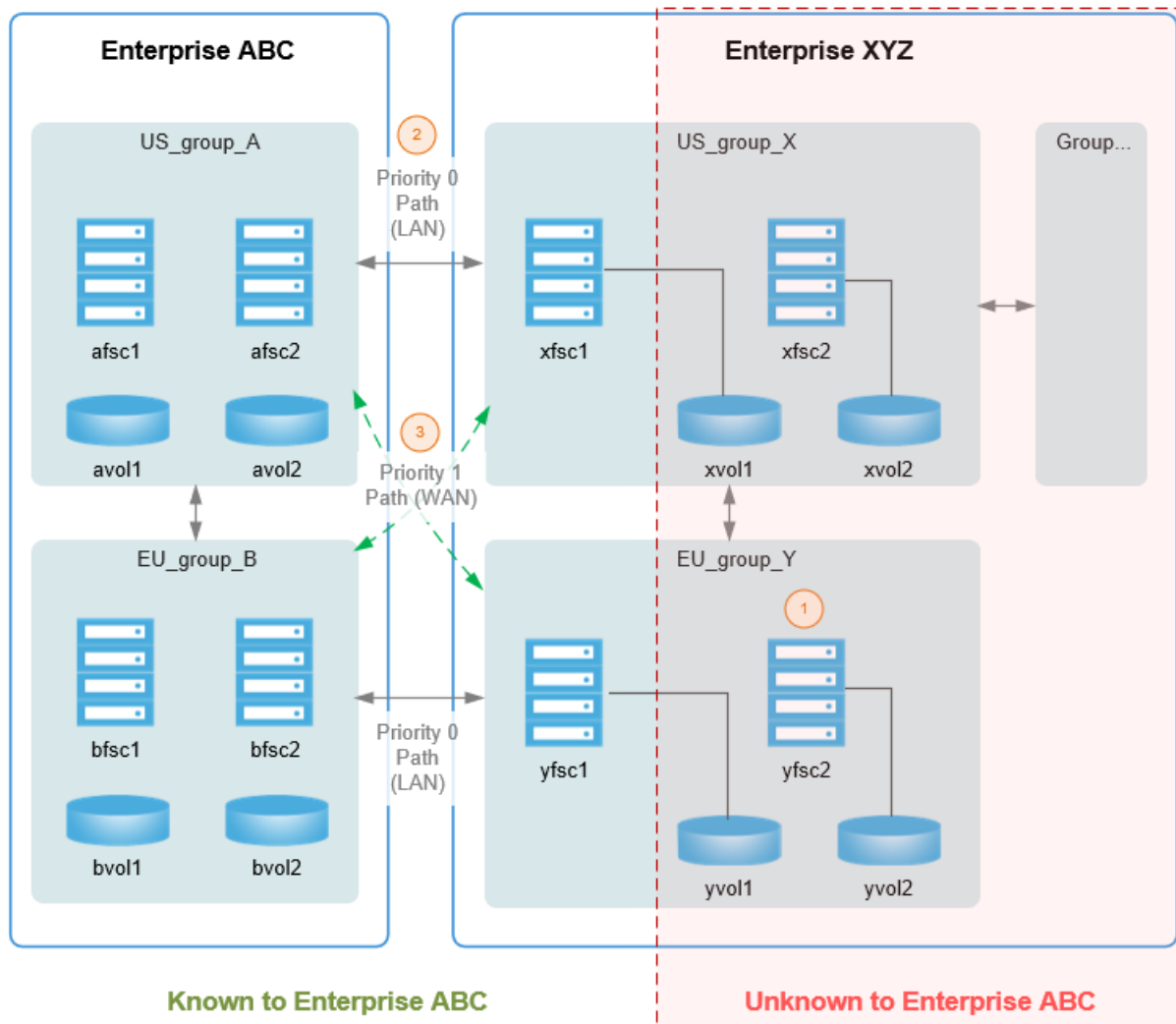
```

FSC group import multisite routing configuration

This configuration illustrates how to import an FSC group from a remote FMS site over a LAN or WAN network.

Note:

1. Not all FSCs or volumes need to be known by Enterprise ABC.
2. multisiteimport for localfscgroup groupA transport LAN priority 0
3. multisiteimport for localfscgroup groupA transport WAN priority 1



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

Use the **localfscgroup** element to express routes between two sites. With this element, you can express multisite routing configuration without exposing the entire network topology of the remote site. Only the gateway FSCs in the remote site need be known by the local site.

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key points of this configuration's configuration file are:

- **fscgroupimport** element

The **fscgroupimport** element defines routes to FSCs in a remote site. This element allows you to define routes to a remote site based on the originating local **fscgroup**. The **fscgroupimport**

contains **defaultfsc** elements, which define the remote FSC address or ID, transport mode and priority for the route. Using **fscgroupimport** elements, a site can express multisite routing configurations without exposing their entire network topology. Only the gateway FSCs in the remote site need be known by the local site.

- **defaultfsc** element

The **fscgroupimport** elements contains **defaultfsc** elements, which can define the remote FSC address or ID, transport mode and priority for the route. Using this element, a site can define a route to a geographically close FSC in the remote site which makes sense for the local site's group.

- **wan** transport attribute

The **wan** attribute allows you to direct remote traffic via the WAN transport mode. Use this attribute to configure WAN routes between geographically distant sites.

```
<fmsworld>
  <multisiteimport siteid="XYZ">
    <defaultfscimport fscid="xfsc1" fscaddress="http://127.100.0.1:5101"
priority="0" />
    <defaultfscimport fscid="yfsc1" fscaddress="http://127.100.0.1:5102"
priority="1" />
    <fscgroupimport groupid="US_group_A">
      <defaultfsc address="http://127.100.0.1:5101" priority="0"/>
      <defaultfsc fscid="yfsc1" priority="1" transport="wan"
maxpipes="4"/>
    </fscgroupimport>
    <fscgroupimport groupid="EU_group_B">
      <defaultfsc fscid="yfsc1" priority="0"/>
      <defaultfsc fscid="xfsc1" priority="1" transport="wan"
maxpipes="4"/>
    </fscgroupimport>
  </multisiteimport>
  <fmsenterprise id="ABC">
    <fscgroup id="US_group_A">
      <fsc id="afsc1" address="http://127.0.0.1:4101">
        <volume id="avol1" root="/avol1"/>
      </fsc>
      <fsc id="afsc2" address="http://127.0.0.1:4102">
        <volume id="avol2" root="/avol2"/>
      </fsc>
    </fscgroup>
    <fscgroup id="EU_group_B">
      <fsc id="bfsc1" address="http://127.0.0.1:4201">
        <volume id="bvol1" root="/bvol1"/>
      </fsc>
      <fsc id="bfsc2" address="http://127.0.0.1:4202">
        <volume id="bvol2" root="/bvol2"/>
      </fsc>
    </fscgroup>
  </fmsenterprise>
</fmsworld>
```

FMS shared network configuration

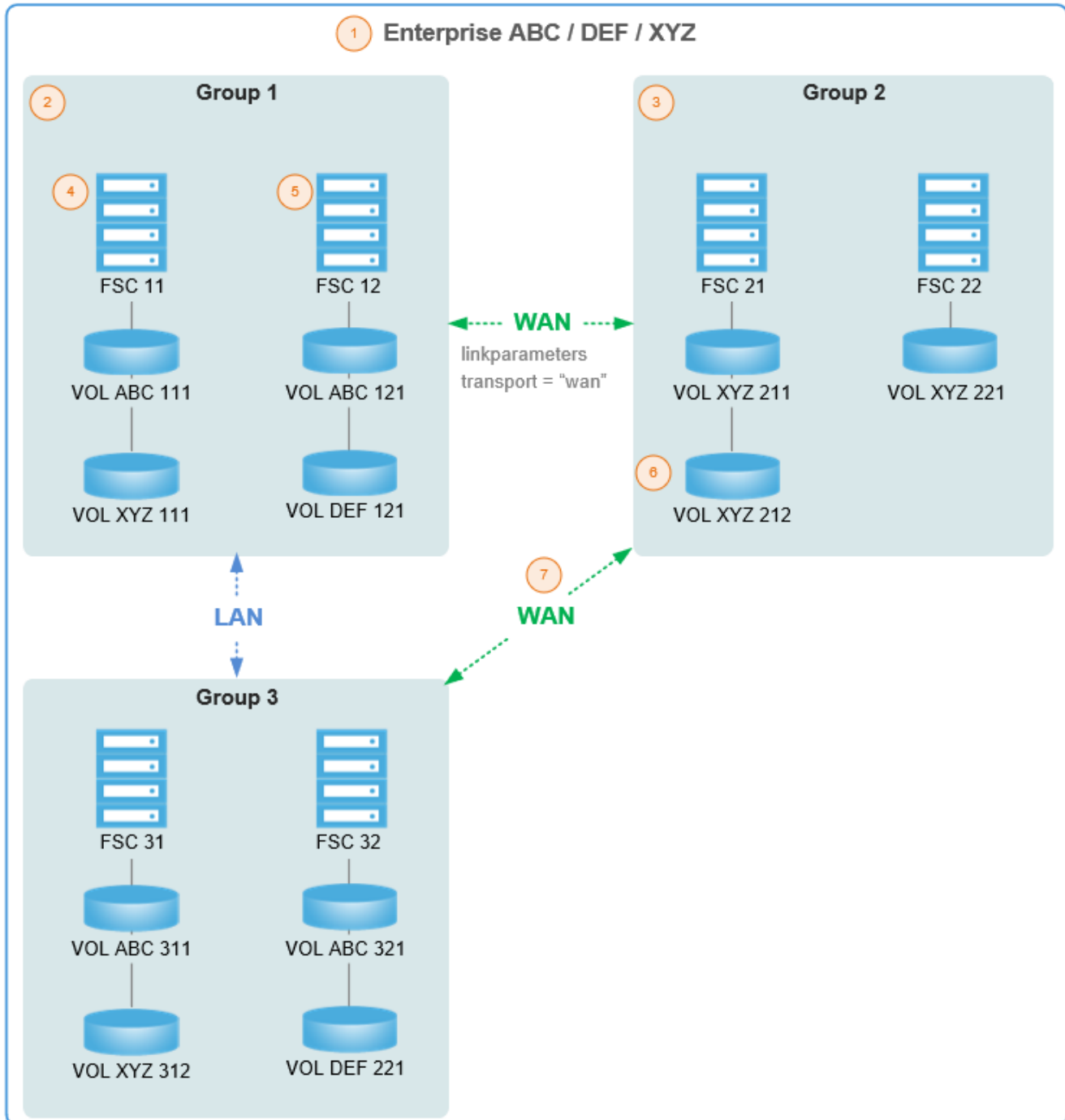
This configuration illustrates how to map other sites onto a local site using a shared network configuration (also known as alias access configuration). In this situation, certain FSCs are shared and capable of managing volumes owned by any defined site. The added benefit of this configuration is that multiple sites/databases can be supported by a single FMS configuration.

- All groups are shared.
- All FSCs are shared.
- All **linkparameters**, **entryfsc**, **exitfsc**, and so on are shared.
- Volumes belong to just one enterprise.

The following diagram demonstrates how to map sites onto a local site using a shared network configuration.

Note:

1. Multiple enterprises share a common FMS configuration.
2. Some groups or FSCs may have volumes for more than one enterprise.
3. Other groups or FSCs may have volumes for only one enterprise.
4. An FSC can mount volumes from any enterprise.
5. Any FSC can handle requests for any enterprise.
6. Any volume is managed (owned) by one of the enterprises.
7. All caching, transport configuration, and routing is common.



- Primary configuration file (*FSC_HOME/fmsmaster_fscid.xml*)

Additional sites can be defined using the **fmsenterprise** element, which uses the same configuration defined by the local enterprise.

This file is configured similar to the previous examples. See the *FSC_HOME/fmsmaster.xml.template* file for all available elements. The key point of this configuration file is:

- **fmsenterprise** element

Define additional sites using the **fmsenterprise** element, which uses the same configuration defined by the local enterprise. This arrangement allows for an FSC to manage volumes owned by either site. Whatever routing is defined in the local site is shared between the sites. (These multisite enterprise elements have the **DEF** and **XYX** attributes in the following configuration file. The advantage of this configuration is that a single FMS configuration can be defined to manage multiple sites (databases). This assumes that the common FSC can be physically shared between the sites.

```
<fmsworld>
  <fmsenterprise id="ABC">
    <fmsenterprise id="DEF"/> <!-- DEF is enterprise site -->
    <fmsenterprise id="XYZ"/>
    <fscgroup id="group1">
      <fsc id="fsc11" address="http://fsc11:4544">
        <volume id="volABC111" enterpriseid="ABC" root="c:/volumes/volABC111"/>
        <volume id="volXYZ111" enterpriseid="XYZ" root="c:/volumes/volXYZ111"/>
      </fsc>
      <fsc id="fsc12" address="http://fsc12:4544">
        <volume id="volABC121" enterpriseid="ABC" root="c:/volumes/volABC121"/>
        <volume id="volDEF121" enterpriseid="DEF" root="c:/volumes/volDEF121"/>
      </fsc>
    </fscgroup>
    <fscgroup id="group2">
      <fsc id="fsc21" address="http://fsc21:4544">
        <volume id="volXYZ211" enterpriseid="XYZ" root="c:/volumes/volXYZ211"/>
        <volume id="volXYZ212" enterpriseid="XYZ" root="c:/volumes/volXYZ212"/>
      </fsc>
      <fsc id="fsc22" address="http://fsc22:4544">
        <volume id="volXYZ221" enterpriseid="XYZ" root="c:/volumes/volXYZ221"/>
      </fsc>
    </fscgroup>
    <fscgroup id="group3">
      <fsc id="fsc31" address="http://fsc31:4544">
        <volume id="volABC311" enterpriseid="ABC" root="c:/volumes/volABC311"/>
        <volume id="volABC312" enterpriseid="ABC" root="c:/volumes/volABC312"/>
      </fsc>
      <fsc id="fsc32" address="http://fsc32:4544">
        <volume id="volABC321" enterpriseid="ABC" root="c:/volumes/volABC321"/>
        <accession id="volDEF321" enterpriseid="DEF" root="c:/volumes/volDEF321"/>
      </fsc>
    </fscgroup>
    <linkparameters fromgroup="group1" togroup="group2" transport="wan" maxpipes="4"/>
    <linkparameters fromgroup="group1" togroup="group2" transport="wan" maxpipes="4"/>
    <linkparameters fromgroup="group1" togroup="group2" transport="wan" maxpipes="8"/>
    <linkparameters fromgroup="group1" togroup="group2" transport="wan" maxpipes="8"/>
  </fmsenterprise>
</fmsworld>
```


29. Server manager

What is the server manager?

The *server manager* (sometimes called a *pool manager*) is a component in the enterprise tier of a 4-tier Teamcenter environment that starts and times out a configurable number of business logic server instances (TcServers). The server manager listens on a single port for incoming requests from web tier clients. As a user logs on to a client, a server assigner process within the server manager assigns an available TcServer. The TcServer performs operations requested by the web tier client and interacts with the Teamcenter database.

The **Teamcenter management console** component can be used to perform many administrative tasks related to the server manager and TcServer instances.

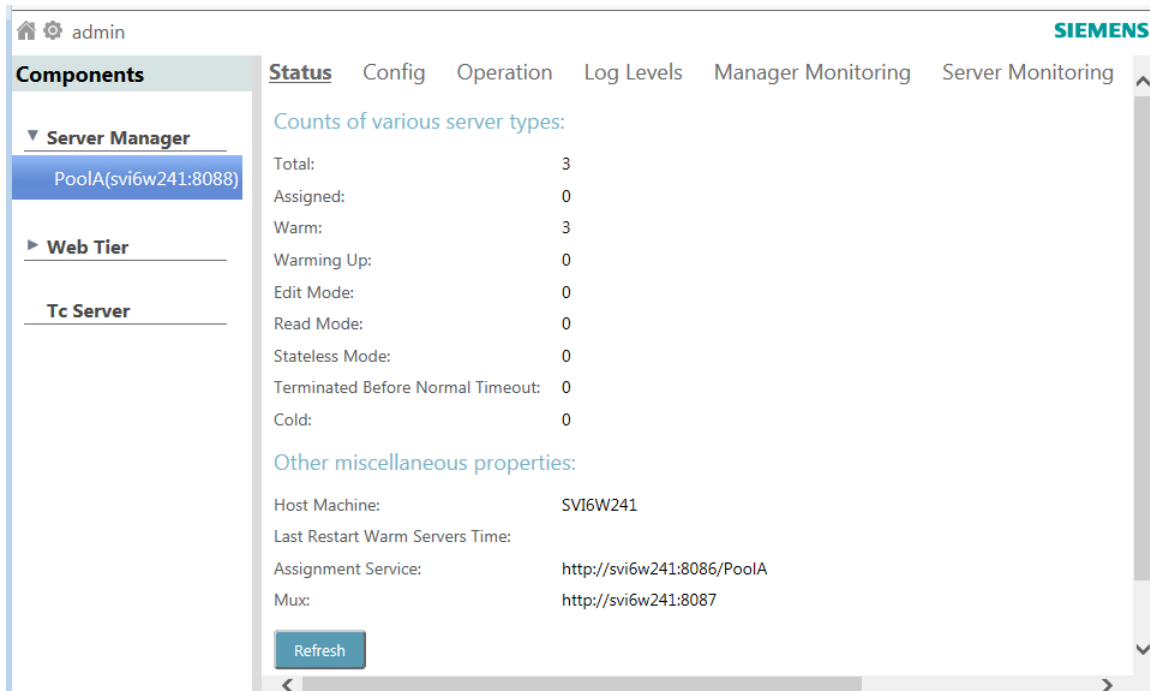
You can use third-party applications to view or manage server manager administration data. Some examples of third-party applications include the following.

JConsole	A monitoring tool complying to Java Management Extensions (JMX) specifications. JConsole uses Java Virtual Machine (Java VM) to provide information about the performance and resource consumption of applications running on the Java platform. This tool is included in the Java development kit (JDK).
Java VisualVM	A monitoring tool that provides a visual interface for viewing detailed information about Java applications while they are running on a Java Virtual Machine (JVM), and for troubleshooting and profiling these applications. Various stand-alone tools, such as JConsole, jstat, jinfo, jstack, and jmap, are part of Java VisualVM. This tool is included in the Java development kit (JDK).
HP Operations Manager	A comprehensive management solution that monitors, controls, automates corrective actions, and reports on the health of all parts of the managed IT infrastructure.

Teamcenter management console

What is the Teamcenter management console?

The Teamcenter management console is a secure web interface for dynamically managing and monitoring server-side components in four-tier Java EE deployments, such as server manager pools, the web tier, and TcServer instances. Administrators can use the console to perform many Teamcenter management tasks from a single page containing tabbed pages for different aspects of components. For security, this console supports SSL.



Changes made in the console affect running Teamcenter components; when the components are stopped and restarted, configurations reset to the values contained in respective configuration files.

Administrative task	Out-of-the-box tools for administering the task
Manage server pools or individual TcServers - Shut down a server pool. - Change pool-specific configurations. - Restart warm TcServer instances.	The Teamcenter management console. The console enables you to monitor and ensure availability of Teamcenter server (TcServer) instances while minimizing resource requirements.
Monitor for critical events - Enable monitoring and configure monitoring metrics to a certain threshold level. - Configure responders for events notifications.	- The Teamcenter management console. - Various server manager monitoring files.
Manage TcServer and server manager logging - Change logging options for showing syslog file and journal output for TcServer instances. - Change the priority of logger levels. - Change the logger levels for the server manager, web tier, and TcServer instances.	- Tool to dynamically view or change logging options for a running component: the Teamcenter management console. - Tool to set startup logging options for the server manager: the log4j2.xml file. - Tool to set startup logging options for all servers in the server pool: the logger.properties file.

Note that the console does not support a .NET (IIS) web tier.

The web-based Teamcenter management console replaced the JMX-based server manager and web tier HTML adapter as of Teamcenter release 11.2. The older interface formerly available at **http://hostname:8082** is obsolete.

Install the Teamcenter management console

This procedure describes using TEM to install the Teamcenter management console. Deployment Center can also be used to install the Teamcenter management console component.

1. In the **Features** panel of Teamcenter Environment Manager (TEM), under **Server Enhancements** select **Teamcenter Management Console** and click **Next** until you reach the **Management Console LDAP Configurations** panel.
2. On the **Management Console LDAP Configurations** panel, do this:
 - a. Select one of **Use Embedded LDAP** or **Use External LDAP**.
 - b. Enter parameters for the selected LDAP.

Use Embedded LDAP

Supplies an embedded lightweight directory access protocol (LDAP) to facilitate authentication and authorization. There is no support for multiple embedded LDAP servers. If you use embedded LDAP, then both the Teamcenter management console and the server manager must reside on the same host.

Use External LDAP

Allows you to point to your own LDAP server. For failover purposes, multiple external LDAPs are supported. To support failover, the multiple LDAP servers must have similar LDAP configurations.

Parameter	Description
Protocol	Select the type of security protocol: ldap , ldaps (secure ldap), or tls (transport layer security).
Host	Type the name where the stand-alone LDAP server resides.
Port	Type the port used by the LDAP instance to listen for connections.
Partition	Type the unique name of the LDAP partition.
Users	Type the name for the list of users to receive LDAP permission. The default value is ou=Users .
User Object Class	Type the class for the users to receive LDAP permission. The default value is inetOrgPerson .

Parameter	Description
User Attribute	Type the attribute that users must have to receive LDAP permission.
User Filter (optional)	Type the LDAP-formatted filter to narrow the pool of authorized users. Use the following format: <i>(attribute1=value1)(attributeN=valueN)</i>

c. Click **Next**.

- In the **JMX Configuration** panel, for each of the server components that you want to view in the Teamcenter management console, add the component and enter its parameter values.

If you add a **Web Tier** component here, then you will later need to **configure its web app server**.

Parameter	Description
Type	Select one of Server Manager or Web Tier .
ID	Type the ID of the component. For example, type PoolA for the default server manager pool or Teamcenter1 for the default web tier. Tip: Obtain the server manager pool name from the smgrPoolId value in the TC_ROOTinstall\configuration.xml file. Obtain the web tier ID from the Web Application Manager used to create the web tier application.
Host	Type the host name of the machine running the server.
JMX RMI Port	Type the JMX RMI port number for the server. To find the correct JMX port for your web server, see your web server documentation. Note: For JBoss web tier configuration, use default port 9999. For Tomcat and WebSphere, use default port 8089.

- In the **Communication Configuration** panel, do one of the following:
 - Accept the default port **8083**.
 - Select the **Enable SSL** check box and enter parameters to use secure communication.

Parameter	Description
Https Port	Type the port to be used for HTTPS communication.
KeyStore	Type the full path and file name to the keystore file.
KeyStore Type	Type the file extension for the keystore.
KeyStore Password	Type the password to the keystore. The password can be entered as plain text or as a password obfuscated using a utility such as jetty.
KeyManager Password	Type the manager password to the keystore.

- After you enter all the installation information, in the **Confirmation** panel click **Start** to install the Teamcenter management console.

Configure the web app server for a Teamcenter management console web tier component

If, when the Teamcenter management console was installed, in the **JMX Configuration panel** a **Web Tier** component was added, then the web application server for the component must be configured to allow authenticated remote JMX connection. Authentication is typically accomplished using LDAP.

This procedure applies only to a Java-based web tier; the console does not support a .NET (IIS) web tier.

Use the following procedures to enable remote JMX connection. The procedure steps vary depending on the web application server software, such as **JBoss**, **Tomcat** or **WebSphere**.

JBoss

JBOSS Enterprise Edition implements its own JMX connector with built-in security to allow JMX Remote Connection. This built-in security does not use the LDAP server used by the Teamcenter management console.

Note:

Siemens Digital Industries Software certifies only JBoss Enterprise Application Platform versions of JBoss. Mapping to appropriate WildFly versions should work, but is not directly tested or certified.

When running in stand-alone mode, the native management endpoint runs with the default port **9999**. To access this native management securely, a management user in the **ManagementRealm** realm is required.

- Edit the JBoss *standalone.xml* configuration file. The file resides in the *JBOSS_HOME/standalone/configuration* directory.
 - In the section `interface`, on the line for the `inet-address`, change the value from the default local host ip address (127.0.0.1) to the actual host name of the machine on which JBoss is running.

```

</profile>
<interfaces>
  <interface name="management">
    <inet-address value="${jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <inet-address value="${jboss.bind.address:127.0.0.1}"/>
  </interface>
</interfaces>

```

- In the section `socket-binding-group`, add the new socket binding:

```

<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
...
  <socket-binding name="management-native" interface="management"
    port="${jboss.management.native.port:9999}" />
...
</socket-binding-group>

```

- In the section `management-interfaces`, depending on JDK version, add native-interface socket binding.

JDK version	
Earlier than JDK 17	<pre> <management-interfaces> <native-interface security-realm="ManagementRealm"> <socket-binding native="management-native"/> </native-interface> ... </management-interfaces> </pre>
JDK 17 or later	<pre> <management-interfaces> <native-interface> <socket-binding native="management-native"/> </native-interface> ... </management-interfaces> </pre>

2. Add a user by running the `JBOSS_HOME/bin/add-user.sh` script (Linux) or `JBOSS_HOME\bin\add-user.bat` script (Windows).

This utility prompts you for the following information:

For this parameter	Do this
Type	Select Management User .
Realm	Enter the name of the realm used to secure the management interface. The default is ManagementRealm .
Username	Type the user name to be created.
Password	Type the password for the user.
Groups	Type the group names that the user should be part of.

For this parameter	Do this
	The default is leave blank, for no groups.
User connects process to another process?	Type no .

Tip:

There is no management user created by default in JBoss. You can add the same user accounts in JBoss as those in the LDAP repository used by the Teamcenter management console.

Teamcenter management console attempts to connect to JBoss Remote Management using its credentials.

- If the same user exists in JBoss Management, then a silent logon occurs in the Teamcenter management console.
- If not, then the Teamcenter management console prompts a logon challenge to connect JBOSS Enterprise Edition Remote Management.

3. Restart JBoss.
4. To allow the Teamcenter management console to connect to JBoss JMX Remote Management, JBoss Client Library is required.
 - a. Stop the Teamcenter management console if it is running.
 - b. Open the Teamcenter management console **com.teamcenter.jeti.mgmt.jmx.jboss-[Teamcenter version].jar** bundle that resides in the *TC_ROOT/mgmt_console/container/bundles* folder using any utility (for example, 7zip).

Inside the bundle, add a new folder called **lib** and add the **jboss-cli-client.jar** to this new folder. The **jboss-cli-client.jar** file can be found in the *JBOSS_HOME/bin/client* directory.

- c. Remove all folders in the *TC_ROOT\mgmt_console\container\data* folder.
- d. Restart the Teamcenter management console.

Tomcat

1. Create an *ldap.config* file with the following content:

```
MgmtLdapConfig {
    com.sun.security.auth.module.LdapLoginModule REQUIRED
    userProvider="ldap://localhost:15389/ou=Users,ou=Management,ou=JETI,
    dc=Teamcenter,dc=PLM,o=Siemens"
```

```

    authIdentity="uid={USERNAME},ou=Users,ou=Management,ou=JETI,
dc=Teamcenter,dc=PLM,o=Siemens"
    authzIdentity=controlRole
    useSSL=false
    debug=false;
};

```

Note:

Ensure that the LDAP settings are configured properly (for example, the LDAP protocol, host, port, and so on.)

2. Establish the *ldap.config* file depending on the web app server and operating system.

For this combination	Do this
Tomcat on Red Hat Enterprise Linux	- Place the <i>ldap.config</i> file in your Tomcat root directory. For example, <i>/apps/apache-tomcat-<version></i>. - Modify the Tomcat startup configuration (<i>catalina.sh</i>) to include the following JAVA_OPTS variable: <pre> JAVA_OPTS="-Dcom.sun.management.jmxremote.port=8089 -Dcom.sun.management.jmxremote.login.config=MgmtLdapConfig -Djava.security.auth.login.config=<PATH>/ldap.config -Dcom.sun.management.jmxremote.ssl=false" export JAVA_OPTS </pre>
Tomcat on other than Red Hat Enterprise Linux	- Place the <i>ldap.config</i> file in your Tomcat root directory. - Modify the Tomcat startup configuration to add the following JAVA_OPTS variable: <pre> set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.port=8089 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.login.config=MgmtLdapConfig -Djava.security.auth.login.config=Tomcat-root-directory\ldap.config" </pre> How you add the options varies depending on how Tomcat is installed. • If Tomcat is installed as a Windows service, then the configuration changes can be made by using the configuration utility. • Otherwise, you must locate the corresponding startup scripts and modify the JVM arguments shown in the preceding example. For example, add the Java arguments in doStart function of the <i>catalina.bat</i> file.

3. Restart the web app server.

WebSphere

The Teamcenter management console does not support secured remote JMX connection to WebSphere due to vendor issues. However, it is possible to configure WebSphere for an unsecured JMX remote connection.

1. Log on to the WebSphere administration console and navigate to the configuration tab of the server instance on which the Teamcenter web tier is deployed.
2. In the **Server Infrastructure** setting group, expand **Java and Process Management** and click **Process definition**.
3. In the **Additional Properties** setting group, click **Java Virtual machine**.
4. Within the **Generic JVM arguments** input field, add the following JVM arguments:

```
-Djavax.management.builder.initial=
-Dcom.sun.management.jmxremote.port=8089
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
```

5. Save and apply the changes and restart the server.

Launch the Teamcenter management console

After you **install the Teamcenter management console**, perform the following steps to launch the console.

Note:

If the Teamcenter management console settings are changed, such as by adding or removing server components, then you must **clear the JETI Management cache** for the changes to reliably take effect.

1. At a system command prompt, set the current directory to the following path as appropriate for the operating system.

Operating system	Directory path
Windows	TC_ROOT\mgmt_console\container\bin\
Linux	TC_ROOT/mgmt_console/container/bin/

2. To start JETI Management, run the following command as appropriate for the operating system:

Operating system	Command
Windows	<code>TC_ROOT\mgmt_console\container\bin\trun</code>
Linux	<code>./trun</code>

Tip:

On Linux platforms, to run the Teamcenter management console in the background, you would expect to run the following command:

```
nohup ./trun &
```

However, **trun** does not support the **nohup** command well, and you must use the following command instead:

```
./start
```

In other words, the **start** script should be used to achieve the same goal. Additionally, the **start** script writes the terminal output to the following file, an equivalent to the **nohup.out** file generated by the **nohup** command:

```
TC_ROOT/mgmt_console/container/data/karaf.out
```

3. To launch the console, in your web browser enter the following URL:

```
http://host:8083/mgmt/console
```

If you encounter difficulties displaying the Teamcenter management console in a Windows Internet Explorer browser, then disable the Internet Explorer **Compatibility View** setting, restart the browser, and reopen the Teamcenter management console.

4. Enter your login credentials and click **Sign in**.

If this is the first time the console is run, then do the following to establish a secure password.

- a. Type **admin** for the user name and **admin** for the password and click **Sign in**.

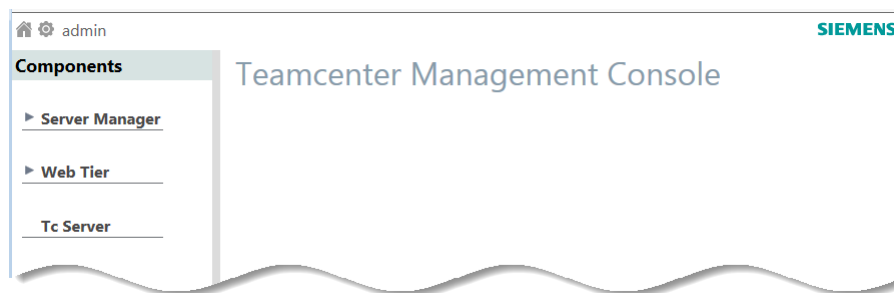
A prompt to change the password appears.

- b. In the prompt to change the password, click **OK**.

- c. In the **Change Password** dialog box, change the password from **admin** to another password and click **Save**.

The password must be a minimum of 8 characters in length and cannot contain the string **admin**.

The console is displayed.



Stopping and restarting JETI Management, clearing the cache

1. To stop JETI Management, in the JETI Management console window press Ctrl+D.

Alternatively, run the following command as appropriate for the operating system.

Operating system	Command
Windows	stop
Linux	./stop

2. To restart JETI Management and clear the JETI Management cache, run the following command as appropriate for the operating system.

Operating system	Command
Windows	trun -clean
Linux	./trun -clean

An alternative method for clearing the cache is to stop JETI Management and then manually delete all the entries in the following directory:

Operating system	Directory
Windows	TC_ROOT\mgmt_console\container\data
Linux	TC_ROOT/mgmt_console/container/data

Run the Teamcenter management console as a service

Teamcenter Environment Manager (TEM) only installs the Teamcenter management console in console mode. Perform the following steps to run the Teamcenter management console as a **Windows service** or as a **Linux service**.

Run the Teamcenter management console as a Windows service

1. **Start the Teamcenter management console** on Windows.
2. At a command prompt, type the following command:

```
feature:install wrapper
```

This provides a new command to install Apache Karaf as a Windows service, for example:

```
karaf@JETIMgmt(>) feature:install wrapper
```

3. At a command prompt, type the following command:

```
wrapper:install -s AUTO_START -n service-name  
-d display_service_name -D service_description
```

For example:

```
karaf@JETIMgmt(>) wrapper:install -s AUTO_START -n Tc-Mgmt-Console-Container  
-d Tc-Mgmt-Console-Container -D "Teamcenter Management Console"
```

Following is an example of the output:

```
Creating file:  
  D:\workdir\user_tc\out\rt\mgmt_console\container\bin\Tc-Mgmt-Console-Container-  
wrapper.exe  
Creating file:  
  D:\workdir\user_tc\out\rt\mgmt_console\container\bin\..\etc\Tc-Mgmt-Console-  
Container-wrapper.conf  
Creating file:  
  D:\workdir\user_tc\out\rt\mgmt_console\container\bin\Tc-Mgmt-Console-Container-  
service.bat  
Creating file:  
  D:\workdir\user_tc\out\rt\mgmt_console\container\lib\wrapper.dll  
Creating file:  
  D:\workdir\user_tc\out\rt\mgmt_console\container\lib\karaf-wrapper.jar  
Creating file:  
  D:\workdir\user_tc\out\rt\mgmt_console\container\lib\karaf-wrapper-main.jar  
  
Setup complete. You may wish to tweak the JVM properties in the wrapper  
configuration file:  
  D:\workdir\user_tc\out\rt\mgmt_console\container\bin\..\etc\Tc-Mgmt-Console-  
Container-wrapper.conf  
before installing and starting the service.  
  
MS Windows system detected:  
  To install the service, run:  
    C:>D:\workdir\user_tc\out\rt\mgmt_console\container\bin\Tc-Mgmt-Console-  
Container-service.bat install  
  
Once installed, to start the service run:  
  C:> net start "Tc-Mgmt-Console-Container"
```

Once running, to stop the service run:
 C:> net stop "Tc-Mgmt-Console-Container"

Once stopped, to remove the installed the service run:
 C:> D:\workdir\user_tc\out\rt\mgmt_console\container\bin\Tc-Mgmt-Console-Container-service.bat remove

4. After setup is completed, follow the instructions to install and start the service on Windows.

For example, to install the Windows service, use a command similar to the following:

```
D:\workdir\user_tc\out\rt\mgmt_console\container\bin\Tc-Mgmt-Console-Container-service.bat install
```

To start the Windows service, use a command similar to the following:

```
net start "Tc-Mgmt-Console-Container"
```

Run the Teamcenter management console as a Linux service

1. **Start the Teamcenter management console** on Linux as root user.
2. Press the **Enter** key to bring up a Karaf command prompt.

Example:

```
karaf@JETIMgmt(>
```

3. At the Karaf command prompt, press the tab key to list all possible commands.
4. Type the following:

```
feature:install wrapper
```

5. At the Karaf command prompt, press the tab key again to list all possible commands.

You will see the wrapper:install command.

Example:

```
feature:install wrapper
```

6. Using the following pattern, type the command to set up your service:

```
wrapper:install -s AUTO_START -n service-name  
-d display_service_name -D service_description
```

For example:

```
karaf@JETIMgmt> wrapper:install -s AUTO_START -n Tc_MgmtConsole_Lnx_Service
-d Tc_MgmtConsole_Lnx -D "Teamcenter Management Console"
```

Following is an example of the output:

```
Creating file:
  /apps/tc/ora/TR/mgmt_console/container/bin/Tc_MgmtConsole_Lnx_Service-wrapper
Creating file:
  /apps/tc/ora/TR/mgmt_console/container/bin/Tc_MgmtConsole_Lnx_Service-service
Creating file:
  /apps/tc/ora/TR/mgmt_console/container/etc/Tc_MgmtConsole_Lnx_Service-
wrapper.conf
Creating file:
  /apps/tc/ora/TR/mgmt_console/container/lib/libwrapper.so
Creating file:
  /apps/tc/ora/TR/mgmt_console/container/lib/karaf-wrapper.jar

Setup complete.
You may wish to tweak the JVM properties in the wrapper configuration file:
  /apps/tc/ora/TR/mgmt_console/container/etc/Tc_MgmtConsole_Lnx_Service-
wrapper.conf
before installing and starting the service.
```

The way the service is installed depends upon your flavor of Linux.

On Redhat/Fedora/CentOS Systems:

To install the service:

```
$ ln -s /apps/tc/ora/TR/mgmt_console/container/bin/
Tc_MgmtConsole_Lnx_Service-service/etc/init.d/
$ chkconfig Tc_MgmtConsole_Lnx_Service-service --add
```

To start the service when the machine is rebooted:

```
$ chkconfig Tc_MgmtConsole_Lnx_Service-service on
```

To disable starting the service when the machine is rebooted:

```
$ chkconfig Tc_MgmtConsole_Lnx_Service-service off
```

To start the service:

```
$ service Tc_MgmtConsole_Lnx_Service-service start
```

To stop the service:

```
$ service Tc_MgmtConsole_Lnx_Service-service stop
```

To uninstall the service:

```
$ chkconfig Tc_MgmtConsole_Lnx_Service-service --del
$ rm /etc/init.d/Tc_MgmtConsole_Lnx_Service-service
```

On Ubuntu/Debian Systems:

To install the service:

```
$ ln -s /apps/tc/ora/TR/mgmt_console/container/bin/
Tc_MgmtConsole_Lnx_Service-service/etc/init.d/
```

To start the service when the machine is rebooted:

```
$ update-rc.d Tc_MgmtConsole_Lnx_Service-service defaults
```

To disable starting the service when the machine is rebooted:

```
$ update-rc.d -f Tc_MgmtConsole_Lnx_Service-service remove
```

To start the service:

```
$/etc/init.d/Tc_MgmtConsole_Lnx_Service-service start
```

To stop the service:

```
$/etc/init.d/Tc_MgmtConsole_Lnx_Service-service stop
```

To uninstall the service:

```
$ rm /etc/init.d/Tc_MgmtConsole_Lnx_Service-service
```

7. After setup is completed, follow the instructions to install and start the service based on your Linux operating system.

For example, on SUSE Linux, enter the following command to install the service:

```
linux-host:/apps/tc/ora/TR/mgmt_console/container # ln
-s /apps/tc/ora/TR/mgmt_console/container/bin/Tc_MgmtConsole_Lnx_Service-service/etc/
init.d/
linux-host:/apps/tc/ora/TR/mgmt_console/container # chkconfig
Tc_MgmtConsole_Lnx_Service-service --add
```

Following is an example of the output to add the service:

Note: This output shows SysV services only and does not include native systemd services.

SysV configuration data might be overridden by native systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.

To see services enabled on particular target use

'systemctl list-dependencies [target]'.

```
Tc_MgmtConsole_Lnx_Service-service 0:off 1:off 2:on 3:on 4:off 5:on 6:off
```

To start the service, use a command similar to the following:

```
linux-host:/apps/tc/ora/TR/mgmt_console/container # service
Tc_MgmtConsole_Lnx_Service-service start
```

Modify settings of the Teamcenter management console

1. Stop JETI Management in one of these ways:

- In the JETI Management console window, press Ctrl+D.
- Run the following command:

```
TC_ROOT\mgmt_console\container\bin\stop
```

2. Launch Teamcenter Environment Manager (TEM). 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 you want to change and click **Next**.
5. In the **Feature Maintenance** panel, under **Teamcenter Management Console** select **Modify settings** and click **Next**.
6. Make changes as needed to the settings. The available panels are the same as those used to **install the Teamcenter management console**.
7. After you make changes, click **Next** in TEM until you arrive at the **Confirmation** panel. Click **Start** to install the changes.
8. Run the following command to stop JETI Management:

```
TC_ROOT\mgmt_console\container\bin\stop
```

9. Run the following command to start JETI Management and to clean out the old settings:

```
TC_ROOT\mgmt_console\container\bin\trun -clean
```

10. Type the following URL in your web browser to run the Teamcenter management console:

```
http://host:8083/mgmt/console
```

The changes you made in TEM should be reflected in the console.

Tip:

If the changes do not appear, there may have been a problem with cleaning out JETI Management information using the **trun -clean** command. To clean out this information, you can remove the files from the following directory:

```
TC_ROOT\mgmt_console\container\data\*.*
```

After removing these files, restart JETI Management.

Administering pools with the Teamcenter management console

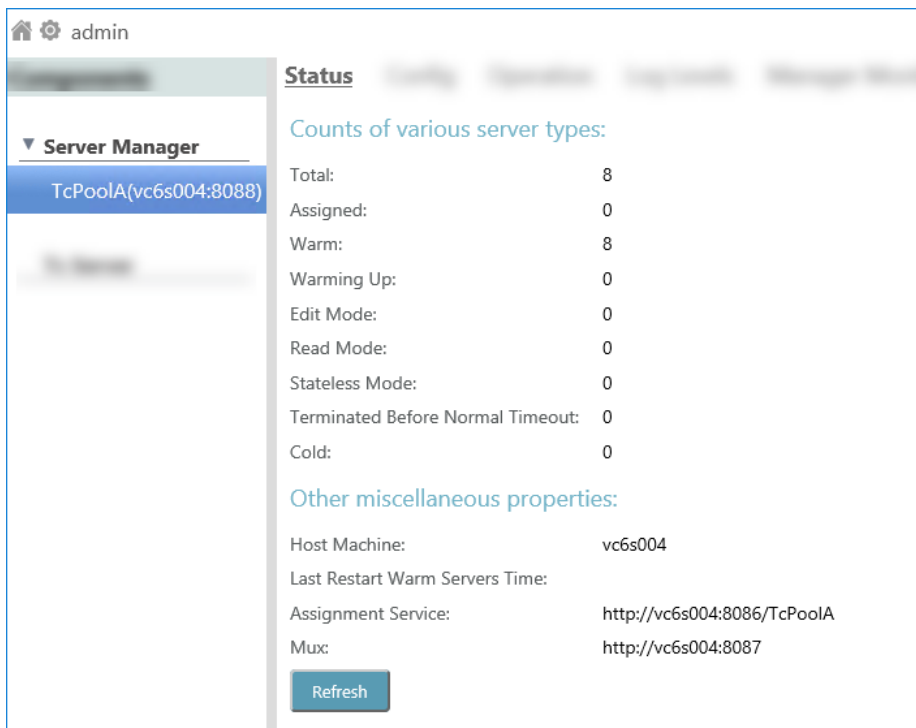
You can use the Teamcenter management console to administer active Teamcenter server pools. To make persistent pool configuration changes, modify the respective `TC_ROOT\pool_manager\confs<config_name>\serverPool.properties` file.

1. **Launch the Teamcenter management console.**

The default address is `http://host:8083/mgmt/console`

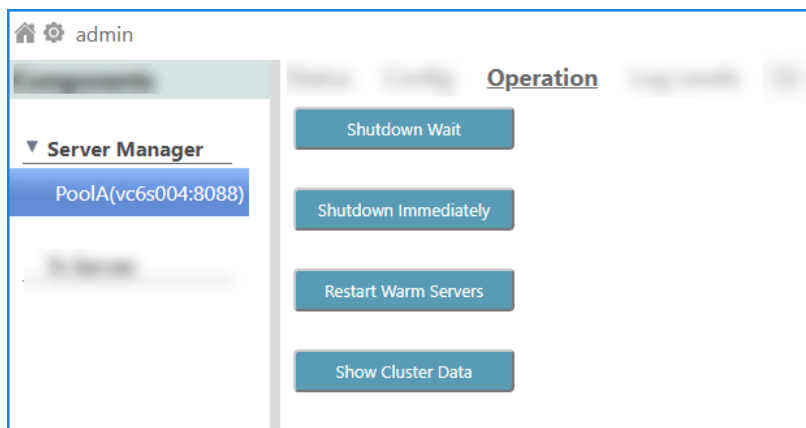
- Under **Server Manager**, select the server pool you want to administer.

The **Status** page of the console displays information for the selected pool.



- Click the **Operation** tab or **Config** tab to show available actions and settings. The following illustrations show the available actions and settings on the tab pages and sub-pages.

Operation tab



Shutdown Wait

Terminate the manager, all of its servers, and TECS with the MUX. Wait for active servers to finish their current request.

Shutdown Immediately Terminate the manager, all of its servers, and TECS with the MUX. Do not wait for active servers to finish their current request.

Restart Warm Servers Restart all warm servers in all pools, so that all unassigned warm servers have a current configuration.

The Restart Warm Servers operation terminates existing warm TcServers (TcServers that are unassigned but ready for use). The Server Manager then spawns new TcServers with the latest configuration, included changes such as server side preference updates.

Show Cluster Data On a new page, show the manager's stored cluster data.

Config tab

Various configuration settings are available on three sub-pages: **Pool Config**, **Advanced Pool Config**, and **Global Pool Config**. To change the selected pool's configuration values, edit the values on a page and then click **Submit** on that page.

Changes made on the **Global Pool Config** page apply to all pools.

The screenshot shows the 'Config' tab for 'PoolA(svi6w241:8088)'. The left sidebar shows 'Server Manager' and 'PoolA(svi6w241:8088)'. The main content area is titled 'Pool Config' and contains the following settings:

Process Warm:	1
Process Target:	0700 3, 1700 2
Process Max:	30
Process Ready Timeout:	300
Process Creation Delay:	2000 2000 8000 16000 30000 60000
Login Per Minute:	0
Max Pool Processing Interval:	600000
Min Pool Processing Interval:	100

At the bottom of the form is a 'Submit' button.

admin

Server Manager

PoolA(svi6w241:8088)

Config

Advanced Pool Config

Writable Configurations:

Acquire Reentrant Lock Timeout:	100
Assignment Retry Limit:	3
Assignment Retry Wait Period:	200
Server Retry Limit:	2
Server Retry Wait Period:	1000
Thread Pool Invoking Servers:	500

☒ Do Warm Logout ☐ Don't do Warm Logout

Submit

Readonly Configurations:

Enable Server Heartbeat:	0
JMX HTTP Adaptor Port:	8082
MGR Protocol:	http
MUX Port:	8087
MUX Protocol:	http
Server Executable:	d:\apps\Siemens\Teamcenter11\bin\tcserver.v
Server Host:	svi6w241
Server ID Prefix:	tcserver
Server Parameter:	

admin

Server Manager

PoolA(svi6w241:8088)

Config

Global Pool Config

Hard Timeout Edit (sec):	28800
Hard Timeout Read (sec):	28800
Hard Timeout Stateless (sec):	28800
Soft Timeout Edit (sec):	28800
Soft Timeout Read (sec):	28800
Soft Timeout Stateless (sec):	1200
Process Max Per User:	0
User Timeout Stateless (sec):	0
Query Timeout (sec):	0
Assignment Max Connections:	100
Manager Availability Interval (sec):	5000
Manager Availability Timeout (sec):	5000

Submit

Configure pool manager monitoring with the Teamcenter management console

This procedure describes how to configure pool manager monitoring with the **Teamcenter management console**. You can also **configure pool manager monitoring with a configuration file**.

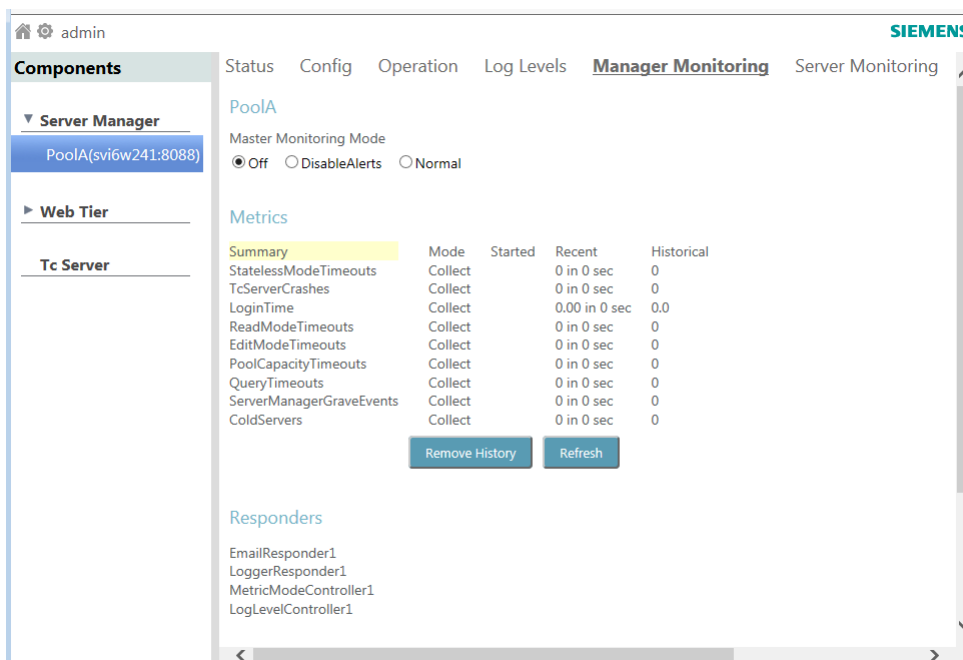
The following dynamic changes are temporary. To make persistent changes, modify the `TC_ROOT\pool_manager\confs\<config_name>\poolMonitorConfig.xml` file.

1. **Start the Teamcenter management console.**

By default, the address is **http://host:8083/mgmt/console**.

2. Under **Server Manager**, select the server pool you want to administer.
3. Click **Manager Monitoring**.

This view lists the monitoring mode and all the metrics available for monitoring.



4. Set the **Master Monitoring Mode** value to one of the following:

- **Off**

Disables monitoring of all the metrics listed in the file.

- **Disable Alerts**

Enables monitoring of all the metrics listed in the file, but disables all notifications of critical events regardless of individual notification settings on any metric.

- **Normal**

Enables monitoring of all the metrics listed in the file.

5. Configure responders.

a. Click **EmailResponder1**.

Responders

EmailResponder1	EmailResponder1	
LoggerResponder1		
MetricModeController1	From_address	CHANGE_TO_VALID_ADDRESS
LogLevelController1	To_address	CHANGE_TO_VALID_ADDRESS
	Host_address	CHANGE_TO_VALID_SMTP_HOST
	Suppression_period	120
	Apply	

(Optional) To be notified when criteria reaches the specified threshold, specify from whom, to whom, and how frequently email notification of critical events are sent by setting the following **EmailResponder1** values. All **EmailResponder1** values in all child monitoring metrics must match the values set here.

- **From_address**

Specify the address from which the notification emails are sent.

- **To_address**

Specify the address to which the notification emails are sent. You can specify multiple email addresses separated by semicolons (;).

- **Host_address**

Specify the server host from which the email notifications are sent. In a large deployment (for example, with multiple server managers) the host address identifies the location of the critical events.

- **Suppression_period**

Specify the amount of time (in seconds) to suppress email notification of critical events.

- b. Click **Apply**.
- c. Click **LoggerResponder1**.

The screenshot shows the 'Responders' configuration interface. On the left, a list of responders includes 'EmailResponder1', 'LoggerResponder1' (which is highlighted), 'MetricModeController1', and 'LogLevelController1'. The 'LoggerResponder1' configuration is displayed on the right, showing the 'Log_filename' field set to 'ServerManagerMonitoring.log' and the 'Suppression_period' field set to '0'. An 'Apply' button is located at the bottom of the configuration area.

(Optional) To be notified when criteria reaches the specified threshold, specify to whom, and to which file, critical events are logged by setting the following **LoggerResponder** values. All **LoggerResponder** values in all child monitoring metrics must match the **LoggerResponder** values set here.

- **Log_filename**

Specify the name of the file to which critical events are logged.

- **Suppression_period**

Specify the amount of time (in minutes) to suppress logging of critical events to the log file.

- d. Click **Apply**.
6. Configure monitoring of any of the server manager metrics listed under the **Metrics** heading.
 - a. Click the desired metric (for example, **ColdServers**). Hover the mouse on each metric to learn about the metric.

Metrics

Summary	Configuration	Events
StatelessModeTimeouts	ColdServers	☐ Off ☒ Collect ☐ Alert
TcServerCrashes	Threshold Value	10
LoginTime	Threshold Period	600
ReadModeTimeouts	History Max Entries	100
EditModeTimeouts	Responder Ids	EmailResponder1 LoggerResponder1 MetricModeController1 LogLevelController1
PoolCapacityTimeouts		
QueryTimeouts		
ServerManagerGraveEvents		
ColdServers		

A cold server is one that did not report back to the server manager within the configured time period (Default: 5 mins) after being started.

Apply

Click **Events** to see events that have occurred for a metric. Hover the mouse over the events to view information about the possible causes and recommended actions for the event.

Metrics

Summary	Configuration	Events
StatelessModeTimeouts	Clear	
TcServerCrashes	Recent Events	
LoginTime	Event Date & Time	Value Server Id Syslog
ReadModeTimeouts	Jun 4, 2015 11:58	Possible Causes:
EditModeTimeouts	tcserver8@PoolA	- High CPU consumption on the machine.
PoolCapacityTimeouts	C:\Temp\tcserver	- Slow responses from DB.
QueryTimeouts		- Trouble getting DB locks.
ServerManagerGraveEvents	Historical Events	Recommended Actions:
ColdServers	Event Date & Time	- Monitor resource consumption in the server pool and the database.
	Jun 4, 2015 11:58	- Check the syslogs for database errors.
	tcserver8@PoolA@16024@SVI6W241	
	C:\Temp\tcserver.exe437c2dd4.syslog	

b. Set the value for the mode (the **Configure_mode** attribute) to one of the following:

- **Off**

Specifies that no metric data is collected.

- **Collect**

Specifies that metric data is collected and displayed in the console view.

This is the default setting.

- **Alert**

Specifies that metric data is collected and displayed in the console view. Email notifications are sent when critical events occur.

- c. (Optional) To be notified when the criteria reaches the specified threshold, specify the **EmailResponder1** and **LoggerResponder1** values for the **Responder Ids** attribute.

By default, these values are set to **EmailResponder1** and **LoggerResponder1**. If you have configured multiple **EmailResponder** IDs, make sure you specify the desired **EmailResponder**.

- d. Set the remaining values to specify criteria that must be met to initiate a critical event for the metric.
- e. After specifying values for each monitoring metric, click **Apply**.

Configure server monitoring with the Teamcenter management console

Administrators can use the following procedure to configure monitoring and alerts of various server events.

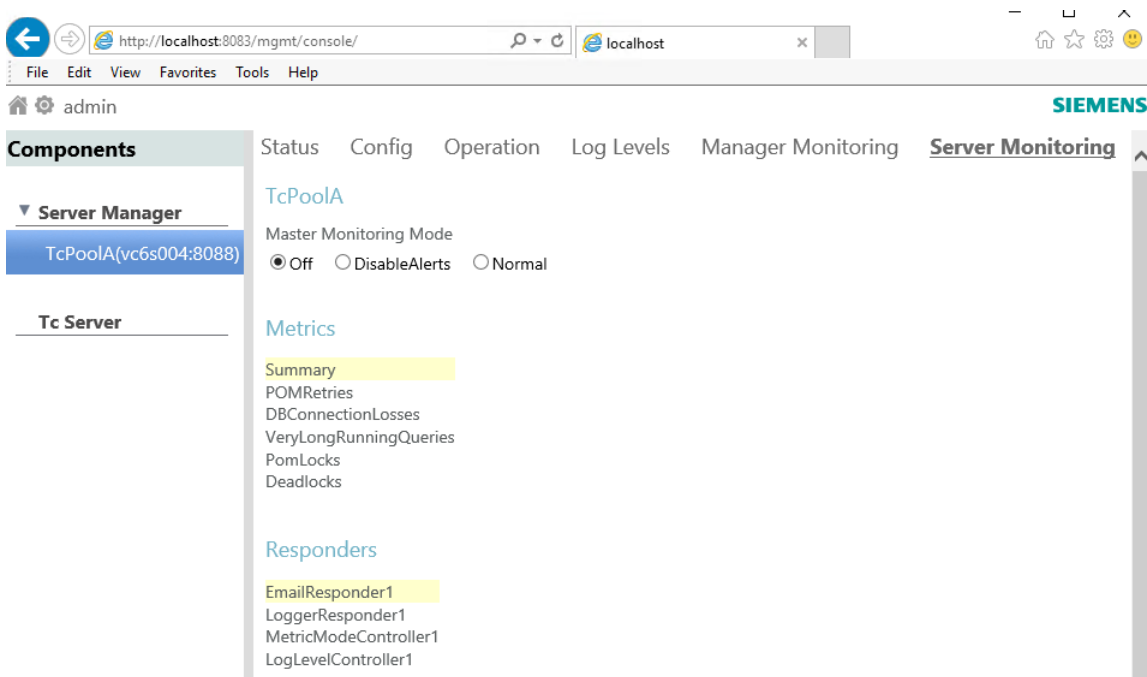
Changes made in the **Server Monitoring** view are persisted in the *TC_DATA\serverMonitorConfig.xml* file.

1. **Start the Teamcenter management console.**

The default address is **http://host:8083/mgmt/console**.

2. Click **Server Monitoring**.

This view lists the master monitoring mode, the event metrics available for monitoring, and responders that can be used for notification of a critical event.



3. Set the **Master Monitoring Mode** value to one of the following:

Value	Description
Off	Disables monitoring of all the metrics listed in the file.
DisableAlerts	Enables monitoring of all the metrics listed in the file, but disables all notifications of critical events regardless of individual notification settings on any metric.
Normal	Enables monitoring of all the metrics listed in the file.

4. To configure monitoring of a server event type (or metric), under the **Metrics** heading, click the desired metric and set its values.

Example:

- a. Click **Deadlocks**.

Metrics

Summary

POMRetries

DBConnectionLosses

VeryLongRunningQueries

PomLocks

Deadlocks

Configuration

Deadlocks

☐ Off ☒ Collect ☐ Alert

Threshold Value

10

History Max Entries

50

Responder Ids

EmailResponder1

LoggerResponder1

MetricModeController1

LogLevelController1

Apply

- b. Set values for the metric.

Value	Description								
Deadlocks	Select one of the following.								
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>Off</td><td>No metric data is collected.</td></tr> <tr> <td>Collect</td><td>Metric data is collected and displayed in the console view. This is the default setting.</td></tr> <tr> <td>Alert</td><td>Metric data is collected and displayed in the console view. Notifications are initiated when critical events occur.</td></tr> </table>	Value	Description	Off	No metric data is collected.	Collect	Metric data is collected and displayed in the console view. This is the default setting.	Alert	Metric data is collected and displayed in the console view. Notifications are initiated when critical events occur.
Value	Description								
Off	No metric data is collected.								
Collect	Metric data is collected and displayed in the console view. This is the default setting.								
Alert	Metric data is collected and displayed in the console view. Notifications are initiated when critical events occur.								
Threshold Value	Enter the event quantity at which to initiate a critical event notification.								
History Max Entries	Enter the event quantity at which to stop logging entries.								
Responder Ids	In the list, the active critical event alert notification responders for the metric are highlighted. To select a								

Value	Description

responder, click its name. Use Ctrl+click to select multiple responders.

- c. Click **Apply**.

5. Configure the responders that can be used for alert notification of a critical event.

Example:

To configure email notification, click **EmailResponder1** and set the values. When you finish specifying values, click **Apply**.

Responders

EmailResponder1	EmailResponder1	
LoggerResponder1	From_address	CHANGE_TO_VALID_ADDRESS
MetricModeController1	To_address	CHANGE_TO_VALID_ADDRESS
LogLevelController1	Host_address	CHANGE_TO_VALID_SMTP_HOST
	Suppression_period	43200
	Apply	

Value	Description
From_address	The address from which notification emails are sent.
To_address	The address to which notification emails are sent. You can specify multiple email addresses separated by semicolons (;).
Host_address	The server host from which the email notifications are sent. In a large deployment (for example, with multiple server managers) the host address identifies the location of the critical events.
Suppression_period	The amount of time (in seconds) to suppress email notification of critical events.

Example:

To configure logging of critical events, click **LoggerResponder1** and set the following values. When you finish specifying values, click **Apply**.

Responders

EmailResponder1	LoggerResponder1	
LoggerResponder1	Log_filename	TcServerMonitoring.log
MetricModeController1	Suppression_period	1000
LogLevelController1		
Apply		

Value	Description
Log_filename	The name of the file to which critical events are logged.
Suppression_period	The amount of time (in seconds) to suppress logging of critical events to the log file.

Administer logging levels with the Teamcenter management console

You can change logging levels dynamically for server manager pools, the web tier, or **tcserver** instances.

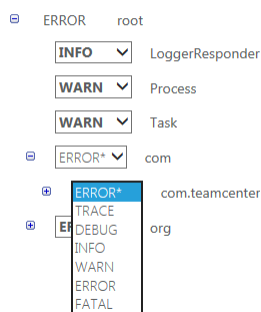
The following dynamic changes are temporary. To make persistent changes, modify the `TC_ROOT\pool_manager\confs\<config_name>\log4j2.xml` file.

Select the component, click **Log Levels**, browse to the element for which you want to change the logging, and click the arrow on the box to select a different **log level**.

The bold text in the priority level indicates the level is configured explicitly, for example, **INFO** in the following example. The entries with an asterisk indicate that the priority level is inherited from its parent, for example, **ERROR***.

Status Config Operation **Log Levels**

PoolA

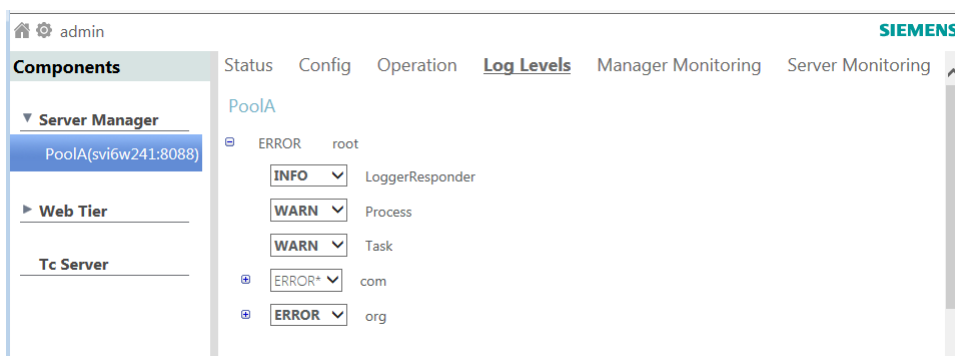


Perform the following steps to learn how to navigate to the logging for each type of component (server manager, web tier, and **tcserver**).

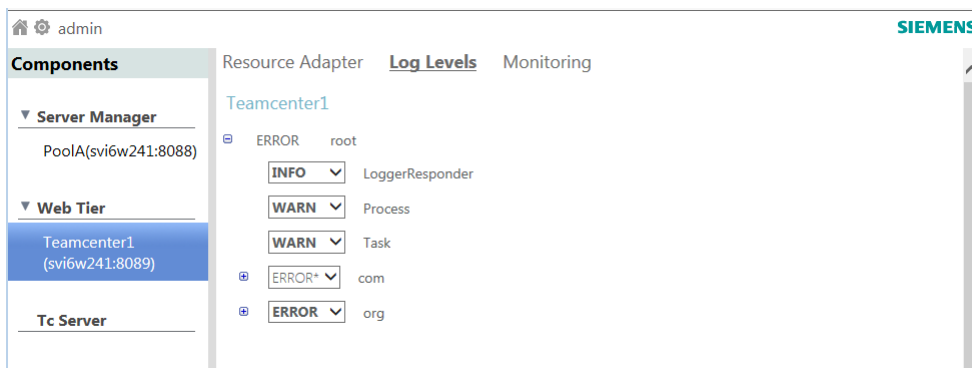
1. **Start the Teamcenter management console.**

By default, the address is **http://host:8083/mgmt/console**.

2. Under **Server Manager**, select a server pool and click **Log Levels**.



3. To access logging for the web tier, select the web tier instance and click **Log Levels**.



4. Perform the following steps to access logging for a **tcserver** instance:

- a. Select **TcServer** and click the **Search** button.

The available instances are displayed.

- b. Select a **tcserver** instance.
- c. Click **Logging** to set the kind of logging you want for the selected instance.

The screenshot shows the Siemens Teamcenter Administration console. On the left, the 'Components' sidebar has 'Tc Server' selected. The main area displays 'TcServer Filter Options' with checkboxes for 'All Pools' and 'PoolA@svi6w241'. Below this is a table of tcserver instances:

Server	PID	Lifecycle	User	Mode	Status	Duration(sec)
tcserver27@PoolA@3732@SVI6W241	9672			Stateless	Idle	116811
tcserver26@PoolA@3732@SVI6W241	17432	Assigned	infodba	Read	Idle	5442
tcserver28@PoolA@3732@SVI6W241	18004	Warm		Stateless	Idle	30411

Below the table, the 'Logging' tab is selected. It shows 'Syslog Options' with a 'Checking Level' of 0 and checkboxes for 'SQL Logging', 'Profile', 'Show Bind Variables', 'Timing', and 'To Journal File'. The 'Syslog Operations' section has a 'Show Syslog' button and a 'Max Size (KB)' field set to 100000. The 'Journaling Options' section has checkboxes for 'Write Journal Entries' and 'Collect Journal Statistics', and a 'Config for modules' button. The 'Journaling Operations' section has a 'Show Journal' button, a 'Max Size (KB)' field set to 100000, and buttons for 'Clear Statistics' and 'Dump Statistics to Syslog'.

- d. Click **Log Levels** to change the levels for Teamcenter elements in the selected **tcserver** instance.

The screenshot shows the Teamcenter management console interface. On the left, the 'Components' sidebar is expanded to 'Web Tier', showing 'Teamcenter1 (svi6w241:8089)' and 'Tc Server'. The main area is titled 'TcServer Filter Options'. It includes checkboxes for 'All Pools' (checked), 'Mode' (Stateless, Read, Edit), 'Status' (Active, Idle), and an 'Assigned User' field. Below this is a table of servers:

Server	PID	Lifecycle	User	Mode	Status	Duration (sec)
tcserver27@PoolA@3732@SVI6W	9672			Stateless	Idle	116811
tcserver26@PoolA@3732@SVI6W	17432	Assigned	infodba	Read	Idle	5442
tcserver28@PoolA@3732@SVI6W	18004	Warm		Stateless	Idle	30411

Below the table, there are tabs for 'Status', 'Logging', 'Log Levels', and 'Monitoring'. The 'Log Levels' tab is selected, showing a tree view of log levels for various components, all set to 'INFO'.

Administer the web tier with the Teamcenter management console

An administrator can use the Teamcenter management console to dynamically view and change server and web tier parameters. The following procedure applies dynamic changes to running components, but does not permanently alter the component configurations and settings.

1. **Start the Teamcenter management console.**

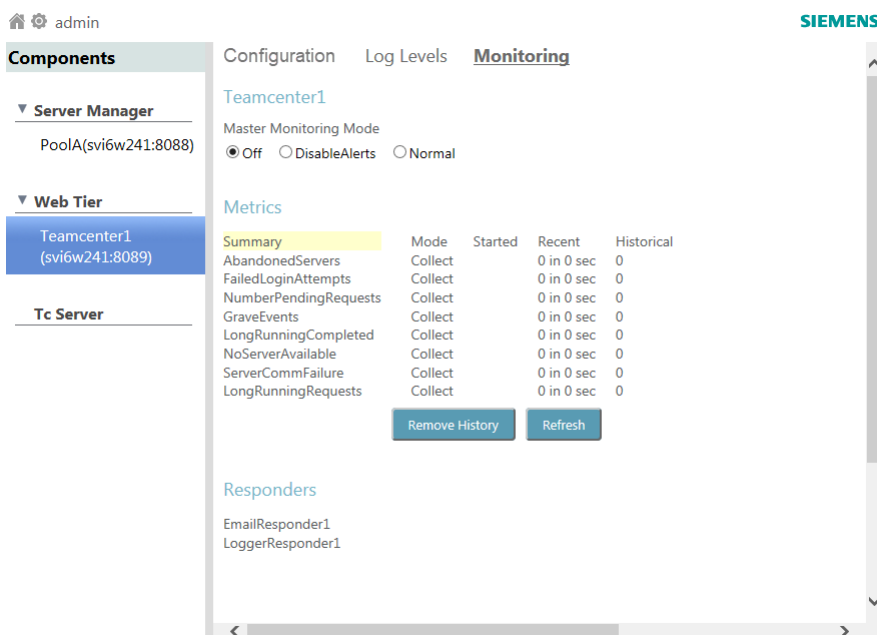
By default, the address is **http://host:8083/mgmt/console**.

2. Under **Web Tier**, select the web tier server you want to administer.

3. Select **Configuration** to administer the web tier. Enter values for **Retry Limit** and **Retry Delay** and then click **Submit** to update the web tier pool.

The screenshot shows the 'Configuration' tab selected in the Teamcenter management console. The 'Components' sidebar on the left shows 'Teamcenter1 (si0vm451)' selected under 'Web Tier'. The main area displays the 'Configurations' section with two input fields: 'Retry Limit' set to 2 and 'Retry Delay' set to 500. A 'Submit' button is located below these fields.

4. Select **Log Levels** to **change logging** for the web tier.
5. Select **Monitoring** to manage and view the web tier monitoring metrics.



See [Web tier monitoring metrics](#) for more information about the metrics.

Web tier monitoring configuration files

To make persistent changes, modify the configuration or setting in the Web Application Manager (insweb), regenerate *tc.war*, and redeploy *tc.war* in the application server.

The web tier monitoring files *webMonitorConfig.xml* and *webMonitorConfigInfo.xml* reside inside the *tc.war* file (*WEB-INF\lib\mldcfg.jar*). An administrator can extract *webMonitorConfig.xml* and *webMonitorConfigInfo.xml* files and place them in the root folder of the application server, or update the *webMonitorConfig.xml* file inside the *tc.war* file.

The web tier configuration file *log4j2.xml* resides inside *tc.war* file (*WEB-INF\lib\mldcfg.jar*). An administrator can extract this *log4j2.xml* file and place it in the root folder of the application server or update this *log4j2.xml* file inside the *tc.war* file.

Administer TcServer instances in the Teamcenter management console

TcServer instances are logical business-logic servers in the server pool. Use the **TcServer** component in the Teamcenter management console to dynamically configure and monitor these servers.

Dynamic changes to tcserver log level are temporary. To make persistent changes to the TcServer log level, an administrator can modify the TcServer logger in the *TC_DATA\logger.properties* file.

1. Start the Teamcenter management console.

By default, the address is **http://host:8083/mgmt/console**.

2. Select TcServer and click the Search button.

The available server instances are displayed along with related information such as their process IDs, modes, and status. Each server's lifecycle stage is also displayed. **Lifecycle** has the following stages:

Warming	The server has started, but is not ready.
Ready	The server is ready to be assigned to a user.
Assigning	The server is in the process of being assigned to a user.
Assigned	The server is assigned to a user session.

3. Select a TcServer instance.

The screenshot shows the Teamcenter management console interface. On the left, the 'Components' sidebar is expanded to 'Server Manager', showing a tree view with 'PoolA(vc6s012:8088)' and a 'Tc Server' button. The main area displays 'TcServer Filter Options' with checkboxes for 'All Pools', 'Mode' (Stateless, Read, Edit), 'Status' (Active, Idle), 'Lifecycle' (Warming, Ready, Assigning, Assigned), and an 'Assigned User' field. Below the filters is a table of TcServer instances. The table has columns: Server, PID, Lifecycle, User, Mode, Status, and Duration(sec). The instance 'tcserver02608@PoolA@2148@v' is selected. At the bottom, there are tabs for 'Status', 'Logging', 'Log Levels', and 'Monitoring', with 'Status' currently active. A 'Hide' button is visible on the right side of the table.

Server	PID	Lifecycle	User	Mode	Status	Duration(sec)
tcserver00001@PoolA@2148@v	8000	Assigned	karia	Stateless	Idle	13
tcserver00022@PoolA@2148@v	3124	Assigned	ed	Stateless	Idle	2
tcserver02477@PoolA@2148@v	29628	Assigned	karia	Read	Idle	25893
tcserver02478@PoolA@2148@v	24812	Assigned	ace_revbom	Read	Idle	25858
tcserver02608@PoolA@2148@v	33656	Assigned	nxqa1	Edit	Idle	17130
tcserver02924@PoolA@2148@v	32660	Assigned	nxqa1	Edit	Idle	17112
tcserver03577@PoolA@2148@v	38584	Assigned	ed	Stateless	Idle	123
tcserver03609@PoolA@2148@v	34980	Assigned	ed	Stateless	Idle	214
tcserver03624@PoolA@2148@v	33072	Assigned	karia	Stateless	Idle	2491
tcserver03658@PoolA@2148@v	35368	Assigned	agarkar	Stateless	Idle	1090

4. Select Logging or Log Levels to change logging for the server instance.

Click **Logging** to view and change the active logging options for the selected **TcServer** instance. The following options are useful for troubleshooting:

- **Show Syslog**

Displays the most recent portions of the **tcserver** system log in a separate browser.

- **Config for modules**

Shows each Teamcenter module known to the journaling system. Select the modules for which journaling is active.

- **Show Journal**

Displays the most recent parts of the journal file in a separate browser.

Status **Logging** Log Levels Monitoring

Syslog Options

Checking Level 0

☐ SQL Logging ☐ Profile ☐ Show Bind Variables ☐ Timing ☐ To Journal File

Syslog Operations

Show Syslog Max Size (KB): 100000

Journaling Options

☐ Write Journal Entries ☐ Collect Journal Statistics [Config for modules](#)

Journaling Operations

Show Journal Max Size (KB): 100000

Clear Statistics

Dump Statistics to Syslog

5. Select **Monitoring** to view the server instance settings.

Tip:

Many of the settings are read-only because the master settings are **configured in the server pool**.

Status Logging Log Levels **Monitoring**

Master Monitoring Mode

☒ Off ☐ DisableAlerts ☐ Normal

Metrics

Summary	Configuration	Events
POMRetries	Deadlocks	☐ Off ☒ Collect ☐ Alert
OsMemoryTotal	Threshold Value	10
OmRollbackData	History Max Entries	50
DBConnectionLosses	Responder Ids	EmailResponder1 LoggerResponder1 MetricModeController1 LogLevelController1
VeryLongRunningQueries		
PomLocks		
Deadlocks		
SqlTripCount		
SqlTotalTime		
OsBsmUndoPool		
OsMemoryPeak		
DetailedSqlStats		
TimesDBOutOfSpace		
BsmUndoPoolInUse		
OmModelData		

Inactivate and activate a server pool

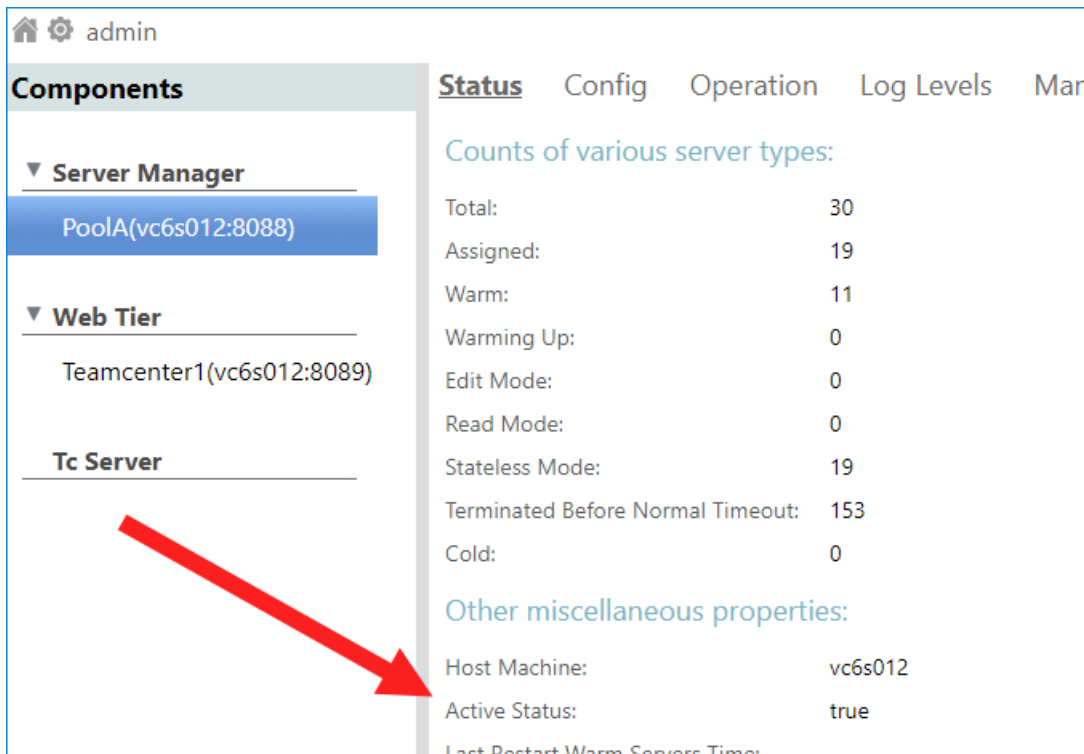
Inactivating a running server pool prevents it from accepting new connections. After notifying users who are connected to the pool to close down or re-login, the administrator can perform maintenance tasks, such as perform patch install or system level maintenance, and then later reactivate the pool. Existing TcServer assignments in the pool continue to serve clients and can be managed as usual, as long as the pool is still running.

1. **Launch the Teamcenter management console.**

By default, the address is **http://host:8083/mgmt/console**.

2. Under **Server Manager**, select the server pool you want to administer.

The **Status** tab of the console displays information on the selected pool. An **Active Status** of **true** means the pool is available for TcServer assignments to new Teamcenter login requests.

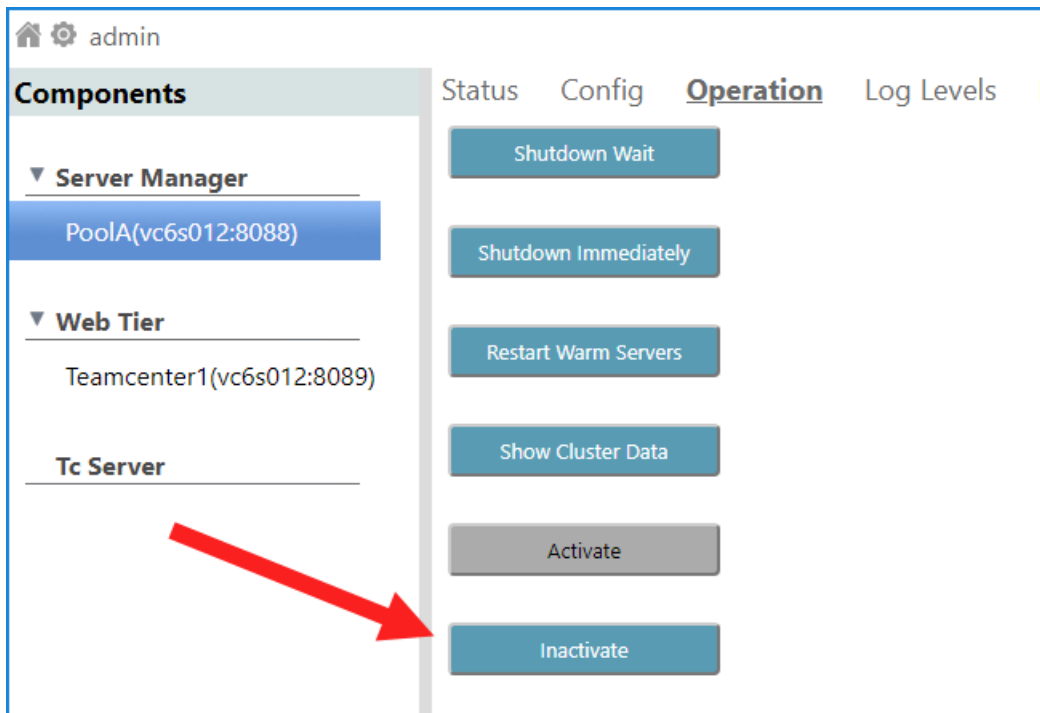


The screenshot shows the Teamcenter management console interface. On the left, under the 'Components' section, 'Server Manager' is expanded, and 'PoolA(vc6s012:8088)' is selected. The main area displays the 'Status' tab for this pool. It includes a table of server counts and a section for miscellaneous properties. A red arrow points to the 'Active Status' property, which is set to 'true'.

Counts of various server types:	
Total:	30
Assigned:	19
Warm:	11
Warming Up:	0
Edit Mode:	0
Read Mode:	0
Stateless Mode:	19
Terminated Before Normal Timeout:	153
Cold:	0

Other miscellaneous properties:	
Host Machine:	vc6s012
Active Status:	true
Last Restart Warm Servers Time:	

3. To change the **Active Status** value, click the **Operation** tab and then click the **Activate** or **Inactivate** button as appropriate.



- To confirm the change in status, click the **Status** tab and check the value of **Active Status**.

Administer TcServer timeouts in the Teamcenter management console

The server manager may timeout (terminate) idle TcServer instances. A TcServer instance is considered idle if it does not receive any requests from the client application.

Use the **Global Pool Config** tab to dynamically configure Teamcenter server timeout values. To reach the tab, choose **Server Manager**>[pool name]>**Config**>**Global Pool Config**.

The screenshot shows the Teamcenter management console interface. On the left, a sidebar lists components: Components, Server Manager (selected), Web Tier, and Tc Server. Under Server Manager, 'PoolA(localhost:8088)' is selected. The main area has tabs: Status, **Config**, Operation, Log Levels, Manager Monitoring, and Server Monitoring. Under the Config tab, there are sub-tabs: Pool Config, Advanced Pool Config, and **Global Pool Config**. The Global Pool Config tab contains the following settings:

Hard Timeout Edit (sec):	28800
Hard Timeout Read (sec):	28800
Hard Timeout Stateless (sec):	28800
Soft Timeout Edit (sec):	28800
Soft Timeout Read (sec):	28800
Soft Timeout Stateless (sec):	28800
Process Max Per User:	0
User Timeout Stateless (sec):	0
Query Timeout (sec):	0

A 'Submit' button is located at the bottom left of the configuration area.

When you modify timeout values on the **Global Pool Config** tab and then click **Submit**, the timeout values are immediately applied to all Server Manager pools and all TcServer instances in the cluster. To make persistent changes to the TcServer timeout values, an administrator can modify the respective setting in `TC_ROOT\pool_manager\confs\<config_name>\serverPool.properties` file.

How server manager computes timeout application

The server manager computes the timeout value to apply to a server instance based on a combination of the server pool load, the instance's server mode, and the applicable timeout parameter value. Hard and Soft timeout parameters specify a range of timeout values for each of three server modes.

Situation	Server mode	Result of timing out the server
The client has made unsaved edits.	Edit	Changes are lost.
Example: Structure Manager uses edit mode to allow users to edit a BOM structure through multiple operations that change temporary		

Situation	Server mode	Result of timing out the server
data in the server and client until the user saves the data.		
	Read	A performance or functional issue may occur, but no loss of edits occurs. The cost of timing out a TcServer in Read mode is the time a user has to spend to get back to the state in existence at the time the server was timed out.
	No persistent data is cached in memory. Stateless	No performance, functional issue, or loss of data occurs. The only cost of timing out a TcServer in Stateless mode is the cost of a server assignment and authentication when the client does become active again.

Edit and Read mode occur mostly with rich client and NX. Stateless mode occurs mostly with Active Workspace.

The server manager never applies timeout values that are longer than the Hard values. Therefore, each Soft value should be set less than or equal to the corresponding Hard value. As the count of TcServer instances exceeds the **Process Target** value, the server manager incrementally reduces the timeout value it applies, from the Hard value to the Soft value. The total number of TcServer instances includes both assigned and warm processes. When the count reaches 110% of **Process Target**, the server manager fully applies the Soft timeout value.

If the total count of TcServers reaches 95% of **Process Max** (PROCESS_MAX), the server manager further incrementally reduces timeout values. This reduction proceeds to as low as 20% of the Soft timeout values.

The following table describes the additional timeout parameters that appear on the **Global Pool Config** tab:

This parameter	Configures this
Process Max Per User	The maximum number of running processes (server instances) allowed per user.
User Timeout Stateless	If a user hits the limit defined by the Process Max Per User value, the timeout value applied to stateless server instances that are assigned to the user.
Query Timeout	The maximum time a server is allowed to process a single request. If this time is exceeded, the server is terminated.

This parameter	Configures this
	A value of 0 turns off the query timeout, allowing a server to continue processing a request indefinitely.

Example log messages

The server manager logs messages for automated changes in idle timeout values and when a TcServer does timeout:

```
PoolA.00041.ServerStarter4Tier - Adjust idle timeouts to 60% of soft
timeouts:
PoolA.00041.ServerStarter4Tier - Edit idle timeout: 04:48:00 (17280s)
PoolA.00041.ServerStarter4Tier - Read idle timeout: 04:48:00 (17280s)
PoolA.00041.ServerStarter4Tier - Stateless idle timeout: 12:00 (720s)
...
vcl6006.96233728.tcserver93.2 - TcServer tcserver93@PoolA@81958@vcl6006
stateless idle timeout.
vcl6006.96233728.tcserver93.3 - TcServer tcserver93@PoolA@81958@vcl6006
removed.
vcl6006.96233728.tcserver93.3 - TcServer tcserver93@PoolA@81958@vcl6006
finished.
```

Each TcServer instance also records in the syslog changes to the idle timeouts and when the TcServer does timeout:

```
EDIT IDLE timeout limit: 04:48:00 (17280s)
READ IDLE timeout limit: 04:48:00 (17280s)
STATELESS IDLE timeout limit: 12:00 (720s)
...
Exceeded the STATELESS IDLE timeout limit: 720 seconds
```

Administer embedded LDAP for Teamcenter management console

Introduction to administering embedded LDAP for the Teamcenter management console

Lightweight directory access protocol (LDAP) is a protocol that facilitates user authentication and authorization. You must use LDAP to provide authorization for user access to the Teamcenter management console.

When you **install the Teamcenter management console**, the **Use Embedded LDAP** option is selected by default on the **Management Console LDAP Configurations** panel in Teamcenter Environment Manager (TEM). You can use this embedded LDAP to manage user authentication instead of having to use your own external LDAP system.

To administer the embedded LDAP, you can use Apache Directory Studio provided in the Teamcenter installation source at **additional_applications\ApacheDirectoryStudio**. Apache Directory Studio is an

optional tool. After you **install and configure Apache Directory Studio**, you can define users and passwords used by the embedded LDAP to permit access to the Teamcenter management console.

The embedded LDAP only has one user account created by default (**admin**) to access the Teamcenter management console. The Teamcenter administrator can add additional users to access the console (for example, **user1**, **user2**, and so on). In addition, you can also set password policies such as the minimum password length, password reset at first-time logon, password aging, the requirement of special characters on passwords, blacklist passwords, and so on.

Install and configure Apache Directory Studio for use with embedded LDAP

Use Apache Directory Studio to administer users and passwords for the **Teamcenter management console** embedded lightweight directory access protocol (LDAP).

1. In the Teamcenter installation source, browse to the *additional_applications\ApacheDirectoryStudio* directory.

- Windows systems

Run the *ApacheDirectoryStudio-platform-version.exe* file.

- Linux systems

Extract the *ApacheDirectoryStudio-platform-version.tar.gz* file and run the resulting installable file.

2. Follow the prompts to install Apache Directory Studio.

Apache Directory Studio stores its information in the following directories:

- Windows systems

user-home-directory\ApacheDirectoryStudio\metadata

- Linux systems

\$HOME/.ApacheDirectoryStudio/metadata/

3. Run Apache Directory Studio.

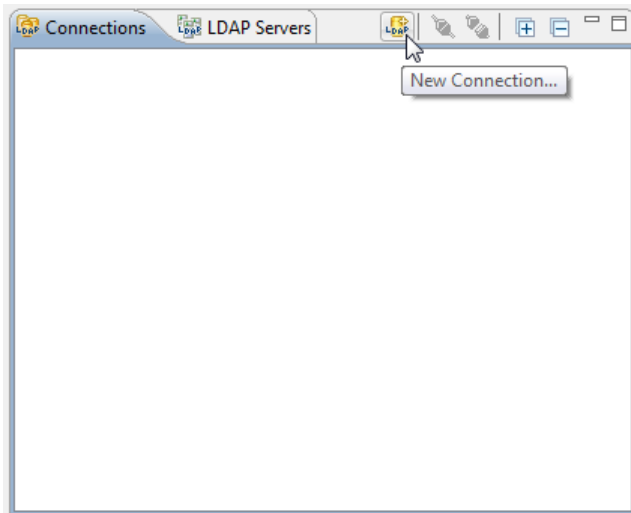
- Windows systems

Run the *installed-location\Apache Directory Studio\Apache Directory Studio.exe* file.

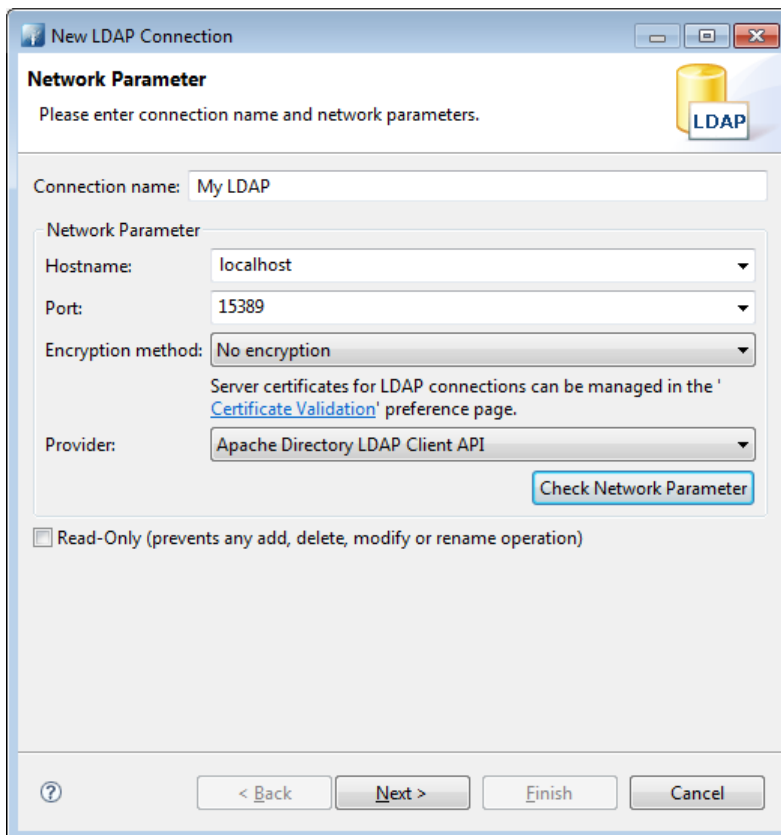
- Linux systems

Run the *installed-location/ApacheDirectoryStudio-platform-version/ApacheDirectoryStudio* file.

4. In the Apache Directory Studio user interface, open the **Connections** tab and click the **New Connection** button.



5. In the **Network Parameter** dialog box, create a connection to the embedded LDAP.



- a. In the **Connection Name** box, type a name to identify this connection.
 - b. In the **Hostname** box, type the name of the host where the embedded LDAP is installed. This is the same host where the Teamcenter management console is installed.
 - c. In the **Port** box, type **15389**. This is the default port for the embedded LDAP.
 - d. In the **Encryption method** box, select the required encryption of the target LDAP. The default for embedded LDAP is **No encryption**. (Do not select **Use StartTLS extension** because StartTLS is not supported in the embedded LDAP.)
 - e. In the **Provider** box, select **Apaxe Directory LDAP Client API**.
 - f. Click **Check Network Parameter** to ensure that the connection works.
 - g. Click **Next**.
6. In the **Authentication** dialog box, configure security for the connection.

New LDAP Connection

Authentication
Please select an authentication method and input authentication data.

Authentication Method: Simple Authentication

Authentication Parameter:
Bind DN or user: uid=admin,ou=system

Bind password:

☐ Save password

► SASL Settings
► Kerberos Settings

? < Back Next > Finish Cancel

- a. In the **Authentication Method** box, select **Simple Authentication**.
- b. In the **Bind DN or user** box, select **uid=admin,ou=system**.

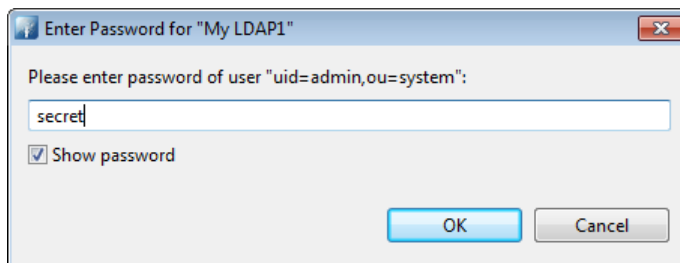
- c. Clear the **Save password** ☐ check box. If it is not selected, a password prompt appears every time you open the connection.

Tip:

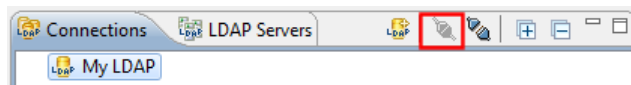
It is good policy to force logon with a password each time a connection is accessed. This prevents unauthorized access to the system.

- d. Click **Finish**.

You are prompted to enter the default password. Enter the default password **secret** and click **OK**.



The connection appears in the **Connections** view of Apache Directory Studio. The **Open Connection**  button is gray, indicating the connection is open.



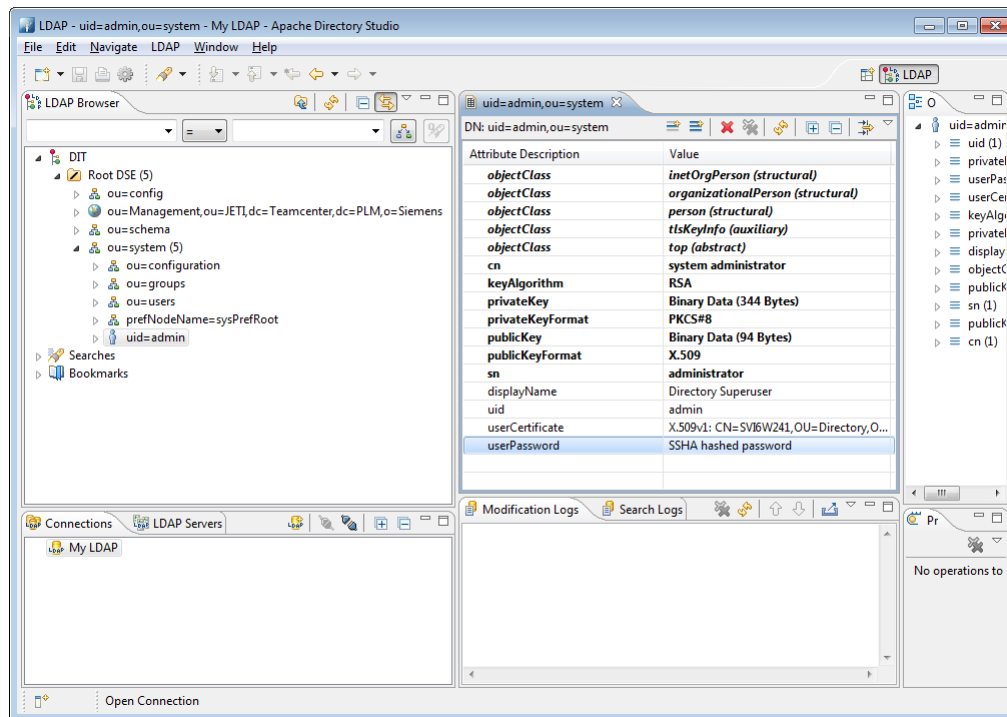
Tip:

If you want to change the values of the connection, right-click the connection and choose **Properties**.

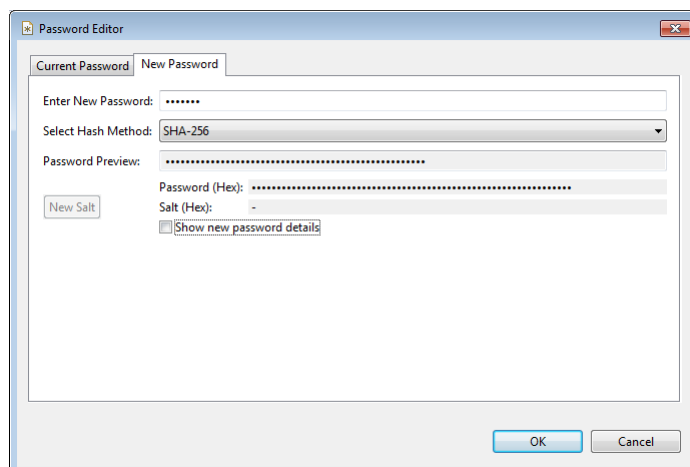
- e. After you create the connection, you must change the password so that the default password is no longer used. Choose the following path in the **LDAP Browser** view:

ou=system →
uid=admin

Double-click the **userPassword** attribute.



- f. Type a new password in the **Enter New Password** box. Click the arrow in the **Select Hash Method** box to select an encryption method. Siemens Digital Industries Software recommends at a minimum that you use the **SHA-256** method. When done, click **OK**.



- g. After creating an LDAP connection, you can use the **LDAP Browser** view to **set password policies for the embedded LDAP**.

Set password policies for embedded LDAP

If you use embedded lightweight directory access protocol (LDAP) to provide authorization for user access to the **Teamcenter management console**, use Apache Directory Studio to administer passwords for the embedded LDAP.

1. Ensure that JETI Management is running. JETI Management must be running to work with embedded LDAP.

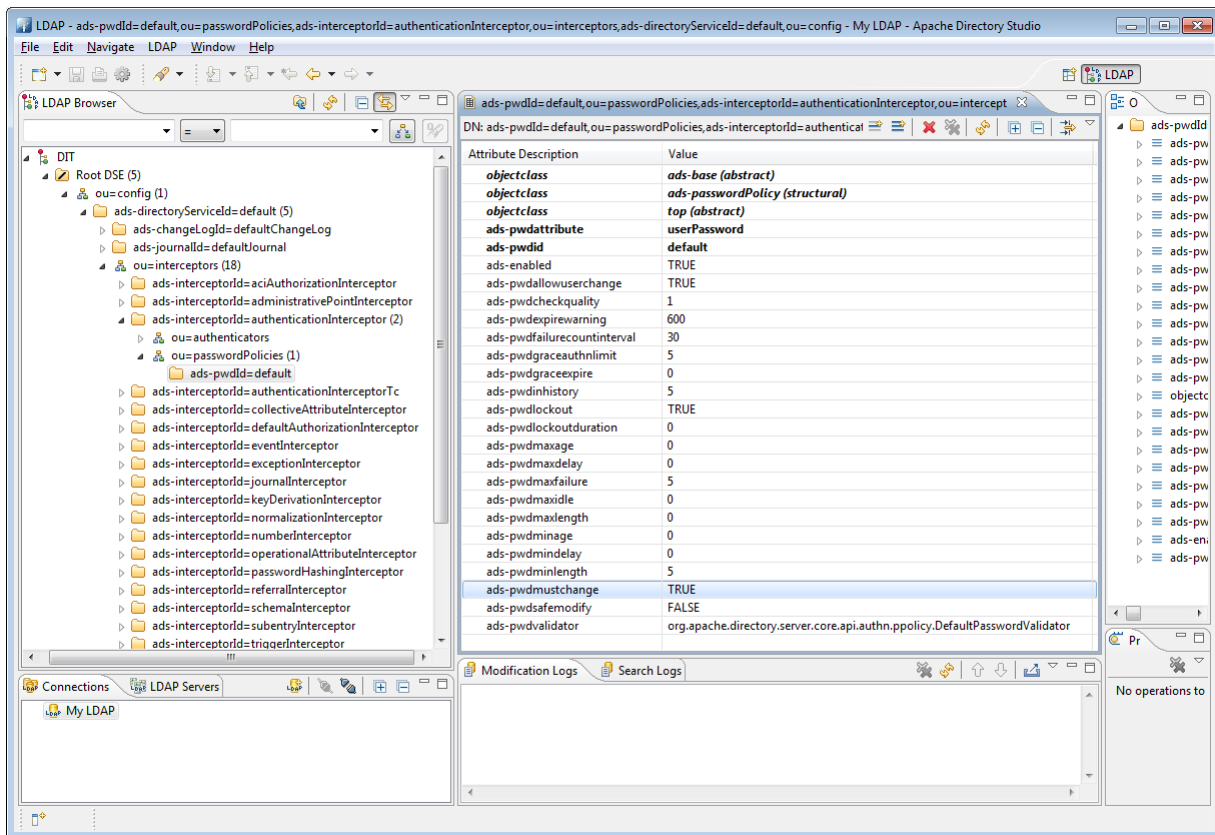
Run the following command to **start JETI Management**:

```
TC_ROOT\mgmt_console\container\bin\trun
```

2. **Install and configure Apache Directory Studio.**
3. Run Apache Directory Studio. For example, run the *installed-location\Apache Directory Studio\Apache Directory Studio.exe* file.
4. To view the default password policy settings provided by Apache Directory Studio, choose the following path in the **LDAP Browser** view:

```
ou=config→
ads-directoryServiceId=default→
ou=interceptors→
ads-interceptorId=authenticationInterceptor→
ou=passwordPolicies→
ads-pwdId=default
```

The default Apache Directory Studio (**ads**) password policy attributes are displayed. For a description of all Apache Directory Studio password policy entries, see Apache Directory help.

**Tip:**

Because the **ads-pwldmustchange** attribute is set to **TRUE**, LDAP forces users to change their password upon the first login after a new Teamcenter user is created or after the password is modified for a Teamcenter user by an LDAP administrator. Following the initial password change, users can subsequently **change their Teamcenter Management Console user password**.

5. The default embedded LDAP is extended with additional Teamcenter password policy features in the **authenticationInterceptorTc** interceptor.

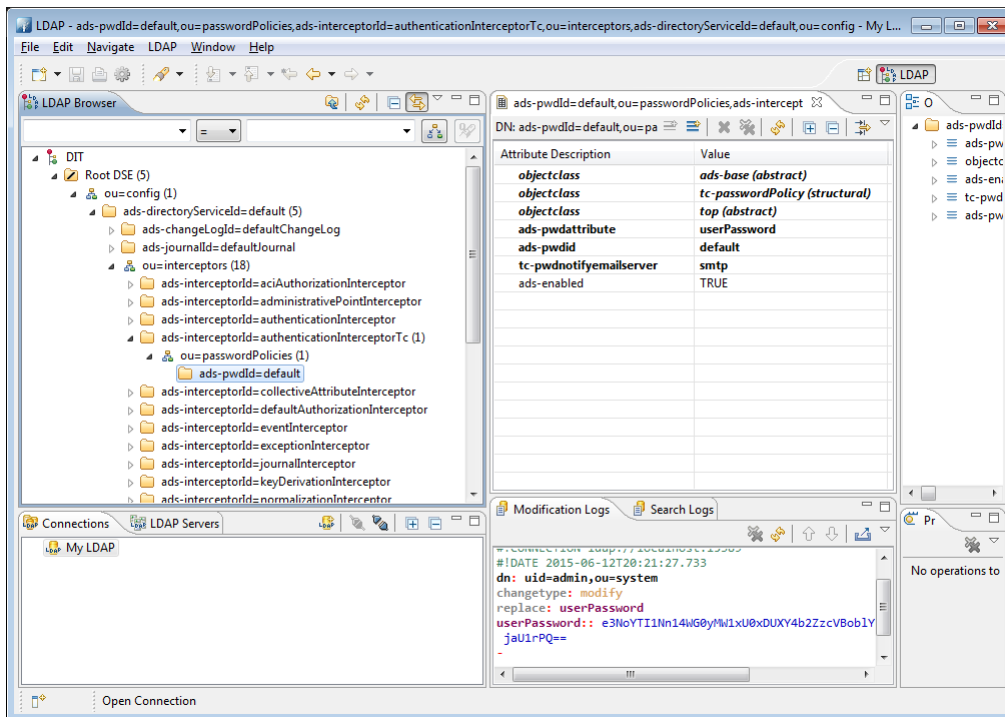
To view these Teamcenter password policy settings, choose the following path in the **LDAP Browser** view:

```

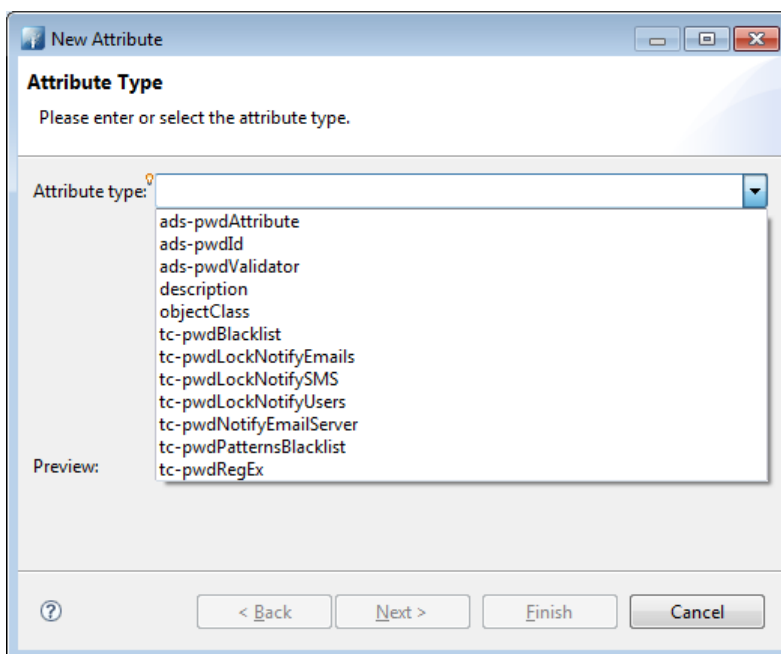
ou=config →
ads-directoryServiceId=default →
ou=interceptors →
ads-interceptorId=authenticationInterceptorTc →
ou=passwordPolicies →
ads-pwldid=default

```

The Teamcenter password policy settings are displayed.



- By default, **tc-pwdnotifyemailserver** is the only Teamcenter-specific password policy attribute shown. To add more Teamcenter policy entries, click the **New Attribute** button on the view toolbar.
- In the **New Attribute** dialog box, click the arrow in the **Attribute type** box to see a list of available attributes.



Following are some of the available Teamcenter attributes:

tc-pwdBlacklist (Optional)	Specifies a list of blocklisted passwords.
tc-pwdLockNotifyEmails (Optional)	Specifies a list of email addresses to send notifications to regarding Teamcenter users locked out of their accounts.
tc-pwdNotifyEmailServer (Required)	Specifies the name of an email server to be used when sending password policy-related email notifications.
pc-pwdPatternsBlacklist (Optional)	Specifies a list of entries that are not allowed to be part of a password.
tc-pwdRegEx (Optional)	Specifies a regular expression identifying a required password pattern.

Any changes to password policies requires you to **restart the Teamcenter management console** before the policy can take effect.

- After changing password policies, you can use the **LDAP Browser** view to **add Teamcenter users to embedded LDAP**.

Add Teamcenter users to embedded LDAP

If you use embedded lightweight directory access protocol (LDAP) to provide authorization for user access to the **Teamcenter management console**, use Apache Directory Studio to add users to the embedded LDAP.

- Ensure that JETI Management is running. JETI Management must be running to work with embedded LDAP.

Run the following command to **start JETI Management**:

```
TC_ROOT\mgmt_console\container\bin\trun
```

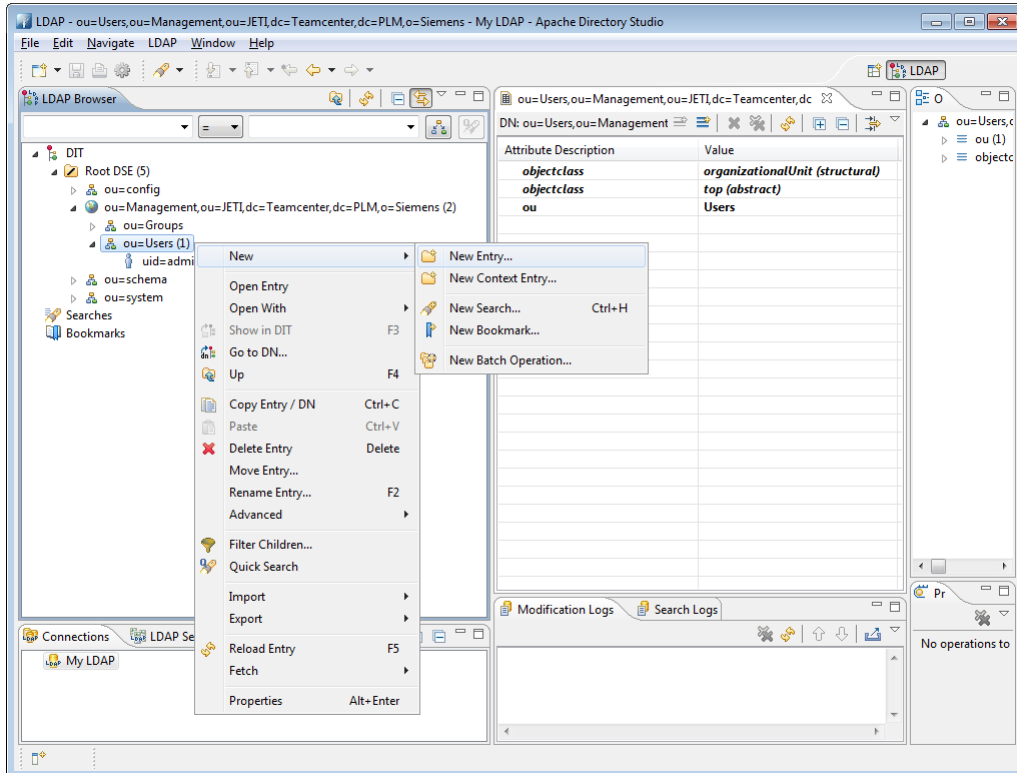
- Install and configure Apache Directory Studio.**
- Run Apache Directory Studio. For example, run the *installed-location\Apache Directory Studio\Apache Directory Studio.exe* file.
- To view the available Teamcenter users, choose the following path in the **LDAP Browser** view:

```
ou=Management,ou=JETI,dc=Teamcenter,dc=PLM,o=Siemens→  
ou=Users
```

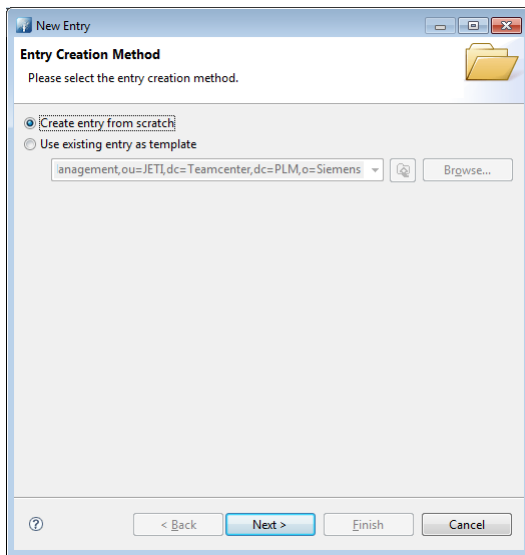
Tip:

If a user is locked out of the system, you can unlock them by deleting the **pwdAccountLockedTime** attribute on the user. To unlock the user, right-click the user, choose **Fetch**→**Fetch Operational Attributes**, right-click the **pwdAccountLockedTime** attribute, and choose **Delete Value**.

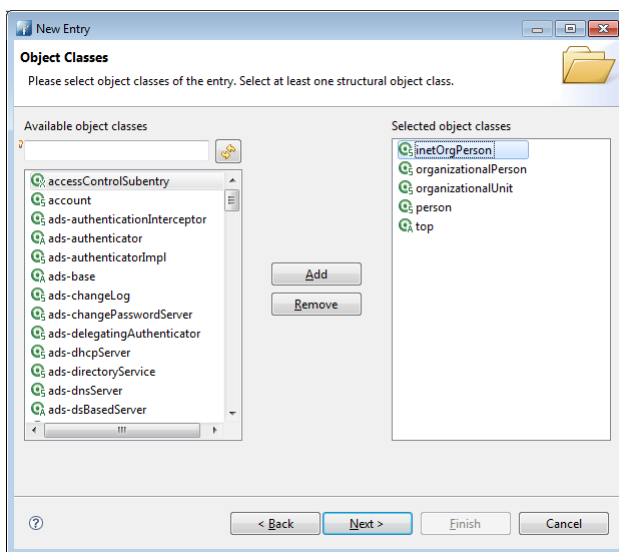
- To add a new user, right-click the **ou=Users** entry and choose **New**→**New Entry**.



- Ensure that **Create entry from scratch** is selected and click **Next**.



7. In the **Available object classes** box, type **inetOrgPerson** and click **Add** to move it to the **Selected object classes** pane. Click **Next**.



8. In the **RDN** box, type **uid**. Type the user's name in the **=** box and click **Next**.

The user name must be in a typical user name format and should not have spaces. For example, user Bob Smith should have a user name like **bsmith**.

New Entry

Distinguished Name
Please select the parent of the new entry and enter the RDN.

Parent:

RDN: =

DN Preview: uid=bsmith,ou=Management,ou=JETI,dc=Teamcenter,dc=PLM,o=Siemens

9. Type the user name in the **cn** and **sn** value boxes.

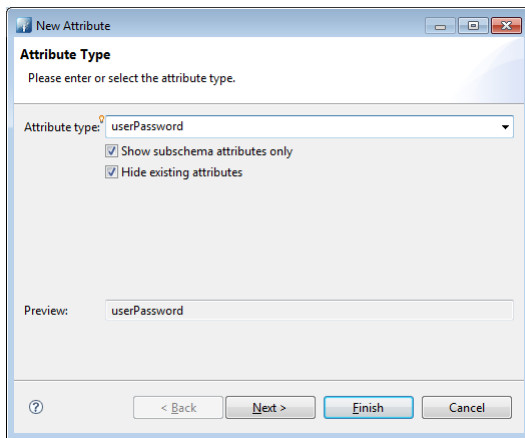
New Entry

Attributes
Please enter the attributes for the entry. Enter at least the MUST attributes.

DN: uid=bsmith,ou=Users,ou=Management,ou=JETI,dc=Team

Attribute Description	Value
objectClass	inetOrgPerson (structural)
objectClass	organizationalPerson (structural)
objectClass	person (structural)
objectClass	top (abstract)
cn	bsmith
sn	bsmith
uid	bsmith

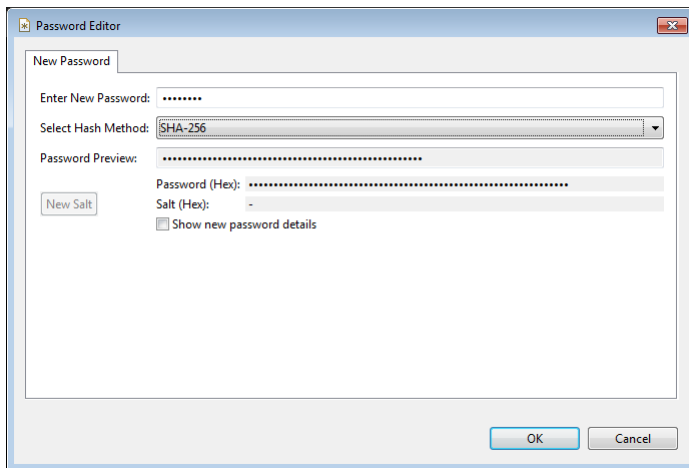
10. Click the **New Attribute**  button on the view toolbar.
11. In the **Attribute type** box, type **userPassword** and click **Next**.



12. Click **Finish**.

The **New Password** dialog box is displayed.

Type the user's password in the **Enter New Password** box. In the **Select Hash Method** box, select the method you want to use to encrypt the password. Siemens Digital Industries Software recommends at a minimum that you use the **SHA-256** method.



13. Click **OK**.

The **userPassword** attribute is displayed.

New Entry

Please enter the attributes for the entry. Enter at least the MUST attributes.

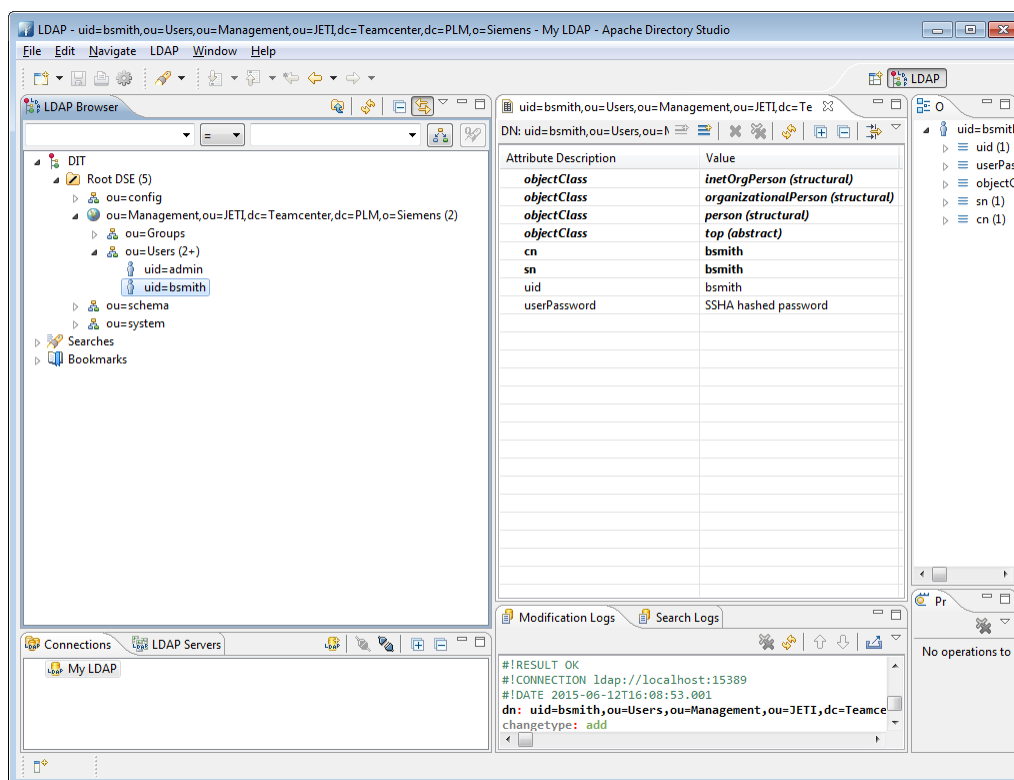
DN: uid=bsmith,ou=Users,ou=Management,ou=JETI,dc=Teamcenter

Attribute Description	Value
objectClass	inetOrgPerson (structural)
objectClass	organizationalPerson (structural)
objectClass	person (structural)
objectClass	top (abstract)
cn	bsmith
sn	bsmith
uid	bsmith
userPassword	SHA-256 hashed password

< Back Next > Finish Cancel

14. Click **Finish**.

The new user is displayed in the **LDAP Browser** view.



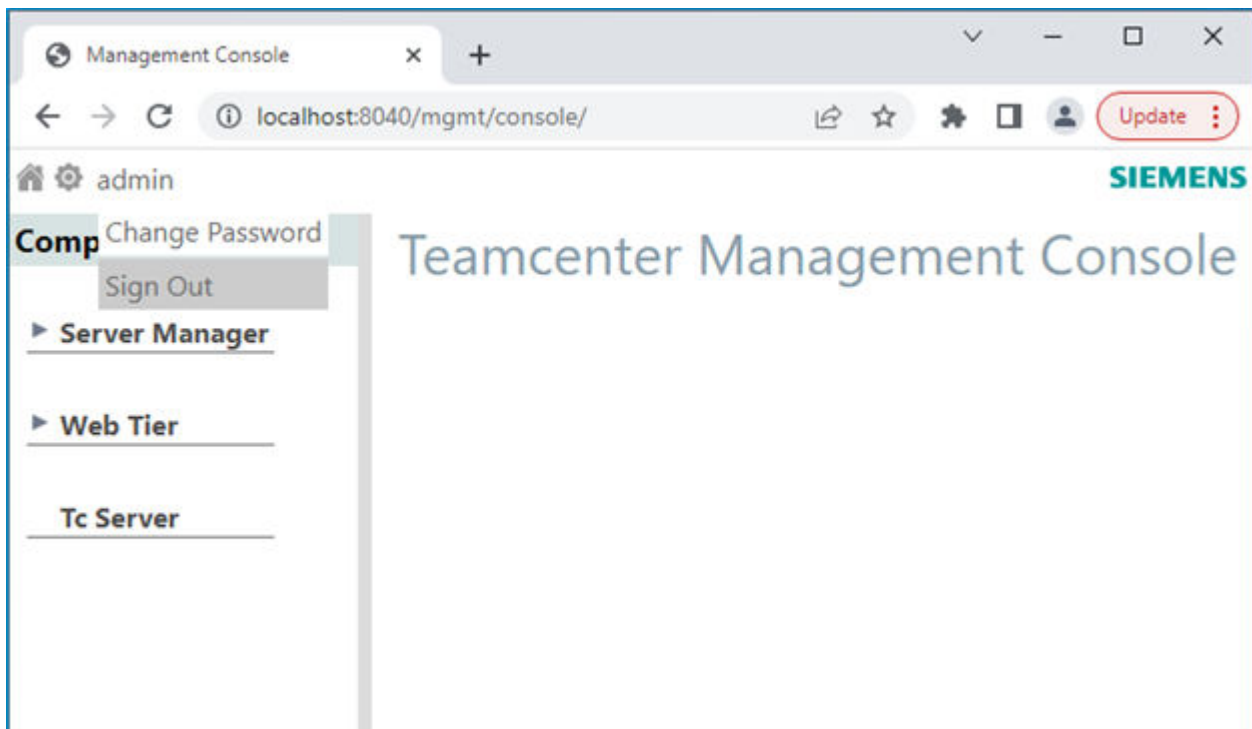
Change your Teamcenter Management Console user password

Teamcenter Management Console users can change their console password from within the Teamcenter Management Console.

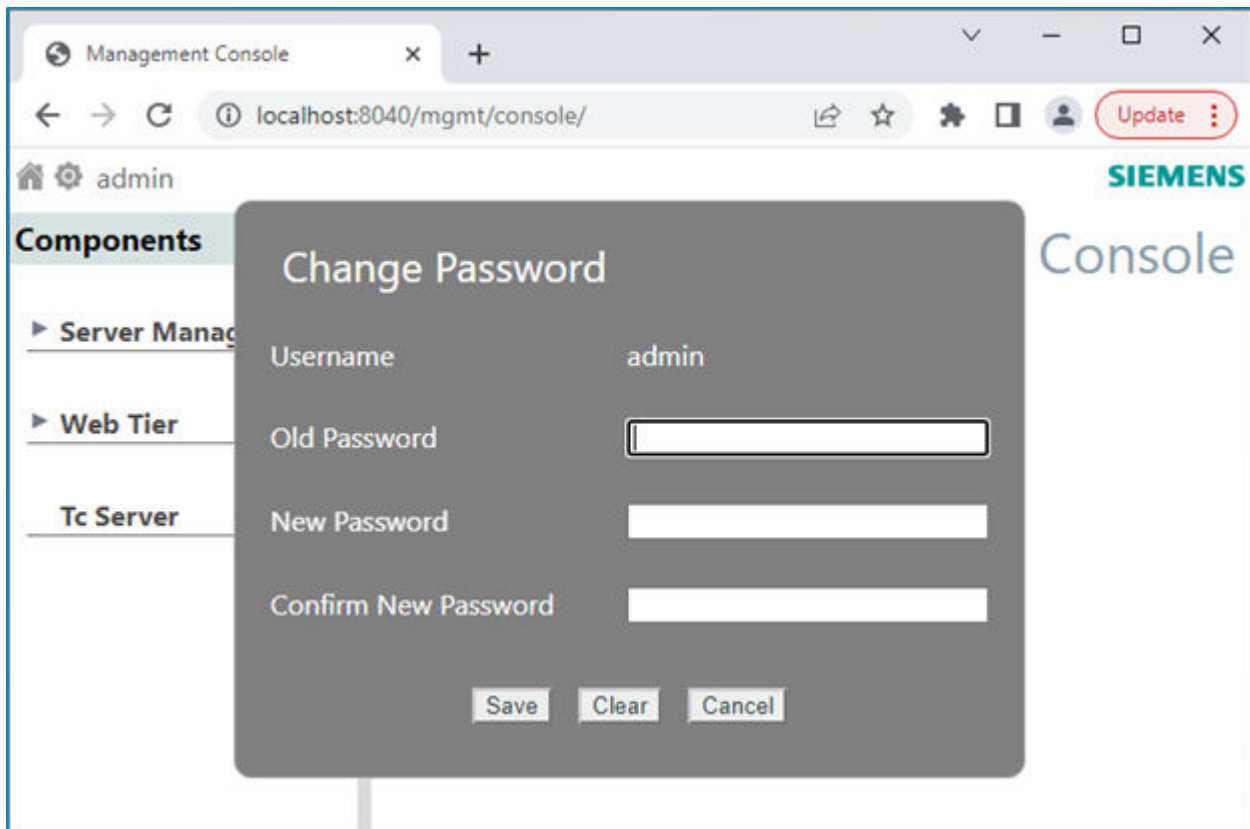
Note:

If the Apache Directory Studio **ads-pwdmustchange** attribute is set to **TRUE**, and authorization for user access to the **Teamcenter management console** is provided by embedded lightweight directory access protocol (LDAP), LDAP forces Teamcenter Management Console users to change their password upon the first logon after a new Teamcenter user is created or after the password is modified for a Teamcenter user by an LDAP administrator.

1. **Launch the Teamcenter management console** using the account whose password is to be changed.
2. On the management console home page, at the top left corner click **admin** and select **Change Password**.



3. Enter the old and new password and click **Save**.



The new password is applicable the next time the user launches the console.

Server manager prerequisites

Before using the server manager, you must:

1. Install the server manager.

In Teamcenter Environment Manager (TEM), you can install the server manager from the **Features** panel under the **Server Enhancements** category.

Note:

To balance the sessions load among multiple server managers, each server manager must have the same **Cluster ID** (that is, use the same server manager database). The Cluster ID value is stored in the **MANAGER_CLUSTER_ID** property in the **TC_ROOT\pool_manager\serverPool\database-name.properties** file.

2. **Configure pool-specific properties.**

Once server pools are configured, pool-specific configuration of each machine included in the pool should be tuned. This is of particular concern if machines differ greatly in the number and power

of CPUs. The pool-specific parameters are set in the `TC_ROOT\pool_manager\serverPool\database-name.properties` file.

3. (Optional) **Configure web tier administration.**

By default, administering of web tier metrics and logging behavior is disabled. Enable administration of web tier metrics and logging by resetting the **mode** attribute in the **pref_export.xml** file.

4. Ensure that the server manager service or daemon is running.

The server manager service or daemon is necessary to enable four-tier rich clients to connect to the corporate server.

Example:

On a Windows system, choose **Start>Control Panel>Administrative Tools>Services** and select **Teamcenter Server Manager**. You can also manually run the server manager service by running the following executable:

```
TC_ROOT\pool_manager\confs\configuration-namemgrstart.bat
```

Tip:

To verify that the server manager is running, launch the server manager user interface from a web browser. For example, to **start the interface**, use the **http://host-name:8083/mgmt/console** address.

Server manager properties files

Overview of server manager property files

The following property files configure server manager behavior.

- **serverPool.properties**

Performs pool-specific tuning of each individual machine. Pool-specific tuning is of particular importance if machines differ greatly in the number and power of CPUs.

This file is stored in the `TC_ROOT/pool_manager/confs/configuration-name` directory.

- **pref_export.xml**

Configures web tier metrics and logging behavior.

This file is stored in the `TC_ROOT/pool_manager/confs/configuration-name` directory.

Server manager pool-specific configuration tuning

Server manager pool-specific tuning parameters

Tune the server manager configuration to maximize server availability while minimizing resources, thereby making the best use of specific system resources.

Each individual machine should have its pool-specific configuration tuned. This is of particular concern if individual machines differ greatly in the number and power of CPUs.

The pool-specific parameters are set in the `TC_ROOT\pool_manager\confs\configuration-name\serverPool.properties` file.

The following parameters most likely require tuning:

PROCESS_TARGET

Provides a time profile of minimum numbers of Teamcenter servers to have running on the machine.

PROCESS_CREATION_DELAY

Determines the interval of time (in milliseconds) between starting each additional **TcServer** process to join the pool. (This parameter does not explicitly appear in the file by default but can be added manually.)

PROCESS_MAX

Sets the upper limit on number of running Teamcenter server processes. For highly scaled up deployments on Suse Linux, please review **Suse Linux UserTasksMax setting may need tuning**.

PROCESS_WARM

Sets the desired number of prestarted but unassigned Teamcenter servers.

Additional parameters provide the following information. Typically, the default settings are adequate and do not require additional tuning. Many of these parameters do not appear in the file by default, but can be added manually.

ACQUIRE_REENTRANT_LOCK_TIMEOUT

Determines the amount of time (in milliseconds) the manager waits before giving up when attempting to lock its internal data object regarding a server process. The default setting is **100**.

LOGINS_PER_MINUTE

Determines the number of logins per minute allowed to the pool. The default setting is **0** (unlimited).

MAX_POOL_PROCESSING_INTERVAL

Determines the maximum time (in milliseconds) any given manager processing thread sleeps before checking for additional work. The default setting is **600000** (10 minutes).

MIN_POOL_PROCESSING_INTERVAL

Determines the minimum amount of time (in milliseconds) any given manager processing thread sleeps before checking for additional work. The default setting is **1000**.

POOL_ID

Determines the ID of the server pool. The default setting is **PoolA**.

PROCESS_READY_TIMEOUT

Determines the time (in seconds) the manager waits for a Teamcenter server to report that it is ready before terminating the server process. The default setting is **300**.

SERVER_EXECUTABLE

Specifies the path to the **tcserver** executable file. The default setting is **TC_ROOT\bin\tcserver.exe**.

SERVER_HOST

Defines the host name (or IP address) on which the Teamcenter server listens for connections. Store and forward is useful on machines with multiple network interfaces. By default, this parameter is unset, causing the server to select an arbitrary network interface from the available network interfaces.

SERVER_PARAMETERS

Defines additional command line parameters supplied when starting a Teamcenter server process. The default setting is **-ORBNegotiateCodesets 0**.

SERVER_RETRY_LIMIT

Determines the number of times the manager retries sending a message (for example, a license heartbeat) to a Teamcenter server before giving up. The default setting is **2**.

SERVER_RETRY_WAIT_PERIOD

Determines the delay (in milliseconds) between retry attempts when sending a message to a Teamcenter server. The default setting is **1000**.

THREAD_POOL_INVOKING_SERVERS

Determines the maximum size of the pool of threads used for sending messages to Teamcenter servers and performing other miscellaneous tasks in the manager. The default setting is **500**.

Setting the PROCESS_CREATION_DELAY parameter

Tip:

Siemens Digital Industries Software recommends that you start your tuning process start by specifying an optimal setting for the **PROCESS_CREATION_DELAY** parameter. The **PROCESS_CREATION_DELAY** parameter is the most important pool server parameter for preventing CPU overload due to starting new server processes. It is critical that creating new Teamcenter server processes in response to logon demand does not overwhelm the processing resources available on the pool server machine.

When logon demand requires that the pool manager create a new server process and add it to the pool, the server pool **PROCESS_CREATION_DELAY** parameter value list determines the interval of time (in

milliseconds) the pool manager waits before attempting to create a new **TcServer** process. The default value for the parameter is the list **2000 2000 8000 16000 30000 60000**.

Example:

If the most recent attempt to create a Teamcenter server processes succeeded, then the pool manager waits the first interval in the list before attempting to create a new server process. If the attempt to create a new process fails, then the manager increments a failure count, waits the second interval in the list, and then again attempts to create a new server process. If a second attempt fails, then the pool manager increments the failure count and waits the third interval in the list (default value is 8000 milliseconds) before a third attempt.

When an attempt to start a server process succeeds, the failure count resets to zero, and for the next creation attempt, the pool manager waits the first value in the list.

Out of the box, the **PROCESS_CREATION_DELAY** parameter value is not explicitly present in the **serverPool.properties** file. To override the default, use the following procedure to find your own optimal first value and explicitly add the parameter to the file.

1. For your specific installation configuration and machine, measure the amount of operating system processing resources consumed by a Teamcenter server process started and added to the warm pool.

This can be done simply by looking at the processing resources consumed by a server in a newly started up (but unused) pool.

This amount can be expressed as *[warmCPUsec]* in units of CPU seconds.

2. Determine what fraction of the machine can be dedicated to creating a new server process in response to logon demand during the peak usage of the system.

Siemens Digital Industries Software recommends you choose a fraction between 0.4 and 0.7.

This fraction can be expressed as *[warmFraction]* (unitless). A lower value for this fraction reserves more of the machine processing resources for concurrent nonstartup activity.

3. Determine how many CPUs are on the machine (*[numCPUs]*).
4. Calculate the optimum process creation delay in units of milliseconds (mSec) as follows:

$$[optimumDelay] = ([warmCPUsec] * [1000 \text{ mSec/Sec}]) / ([warmFraction] * [numCPUs])$$

Example:

[warmCPUsec] = 8 seconds
[warmFraction] = 0.6

```
[num_CPUs] = 2
[optimumDelay] = (8 * 1000) / (0.6 * 2) = 6667
```

5. Add the **PROCESS_CREATION_DELAY** parameter with a value list of increasing delays to the **serverPool.properties** file, starting with the optimum (or slightly under) delay, and additional delays ramping up to about 60000 milliseconds.

Example:

```
PROCESS_CREATION_DELAY = 6667 7000 12000 24000 40000 60000
```

Note:

In our example of a low-powered machine where each server launch occupies a core for eight seconds, the machine is overloaded if the pool manager needs to start at least two Teamcenter server processes at the default rate of one every four seconds, even with no actual request processing. Therefore, setting the **PROCESS_CREATION_DELAY** parameter to greater than the default values list is critical to avoid overloading the machine.

Setting the PROCESS_MAX parameter

The **PROCESS_MAX** parameter in the **serverPool.properties** file sets the upper limit on number of running Teamcenter server processes. The only concern for limiting the maximum number of Teamcenter servers on a machine may be memory consumption. The incremental free memory per server can be expected to vary from installation to installation but is on the order of 30 megabytes.

Determine the maximum amount of RAM on the machine that you want to provide to the Teamcenter servers in the pool. This can be expressed as *[maxTCmem]* in gigabytes.

Set **PROCESS_MAX** = *[maxTCmem]* * [1024 MB/GB] / [30 MB].

For example:

```
[maxTCmem] = 4 Gig
PROCESS_MAX = 4 * 1024 / 30 = 136
```

Setting the PROCESS_TARGET parameter

The **PROCESS_TARGET** parameter in the **serverPool.properties** file provides a time profile of minimum numbers of Teamcenter servers to have running on the machine. **PROCESS_TARGET** is a series of comma-delimited local time and pool minimum pairs.

For example:

```
0700 20, 0730 50, 1100 20, 1300 50, 1900 10
```

This specifies that at 0700 local time, the pool manager should ensure that a minimum of 20 Teamcenter servers are in the pool, and at 0730 at least 50, and so on, until 1900 when the pool minimum drops to 10.

A second example is the single pair **0000 5**. This specifies that the pool manager maintain a minimum of 5 at all times.

It is desirable that this time profile be a reasonably good estimate of the number of servers needed by user demand. This can be estimated by occasionally monitoring the number of servers assigned to users throughout a representative workday. If this profile is well configured, the value of tuning the next parameter, **PROCESS_WARM**, becomes low.

When choosing the time to set a new minimum, realize that it takes time (based upon **PROCESS_CREATION_DELAY**) to ramp up from one level to another much higher level. For example, if it is determined that there should be a minimum of 200 at time 1300, and the previous minimum was 100, and **PROCESS_CREATION_DELAY** has been tuned to 5000 milliseconds, then it could take about $(200 - 100) * 5000 = 500,000$ milliseconds = 500 seconds = 8.3 minutes to ramp up the additional 100 processes. Thus the limit should be configured to 200 at around time 1250 to ensure 200 are all warm or in use at time 1300.

Setting the `PROCESS_WARM` parameter

The **PROCESS_WARM** parameter in the **serverPool.properties** file sets the desired number of prestarted but unassigned Teamcenter servers.

A *warm* Teamcenter server is one that has been started and established its database connection, and then is held in readiness for a user for logon. If there are no warm Teamcenter servers, a new logon attempt displays a message that a server is not available, and to try later.

It takes a certain amount of processing to start up a new Teamcenter server process to the point it can be added to the warm pool. A major part of this processing can be consumed simply by loading the shared libraries into the process. Different machines consume different amounts of processing resources to do this. Other factors can also come into play. On Solaris, for example, loading the shared libraries over NFS paths can consume more processing resources than loading them from local disks.

The **PROCESS_WARM** parameter is the most difficult to optimize, because it requires an estimate of the burst rate of logons that may occur.

If the **PROCESS_WARM** value is set too low, users in the rear of a burst of logon requests may encounter the `tcserver is not available, try again later` message, and a later logon will be successful.

To minimize this situation, the **PROCESS_WARM** value can be increased.

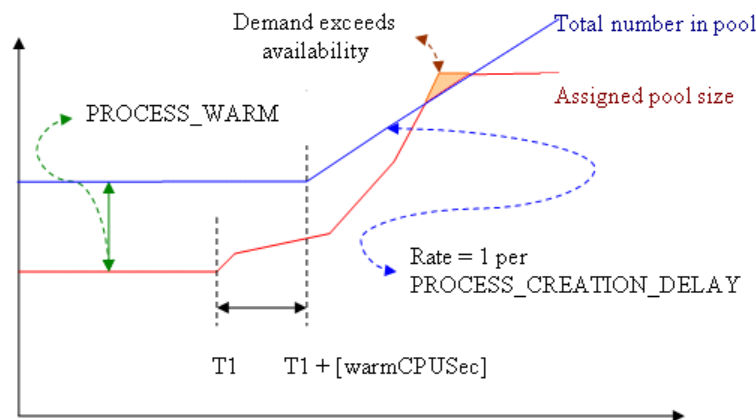
As defined at the start, **PROCESS_WARM** is a desired number of Teamcenter servers to be warm, but not assigned.

This configuration value comes into play if the sum of *assigned servers* + **PROCESS_WARM** is greater than the **PROCESS_TARGET** for that time of day. In this case, the pool manager responds to a logon (which moves a server state from warm to assigned) by starting one or more servers (not to exceed the rate dictated by **PROCESS_CREATION_DELAY**) to make up the **PROCESS_WARM** deficit.

A *burst of logon requests* begins in a state with the warm pool fully populated, extends until the warm pool is again fully populated, and contains one or more logon intervals faster than the rate dictated by the **PROCESS_CREATION_DELAY**.

A new Teamcenter server takes a minimum of $[warmCPUsec]$ seconds to become available, and then servers are added to the pool at a maximum rate of one per the setting in the **PROCESS_CREATION_DELAY** parameter.

The following graphic illustrates an assignment burst that starts at time T_1 , and continues until it exhausts the number of available warm servers sometime later during the time new servers are being added to the pool.



The number of Teamcenter servers added as warm per second after the delay $[warmCPUsec]$ is $(1000 \text{ mSec/Sec}) / \text{PROCESS\_CREATION\_DELAY}$. Keep in mind the following:

- If the burst interval is less than $[warmCPUsec]$, a burst of logons greater than **PROCESS_WARM** encounters an empty warm pool.
- If the burst interval is greater than $[warmCPUsec]$, a burst of logons greater than **PROCESS_WARM** + $(1000 * ([burst\_interval] - [warmCPUsec])) / \text{PROCESS\_CREATION\_DELAY}$ encounters an empty warm pool.

Typically, you do not need to perform this level of detailed analysis on pool logon demands. Note that:

- Large **PROCESS_CREATION_DELAY** values should prompt you to configure larger **PROCESS_WARM** pools.

Note:

Large **PROCESS_CREATION_DELAY** values should be configured for low numbers of server CPUs or long values of *[warmCPUsec]*. If you chose a low fraction of the machine to be used to start up new servers to compute optimum **PROCESS_CREATION_DELAY**, you may want to configure a larger **PROCESS_WARM** value to compensate for the longer startup delays.

- Excessive occurrences of low or exhausted warm margins should prompt either an upward adjustment of the **PROCESS_WARM** value, the **PROCESS_TARGET** minimum for that time of day, or both.

Tuning considerations for the Active Workspace indexer

In some cases, settings in clients rely on the number of warmed servers. For example, in the Active Workspace, the **tc.maxConnections** parameter specifies the maximum number of connections between the Teamcenter server and the indexer to be open at a given time and controls the quickness of the indexing process through parallelization of steps. This value should be less than the number of warmed-up Teamcenter servers available in the pool. For example, if you have 100 warmed-up servers in the pool manager, you may want to set the **tc.maxConnections** value to **50** to provide an adequate number of maximum connections.

During a period of heavy indexing, such as the first time the indexer is run, first check that the Teamcenter server pools have sufficient servers to handle the indexing load in addition to other loads such as from interactive user sessions. For example, if the **tc.maxConnections** parameter is set to **50**, the indexer needs 50 available warm servers. This can be achieved by increasing the **PROCESS_TARGET** parameter of the server pools to prestart an appropriate number of servers.

If you use Active Workspace, you can change the value of the **tc.maxConnections** parameter 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 in the **Maximum Teamcenter Connections** box.

To set the maximum connections value dynamically during the indexing run, perform the following command from the *FTS_INDEXER_HOME* directory (*TC_ROOT\TcFTSIndexer\bin*):

```
runTcFTSIndexer -maxConnections=connections
```

Server manager database password

When the server manager is installed, the supplied server manager database password is obfuscated and added to the *serverPool.properties* file. In case the database password is changed, use the **mgradmin.bat** utility to obfuscate the new password and replace the one stored in the properties file.

1. Open a Windows command prompt as an administrator.
2. Change directory to the *TC_ROOT\pool_manager\confs\<confName>* directory.

3. Enter the following command:

```
mgradmin.bat -password
```

4. At the prompt, enter the new Server Manager Database password and then press **Enter**.

The utility obfuscates the password and updates the *serverPool.properties* file.

Server manager monitoring

Introduction to server manager monitoring

In a four-tier environment, you can monitor critical event data and optionally receive notification when critical events exceed specified thresholds. This information is useful for determining the cause of an issue, thereby indicating corrective action.

Critical events can be monitored for the following elements of the Teamcenter system:

Web tier	Monitor web tier server events such as abandoned servers, no servers available, and so forth.
Server pool	Monitor pool manager events such as login time, server crashes, and various types of server time-outs.
Teamcenter servers	Monitor server events on specific Teamcenter servers such as out of memory errors, long running queries, and memory use.
File Management System (FMS) components	Monitor Multi-Site events such as global memory pool events, route events, and ticket events.
Translation components	Monitor translation events such as dispatcher client events, dispatcher scheduler events, and dispatcher module events.

You should review all monitoring settings, ensuring the thresholds are set correctly for your site. If you do not know the optimum monitoring setting for any given critical event, set the value to **Collect**. Collect the data and review to determine normal activity levels. Then set threshold values slightly higher than normal activity levels.

You can configure monitoring behavior either temporarily in an interactive administrative interface, or persistently in XML configuration files. Both configuration methods require you to first define the **EmailResponder** element (specifying how and where email notification is set) and the **LoggerResponder** element (where critical event data is logged). You can then set values for the different types of critical events of which you want to be notified.

You can control the frequency of email notifications and event logging by specifying suppression periods.

Example:

You have configured the monitoring of business logic server crashes to send email notification when server crashes exceed 10 in 600 seconds.

At 10:00, you set suppression of this notification to 4 hours.

At 10:05, the notification threshold is met. Email notification is sent. The suppression clock starts to count down.

The notification threshold is reached 22 times between 10:20 and 13:50. Notification is suppressed each time.

At 14:00 the suppression clock reaches its 4-hour threshold. Another notification threshold is met for business logic server crashes at 14:20. Email notification is sent. Once again, the suppression clock starts.

Pool manager monitoring

Pool manager monitoring metrics

Configure pool manager monitoring using either:

- `TC_ROOT/pool_manager/confs/configuration-name/poolMonitorConfig.xml`
- **Teamcenter Management Console**

You can configure the following metrics to provide specified levels of monitoring for specified threshold levels. Optionally, you can receive email notification when specified metrics cross specified thresholds.

Metric	Description
Login Time	The response time for users logging on through the server manager to the server.
Edit Mode Timeouts	The number of assigned servers in edit mode that are timed out.
Read Mode Timeouts	The number of assigned servers in read mode that are timed out.
Stateless Mode Timeouts	The number of assigned servers in stateless mode that are timed out.
Query Timeouts	The number of assigned servers performing queries that are timed out.

Metric	Description
Business Logic Server Crashes	The number of each server crash in the server pool.
Cold Servers	The number of servers launched but not reporting back in a specified amount of time.
Pool Capacity Timeouts	The number of servers terminated by the server manager before its configured amount of time, due to insufficient capacity in the server pool.
Grave Events	Fatal or unexpected errors occurring in the server manager.
Log Level	The duration and log level that is applied to the target logger when triggered by an alert.
Metric Mode	The list of target metrics that are collected when triggered by automatic alerts.
Monitoring Notification	The list of registered client listeners to notify when a system alert occurs.

Tip:

- You should review all monitoring settings, ensuring the thresholds are set correctly for your site.
If you do not know the optimum monitoring setting for any given critical event, set the value to **COLLECT**. Collect the data and review to determine normal activity levels. Then set notification values slightly higher than normal activity levels.
- The contents of the email notifications are generated from the **poolMonitorConfigInfo.xml** file (located in the *TC_ROOT/pool_manager/confs/configuration-name* directory). This is a companion file to the **poolMonitorConfig.xml** file. For a complete list of possible causes and recommended actions for server manager monitoring, see this file.

Configure pool monitoring with the poolMonitorConfig.xml file

This procedure describes how to configure pool monitoring with a configuration file. You can also [configure pool monitoring with the server manager interface](#).

1. Open the *TC_ROOT/pool_manager/confs/configuration-name/poolMonitorConfig.xml* file.
2. At the top of the file, set **mode** to one of the following:

- **Normal**

Enables monitoring of all the metrics listed in the file.

- **Disable_Alerts**

Enables monitoring of all the metrics listed in the file, but disables all notifications of critical events, regardless of individual notification settings on any metric.

- **Off**

Disables monitoring of all the metrics listed in the file.

3. (Optional) To be notified when criteria reaches the specified threshold, specify from whom, to whom, and how frequently email notification of critical events are sent by setting the following **EmailResponder** values. For example:

```
<EmailResponder id="EmailResponder1">
  <protocol value="smtp"/>
  <hostAddress value="CHANGE_TO_VALID_SMTP_HOST"/>
  <fromAddress value="CHANGE_TO_VALID_ADDRESS" />
  <toAddress value="CHANGE_TO_VALID_ADDRESS" />
  <suppressionPeriod value="120"/>
  <emailFormat value="html"/>
</EmailResponder>
```

You can specify more than one **EmailResponder id** value.

All **EmailResponder id** values in all subsequent monitoring metrics in this file must match one of the **EmailResponder id** values set here.

- **EmailResponder id**

Specify an identification for this email responder. Multiple email responders can be configured, in which case, the identifiers must be unique.

- **protocol**

Specify the email protocol by which notifications are sent. The only supported protocol is SMTP.

- **hostAddress**

Specify the server host from which the email notifications are sent. In a large deployment (with multiple server managers, or the web tier running on different hosts) the host address identifies the location of the critical events.

- **fromAddress**

Specify the address from which the notification emails are sent.

- **toAddress**

Specify the address to which the notification emails are sent.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress email notification of critical events.

- **emailFormat**

Specify the format in which the email is delivered. Valid values are **html** and **text**.

4. (Optional) To be notified when criteria reaches the specified threshold, specify to whom, and to which file, critical events are logged by setting the following **LoggerResponder** values. For example:

```
<LoggerResponder id="LoggerResponder1">
  <logFileName value="ServerManagerMonitoring.log" />
  <suppressionPeriod value="0"/>
</LoggerResponder>
```

All **LoggerResponder** values in all subsequent monitoring metrics in this file must match the **LoggerResponder id** value set here.

- **LoggerResponder id**

Specify an identification for this logger responder. Multiple logger responders can be configured, in which case, the identifiers must be unique.

- **logFileName**

Specify the name of the file to which critical events are logged.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress logging of critical events to the log file.

5. Configure the criteria for a critical event for any of the metrics in the file by:
 - a. Specifying a particular **EmailResponder**, if desired.
 - b. Specifying a particular **LoggerResponder**, if desired.
 - c. Setting the metric's monitoring mode to one of the following:

- **Collect**

Collect metric data and display results in the **server manager administrative interfaces** for this metric.

This is the default setting.

- **Alert**

When critical events occur, collect metric data, send email notifications (set up with **EmailResponder** values), write messages to log files (set up with **LoggerResponder** values), and display results in the server manager administrative interfaces for this metric. Log file messages and email notifications are only issued when monitoring mode is set to **Alert**.

- **Off**

No metric data is collected.

- Setting the remaining values to specify criteria that must be met to initiate a critical event for the metric.

- Save the file.

Server manager monitoring is enabled for the metrics you configured.

Sample poolMonitorConfig.xml file

In the following example, two **EmailResponder** elements are configured. The majority of email notifications are sent to **admin1@company.com**, but email notifications regarding Teamcenter server crashes and grave events are sent to **admin2@company.com**. Information regarding pool capacity and query time-outs is only collected; no email notifications are sent.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
@<COPYRIGHT>@
=====
Copyright 2011.
Siemens Product Lifecycle Management Software Inc.
All Rights Reserved.
=====
@<COPYRIGHT>@
-->
<!-- Server Manager Health Monitoring Configuration -->
<ApplicationConfig mode="Normal" version="1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="healthMonitorV1.0.xsd">
  <RespondersConfig>
    <EmailResponder id="EmailResponder1">
      <protocol value="smtp"/>
    </EmailResponder>
  </RespondersConfig>
</ApplicationConfig>
```

```

    <hostAddress value="svc1smtp.company.com"/>
    <fromAddress value="tcsys@company.com" />
    <toAddress value="admin1@company.com" />
    <suppressionPeriod value="4200"/>
    <emailFormat value="html"/>
  </EmailResponder>
  <EmailResponder id="EmailResponder2">
    <protocol value="smtp"/>
    <hostAddress value="svc1smtp.company.com"/>
    <fromAddress value="tcsys@company.com" />
    <toAddress value="admin2@company.com" />
    <suppressionPeriod value="4200"/>
    <emailFormat value="html"/>
  </EmailResponder>
  <LoggerResponder id="LoggerResponder1">
    <logFileName value="ServerManagerMonitoring.log" />
    <suppressionPeriod value="0"/>
  </LoggerResponder>
</RespondersConfig>
<MetricsConfig>
  <Metric name="Login Time" id="LoginTime" maxEntries="100" mode="Alert"
metricType="double"
    description="Average time taken for user to login during recent time period">
    <ThresholdWithPeriod minEvents="3">
      <ThresholdValue name="AverageTime" value="60"
        description="Average time of the login request" />
      <ThresholdPeriod name="TimePeriodInSec" value="600"
        description="Periods for which login time will be monitored." />
    </ThresholdWithPeriod>
    <Responders>
      <ResponderRef id="EmailResponder1"/>
      <ResponderRef id="LoggerResponder1"/>
    </Responders>
  </Metric>
  <Metric name="Edit Mode Timeouts" id="EditModeTimeouts" maxEntries="100" mode="Alert"
metricType="integer"
    description="Edit mode timeouts may result in lost user data.">
    <ThresholdWithPeriod>
      <ThresholdValue name="NumberOfEditModeTimeOuts" value="10"
        description="If number of timeouts exceeds this limit, notify the
administrator." />
      <ThresholdPeriod name="TimePeriodInSec" value="600"
        description="Periods for which timeouts will be monitored." />
    </ThresholdWithPeriod>
    <Responders>
      <ResponderRef id="EmailResponder1"/>
      <ResponderRef id="LoggerResponder1"/>
    </Responders>
  </Metric>
  <Metric name="Read Mode Timeouts" id="ReadModeTimeouts" maxEntries="100" mode="Alert"
metricType="integer"
    description="Read mode timeouts may force Rich Client users to restart client
sessions.">
    <ThresholdWithPeriod>
      <ThresholdValue name="NumberOfReadModeTimeOuts" value="10"
        description="If number of timeouts exceeds this limit, notify the
administrator." />
      <ThresholdPeriod name="TimePeriodInSec" value="600"
        description="Periods for which timeouts will be monitored." />
    </ThresholdWithPeriod>

```

```

    <Responders>
      <ResponderRef id="EmailResponder1"/>
      <ResponderRef id="LoggerResponder1"/>
    </Responders>
  </Metric>
  <Metric name="Stateless Mode Timeouts" id="StatelessModeTimeouts" maxEntries="100"
    mode="Alert" metricType="integer"
    description="Stateless mode timeouts generally have low impact on users and are a
good way to control resource consumption in the server pool. However, an excessive rate
might lead to high CPU consumption due to continually starting new servers.">
    <ThresholdWithPeriod>
      <ThresholdValue name="NumberOfStatelessModeTimeOuts" value="10"
        description="If number of timeouts exceeds this limit, notify the
administrator." />
      <ThresholdPeriod name="TimePeriodInSec" value="600"
        description="Periods for which timeouts will be monitored." />
    </ThresholdWithPeriod>
    <Responders>
      <ResponderRef id="EmailResponder1"/>
      <ResponderRef id="LoggerResponder1"/>
    </Responders>
  </Metric>
  <Metric name="Query Timeouts" id="QueryTimeouts" maxEntries="100" mode="Collect"
    metricType="integer"
    description="A query time out indicates that a single server request has taken longer
than the configured timeout. ">
    <ThresholdWithPeriod>
      <ThresholdValue name="NumberOfQueryTimeOuts" value="10"
        description="If number of timeouts exceeds this limit, notify the
administrator." />
      <ThresholdPeriod name="TimePeriodInSec" value="600"
        description="Periods for which timeouts will be monitored." />
    </ThresholdWithPeriod>
    <Responders>
      <ResponderRef id="EmailResponder1"/>
      <ResponderRef id="LoggerResponder1"/>
    </Responders>
  </Metric>
  <Metric name="Business Logic Server Crashes" id="TcServerCrashes" maxEntries="100"
    mode="Alert" metricType="integer"
    description="Number of tcservers that have crashed for any reason.">
    <ThresholdWithPeriod>
      <ThresholdValue name="NumberOfCrashes" value="10"
        description="If number of crashes exceeds this limit, notify the
administrator." />
      <ThresholdPeriod name="TimePeriodInSec" value="600"
        description="Periods for which timeouts will be monitored." />
    </ThresholdWithPeriod>
    <Responders>
      <ResponderRef id="EmailResponder2"/>
      <ResponderRef id="LoggerResponder1"/>
    </Responders>
  </Metric>
  <Metric name="Cold Servers" id="ColdServers" maxEntries="100" mode="Alert"
    metricType="integer"
    description="A cold server is one that did not report back to the server manager
within the configured time period (Default: 5 mins) after being started.">

```

```

        <ThresholdWithPeriod>
            <ThresholdValue name="NumberOfColdServers" value="10"
                description="If number of cold servers exceeds this limit, notify
administrator." />
            <ThresholdPeriod name="TimePeriodInSec" value="600"
                description="Periods for which cold servers will be monitored." />
        </ThresholdWithPeriod>
        <Responders>
            <ResponderRef id="EmailResponder1"/>
            <ResponderRef id="LoggerResponder1"/>
        </Responders>
    </Metric>
    <Metric name="Pool Capacity Timeouts" id="PoolCapacityTimeouts" maxEntries="100"
        mode="Collect" metricType="integer"
        description="Servers are being terminated by the manager before their normal timeout
period
        due to insufficient capacity in the server pool.">
        <ThresholdWithPeriod>
            <ThresholdValue name="NumberOfPoolCapacityTimeouts" value="10"
                description="If number of pool capacity timeouts exceeds this limit, notify the
administrator." />
            <ThresholdPeriod name="TimePeriodInSec" value="600"
                description="Periods for which pool capacity timeouts will be monitored." />
        </ThresholdWithPeriod>
        <Responders>
            <ResponderRef id="EmailResponder1"/>
            <ResponderRef id="LoggerResponder1"/>
        </Responders>
    </Metric>
    <Metric name="Grave Events" id="GraveEvents" maxEntries="100" mode="Alert"
        description="Something has happened causing the server manager to malfunction.">
        <Responders>
            <ResponderRef id="EmailResponder2"/>
            <ResponderRef id="LoggerResponder1"/>
        </Responders>
    </Metric>
</MetricsConfig>
</ApplicationConfig>

```

Teamcenter server monitoring

Teamcenter server monitoring metrics

Configure Teamcenter server monitoring using either:

- `TC_DATA\serverMonitorConfig.xml`
- **Teamcenter Management Console**

You can configure the following metrics to provide specified levels of monitoring for specified threshold levels. Optionally, you can receive email notification when specified metrics cross specified thresholds.

Metric	Description	Comment
Deadlocks	Report when the Teamcenter server encounters a database deadlock error.	Deadlock errors should be rare, and are often transparent if handled on retry. To get more information on the Teamcenter operation involved, enable monitoring if deadlock errors are found in database server logs.
VeryLongRunningQueries	Report SQL, elapsed time and operation when either: - Individual query time is more than the "slow" criterion. - An SQL statement matches a pattern contained in the POM_SQL_PLAN environment variable.	Avoid enabling during upgrade, where many queries are expected to be slow. This metric could be useful when slow query is intermittent, and so finding the Teamcenter operation context offers further clues on origin. The default "slow" criterion value is 10 seconds. This value can be overridden using the TC_SLOW_SQL environment variable.
DBConnectionLosses	Report when the database connection is lost.	This metric is useful if a network issue is intermittent and unexpected (that is, the network issue is not due to firewall / idle-connection tidy-up or other expected event).
PomLocks	Report Teamcenter object locks at the end of an SOA operation.	Enable this metric only for testing sessions and debugging information.
POMRetries	Report Teamcenter operations where database level functions failed to get a lock and retried.	This event may happen frequently in busy environments and may be expected, so the metric is only appropriate for testing.

Tip:

- You should review all monitoring settings, ensuring the thresholds are set correctly for your site.

If you do not know the optimum monitoring setting for any given critical event, set the value to **Collect**. Collect the data and review to determine normal activity levels. Then set notification values slightly higher than normal activity levels.

- The contents of the email notifications are generated from the *TC_DATA/serverMonitorConfigInfo.xml* file. This is a companion file to the **serverMonitorConfig.xml** file. For a complete list of possible causes and recommended actions for Teamcenter server monitoring, see this file.

Configure server monitoring with the **serverMonitorConfig.xml** file

To configure server monitoring with a configuration file, edit the *TC_DATA/serverMonitorConfig.xml* file provided as a template for configuring server monitoring.

1. Open the **serverMonitorConfig.xml** file.
2. In the **ApplicationConfig** tag near the beginning of the file, set the **mode** attribute to one of the following:

- **Normal**

Enables monitoring of all the metrics listed in the file.

- **Disable_Alerts**

Enables monitoring of all the metrics listed in the file, but disables all notifications of critical events, regardless of individual notification settings on any metric.

- **Off**

Disables monitoring of all the metrics listed in the file.

3. (Optional) To be notified when criteria reaches the specified threshold, specify from whom, to whom, and how frequently email notification of critical events are sent by setting the following **EmailResponder** values. For example:

```
<EmailResponder id="EmailResponder1">
  <protocol value = "smtp"/>
  <hostAddress value ="CHANGE_TO_VALID_SMTP_HOST"/>
  <fromAddress value = "CHANGE_TO_VALID_ADDRESS" />
  <toAddress value = "CHANGE_TO_VALID_ADDRESS" />
  <suppressionPeriod value = "43200"/>
  <emailFormat value = "html" />
</EmailResponder>
```

You can specify more than one **EmailResponder id**.

All **EmailResponder id** values in all subsequent monitoring metrics in this file must match one of the **EmailResponder id** values set here.

- **EmailResponder id**

Specify an identification for this email responder. Multiple email responders can be configured, in which case, the identifiers must be unique.

- **protocol**

Specify the email protocol by which notifications are sent. The only supported protocol is SMTP.

- **hostAddress**

Specify the server host from which the email notifications are sent. In a large deployment (with multiple server managers, or the web tier running on different hosts) the host address identifies the location of the critical events.

- **fromAddress**

Specify the address from which the notification emails are sent.

- **toAddress**

Specify the address to which the notification emails are sent.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress email notification of critical events.

- **emailFormat**

Specify the format in which the email is delivered. Valid values are **html** and **text**.

4. (Optional) To be notified when criteria reaches the specified threshold, specify to whom, and to which file, critical events are logged by setting the following **LoggerResponder** values. For example:

```
<LoggerResponder id="LoggerResponder1">
  <logFileName value="TcServerMonitoring.log" />
  <suppressionPeriod value="1000"/>
</LoggerResponder>
```

All **LoggerResponder** values in all subsequent monitoring metrics in this file must match the **LoggerResponder id** value set here.

- **LoggerResponder id**

Specify an identification for this logger responder. Multiple logger responders can be configured, in which case the identifiers must be unique.

- **logFileName**

Specify the name of the file to which critical events are logged.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress logging of critical events to the log file.

5. Configure the criteria for a critical event for any of the metrics in the file by:

- Specifying a particular **EmailResponder**, if desired.
- Specifying a particular **LoggerResponder**, if desired.
- Setting the metric's monitoring mode to one of the following:

- **Collect**

Collect metric data and display results in the **server manager administrative interfaces** for this metric.

This is the default setting.

- **Alert**

When critical events occur, collect metric data, send email notifications (set up with **EmailResponder** values), write messages to log files (set up with **LoggerResponder** values), and display results in the server manager administrative interfaces for this metric. Log file messages and email notifications are only issued when monitoring mode is set to **Alert**.

- **Off**

No metric data is collected.

- Setting the remaining values to specify criteria that must be met to initiate a critical event for the metric.

6. Save the file.

Server monitoring is enabled for the metrics you configured.

Server monitoring system alerts

Client applications can register their implementation of standard JMX notification listeners with the JMX monitoring system. When a monitoring system alert occurs, registered listeners issue a notification that contains the following information:

- Source

The object name that generated the notification. The client application uses this source to communicate with the component that raised the alert to request additional information about the alert.

- Sequence number

An incremental integer used to order notifications.

- A string that indicates the monitoring component that raised the alert in a Java component format, for example"

```
com.teamcenter.mld.healthmonitoring.ServerManager.threshold
```

- Message

A summary of the alert from the originating metric, for example:

```
Teamcenter Alert: Business Logic Server Crashes exceeded threshold.
```

- User data

A string containing the **MetricID** value.

The client can request the following additional information about the alert:

- **AlertSummary** data

This is information specific to the alert raised.

- **AlertSubject** data

This is information specific to the alert raised.

- **AlertEventData** data

This is information specific to the alert raised.

- **MetricID** value

Defined in the **webtierMonitorConfig.xml** file. This file is located in the deployed .war file in the **lib\mldcfg.jar** file.

- **MetricName** value

Defined in the **webtierMonitorConfig.xml** file.

- **MetricDescription** data

Defined in the **webtierMonitorConfig.xml** file.

- **PossibleCauses** data

Defined in the **webtierMonitorConfigInfo.xml** file.

- **RecommendedActions** data

Defined in the **webtierMonitorConfigInfo.xml** file.

Server manager automatic metric collection

You can enable automatic collection of metrics based on the occurrence of an alert. When the alert occurs, all events for the metric are captured and stored in memory. The maximum number of records to keep in memory is configured by the **maxEntries** attribute in the **webtierMonitorConfigInfo.xml** file. After collection is initiated, it remains in effect until you manually change the monitor mode for the metric to any other supported mode.

The following is a sample configuration for automatic collection of **DBConnectionLosses** and **Deadlock** metrics when the **POMRetries** alert occurs:

```
<MetricModeController id=" MetricModeController1">
    <targetedMetrics>
        <MetricId value="DBConnectionLosses"/>
        <MetricId value="DeadLock"/>
    </targetedMetrics>
</MetricModeController>
    <Metric name=" POMRetries" id=" POMRetries" >
    <Responders>
        <ResponderRef id="MetricModeController1" />
    </Responders>
</Metric >
```

If the mode for a metric is already set to **Collect** or **Alert**, subsequent alerts are ignored.

Server manager automatic log level change

You can configure a logger to automatically change log level to a specific value when an alert occurs. If multiple instances of a responder have different target levels for a logger, the logger is set the highest value (larger number) using the following order:

1. **FATAL**
2. **ERROR**
3. **WARN**
4. **INFO**
5. **DEBUG**
6. **TRACE**

If you specify a log level for a logger that has been adjusted due to an alert on a metric, your value supersedes the responder setting and clears any log level changes in queue due to the alert. The following is a sample configuration for automatic log level change to **Debug** for the **LogLevelController1** responder with a duration of 1000 seconds:

```
<LogLevelController id="LogLevelController1">
  <targetedLevel value="Debug"/>
  <duration value ="1000"/>
  <targetedLoggers>
    <loggerName value="Teamcenter.pom"/>
    <loggerName value="Teamcenter.bom"/>
  </targetedLoggers>
</LogLevelController>
```

Server manager logging

Server manager logging levels

In a four-tier environment, you can dynamically change logging levels for the web tier, server manager, and Teamcenter servers.

Logging level	Description
FATAL	Logs only severe error events that cause the application to abort. This is the least verbose logging level.
ERROR	Logs error events that may allow the application to continue running.
WARN	Logs potentially harmful situations, such as incomplete configuration, use of deprecated APIs, poor use of APIs, and other run-time situations that are undesirable or unexpected but do not prevent correct execution.

Logging level	Description
INFO	Logs informational messages highlighting the progress of the application at a coarse-grained level.
DEBUG	Logs fine-grained informational events that are useful for debugging an application.
TRACE	Logs detailed information, tracing any significant step of execution. This is the most verbose logging level.

Configure server manager logging

Administrators can change startup logging levels for server manager, and can change session logging levels for a running server manager.

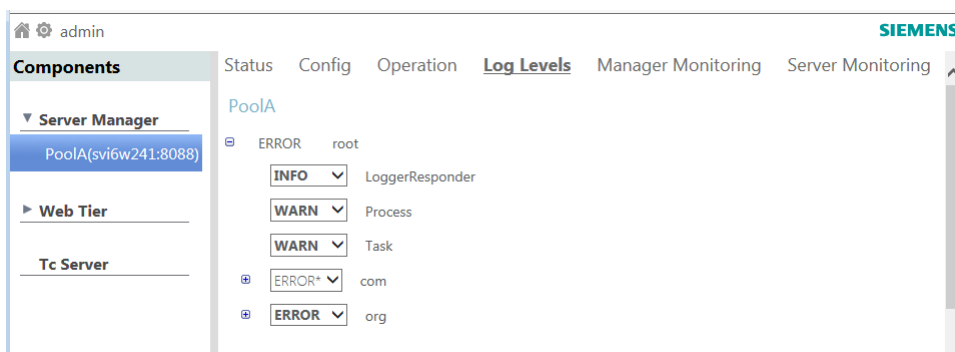
Duration	Method
Persistent	Modify the log4j2.xml file, stored in the <i>TC_ROOT/pool_manager/confs/configuration-name</i> directory. This method sets start-up logging levels for the server manager.
Current session only	Use the Teamcenter Management Console described below. This method dynamically changes logging levels for the server manager until the server manager is restarted. This method is useful for testing sandbox environments.

Perform the following steps to temporarily configure logging for a running server manager.

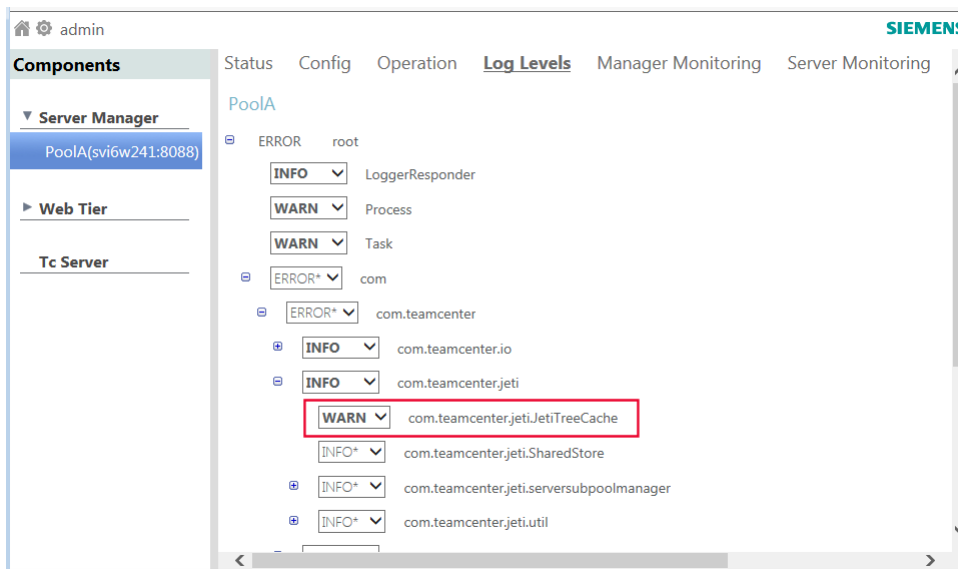
1. Start the Teamcenter Management Console.

By default, the address is **http://host:8083/mgmt/console**.

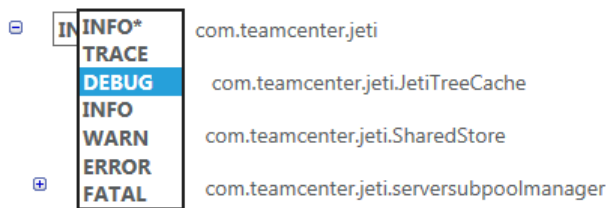
The list of loggers is displayed under the **Log Levels** heading.



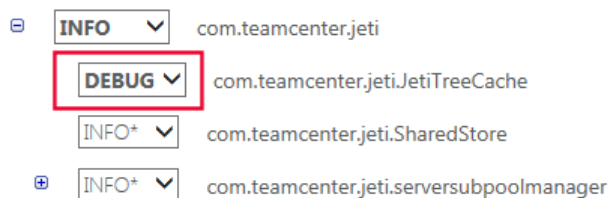
- Browse to the logger whose logging level you want to configure.



- Click the arrow in the box to select a new logging level, for example, **DEBUG**.



The logging level for the selected logger is changed for the server manager until the server manager is restarted.



Restarting warm servers

Click the **Restart Warm Servers** button to recycle warm servers in all server manager pools without shutting down the server manager. This is useful for updating cached values on a warm server.

The **Restart Warm Servers** button can be found in the **Teamcenter Management Console** under the **Operations** link.

Example:

In a four-tier configuration, Host A and Host B use a common Teamcenter database.

Changes made to preferences with a protection scope of **Site** on Host A affect all existing Teamcenter server processes running on Host A. Because Host B caches such preferences when it starts, the changes to these preferences are not received by Host B if it was running when the changes are made through Host A. The changes are not immediately received by Host B.

In a four-tier environment, the server manager can be configured with additional warm servers, ready for use by the next user to log on. Warm servers cache preferences when they are started.

If warm servers are configured, and Host B logs off and then logs on through a warm server, Host B still does not receive the preference changes because the warm server cached the preference settings when it started.

To ensure that all hosts receive the latest information at logon, use the **Restart Warm Servers** button. This operation stops and restarts all warm servers, ensuring each warm server receives the latest preference settings.

If you use this functionality to restart most (or all) the warm servers at your site, this task should be performed with very few users on the system. Otherwise, there is a risk that no warm servers are available during the short time it takes to recycle the servers.

Example:

The server manager is configured to support 100 users during 08:00 to 17:00. At 07:55, there are no users on the system; therefore, there are 100 warm servers available. Choosing to recycle all 100 servers at 07:55, in order to refresh cached server values, may not allow sufficient time for the servers to restart by 08:00. Any users attempting to log on before the servers have restarted receive a message that no server is available.

Server Manager Troubleshooting

Windows desktop heap may cause hang of TcServer processes

If you are planning to run large numbers of processes using Microsoft Services, you will likely run out of non-interactive desktop heap resources.

A limitation exists in the size of the Windows desktop heap that effectively limits the number of processes that the server manager can manage. When this particular heap allocation is consumed, processes may hang in several ways (including during execution of a child process). If the **TcServer** process hangs after spawning any new subprocess, the client appears hung to the user.

Everything may work fine on a system that is not heavily populated with large numbers of **TcServer** processes. The size of the desktop heap is independent of the amount of RAM installed.

The size of the desktop heap is by default smaller for processes started with services, so this is more likely to be encountered when Teamcenter server manager is started as a service.

The Microsoft NtDebugging Blog article *Desktop Heap Overview* provides more information regarding this resource.

On some operating systems, a default non-interactive heap setting of **768** for the **SharedSection** settings results in limitations of about 100 **TcServer** processes for a server pool started as a Windows service.

If you encounter this limitation when running the server manager pool as a service, you can work around it by starting the server manager from a user console. The desktop heap **SharedSection** registry values associated with an interactive window are by default many times larger than the **SharedSection** associated with a non-interactive window service. For example, on a Windows 7 machine, the value is:

```
SharedSection=1024, 20480, 768
```

If you have no other recourse, you may carefully modify the controlling registry values per the Microsoft Knowledge Base article 184802.

Caution:

Pay close attention to the risks of modifying these particular registry values spelled out in the article. If not done properly, you may have to reinstall the operating system (OS).

You should only edit the third value of the **SharedSection** value, which is the value associated with non-interactive desktop heap. If you run a 32-bit OS (only applicable for earlier versions of Teamcenter on 32-bit), the **boot.ini** setting of **/3GB** for a 32-bit OS should not be used with modifications to this registry setting, because it causes the first value of the **SharedSection** (for **SessionViewSize**) to be ignored. Setting **/3GB** after this registry has been edited may make the system unusable.

When editing this registry value, Siemens Digital Industries Software recommends that the third value for non-interactive desktop be increased in small increments such as **512** until the processes started with services run well. You should not have to increase it larger than the second value of the **SharedSection** that applies to interactive desktop heap.

Suse Linux UserTasksMax setting may need tuning

By means of the system setting **UserTasksMax**, Suse Linux limits the number of concurrent processes/threads per operating system user. In highly scaled up deployments, the server manager may encroach on the default value **UserTasksMax=12,288**.

For example, a single TcServer process launched by the server manager can use up to six concurrent threads, and thus an installation configuration where **PROCESS_MAX=2000** can consume up to 12,000 threads. In this case, if **UserTasksMax** is set to the default value 12,288, few threads are available for other processes started using the same user account that launched the server manager, resulting in abnormal behavior for processes that cannot acquire threads.

In addition to the operating system limitation per user, several Linux distributions such as Suse Linux Enterprise Server use the systemd service, which has parameters to limit the concurrent number of processes and tasks. If the server manager is running in a systemd service, the default systemd configuration limits the amount of tcserver processes that can be started. To resolve this issue, update the following systemd settings:

TasksMax The maximum number of tasks that can be created in the service.

DefaultTasksMax The default tasks maximum set by the operating system.

Server manager failover limitations

The failover abilities of server managers are limited. When a server manager becomes unavailable in a hard way such as a network interruption or a machine crash, the reference to the server manager with all of its tcservers remains in the cluster database. If the connected clients use a static session discriminator (such as rich client shared sessions do), a user attempting to reconnect or log in might not be able to do so.

To prevent this kind of failure, an administrator can do either of the following:

- Configure the rich client to use non-shared sessions.
- Remove the unavailable pool and its tcservers from the cluster database (for example by running **mgradmin.bat -clean <poolid>** from a healthy server manager).

Other SOA clients such as CAD clients, NX being started without rich client, rich clients with SharedSession off, or custom SOA clients with random SOA session discriminator are not affected by this limitation.

JMX connection error: MBean Server

In complex deployment configurations the management console may not automatically connect to the JMX server for a component such as a pool manager or a web tier.

An indicator of connection failure is an error message in this format:

```
Problem connecting to MBean Server using RMI protocol at [host]:[port]
```

To ensure connection, in the configuration file of the component that the management console is not able to reach, administrators can add the parameter **java.rmi.server.hostname**. The parameter should be set to the host name identified in the error message.

Example:

The management console installed on machine **server1.example.com** is not able to connect to pool manager **x** installed on **poolx.example.com**.

Try adding the following setting to the JMX_OPTS line of the pool manager configuration file *mgrenv (mgrenv.bat)* installed on **poolx.example.com**:

```
-Djava.rmi.server.hostname=poolx.example.com
```

For reference, before the hostname parameter is added, the JMX_OPTS line might look like this in the file:

```
set JMX_OPTS=-Dcom.sun.management.jmxremote.port=8088
-Dcom.sun.management.jmxremote.login.config=MgmtLdapConfig
-Djava.security.auth.login.config=ldap.config
-Dcom.sun.management.jmxremote.ssl=false
```

Parse errors warning in Oracle trace log

If the Teamcenter database is in Oracle, and any JMX client including the Teamcenter Management Console frequently queries the state of Teamcenter servers, then the Oracle trace log may show a parse errors warning related to Server Manager SQL. This is caused by the JDBC driver appending a row ID to the query.

For example:

```
2021-01-29T20:10:15.281248-05:00
WARNING: too many parse errors, count=426 SQL hash=0x60c18596
PARSE ERROR: ospid=8560, error=937 for statement:
2021-01-29T20:10:15.282248-05:00 select rowid as
"__Oracle_JDBC_internal_ROWID__", COUNT(*) as ELEMENT_COUNT FROM
SERVERS_1773957227_357170909 WHERE POOL_ID = 'PoolA'
Additional information: hd=00007FF8E079EEE8 phd=00007FF8ECA7D748
flg=0x20 cisid=94 sid=94 ciuid=94 uid=94 sqlid=c7yd2hjhc31cq
...Current username=TCCLUSTERDB
...Application: JDBC Thin Client Action:
```

```
2021-01-29T20:11:32.725724-05:00 WARNING: too many parse errors,
count=312 SQL hash=0x96b6d88d
PARSE ERROR: ospid=10028, error=937 for statement:
2021-01-29T20:11:32.725724-05:00 select rowid as
"__Oracle_JDBC_internal_ROWID__", COUNT(*) as ELEMENT_COUNT FROM
SERVERS_1773957227_357170909 WHERE POOL_ID = 'PoolA' AND STATE =
'warming'
Additional information: hd=00007FF8EA954610 phd=00007FF8EC900930
flg=0x20 cisid=94 sid=94 ciuid=94 uid=94 sqlid=0bbcb7abbdq4d
```

```
...Current username=TCCLUSTERDB  
...Application: JDBC Thin Client Action:
```

The parse errors do not affect performance.

To eliminate the parse errors warning, the workaround suggested by Oracle is to set the following parameter on the Oracle server: **`_kks_parse_error_warning=0`**

This setting is available for Oracle Database version 12.2 and later.

1. Login to SQLPlus using a system administrator account.
2. Run the following command:

```
alter system set "_kks_parse_error_warning"=0 scope=both;
```

Running the server manager on a machine separate from the corporate server

If you install the Teamcenter server manager (pool manager) on a machine other than the corporate server, the pool manager machine must have the same data model as the corporate server machine. To accomplish this, use Teamcenter Environment Manager (TEM) to select the same features on the pool manager machine. This ensures that the pool manager machine has an up-to-date *TC_DATA* directory and binary files. If the *TC_DATA* directory is shared, this same process is still required to keep the binary files synchronized.

Operations such as patching, upgrading, and installation of custom or Teamcenter data models (and updates to those data models) must be performed on the pool manager server machine after being done on the corporate server machine. This process ensures that the pool manager machine is synchronized with the corporate server machine.

30. System maintenance

Maintaining Teamcenter servers

Scheduled maintenance for Teamcenter servers

Run the following maintenance utilities on Teamcenter servers as scheduled for daily, weekly, or monthly frequency.

To maximize system up-time, set the weekly maintenance window for Teamcenter servers and applications to a particular day and time each week. Changes that require taking the system off-line, such as service account password reset, certificate renewal, and database re-indexing, should be done during this window. Any change outside the maintenance window that requires taking the system off-line should require business and production manager approval.

Run Daily

clearlocks

Clears dead process locks from the database. Dead process locks typically occur when a Teamcenter session terminates abnormally. Process locks are set on an object when it is being modified or deleted. If a Teamcenter session does not terminate gracefully (by logging off), these locks can remain in place.

Run Weekly

dataset_cleanup

Repairs corrupted datasets and removes orphaned revision anchors. Run during off-hours. A dataset is identified as corrupt if any of the following are found:

- The dataset has no reference to an **imanFile** object.
- The dataset has a reference to an **imanFile** corresponding operating system file that does not exist and the dataset is not archived.
- The dataset is an orphan (that is, the dataset refers to the anchor, but the anchor does not go to the dataset).
- The anchor refers to datasets that do not exist.
- The anchor size is **0**.

index_verifier

Detects missing indexes and describes how to create the indexes.

purge_datasets

Removes old dataset versions from Teamcenter database. Normally, Teamcenter keeps a fixed limit of dataset versions in the database that is controlled by the **AE_dataset_default_keep_limit** preference. The **purge_datasets** utility can be run during normal working hours, with users logged on to the system, and can process approximately 500 anchors per hour. A listing is produced that shows each dataset purged, along with the owning user and group.

purge_invalid_subscriptions

Reports or deletes invalid and expired subscription objects.

purge_volumes

Unlocks files left behind in the Teamcenter volume after associated Teamcenter objects are deleted, so that the files can be deleted from the volume.

Files are left behind if, during a Teamcenter session, a user deletes an object, but does not have the necessary privileges at the operating system level in the Teamcenter volume to delete the associated file.

report_volume

Reports on the available volumes as part of general housekeeping. Use this utility to create a checklist of volumes for the users. The path does not include the network node name.

review_volumes

Generates a report file describing volume usage by various groups and users, as well as any unreferenced operating system files, missing operating system files, and unreferenced Teamcenter files. Unreferenced operating system files can be deleted at the time a report file is generated or at a later time using a previously generated report file as an input. The report file format is plain text (ASCII) and can be manually edited to not delete certain files. Simply remove any file names that you do not want to delete before using the report file as input. You can also save any deleted files to a ZIP format compressed file.

verify_tasks

Finds all corrupted Change Admin (CM) tasks, jobs, and other associated internal task model objects in order to delete them from the database. If a corrupted object, such as a job, is referenced in a folder, the reference is removed and the job is deleted.

Run Monthly

audit_archive

Searches the database for audit log records based on input criteria. Once it finds the records to archive, it processes the archive. If the **-delete_record** argument is specified, the utility deletes the audit log records. Audit log entries with an audit definition with a day value of **-I** are not archived because **-I** indicates that the log record is permanent.

cleanup_recovery_table

Locates and removes all recipe objects resulting from time-out sessions and client crashes.

Teamcenter recipe objects are used to facilitate transparent recovery when a **TcServer** instance crashes. Typically, recipe objects are deleted when users log off. But during client crashes or session time-outs, recipe objects may remain in the database if the session or objects are not recovered.

Useful administration utilities

The following utilities are useful when performing maintenance activities.

Use this utility	To do this
am_install_tree	Install the Access Manager tree into the database.
backup_modes	Set the Teamcenter TCFS process in different modes. Blobby volume mode is used for Teamcenter hot backup.
backup_xmlinfo	Generate File Management System (FMS) ID tags for transient volumes in Teamcenter. Also create XML files for volume FMS IDs in the folder from where it is run.
clearlocks	Clear dead processes from the database. The -verbose option is recommended. Use the -assert_all_dead argument with extreme caution.
dataset_cleanup	Remove unreferenced dataset objects from the database. Run this utility periodically to clean up orphaned database objects.
index_verifier	Verify indexes on database tables. Custom indexes added to the database using the install utility can also be tracked using this tool.
install -lock_db	Locks the Teamcenter database until you unlock it using -unlock_db . While the database is locked, new logins are not allowed. Existing logins are not affected. Use this before a maintenance window in which you do not want users to log in. Commonly followed by clearlocks .
make_user	Create Person , User , Group , and Role objects in the Teamcenter Organization application.
purge_datasets	Remove old versions of datasets from the database. Run this utility periodically on the Teamcenter database.
purge_invalid_subscriptions	Remove invalid subscriptions on objects from the Teamcenter database. Run this utility periodically to clean up invalid subscription objects from the Teamcenter database.
purge_volumes	Remove unreferenced files from the operating system. These files are not attached to any valid dataset in the Teamcenter database. Run this utility periodically to clean up files on the operating system to increase disk space.
report_volume	List the operating system paths of all existing Teamcenter volumes. The path does not include the network node name.
reset_user_home_folder	Repair corruption that may occur when deleting a user from the database. The corruption redirects the user's home folder, mailbox

Use this utility	To do this
	folder, and new stuff folder to the folder of the individual performing the deletion.
review_volumes	Display detailed information about Teamcenter volumes and optionally remove unreferenced operating system files from these volumes.
verify_tasks	Find all corrupted CM tasks, jobs, and other associated internal task model objects to delete them from the database. If a corrupted object (such as a job) is referenced in a folder, the reference is removed and the job is deleted.

Using process daemons

Process daemons are small programs used to perform system processes. Use Teamcenter process daemons to manage system events, configure data sharing, monitor user subscriptions to system events, and monitor user inboxes for overdue tasks.

Process daemons include the following:

- Action Manager

Dispatches events that have a specified execution time or subscription events that have failed to process.

- Subscription Manager

Monitors the event queue for **TcEvent** objects.

- Task Manager

Checks a user's inbox for tasks that have passed their due date. If such a task is found, the daemon notifies the delegated recipients and marks the task as late.

The frequency of the daemon's monitoring is controlled by setting the **TASK_MONITOR_SLEEP_TIME** preference. To kill the daemon at any time, create an empty file as **TC_ROOT\logs\taskmonitor_graceful_exit.tmp**.

- Tessellation Manager

Tessellates **UGMASTER** and **UGALTREP** datasets to the JT (**DirectModel**) dataset and attaches the JT dataset back to the item revision and **UGMASTER** and **UGALTREP** datasets. Installing the Tessellation service is required to create the tessellated representations in Repeatable Digital Validation (RDV) that enable users of the Design Context application to quickly visualize components in context. The tessellated representations are created during the workflow release process, ensuring that JT files of the **DirectModel** datasets are updated as the NX files are released.

- **Shared Metadata Cache Service**

Synchronizes the shared server metadata memory cache.

You can install process daemons from Teamcenter Environment Manager (TEM) during installation or maintenance. In the **Features** panel, under **Server Enhancements > Database Daemons** select the daemon features to install.

Note:

You can provide additional security for process daemons by placing passwords in an encrypted file. The daemons must then be configured to run under an operating system user ID with read permissions on this password file.

Set Oracle-dependent services to automatically restart

The following services may not automatically restart after a server reboot. They are dependent on the Oracle service (for example, **OracleServiceTC**), a dependency that cannot be established when installing the services.

- Action Manager service
- Shared Metadata Cache service
- Subscription Manager service
- Task Monitor service
- Tessellation service

Set the dependency using the service **depend** command. For example, run the following command for the **actionmgrd** service:

```
sc config actionmgrd depend=OracleServiceTC
```

To see service names on Windows systems, open the **Control Panel** and choose **Administrative Tools→Services**.

Application process monitoring

Teamcenter runs various critical processes. Monitor these processes to ensure system availability to the end user.

The following table lists critical processes. If any of the processes running as a Windows service is down, an error is logged to the Windows **Event Viewer** with information on date, time, error type, source

(process), and server name. Set up automated log reading tasks to determine if a service is down and generate a critical event notification for the Teamcenter support group.

Process	Service name	Source (in event viewer)	Server type	Typical log file location
Server manager(s)	Teamcenter Server Manager <i>configuration</i>	EventID is 7034 and the description contains the Teamcenter service name	Server manager	As described in Enterprise tier logs .
FMS (File Management System)	Teamcenter FSC service <i>configuration</i>	EventID is 7034 and the description contains the Teamcenter service name	File management servers	As described in Resource tier logs .
Subscription Manager	Teamcenter Subscription Manager Service	In Application tab with Source as Subscriptionmgrd	Application business logic server	<i>TEMP</i> \subscriptionmanager.syslog
Action Manager	Teamcenter Action Manager Service	In Application tab with Source as actionmgrd	Application business logic server	<i>TEMP</i> \actionmanager.syslog
License Manager	Siemens License Server (ugslmd)	The service should be up and running ugslmd	License manager server	NA
TcWeb tier	TcWeb tier	The Source is TcWebTier and the Event Type is Error or Critical	Application business logic server	NA

Maintaining the database server

Maintaining the Oracle database

Tune the Oracle database

For best performance, maintain and tune Oracle database settings and services for your site. Tuning any database management system is an iterative process requiring careful record keeping and patience to measure, make configuration changes, and measure again, until optimal performance is achieved.

Set the Oracle environment

Manually set the Oracle environment

Before running Oracle database administration utilities, you must set the **ORACLE_HOME** and **ORACLE_SID** environment variables. Oracle Corporation recommends that you do not define Oracle environment variables in the system environment scope. Rather, define them manually when you use Oracle utilities. Defining Oracle environment variables at the system environment scope can cause conflict when running multiple versions of Oracle on the same machine.

Note:

Do not set **ORACLE_HOME** on SUSE Linux platforms.

Environment variable	Description
ORACLE_HOME	This environment variable must be set before any of the commands are started. ORACLE_HOME must be set to the path of the top-level (root) directory containing the Oracle application files. Note: Be consistent with the setting of ORACLE_HOME for a database. Certain database functions fail if this variable is set incorrectly, even if it is set to a valid path to the Oracle home directory (for example, a symbolic link). ORACLE_HOME must always be set to the same value as it was when the database was created.
ORACLE_SID	This environment variable is used to distinguish each unique database instance and therefore must be set to the database instance that you want to maintain or administer.

To manually set the Oracle environment, enter one of the following sets of commands:

Windows systems:

```
set ORACLE_HOME=ORACLE_HOME
set ORACLE_SID=tc
```

Linux systems:

```
export ORACLE_HOME=ORACLE_HOME
export ORACLE_SID=tc
```

Replace **ORACLE_HOME** with the path to Oracle. All examples assume that the database SID is **tc**.

Manually set the PATH environment variable for Oracle

It is also useful to add the Oracle **bin** directory to the **PATH** environment variable to allow Oracle scripts and utilities to be run from the command line without adding a directory path qualification. Additionally, many of the Oracle administration scripts require this as they often invoke other Oracle commands without using a fully qualified directory path name.

To add the Oracle **bin** directory to the **PATH** environment variable, enter one of the following commands:

Windows systems:

```
set PATH=%PATH%;%ORACLE_HOME%\bin
```

Linux systems:

```
export PATH=$PATH:$ORACLE_HOME/bin
```

Manually set the shared library path for Oracle (Linux systems)

Certain Oracle executables use shared library. To run these programs, set the platform specific, shared library path environment variable to include the Oracle **lib** directory.

Consider defining **ORACLE_HOME**, **ORACLE_SID**, **PATH**, and shared library path in this manner in the logon startup scripts of the Oracle user (**.cshrc** for C shell users, **.profile** for Bourne or Korn shell users).

Oracle services (Windows systems)

Introduction to Oracle services for Windows systems

There are two types of services associated with running an Oracle database: Oracle listener and database instance. Both types must be running on the system when running Teamcenter.

Process	Description
Oracle listener	The Oracle listener (OracleTNSListener) service monitors database connections from remote clients. This is a SQL*Net V2/Net process and is required for Teamcenter to connect. Under SQL*Net V2/Net, several listeners may run on the same system, each listening for connect requests to particular databases. Each listener must be configured to listen on a different port. Even if Teamcenter is run

Process	Description
	on the Oracle server, it is necessary to start this service because all Teamcenter database requests use the remote connect mechanism.
Database instance	There is one Oracle database service for an Oracle database instance. It must be running for Teamcenter to function properly. The service is: OracleServiceSID <i>SID</i> represents the database instance system identifier.

Starting the instance of an Oracle database is referred to as *initializing a database instance*. To initialize a database instance, use the Oracle SQL*Plus utility to manually start one database instance defined by the **ORACLE_SID** environment variable.

Caution:

Never shut down a database instance by killing database processes from the Windows Task Manager. Oracle databases require orderly shutdowns to ensure that all necessary database transactions are completed. Failure to observe this may result in the corruption of the database. Manual termination of processes also prevents the Oracle Relational Database Management System (RDBMS) from releasing memory that is no longer needed, and could require additional database recovery procedures at the next database startup.

Use the Oracle SQL*Plus utility to stop a single database instance defined by the **ORACLE_SID** environment variable.

Manually start Oracle services (Windows systems)

Before manually starting Oracle Services (Windows systems)

These instructions for manually starting the Oracle listener and database services are applicable only if a service or instance is not configured to start automatically when the system is restarted or if a service or instance terminates unexpectedly.

Note:

You cannot start all database instances at the same time after the system is started. To start all database instances at the same time, configure each database individually to start automatically following a system restart.

Start the Oracle listener (Windows systems)

You can start the listener service either from the **Services** dialog box in the Windows Control Panel or manually from a command prompt.

Note:

This example shows the startup of a listener service called **LISTENER**. More than one listener service can be run on a system and each listener should be defined in a configuration file called **listener.ora**.

- Control panel startup

1. Log on to the operating as a user with administrator privilege.
2. In the Windows **Services** dialog box, select the service named **Oraclehome-nameTNSListener**.
3. Click **Start**.
4. Verify that the **OracleTNSListener** service is running by checking the Windows **Services** dialog box for the following entry:

Oraclehome-nameTNSListener **Started** *startup-mode*

startup-mode is either **Automatic** or **Manual**.

5. Click **Close**.

- Manual startup

1. Log on to the operating system as a user with administrator privilege.
2. Manually set the Oracle environment:

```
set ORACLE_HOME=Oracle-home-path
set PATH=%PATH%;%ORACLE_HOME%\bin
```

3. Start the listener service:

```
%ORACLE_HOME%\bin\lsnrctl start LISTENER
```

4. Verify that the **OracleTNSListener** service is running by checking the Windows **Services** dialog box for the following entry:

Oraclehome-nameTNSListener **Started** *startup-mode*

startup-mode is either **Automatic** or **Manual**.

Start Oracle database services (Windows systems)

1. Log on to the operating system as a user with administrator privilege.
2. In the Windows **Services** dialog box, select the service named **OracleServiceSID** (*SID* is the instance system identifier, for example, **tc**).
3. Click **Start**.
4. Verify that the **OracleServiceSID** service is running by checking the Windows **Services** dialog box for the following entry:

OracleServiceSID **Started** *startup-mode*

startup-mode is either **Automatic** or **Manual**.

5. Click **Close**.

Initialize an Oracle database instance using SQL*Plus utility (Windows systems)

1. Log on to the operating system as a user with **administrator** privilege. To connect as internal (without a password), this account must be part of **ORA_DBA** Windows local group.
2. Manually set the Oracle environment by entering one of the following sets of commands:

```
set ORACLE_HOME=Oracle-home-path
set ORACLE_SID=tc
set PATH=%PATH%;%ORACLE_HOME%\bin
```

3. Start the Oracle SQL*Plus utility:

```
%ORACLE_HOME%\bin\sqlplus
```

4. Log on to SQL. At the **SQLPLUS** prompt, type the following command:

```
connect / as sysdba
```

The following system message is displayed:

```
Connected to an idle instance.
```

5. At the **SQLPLUS** prompt, type the following command:

```
startup
```

A system message similar to the following is displayed:

```
ORACLE instance started.
Total System Global Area 35042572 bytes
Fixed Size 70924 bytes
Variable Size 30294016 bytes
Database Buffers 4505600 bytes
Redo Buffers 172032 bytes
Database mounted.
Database opened.
```

6. The Oracle database is initialized. Exit the Oracle SQL*Plus utility by entering the following at the **SQLPLUS** prompt:

```
exit
```

Manually stop Oracle services (Windows systems)

Stop the Oracle listener (Windows systems)

The listener service can be shut down either from the **Services** dialog window in the Windows Control Panel or manually from a command prompt.

Note:

This example shows the shutdown of a listener service called **LISTENER**. More than one listener service can be run on a system and each listener should be defined in a configuration file called **listener.ora**.

- Control panel shutdown
 1. Log on to the operating system as a user with administrator privilege.
 2. In the Windows **Services** dialog box, select the service named **Oraclehome-nameTNSListener**.
 3. Click **Stop**.
 4. Verify that the **OracleTNSListener** service has stopped running by checking the Windows **Services** dialog box for the following entry:

Oraclehome-nameTNSListener *startup-mode*

startup-mode is either **Automatic** or **Manual**.

5. Click **Close**.
- Manual shutdown

1. Log on to the operating system as a user with administrator privilege. To connect as internal (without a password), this account must be part of **ORA_DBA** Windows local group.
2. Manually set the Oracle environment:

```
set ORACLE_HOME=Oracle-home-path
set PATH=%PATH%;%ORACLE_HOME%\bin
```

3. Stop the **OracleTNSListener** service:

```
%ORACLE_HOME%\bin\lsnrctl stop LISTENER
```

4. Verify that the **OracleTNSListener** service has stopped running by checking the Windows **Services** dialog box for the following entry:

Oraclehome-nameTNSListener *startup-mode*

startup-mode is either **Automatic** or **Manual**.

Stop Oracle database services (Windows systems)

1. Log on to the operating system as a user with administrator privilege.
2. In the Windows **Services** dialog box, select the service named **OracleServiceSID** (*SID* is the instance system identifier, for example, **tc**).
3. Click **Stop**.
4. Verify that the **OracleServiceSID** service is running by checking the Windows **Services** dialog box for the following entry:

OracleServiceSID *startup-mode*

startup-mode is either **Automatic** or **Manual**.

5. Click **Close**.

Shut down an Oracle database instance using SQL*Plus (Windows systems)

1. Log on to the operating system as a user with administrator privilege. To connect as internal (without a password), this account must be part of the Windows **ORA_DBA** local group.
2. Manually set the Oracle environment by entering the following commands:

```
set ORACLE_HOME=Oracle-home-path
set ORACLE_SID=tc
set PATH=%PATH%;%ORACLE_HOME%\bin
```

3. Start the Oracle SQL*Plus utility:

```
%ORACLE_HOME%\bin\sqlplus
```

4. Log on to SQL. At the **SQLPLUS** prompt, type the following command:

```
connect / as sysdba
```

The following system message is displayed:

```
Connected.
```

5. At the **SQLPLUS** prompt, type the following command:

```
shutdown
```

A system message similar to the following is displayed:

```
Database closed.  
Database dismounted.  
ORACLE instance shut down.
```

6. The Oracle database is shut down. Exit the SQL*Plus utility by entering the following command at the **SQLPLUS** prompt:

```
exit
```

Automate Oracle Service startup and shutdown (Windows systems)

Oracle services automatically start when the system is restarted and shut down when the system is shut down.

Note:

Oracle automatically shuts down Oracle databases when you shut down the Windows operating system. No configuration is required.

The Oracle listener service is configured to start automatically when the system is restarted during the Oracle installation process:

1. Log on to the operating system as a user with administrator privilege.
2. Choose **Start→Control Panel→Administrative Tools→Services**.
3. Select the desired service, for example, **Oraclehome-nameTNSListener** for the Oracle listener service or **OracleServiceSID** (SID is the instance system identifier, for example **tc**).

- Verify the startup mode of the service is running by checking the Windows **Services** dialog box for the following entry:

service-name *status* *startup-mode*

service-name is the name of the selected service, *status* is either **Started** if the service is running or blank if it is inactive, and *startup-mode* is the current startup mode.

- If the startup mode is not **Automatic**, click **Startup** in the **Services** dialog box, change the **Startup Type** to **Automatic**, and click **OK**.
- Click **Close**.

Oracle processes (Linux systems)

Introduction to Oracle processes for Linux systems

There are two types of processes associated with running an Oracle database: Oracle listener and database instance. Both types must be running on the system when running Teamcenter.

Process	Description
Oracle listener	The Oracle listener (tnslsnr) process monitors database connections from remote clients. This is a SQL*Net V2/Net process and is required for Teamcenter to connect. Under SQL*Net V2/Net, several listeners may run on the same system, each listening for connect requests to particular databases. Even if Teamcenter is run on the Oracle server, it is necessary to start this process because all Teamcenter database requests use the remote connect mechanism.
Database instance	Database processes are associated with a particular Oracle database instance. There are six processes associated with each database instance when it is first started. The processes are: • ora_pmon_SID Process monitor; performs Oracle process recovery when a user process fails. • ora_dbw0_SID Database writer process; writes dirty Oracle buffers to disk. • ora_ckpt_SID Checkpoint process; updates the headers of all Oracle data files to record the details of the checkpoint. • ora_smon_SID

Process	Description
	System monitor process; performs Oracle instance recovery at instance startup. • ora_reco_SID Oracle recovery process; process used with distributed database configuration that automatically resolves failures involving distributed transactions. • ora_lgwr_SID Log writer process; writes the Oracle redo log buffer to a redo log file on disk. <i>SID</i> represents the database instance system identifier.

Starting these processes on an Oracle database server is referred to as *initializing a database instance*. Use one of the following methods to initialize a database instance:

- To start all database instances flagged in the **oratab** file, use the Oracle **dbstart** utility. Only those instances marked with a **Y** flag are started.
- To start a single database instance defined by the **ORACLE_SID** environment variable, use the Oracle SQL*Plus utility.

All databases instances present on the system should be listed in the **oratab** configuration file. This file is located in the **/var/opt/oracle** directory on Solaris systems and in the **/etc** directory on all other platforms. Each instance should be listed on a separate line and conform to the following format:

ORACLE_SID:ORACLE_HOME:FLAG

ORACLE_SID is the system identifier of the instance (for example **tc**), **ORACLE_HOME** is the Oracle home directory associated with that instance (for example, **/u01/app/oracle/product/oracle-version**), and **FLAG** is **Y** or **N**. These flags are used by the Oracle **dbstart** and **dbshut** utilities to determine which instances to start or stop.

Caution:

Never shut down a database instance by killing database processes or by restarting the system. Oracle databases require orderly shutdowns to ensure that all necessary database transactions are completed. Failure to observe this may result in the corruption of the database. Manual termination of processes also prevents the Oracle Relational Database Management System (RDBMS) from releasing memory that is no longer needed, and could require additional database recovery procedures at the next database startup.

There are two methods for shutting down a database instance:

- Use the Oracle **dbshut** utility to shut down all database instances flagged in the **oratab** file. Only those instances marked with a **Y** flag are stopped.
- Use the Oracle SQL*Plus utility to stop a single database instance defined by the **ORACLE_SID** environment variable.

Manually start Oracle processes (Linux systems)

Start the Oracle listener process (Linux systems)

This example shows the startup of a listener process called **LISTENER**. More than one listener process can be run on a system. Each listener should be defined in a configuration file called **listener.ora**.

Manually start the **tnslsnr** process by performing the following procedure:

1. Log on to the operating system as **oracle**, or switch user to **oracle** by typing **su - oracle** followed by the **oracle** password.
2. Manually set the Oracle environment by typing one of the following commands:

```
export ORACLE_HOME=Oracle-home-path
export PATH=$PATH:$ORACLE_HOME/bin
```

3. Start **tnslsnr** by typing the following command:

```
$ORACLE_HOME/bin/lsnrctl start LISTENER
```

4. Verify that **tnslsnr** is running by typing the following command:

```
ps -ef | grep tnslsnr
```

The following process information is displayed:

```
oracle 1833 1 80 date time tnslsnr -inherit LISTENER
```

Replace *date* and *time* with the operating system date and time that **tnslsnr** was started.

Initialize all flagged Oracle database instances using dbstart (Linux systems)

1. Log on to the operating system as **oracle**, or switch user to **oracle** by typing the following command, followed by the **oracle** password:

```
su - oracle
```

2. Manually set the Oracle environment by typing one of the following commands:

```
export ORACLE_HOME=Oracle-home-path
export PATH=$PATH:$ORACLE_HOME/bin
```

3. Start all Oracle database instances flagged in the **oratab** file by typing the following command:

```
$ORACLE_HOME/bin/dbstart
```

4. Verify that the database processes are running by typing the following command:

```
ps -ef | grep ora
```

The following process information is displayed for each database instance:

```
oracle 1830 1 80 date time ora_dbw0_tc
oracle 1831 1 80 date time ora_pmon_tc
oracle 1832 1 80 date time ora_lgwr_tc
oracle 1833 1 80 date time ora_smon_tc
oracle 1832 1 80 date time ora_reco_tc
oracle 1833 1 80 date time ora_ckpt_tc
```

Replace *date* and *time* with the operating system date and time that the database process was started. This example shows the background database processes associated with an Oracle instance called **tc**.

Initialize an Oracle database instance using SQL*Plus utility (Linux systems)

1. Log on to the operating system as **oracle**, or switch user to **oracle** by typing the following command, followed by the **oracle** password:

```
su - oracle
```

2. Manually set the Oracle environment by typing one of the following sets of commands:

```
export ORACLE_HOME=Oracle-home-path
export ORACLE_SID=tc
```

3. Start the Oracle SQL*Plus utility:

```
$ORACLE_HOME/bin/sqlplus
```

4. Log on to SQL. At the **SQLPLUS** prompt, type the following command:

```
connect / as sysdba
```

The following system message is displayed:

```
Connected.
```

5. At the **SQLPLUS** prompt, type the following command:

```
startup
```

The following system message is displayed:

```
ORACLE instance started.
Total System Global Area 35069936 bytes
Fixed Size 69916 bytes
Variable Size 30314496 bytes
Database Buffers 4505600 bytes
Redo Buffers 180224 bytes
Database mounted.
Database opened.
```

6. The Oracle database is initialized. Exit the Oracle SQL*Plus utility by entering the following at the **SQLPLUS** prompt:

```
exit
```

Manually stop Oracle processes (Linux systems)

Stop the Oracle listener (Linux systems)

This example shows the shutdown of a listener process called **LISTENER**. More than one listener process may be run on a system and each listener should be defined in a configuration file called **listener.ora**.

1. Log on to the operating system as **oracle**, or switch user to **oracle** by entering **su - oracle** followed by the **oracle** password.
2. Manually set the Oracle environment by typing one of the following commands:

```
export ORACLE_HOME=Oracle-home-path
export PATH=$PATH:$ORACLE_HOME/bin
```

3. Stop **tnslsnr** by typing the following command:

```
$ORACLE_HOME/bin/lsnrctl stop listener
```

4. Verify that the **tnslsnr** process is no longer running by entering the following command:

```
ps -ef | grep -v grep | grep tnslsnr
```

There should be no output returned by this command.

Shut down all flagged Oracle database instances using dbshut (Linux systems)

1. Log on to the operating system as **oracle**, or switch user to **oracle** by entering the following command, followed by the **oracle** password:

```
su - oracle
```

2. Manually set the Oracle environment by entering one of the following commands:

```
export ORACLE_HOME=Oracle-home-path
export PATH=$PATH:$ORACLE_HOME/bin
```

3. Stop all Oracle database instances flagged in the **oratab** file:

```
$ORACLE_HOME/bin/dbshut
```

4. Verify that the database processes are no longer running:

```
ps -ef | grep -v grep | grep ora
```

There should be no output returned by this command.

Shut down an Oracle database instance using SQL*Plus (Linux systems)

1. Log on to the operating system as **oracle**, or switch user to **oracle** by typing the following command, followed by the **oracle** password:

```
su - oracle
```

2. Manually set the Oracle environment by typing one of the following commands:

```
export ORACLE_HOME=Oracle-home-path
export ORACLE_SID=tc
```

3. Start the Oracle SQL*Plus utility:

```
$ORACLE_HOME/bin/sqlplus
```

4. Log on to SQL. At the **SQLPLUS** prompt, type the following command:

```
connect / as sysdba
```

The following system message is displayed:

```
Connected.
```

5. At the **SQLPLUS** prompt, type the following command:

```
shutdown immediate
```

The following system message is displayed:

```
Database closed.
Database dismounted.
ORACLE instance shut down.
```


6. The Oracle database is shut down. Exit the SQL*Plus utility by typing the following command at the **SQLPLUS** prompt:

```
exit
```

Automate Oracle startup and shutdown processes (Linux systems)

Automate Oracle startup processes (Linux systems)

Most system administrators find it helpful to automatically start and stop the Oracle processes when starting and stopping the system, respectively. This ensures a graceful and orderly shutdown of Oracle processes.

Automatic startup of Oracle server and database processes is achieved by scripting startup commands so that they are started automatically each time the system is restarted. There are two methods of doing this:

- Modify an existing system run control (**rc**) script to include the startup commands.
- Create a new system **rc** script using sample scripts provided by Siemens Digital Industries Software.

Automate Oracle shutdown processes (Linux systems)

Automatic shutdown of Oracle server and database processes is achieved by scripting shutdown commands so that they are run automatically each time the system is shutdown. There are two methods of doing this:

- Modify an existing system run control (**rc**) script to include the shutdown commands
- Create a new system **rc** script using sample scripts provided by Siemens Digital Industries Software

Oracle database security (Windows systems)

With Oracle, the Oracle internal account does not require a password. It uses Windows native authentication, by using Windows user logon credentials to authenticate privileged database users.

During installation of Oracle server software, the Oracle Universal Installer creates the Windows local **ORA_DBA** group and adds your Windows user name to this group, giving you **SYSDBA** privilege. For anyone else to connect as **internal** without a password, the Windows user name must be added manually to this **ORA_DBA** Windows group.

If you connect to a database using **sqlplus**, and connect as **internal**, and the database requests a password, check whether your Windows user name is part of the Windows local **ORA_DBA** group and that the Oracle server **ORACLE_HOME\networkadmin\sqlnet.ora** file has the **SQLNET.AUTHENTICATION_SERVICES** parameter set to **NTS**.

Note:

If you use an Oracle database and want to change the password Teamcenter uses to connect to the database, you must temporarily set the **TC_DB_CONNECT** environment variable and then re-encrypt the password.

Oracle database management**View Oracle database information (Windows systems)**

You can view and control the size and content of the Oracle database.

The following SQL commands are provided to view general database information. Prior to entering any SQL statements, you must run the Oracle SQL*Plus utility as follows:

1. Log on to the operating system as a user with **administrator** privilege.
2. Manually set the Oracle environment by typing the following commands:

```
set ORACLE_HOME=Oracle-home-path
set ORACLE_SID=tc
set PATH=%PATH%;%ORACLE_HOME%\bin
```

3. Start the Oracle SQL*Plus utility:

```
%ORACLE_HOME%\bin\sqlplus
```

4. Log on to SQL. At the **SQLPLUS** prompt, type the following command:

```
connect db-user/password
```

Replace *db-user* with the Teamcenter database user name; replace *password* with the database user password.

5. The following system message is displayed:

```
Connected.
```

6. Issue any of the following commands to obtain desired information about your database.

- To list a summary of the Teamcenter database files, type the following command from the **SQLPLUS** prompt:

```
select * from sys.dba_data_files;
```

- To list available space in the tablespace in bytes, type the following command from the **SQLPLUS** prompt:

```
select sum (bytes) from sys.dba_free_space where tablespace_name='IDATA';
```

- To list available space in the **SYSTEM** tablespace in bytes, type the following command from the **SQLPLUS** prompt:

```
select sum (bytes) from sys.dba_free_space where tablespace_name ='SYSTEM';
```

7. To exit the SQL*Plus utility, type the following command from the **SQLPLUS** prompt:

```
exit
```

View Oracle database information (Linux systems)

You can view and control the size and content of the Oracle database.

The following SQL commands are provided to view general database information. Prior to entering any SQL statements, you must run the Oracle SQL*Plus utility as follows:

1. Log on to the operating system as **oracle**, or switch user to **oracle** by typing the following command, followed by the **oracle** password:

```
su - oracle
```

2. Manually set the Oracle environment by typing the following commands:

```
export ORACLE_HOME=Oracle-home-path
export ORACLE_SID=tc
```

3. Start the Oracle SQL*Plus utility:

```
$ORACLE_HOME/bin/sqlplus
```

4. Log on to SQL. At the **SQLPLUS** prompt, type the following command:

```
connect db-user/password
```

Replace *db-user* with the Teamcenter database user name; replace *password* with the database user password.

5. The following system message is displayed:

```
Connected.
```

6. Issue any of the following commands to obtain desired information about your database.

- To list a summary of the Teamcenter database files, type the following command from the **SQLPLUS** prompt:

```
select * from sys.dba_data_files;
```

- To list available space in the tablespace in bytes, type the following command from the **SQLPLUS** prompt:

```
select sum (bytes) from sys.dba_free_space where tablespace_name='IDATA';
```

- To list available space in the **SYSTEM** tablespace in bytes, type the following command from the **SQLPLUS** prompt:

```
select sum (bytes) from sys.dba_free_space where tablespace_name = 'SYSTEM';
```

7. To exit the SQL*Plus utility, type the following command from the **SQLPLUS** prompt:

```
exit
```

Delete Oracle databases (Windows systems)

You can delete Oracle databases from a server using the Oracle Database Configuration Assistant. Remove the Oracle database services and files as follows:

1. Log on to the operating system as a user with administrator privilege.
2. Choose **Start→All Programs→Oracle_{home-name}→Database Configuration Assistant**.
3. Select **Delete a database** and click **Next**.
4. Select the database to be deleted (for example, **TC**) and click **Finish**.

All the database files and administrator files are deleted.

You may need to manually remove remaining files relating to this database instance from the **ORACLE_HOME\database** directory. These files have the instance identifier as parts of their names, for example, **iniiman0.ora**, **strtiman.cmd**, **imandb1.lst**, **imandb3.lst**, and **imandb3.sql**.

Oracle semaphores (Linux systems)

Introduction to Linux semaphores in Oracle

Linux semaphores are designed to allow processes to synchronize execution by allowing only one process to perform an operation on the semaphore at a time. The other processes sleep until the semaphores values are either incremented or reset to 0.

Linux semaphores are integer-valued objects set aside by the operating system that can be incremented or decremented automatically. They are designed to allow processes to synchronize execution by allowing only one process to perform an operation on the semaphore at a time. The other processes sleep until the semaphores values are either incremented or reset to 0.

Linux typically uses many semaphores and allocates them to the system in sets. When the Linux kernel is configured, the following semaphore parameters are rigidly set and cannot be changed without rebuilding the Linux kernel and restarting the system:

- Maximum number of semaphores (**SEMMNS**)

The Linux kernel parameter **SEMMNS** is used to specify the maximum number of semaphores in the system.

To increase this parameter, set **SEMMNS** to the sum of the **PROCESSES** parameter for each Oracle database, adding the largest one twice, then add an additional 10 for each database. For example:

```
ORACLE_SID=A, PROCESSES=100
ORACLE_SID=B, PROCESSES=100
ORACLE_SID=C, PROCESSES=200
```

The value for **SEMMNS** is calculated as follows:

$$\text{SEMMNS} = [(A=100) + (B=100) + [(C=200) * 2] + [(\# \text{ of instances}=3) * 10] = 630$$

See the operating system documentation for instructions about allocating semaphores and rebuilding the Linux kernel.

- Maximum number of semaphores per set (**SEMMSL**)

The Linux semaphore kernel parameter **SEMMSL** is used to specify the number of maximum number of semaphores in a semaphore set. To increase this parameter, set **SEMMSL** to 10 plus the largest **PROCESSES** parameter of any Oracle database on the system. The **PROCESSES** parameter can be found in each **initsid.ora** file, located in the **ORACLE_HOME/dbs** directory.

Oracle use of Linux semaphores

Oracle uses **semaphores** to control concurrency between all the background processes (**pmon**, **smon**, **dbw0**, **lgwr**, **reco**, **ckpt**, and **oracle shadows**) and to control communication between the user process and shadow process.

Typing **ipcs -sb** in a shell displays a list of semaphores allocated to the system at the moment. This list includes all the semaphore sets allocated, their identifying number, the owner, and the number of semaphores in each set.

Occasionally, unexpected termination of Oracle processes leaves semaphore resources locked. If the database is not running, but **ipcs -sb** lists semaphore sets owned by **oracle**, these must be reallocated. If this is not done, semaphore resources may not be sufficient to allow restarting the database.

Freeing semaphore sets is done by either using the **ipcrm** command or by restarting the system. Normally, system administrators do not want to restart the system only to free semaphore resources. Semaphore sets can be freed one at a time by performing the following procedure:

Warning:

Do not attempt to reallocate semaphore resources from Oracle if the Oracle server process (**orasrv**) is running. Corrupted data may result.

1. Log on as root.
2. Type the following command to display a list of semaphores owned by **oracle**:

```
ipcs -sb |grep ora
```

3. Free each semaphore set by typing:

```
ipcrm -s ID
```

Replace *ID* with the set identifying number from semaphores listed in step 2.

4. Repeat step 3 until all semaphores owned by Oracle are reallocated.

Common Linux semaphore problems in Oracle

Oracle problems and errors involving **Linux semaphores** often indicate insufficient or improperly configured semaphore resources on that Linux system. Semaphore resources may need to be optimized before Oracle runs properly.

- Semaphore problems during startup

Oracle allocates all the semaphores it needs for the background processes at database startup. The Oracle **init.ora** processes parameter determines the number of semaphores that will be allocated for Oracle use. If Oracle requires more semaphores than are allowed in one set, additional sets are allocated to Oracle. The following error codes are common startup errors.

Error	Cause
ORA-7251 spcre: semget error, could not allocate any semaphores	No semaphores are configured, or every semaphore is currently allocated.
ORA-7252 spcre: semget error, could not allocate semaphores	The first full set of semaphores was successfully allocated, but additional sets could not be allocated.
ORA-7279 spcre: semget error, unable to get first semaphore set	The system is attempting to allocate the first set of semaphores. The system either does not have sufficient semaphore resources or

Error	Cause
	too many semaphores or semaphore sets are already allocated.

The corrective actions for all three of the preceding errors are the same:

1. Check semaphores in use. Verify all unused semaphores are reallocated.
 2. Rebuild Linux kernel to allocate additional semaphore resources.
- Semaphore during shutdown abort

When a shutdown abort is performed, Oracle background processes are killed and semaphore sets are reallocated, without waiting for the user processes to finish. The following error codes are common shutdown abort errors:

- ORA-7264 spwat: semop error, unable to decrement semaphore
- ORA-7265 sppst: semop error, unable to increment semaphore

Cause	Action
One or both of these error codes is displayed as a result of the following scenario: after a shutdown abort, one or more users ends a request to the database and the request process dies. This occurs because the attempt to increment or decrement the semaphore fails.	This is an effective (though ungraceful) way of letting the users know that the database has been shut down with the abort option.

NLS_LANG environment variable

The **NLS_LANG** environment variable is an Oracle variable used to define language, locale, and character set properties.

When you perform Oracle export or import, you must set the **NLS_LANG** environment variable. The **NLS_LANG** environment variable controls character-set conversion between the source database and the target database. The **NLS_LANG** environment variable has the following format:

NLS_LANG=*language_territory.character-set*

For example:

```
NLS_LANG=AMERICAN_AMERICA.US7ASCII
```

For export and import, only the *character-set* portion is important. For *language_territory*, you can always use **AMERICAN_AMERICA**.

You must set the **NLS_LANG** environment variable explicitly.

Note:

If you do not explicitly set **NLS_LANG**, the system uses the default value. On Linux systems, the default **NLS_LANG** value is **AMERICAN_AMERICA.US7ASCII**, which may cause export or import to issue warnings or errors. On Windows systems, the default **NLS_LANG** is obtained from the following Windows registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME id

id is 0, 1, 2, and so forth. The setting in the registry for the character set reflects the character set you selected when you created the database using Oracle DBCA.

- Set **NLS_LANG** for export

When using Oracle export, set the character set on the **NLS_LANG** environment variable to the same character set as the database you are exporting. No conversion occurs, and the export file is created in the same character set as the original database. If you plan to import the file into a database with a different character set, you can postpone the conversion until the import.

To determine the character set of the current database, type the following command in SQL*Plus:

```
select value from nls_database_parameters where parameter='NLS_CHARACTERSET';
```

- Set **NLS_LANG** for import

When using Oracle import, setting the **NLS_LANG** environment variable depends on whether the source database and target database use the same character set:

- If the source database and target database use the same character set, set the **NLS_LANG** environment variable to use that character set.
- If the source database and target database use different character sets, leave the character set part of **NLS_LANG** the same on both export and import. Set it either to the same character set as the source database (preferred) or to the same character set as the target database. Conversion occurs only once, either on export or on import.

To determine the character set of the current database, type the following command in SQL*Plus:

```
select value from nls_database_parameters where parameter='NLS_CHARACTERSET';
```


Oracle initialization parameter files

You can start Oracle instances using either of two initialization files: an ASCII text file or a binary file. The binary file is called the server parameter file (SPFILE). The server parameter file is stored on the database server; changes applied to the instance parameters are persistent across all startups and shutdowns.

The default server parameter file is named **spfileSID.ora** and is located in the following directory:

Windows systems:

ORACLE_HOME\database

Linux systems:

ORACLE_HOME/dbs

The default ASCII text file is named **initSID.ora** and is located in the following directory:

Windows systems:

ORACLE_BASE\admin\ORACLE_SID\pfile

Linux systems:

ORACLE_BASE\admin\ORACLE_SID\pfile

You can also start Oracle instances using a specified ASCII text file or server parameter file, rather than the default file, or without specifying a file.

- Example: specifying no initialization file

The following example starts an Oracle instance without specifying a file:

```
sqlplus /nolog
SQL> connect / as sysdba
SQL> startup
```

Oracle first searches for the **spfileSID.ora** file. If it does not exist, Oracle searches for the **spfile.ora** file. If neither exists, Oracle uses the **initSID.ora** file. If none of these files exist, Oracle displays messages similar to the following example:

```
SQL> startup
ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file 'D:\ORA101\DATABASE\INITORA101.ORA'
```

- Example: specifying ASCII text file

The following example starts an Oracle instance using the **initSID.ora** file:

```
SQL> startup pfile=d:\ora101\database\initORA101.ora
ORACLE instance started.
Total System Global Area  118255568 bytes
Fixed Size                  282576 bytes
Variable Size               83886080 bytes
Database Buffers            33554432 bytes
Redo Buffers                 532480 bytes
Database mounted.
Database opened.
```

This option is not available if you are using a server parameter file. If you try to start an Oracle instance using this option and specifying a server parameter file, Oracle displays the following error message:

```
SQL> startup spfile=d:\ora101\database\spfileORA101.ora
SP2-0714: invalid combination of STARTUP options
```

If you start the database by specifying an ASCII text file, the **spfile** parameter is displayed as empty:

```
SQL> show parameter spfile
NAME                                TYPE        VALUE
-----
spfile                              string
```

- Example: specifying server parameter file

To start an Oracle instance using a server parameter file, you must use an **initSID.ora** file in which you specify only the **spfile** parameter:

Windows systems:

```
spfile=d:/ora101/database/spfiletest.ora
SQL> startup pfile=d:/ora101/database/inittest.ora
```

Linux systems:

```
spfile=d:\ora101\database\spfiletest.ora
SQL> startup pfile=d:\ora101\database\inittest.ora
```

```
ORACLE instance started.
Total System Global Area  122449892 bytes
Fixed Size                  282596 bytes
Variable Size               88080384 bytes
Database Buffers            33554432 bytes
Redo Buffers                 532480 bytes
Database mounted.
Database opened.
```

To determine whether you used the server parameter file, enter the following command in SQL*Plus:

```
SQL> show parameter spfile
```

NAME	TYPE	VALUE
spfile	string	d:\ora101\database\spfiletest.ora

Changing a parameter in the server parameter file depends on whether the parameter is static or dynamic. Changes to static parameters do not take effect until the database is restarted. Dynamic parameters can be changed while the database is running and do not require restarting the database.

To change a dynamic parameter:

```
SQL> alter system set open_cursors=400;
System altered.
```

To change a static parameter, qualify the command with **scope=spfile**:

```
SQL> alter system set processes=200 scope=spfile;
System altered.
```

Oracle online documentation

Oracle online documentation is included on a separate CD-ROM. Documentation is available in both hypertext markup language (HTML) and portable document format (PDF) formats. To view Oracle online documentation directly from the Documentation CD-ROM, click the **index.html** file.

Oracle Net implementation

Oracle Net features

Oracle Net uses its own set of configuration files and processes to listen for and accept database connection requests. The process which listens for connect requests is the **tnslsnr** (Linux) or the **OracleTNSListener** service (Windows). Several listener processes may be configured on a system if required. The connect string used by an Oracle client to make a remote connection request uses a short alias to represent a larger collection of server connect information. This information is referred to as the *connect descriptor*.

The following is an example of a connect descriptor on Linux:

```
test=(DESCRIPTION=
  (ADDRESS_LIST=
    (ADDRESS=
      (COMMUNITY=IMANTCP)
      (PROTOCOL=TCP)
      (HOST=infosun32)
      (PORT=1521)
    )
  )
  (CONNECT_DATA=
    (SID=imandev)
  )
)
```

The following is an example of a connect descriptor on Windows:

```
test=(DESCRIPTION=
  (ADDRESS_LIST=
    (ADDRESS=
      (COMMUNITY=TCP.world)
      (PROTOCOL=TCP)
      (HOST=infosun32)
      (PORT=1521)
    )
  )
  (CONNECT_DATA=
    (SID=test)
  )
)
```

test is the alias for this connect descriptor. Using the **tcdba** user as an example, the resulting connect string within Oracle would be:

```
tcdba/tcdba@test
```

The community defined in the above descriptor indicates the group of nodes to which this system belongs. Communities are used by Oracle to indicate like groups of nodes for use with the Multi-Protocol Interchange, which allows nodes on different types of networks (for example, TCP/IP, SPX) to communicate.

Oracle Net configuration files

Oracle Net may be configured via several files, but only two are mandatory files:

- The **listener.ora** file must reside on the Oracle server. It contains configuration data for running the **tnslsnr** listener process (Linux) and **OracleTNSListener** Windows service (Windows) including a list of databases for which it should listen for connection requests.
- The **tnsnames.ora** file must reside on an Oracle client. It contains the connect descriptors and their aliases for the databases to which this client connects. This file must also reside on an Oracle server.

On Windows, Siemens Digital Industries Software also configures the **sqlnet.ora** file, which must reside on the Oracle server and client and contains additional configuration parameters.

There are other files that can be created but if they do not exist, Oracle assumes defaults. Additionally, certain files only apply to products and functions which Teamcenter is not likely to use. These files may reside in several locations.

The search path that Oracle uses to locate these files on Linux is as follows:

```
HOME/.file-name.ora
```

```
TNS_ADMIN/file-name.ora
```

ORACLE_HOME/network/admin/file-name.ora

The search path that Oracle uses to locate these files on Windows is as follows:

file-name.ora in the current directory
TNS_ADMIN\file-name.ora
ORACLE_HOME\network\admin\file-name.ora

TNS_ADMIN is an environment variable that can point to any directory on the system where you want to keep these files.

- On Windows, *file-name.ora* implies the **listener.ora**, **tnsnames.ora** and **sqlnet.ora** files.
- On Linux, *file-name.ora* implies both the **listener.ora** and **tnsnames.ora** files. Note that in the case of the *HOME* directory location, these files are hidden.

Oracle Net Configuration Assistant

Due to the complexity of the configuration files, Oracle Corporation recommends that customers do not build or edit the files manually. Oracle Corporation supplies a product called Oracle Net Configuration Assistant, which allows you to enter information about the functions of various nodes on your network and then builds the configuration files for each of those nodes and a generic set of files for all possible client nodes.

On Windows, the files it generates must be copied to the appropriate nodes manually (usually using **ftp**).

Oracle Net service resolution on Teamcenter clients

Under Oracle Net, there is a requirement that each Oracle client have some means of translating the service alias supplied in the database connect string into its equivalent connect descriptor.

- On Windows, Teamcenter accomplishes this by creating a copy of the **tnsnames.ora** and **sqlnet.ora** files in the Teamcenter data directory during installation and setting the value of the **TNS_ADMIN** environment variable to **%TC_DATA%** in the **tc_profilevars.bat** file, to force Teamcenter to use this file.
- On Linux, Teamcenter accomplishes this by creating a copy of the **tnsnames.ora** file in the Teamcenter data directory during installation and setting the value of the **TNS_ADMIN** environment variable to **\$TC_DATA** in the **tc_profilevars** file, to force Teamcenter to use this file.

The Teamcenter data directory was chosen as the location for this file as it makes the file immediately visible to all Teamcenter clients who want to use this database and because this directory is guaranteed writable by the installer.

The **tnsnames.ora** file contains one entry—the entry for the database associated with that Teamcenter data directory. This file is regenerated every time the data directory is reconfigured.

You can override the setting of the **TNS_ADMIN** environment variable from the external environment or commented out in the `TC_DATA\tc_profilevars` file, depending on the requirements of the site.

Caution:

If you maintain copies of the Oracle Net configuration files in various locations in the system, be careful not to duplicate service aliases within those files. Duplication of service aliases could lead to connection to the wrong database and can result in lost or corrupted data. Ensure that the **TNS_ADMIN** environment variable is set correctly before running Teamcenter Integration for NX/NX Integration.

Maintaining the Microsoft SQL Server database

Tune the Microsoft SQL Server database

For best performance, maintain and tune Microsoft SQL Server database settings and services for your site. Tuning any database management system is an iterative process requiring careful record keeping and patience to measure, make configuration changes, and measure again, until optimal performance is achieved.

Start the SQL Server service

1. Log on to the operating system as a user with administrator privileges.
2. Open the Windows Control Panel.
3. From the Control Panel, **Administrative Tools**→**Services**.
4. From the list of services, select **SQL SERVER (MSSQLSERVER)** and click **Start**.
5. To verify that the SQL Server service is running, check the **Status** column. When the service is running, the status is **Started**.

Shut down the SQL Server service

1. Log on to the operating system as a user with administrator privileges.
2. Open the Windows Control Panel.
3. From the Control Panel, double-click **Services**.
4. From the list of services, select **MS SQLSERVER** and click **Stop**.
5. To verify that the SQL Server service has stopped, check the **Status** column. When the service not running, the status is blank.

SQL Server database security

SQL Server operates in one of two security (authentication) modes:

- Windows authentication mode

Windows authentication mode allows a user to connect through a Windows user account.

- Mixed mode

Mixed mode allows users to connect to an instance of SQL Server using either Windows authentication or SQL Server authentication. Users who connect through a Windows user account can use trusted connections in either Windows authentication mode or mixed mode.

Siemens Digital Industries Software recommends using mixed mode authentication, where the user is required to input a database user logon ID and password to be able to connect to the SQL Server.

Delete the SQL Server database

If a Teamcenter database is corrupted beyond repair, the simplest solution may be to delete all information from the database and start again.

1. Start SQL Server Enterprise Manager and expand **Microsoft SQL Servers**→**SQL Server group**→**Local Server**.
2. Select and right-click the **Database** menu under **LOCAL**.
3. Select the corrupted database and delete it.

View the SQL Server error logs

You can view the SQL Server error log using SQL Server Management Studio or any text editor.

1. Expand a server group and then expand a server.
2. Expand **Management** and then expand **SQL Server Logs**.
3. Double-click the SQL Server log.

Error log information is displayed in the details pane.